DEEP CONVOLUTIONAL NEURAL NETWORKS- A REVIEW

Tasneem Gorach

Student, Department of Computer Science, SVKM'S NMIMS Mukesh Patel School of Technology Management and Engineering, Maharashtra, India

---

ABSTRACT: Deep learning has progressed quite fast over the past few years and has become omnipresent in almost every field today. Convolutional neural networks are one of the most extensively used and researched neural networks which have found applications in almost every domain from image processing to natural language processing. They have mastered providing exemplary performances in visual challenges and proved to be better at major recognition tasks than most of their neural network contemporaries, along with being cost-effective and comparatively easier to train and having less number of parameters and lower error rates.

This paper discusses the structure and working of convolutional neural networks along with techniques advanced to improve its efficiency. Also, various applications of convolutional neural networks in separate fields and their development over time have been discussed and the effect of depth on their performance is analyzed.

KEYWORDS- *Convolutional neural networks, deconvolutions, natural language processing, activation functions, regularization techniques, image recognition*

1. INTRODUCTION

Convolutional neural networks (CNNs) are one of the most widely used type of deep artificial neural networks that are used in various fields such as image and video recognition, speech processing as well as natural language processing. These networks have been inspired by biological processes, such as working of the visual cortex in cats and spider monkeys [1]. Hubel and Wiesel, in the year 1969, studied the same to classify the cells in the cortex of these as- simple, complex and hypercomplex.

The complex cells were found to have a receptive field, i.e., the area of response to stimulus, approximately twice as large as that of a simple cell. Hence the idea of using translational invariance in order to recognize visual imagery was realized [2]. This property stated that the exact location of an object in an image was of less importance rather than detecting the object.

Using convolutional neural networks is better than fully connected networks in many applications since instead of every node in a layer being connected to every other node in the previous layer, the former has every node in the m^{th} layer being connected to n nodes in the $(m-1)^{th}$ layer, where n is the size of the receptive field of the CNN. This reduces the total number of parameters in the network and hence prevents overfitting and also ensures a built-in invariance.

The first convolutional network was introduced by LeCun et al, in 1998, known as Lenet-5.It was a 7 layered network which was used to digitize hand written digits on checks in banks. It introduced 3 basic architectures used in the network - shared weights, local receptive fields and pooling [2].

2. RELATED WORK

Convolutional neural networks have been discussed along with its various applications where image recognition continues to be the primary field where they are used.

One of the major revolutions in the applications of convolutional neural networks was made in 2012 when Alex Krizhevsky et al, applied CNNs to image classification - the most widely used application of CNN in today's scenario[3]. One of the major reasons that this had been achieved was the introduction of GPU computing which ensured that a vast dataset of images could be used to train and test the networks, which allowed a good case of generality and also fast computing due to increased degree of parallel processing. The network proposed by them, called AlexNet, was used to classify a total of 1.2 million images in a 1000 different classes. The error rates recorded were 17.0%(top-5) and 37.5%(top-1) which were a lot better than the previous technologies used in the ILSVRC(ImageNet Large Scale Visual Recognition Challenge) for image classification. Regularization techniques such as drop-out were used to prevent over-fitting in the fully connected layers of the network along with data augmentation techniques.

In 2013, Matt Zieler and Rob Fergus, proposed a CNN called ZFNet which had a top-5 error rate of 14.8% and was the winner of ILVSRC [4].Here, instead of relying on just trial and error, more focus was given on how the processing is exactly done by the layers. This was achieved by using deconvolutional networks which used the components of the convolutional network in reverse, in order to visualize the exact working of the CNN by mapping feature activations to the input pixels. In this network, the output image pixels had a path back to the input. Filter size was reduced from 11x11 (of AlexNet) to 7x7, receptive window was smaller and stride of convolution reduced from 4 to2.

2.COMPONENTS OF CONVOLUTIONAL NEURAL NETWORKS

The network architecture of CNNs comprises of various different components which are described below.

2.1 Layers In A Convolutional Neural Network

Artificial neural networks consist of various layers between the input and output layers. These layers known as the hidden layers, majorly consist of the following in the case of convolutional neural networks-

2.1.1 Convolutional Layer

Each convolutional neural network consists of different number of convolution layers depending on network requirements. The first convolutional layers are responsible for learning low level features such as edges, corners, etc. The output of these layers are often fed to other convolutional layers which learn higher level features.

Each neuron in this layer is connected to only a limited no. of neurons in the previous layer.The no. of neurons they are connected to is known as the receptive field of the convolutional layer.

These layers comprise of a filter which has a typical size of nxn. The filter is slid (or more precisely convolved) during the forward pass across the width and height of the input volume and dot products are computed between the entries of filters and the input position which comprises the final feature map .However, multiple filters are used for convolution since images have multiple features, and this signifies the depth of the feature maps (which is a hyper parameter) which is equal to the no. of filters used.

2.1.2 Pooling Layer

The pooling layer is used to decrease the spatial size or the resolution of the image in order to decrease the number of parameters, hence decrease the computational burden. It does so by reducing the no. of connections between the convolutional layers [5].They are usually alternated between the convolutional layers. The most common types of pooling are max pooling and average pooling. Though the use of average pooling has been decreased substantially lately, max pooling is still one of the most common method.

Lp pooling is a biologically inspired process [1] from complex cells of visual cortex.It is represented by

$$y_{i,j,k} = [\sum_{(m,n)\in R_{ij}} (a_{m,n,k})^p]^{1/p}$$

It can be noted that when p=1, Lp corresponds to average pooling. When p= ∞, Lp corresponds to max pooling.[5]

Another form of pooling is stochastic pooling, which is also used for regularization. It is a pooling method inspired from dropout [6].It picks a random activation within each pooling region based upon a multinomial distribution rather than the maximum value, like in max-pooling, which ensures that non-maximal activations present within the feature maps can also be utilized.

Though the training time error rates of max pooling are lesser but test time error of stochastic pooling is lesser than max pooling.

Another form of pooling is mixed pooling [7] which is the combination of average and max pooling. This method is proposed to perform better than max pooling and average pooling since it can solve problems related to over fitting more efficiently.

2.1.3 Fully Connected Layer

These layers are present in varying number in a CNN, generally right before the softmax layers. Every neuron in this layer is connected to every neuron in the previous layer. It is used to achieve linearity in the networks but can be replaced or converted into convolutional layers to turn the system into a fully convolutional network (FCN).

2.1.4 Softmax Layer

The softmax layer is generally a linear layer with a softmax classifier which converts the activations into values between 0 and 1, such that their sum is 1.It helps determine the final output of the CNN by finding the output of a recognition or classification task by determining the class or category having the highest probability value. It basically represents a probability distribution.

It could be represented by-

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

Where z is a vector of input units, j is an index for output values while k is an index to input values with the total number of elements in j being K.

2.2ACTIVATION FUNCTIONS

The most necessary role of activation functions in convolutional neural networks is to ensure non-linearity, i.e., multiple neurons are activated as a result of activating a single neuron.

Most common non-linear functions used in convolutional neural networks are listed below.

2.2.1 Logistic Function (Sigmoid Curve)

A sigmoid function is a function that is defined for all real input values. Also, it has a non-negative derivative at each point.

It is defined by

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}.$$

It has non-zero values and an activation range of [0,1]. This function updates at every point and hence is very smooth.

2.2.2 Hyperbolic tangent function

It is represented by

$$\tanh(y) = \frac{e^y - e^{-y}}{e^y + e^{-y}}$$

Its activation range is from [-1,1].

One of the major advantages of tanh is that it gives faster convergence during backpropogation, if the initial weight values are lesser, distributed quite uniformly and centered on 0. However in the case of sigmoid function, the output activations are biased towards lower half of the activation range [8].

2.2.3 ReLU function

ReLU activation function is represented as:

$$A[i, j, l] = \max(z[i, j, l], 0)$$

ReLU is a linear function which retains the positive values of the input and the negative values are turned to zero . Using ReLU ensures faster computations compared to both tanh and sigmoid functions. Training larger networks with more parameters becomes easier as a result. Also it is a numerically less complex function.

A problem in ReLU is that the function is not differentiable at the value z=0. This, however, is overcome easily by defining the gradient at that point to be 0.

ReLUs are really useful if sparsity in the hidden units is desired.

2.3 Regularization Functions

The use of too many parameters in a CNN leads to a problem called overfitting. In order to overcome that, various regularization techniques are used. Some major techniques used for regularization are-

2.3.1 Dropout

It's a regularization technique in which some neurons are randomly removed from the network. As a result of this, the learned weights of other neurons become insensitive to them. This helps the network to generalize better and the influence of an individual neuron on the output produced is reduced.
The probability that a neuron is not dropped from the network is p (p is a hyper parameter) where p is decided using a validation set. As a result, (1-p) becomes the probability of a neuron being dropped and its activation turned to 0 [9]. The probability p of each neuron is completely independent of the probabilities of all the other neurons.
If half the activations in each layer are randomly deleted, the errors too will be calculated using back propagation on only half the activations.

The error values are lower when 0.4<=p<=0.8 and increase as p becomes greater than this and close to 1.The lowest error value is for 0.6, however the default value considered is p=0.5 which is very close to optimal.
The equation for a feed-forward operation for a layer (m-1) in dropout can be given by-

$$z_i^m = w_i^m y'^{(m-1)} + b_i^m$$

Where

$$y'^{m-1} = r^{m-1} * y^{m-1}$$

And r^{m-1} is an independent Bernoulli variable vector where each variable has the probability p of being 1.

2.3.2 Drop-Connect

In this method, the weights connected to a certain neuron are set to zero with some probability .The probability of keeping a connection in the network is p.

The probability of dropping a connection in this case rather than an output unit is (1-p).This model introduces sparsity in the weights compared to DropOut which introduces dynamic sparsity to the output vector layer. As a result, the fully connected layers become sparsely connected layers since only certain connections are chosen at random.

The output of a Drop Connect layer is given by-

$$r = a((M * W) v)$$

Where M is a binary matrix containing connection information where elements in M are obtained by Bernoulli distribution of p[10].

2.3.3 Data Augmentation

Data augmentation is another technique used for the reduction of overfitting in a network. The total amount of information available in the training set is increased by manipulating the data already available in the training set. Various techniques are used to achieve them, many of which can be categorized by data warping which includes augmenting the input data directly to the model available in the data space ,the idea behind which was conceptualized in [11].
Most commonly used data augmenting practices include flipping, cropping, translating, rotating the image along with changing or inverting the colors of the image.

2.3.4 Stochastic Pooling

One of the drawbacks of dropout are that it does not benefit convolutional layers much which are pivotal in the working of CNNs. Hence, this approach aims at making pooling, which occurs at all convolutional layers stochastic[6].A pooled map response in this case is selected in a probabilistic manner using multinomial distribution of the activations instead of averaging or calculating the maximum of the activations in a particular pooling region.
The probabilities p_i for every pooling region j is given by

$$p_i = \frac{a_i}{\sum_{k \in R_j} a_k}$$

A location l is then picked from within the region which is specified based on multinomial distribution formed by probabilities. The final pooled activation is then given by a_l,*where*

$$l \sim \mathrm{P}(p_1, \ldots, p_{|R_j|})$$

3. BASIC WORKING OF CONVOLUTIONAL NEURAL NETWORKS

Each neuron in a convolutional neural network in the m^{th} layer is connected to n neurons in the $(m-1)^{th}$ layer, where n is known as the receptive field of the CNN. Stride of a convolution basically refers to the step size used in convolution operation.A new feature map is obtained by first convolving the input with a weight/kernel. The convolution occurs as follows-

$$z^{m}_{i,j,k} = w_k^m * x_{i,j}^{m-1} + b_k^m$$

For the k^{th} feature map in the m^{th} layer. (i, j) is the location in the m^{th} layer ,where the value in the feature map has to be calculated [5].

w_k^m and b_k^m are the weights and bias for the kth kernel in the mth layer and $x_{i,j}$ is the input to be convolved.

The convolved output is then subjected to a non-linear activation function which yields the final feature map output.

Note that the complete set of feature maps are obtained by convolving the input with several weights/kernel. For e.g, in the case of colored images using an RGB scale, separate kernels representing different intensities of red, green and blue colors are used. However, at a time, the same weights are shared throughout the entire spatial locations of the input. This refers to one of the most central characteristics of CNNs, i .e, the shared weights concept. This concept makes the network less complex and results in fewer parameters.

Several convolutional layers are arranged such that the first layer detects low level features such as edges and corners and the latter layers detect high level features such as objects. The output of the convolutional layers after having the pooling and regularization functions performed on them passes on to the Softmax layer which usually has a fully connected layer as its previous layer. The fully connected layer can also be replaced by a 1x1 convolutional layer. The Softmax layer, then classifies the input provided to it into different classes, which becomes the final output.

Deconvolutional layers have also been used in convolutional networks in order to visualize the exact way that the CNN works. The layers perform the reverse of the convolution operation by mapping features back to the input pixels by providing a path for the output to be projected back to the input. A sensitivity analysis is also performed to see which parts of the input are more necessary for the recognition or classification process [6].

Any dataset that is used for training the network is usually divided into 3 sets- the training set, the test set and validation set which are used to initially train the network on some images, test with fresh set of images and validate the result obtained using another set of images respectively.

4. APPLICATIONS

CNNs have been proven to be useful in many applications and have achieved better performances in various fields compared to other neural network architectures.

4.1 Image Recognition

The use of CNNs for image detection and recognition has been in place since a long time [11][2]. Use of CNNs for classification purposes, though introduced later and has been widespread. [13] used CNNs for generic object detection and recognition while SVMs(Support Vector Machines) were trained on the features learned by the CNN to be used for classification purposes.[14] adapted a GPU implementation of image classification task using CNNs , such that the network was flexible as well as fully online, i.e., the weights were updated after each image. This implementation performed 10-60 times faster computationally than a compiler optimized CPU implementation. However, a revolutionary change in the image classification task by CNNs was brought by [3] in 2012.It achieved significantly lower error rates on ILSVRC(2012) than the previous state-of-the-art.It used a very efficient GPU implementation which involved using multiple GPUs in parallel. Use of efficient regularization techniques like dropout was made along with data-augmentation to reduce overfitting. Various improvements on this classification attempt have been made by reducing the filter size and stride of convolution [6], or increasing the depth [14] as well as width [15] of the network to achieve an even lower error rate on the classification task in the ILSVRC.

Fine-grained recognition refers to differentiating between the subordinate classifications categories such that they have small visual differences. Examples of this are dog breeds and bird species. Though, for a long amount of time, this process has been dominated by part- based classification or part annotations, various techniques without using these part annotation have since been introduced. The former consists of classification into sub-categories using certain parts such as nose and base of ears in the case of the identification of a dog breed, based on the idea that objects within the same basic category would have the same parts[17].These part annotations or keypoints make scaling up to hundreds of finer sub-domains rather difficult as well as cost prohibitive[16].In [25], a fine-grained recognition approach is used which focuses on both part discovery as well as feature learning but without the use of part annotations. The part discovery is done in an unsupervised manner by discovering sets of aligned images with similar poses. The same parts in these cases have similar locations in the images provided the images are well aligned in a given set.

One of the keypoints in fine-grained recognition is pose. In [18], the pose of different birds was used to determine the species of the bird using part annotations by CNNs extracting features from various pose normalized regions.[19] proposes a bilinear CNN model for fine-grained recognition which uses 2 feature extractors such that their outputs are multiplied using outer product and later pooled to obtain an image descriptor. The result of this is passed through a classifier in order to obtain predictions.

Scene recognition did not achieve the same kind of success as object recognition after [3].Since ImageNet was not a scene-centric database and there was a need for such a database. B.Zhou et al [20] used such a database called Places which consisted of 7 million labeled pictures of scenes. Comparing similarity of 2 scenes is more difficult than comparing the object similarity. Hence, [20] used 2 similarity measures, density and diversity. Given 2 databases A and B, A would be more dense if A has more similar nearest neighbors than B. On the other hand, A would be more diverse if two images from B are more similar to each other visually than A. [21] developed a scene parsing system in which every pixel in the image is labeled depending upon its object category. The CNN used a supervised training technique to learn features that were characterized as being low or mid-level. Furthermore, the CNN needs to be trained on a dataset that has pixel-level labeled images.

Object detection deals with localization of objects as well the ability to deal with the cases where the same object occurs in an image at more than one instances. [22] uses a DCNN for a base feature extraction. The network is used to predict a series of bounding boxes such that a fixed score exists for each box depending on the probability of finding an object of interest. A similar DNN based regression approach was used in [23], however it could not scale up such that it would include multiple classes as the cost of performing a single regression was high. Another approach is to generate certain region proposals during test time[24] which are category independent. A CNN computes the features for each region proposal by using image warping for a fixed size input regardless of the region shape using a linear SVM classifier that is category-specific for classification. It also uses a bounding box regression to reduce errors.

4.2 Face Recognition

Face detection using CNNs has been used since a long time [25] [26].Local image sampling was combined with a convolutional neural network to extract larger features successively with layers arranged in a hierarchical manner along with a self-organizing map (SOM) neural network which primarily provided dimensionality reduction and invariance to minor changes in images to perform face recognition. However, it used a rather small dataset with limited individuals. Other approaches consisted of facial detection, rather than recognition which showed whether a given image has a face in it or not by dividing the image into small windows and detecting the presence of a face within each window[26].However, this approach detected only frontal views of faces and also only if the faces were upright. In [27], Garcia et al used CNNs to

detect faces as an improvement to their previous effort in [28] which used skin color for detection since, the previous method could not be used in many applications such as gray scaled images. Hence, CNNs proved to be robust even in the cases of various poses, lighting as well as facial expressions and also faster than previous approaches [26], one of which involved using SVMs [29].

However, with the advancement in the size of datasets and the detection to be performed in a rather uncontrolled environment, improvements in the efficiency of the detection task, given large visual variations was required. [30] used a cascade architecture that was built on CNNs to obtain a high discriminative capacity and fast detection. The CNN cascade rejects the background regions quickly in the fast low resolution stages while it evaluates only some candidates, to detect the object, which is very small in number in the high resolution stage at the end. Another problem in this domain was multi-view face detection, i.e., to capture faces in a wide range of possible orientations. Though various approaches to this used annotation to facial landmarks, [31] proposed a Deep Dense Face Detector (DDFD) which could detect faces without the use of annotations using a single model based on a CNN.

[32] used a VGG face algorithm which consisted of a DCNN for face recognition and DPM (Deformable Parts Model) for localization as well as detection of faces. It resulted in recognition of faces over eight different benchmarks set by the NIST (National Institute of Standards and Technology) in variable illumination with good but different accuracy for different benchmarks. [33] uses convolutional neural networks for facial recognition purposes by using it as a feature extractor. The output of the CNN is fed into a simple logistic classifier in order to classify the images. [34] tried to investigate the reason why certain architectures work better than the others in facial recognition. Fusion of multiple CNN architectures, which provides a combination of features extracted from a lot of different networks and thus a better representation of the face, and the use of metric learning are some of the parameters which improve performance. Also, an architecture with a number of convolutional layers followed by a few fully connected layers provided better results while the filter size, pooling layers, receptive fields were determined by trial and error.

4.3 Video Recognition

CNNs generally use 2 spatial axes for the representation of images. However, representation of videos using this configuration becomes difficult since they consist of still images changing with respect to time. Videos also provide natural data augmentation (jittering) for a single image classification.

Action recognition in videos primarily consisted of two steps. First, performing motion recognition using certain handcrafted features and second, classifying these features [35]. These approaches also tend to make certain assumptions which would certainly not always exist in a real world environment. Furthermore, the handcrafted features are highly problem dependent and cannot be generalized.

The earlier approaches to using CNNs for action recognition comprised of dividing the video into multiple contiguous frames of images and object detection took place within these still frames [37].However, there was no representation of information of the motion that took place within these frames, i.e, motion detection did not exist.

Hence, various approaches have been proposed for representation of videos using CNNs incorporating the information in the time domain.

One of the approaches is introducing a 3-D convolution technique, in which, a third temporal axis apart from the 2 spatial axes is introduced. This is obtained by convolving a 3-dimensional kernel with the cube of contiguous still frames obtained by stacking them together. As a result of this, the feature maps in any convolutional layer are now connected to multiple frames in their previous layer [38].-

$$v_{ij}^{xyz} = \tanh\left(b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijk}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)} \right)$$

where P_i and Q_i are the height and width of the kernel of a particular feature map. b_{ij} represents the bias term while w_{ijk}^{pqr} represents the weight value at the position (p,q). v_{ij}^{xyz} represents the result of the convolution operation in the i^{th} layer for the j^{th} feature map at position (x,y,z).

Another approach is to use 2-D convolution technique and to fuse the feature maps in time. It consists of two streams. The first stream captures the information about the still frames/images, i.e., the detection of objects and scenes in the image

and the second stream captures the motion between the frames, like movement of the observer with respect to the objects [36]. This consists of 3 approaches-early fusion, late fusion, and mixed fusion [37].

In the first approach, the first convolutional layer itself is extended in time. In the second approach, two separate 2-dimensional single frame networks are developed and merged later at the first fully connected layer.

The third approach uses a mixture of both of the above defined approaches. Every convolutional layer is extended in time and at each layer, temporal convolutions along with spatial convolutions occur.

The input for motion detection, i.e., the temporal ConvNet in [37] is raw stacked frames. However,[36] uses optical flow for temporal recognition which provides better results in terms of mean accuracy.

4.4 Speech Synthesis

Speech synthesis had long been dominated by Hidden Markov and GMM models[39]. However, DNNs later started being proven better[40].Speech processing required some translational invariance within speech signals since different speakers have different speaking styles, which DNNs could achieve given sufficiently large networks with a large amount of training samples[45].

But CNNs were capable of achieving translational invariance with far fewer parameters.

CNNs were first proposed to be used in speech synthesis in [41], where convolution operation was theoretically proposed to be done in time domain to achieve small temporal shifts. Also, a convolutional RBM(CRBM) was then proposed in [42] to perform one dimensional convolution in time for speech processing to learn speech features without supervision as documented in [43].

However, it was later found, that the convolution performed along the frequency domain proved to be more beneficial since it is invariant to small shifts in speech frequency which is often the case with different speakers and also for the different emotions and pitches for the same speaker[44].

Similar to convolution, pooling also was desired to be performed in the frequency domain.

However, later a temporal convolution along with a convolution over frequency was used in order to reduce temporal variability[43]. Combination of convolution operations for both time and frequency could result in a 2 dimensional CNN similar to that used in image classification. The axes in this case though, were independent of each other, which is not the case in image analysis.

There was a significant improvement(5% and 10% for complete and limited weight sharing respectively) in phone error rate (PER) in the case of CNNs compared to DNNs.

However, the introduction of the convolution through time only resulted in a very small improvement, indicating that convolution in the frequency domain is more important.

The CNNs used in [45]and [44] had lesser or one convolutional layer and hence were not deep with many hidden units like most convolutional networks used for image/video processing. Hence,[46] explored spatial modeling such that multiple convolutional layers for a deeper network could be created. As a result, the balance between the no. of convolutional and fully connected layers along with the ideal no. of hidden units per layer was calculated.. [47] used a combination of RNNs and CNNs for speech emotion recognition. This problem is very challenging since emotion expression are very different for every person and the features extracted from the variations in voice cannot always accurately determine the emotion expressed. The speech input are converted into a 2D representation using Short Time Fourier Transform. The 2D representation is then analyzed by the CNNs and RNNs, the combination of which results in a time-distributed CNN. The accuracy results of the detection of various signals are compared to normal CNNs and RNNs where time distributed CNNs achieve the best results.

4.5 Text Recognition

CNNs have proved to be useful in many natural language processing(NLP) tasks such as sentence modeling, search query retrieval, various text classification tasks, etc.

Sentence classification by CNNs is done using a single convolution layer along with the use of pre-trained word vectors [49].It consists of different model variations, one of which (referred to as CNN-rad) randomly initializes all the words in

the beginning and are then slowly modified as training continues. Other models(one of which is CNN-static) use a set of vectors pre-trained on a 100 billion words from Google News in this case. The classification model can be used for tasks such as finding four similar words to a given word and performed remarkably well. Most sentence classification applications constitute splitting a sentence in sub-sentences and then performing various classification tasks using labels on these sub-sentences, since the granularity of the sub-sentence can positively be used to represent the entire sentence However, the granularity of any given sub-sentence isn't good enough to be used across all sentences, hence [50] uses a multi-channel variable size convolutional network(MVCNN) which uses variable-sized filters to be used with multigranular phrases.

Text recognition generally consists of two steps. First, detecting text in an image or a document, which is usually done using bounding box algorithms[51]. The second step actually comprises of actually recognizing the text within these bounding boxes.

Unlike other tasks like character recognition for scanned documents, recognizing text from an unconstrained image consists of various variations of font, texture, background, etc. As a result, many such applications use hand-crafted features to detect and later recognize these images. [52] uses CNNs to recognize text by implementing unsupervised feature learning algorithms which extract features from the images automatically. Text detection is performed using a detector window which is slid across the entire image to identify texts on which operations like word segmentation are performed and later recognition The network uses a lexicon(which a collection of pre-defined candidate words).The lexicon is used to recognize the text techniques like NMS and beam search along with the application of two separate CNNs with 2 convolutional layers each, for both detection and recognition operation. However, the recognition step in the approach above uses a set of known words(lexicons) and would not generalize well to an image containing unseen text or text structurally different than that used for training, like alpha-numeric values used for addresses are license plates. The approach in [51] , performs unconstrained text recognition, i.e , there are no set lexicons or known word length. A CNN structure incorporates a CRF(Conditional Random Field) model which is used for recognizing characters and its unary output terms are provided by a CNN character predictor. Higher-order terms are provided by an N-gram predictor CNN which works by calculating subsets of a given word. The network is also tested with alpha-numeric texts to evaluate efficiency in non-language scenarios yielding positive results .Despite not being trained for constrained recognition tasks, the model gives state-of-the-art results when provided with a fixed lexicon.

Most of the approaches above, use CNNs with one or two convolutional layers.[53] draws comparison between NLP and image processing to implement deep convolutional neural networks for text classification since depth has been proven to show better results with image recognition. It was found that extending the no. of layers to 17 and then 29 helped improve performance than other shallower CNNs as well as certain recurrent networks(LSTMs).However going deeper upto 49 layers decreased performance. Even though shortcut layers lessened the degradation, they were not able to achieve state-of-the-art results.

4.6 Handwriting Recognition

The task of recognizing handwriting was one of the first applications of image analysis by convolutional neural networks [2].Le Cun et al used CNNs to recognize handwritten digits on checks and hence, relied on automatic learning rather than hand-crafted features which improved the task efficiency in terms of computational cost as well as time.One of the major problems for recognizing handwriting wasn't identifying individual characters, but separating the characters from the ones near them in order to form a word or a sentence. This used to be done with the help of Heuristic Over-Segmentation which consisted of generating a large no. of potential cuts between characters and selecting the best combination by manually labeling all the character hypothesis. This proved to be cost-ineffective as well as time consuming. Handwriting can contain size, position, writing style as well as slant variations. Invariance to these features could in principle be obtained by training fully-connected neural networks but it would require a very large no. of training instances and a large no. of parameters. However CNNs have a built-in shift invariance which can achieve this task easily.

Tthe handwriting recognition task today is not just recognizing the handwriting and converting it into digital text but identifying the identity of the writer. The recognition can be offline(static) or online(dynamic).In the former, the handwriting is obtained after the writing process has been completed by scanning the document containing the handwritten text. In the latter case, the identification process occurs while the handwritten text is being obtained.[54] used CNN to perform signature verification of individuals such that it could differentiate between genuine signatures and skilled forgeries. The system worked by using skilled forgeries as well as genuine signatures for a subset of users to train the network by identifying gestural cues as difference between the two ,so that the learned features can perform well in identifying forgeries for unseen users. Handwriting analysis can be Writer Dependent(WD) or Writer Independent(WI). For Writer Dependent classifiers, a training set is constructed separately for an individual user with positive and negative

samples being fed to it with respect to a particular task for an individual user. Writer Independent classifiers, on the other hand, use a single model for training for all users by using a dissimilarity vector (which represents the difference between features of the query and template handwriting) as input.

Generally, the recognition approach constitutes segmenting handwritten text or words into smaller parts for recognition after which the final recognized result consists a combination of individually recognized parts or character which are then fed into a word recognition module which mostly implements a dictionary search algorithm. [55] used multiple convolutional neural networks in order to identify a large character set. Each CNN recognized a part of the character class(such as alphabets, digits, etc.) and also different languages which can then be combined by programming algorithms to create a rather flexible classifier which could be used to recognize differential big character classes.

5. DEPTH ANALYSIS

Krizhevsky et al found that decreasing the no. of convolutional layers by 1 in AlexNet increased the error rate by 2%.Hence depth plays a major role in determining the efficiency of the CNN. In 2014,Szegedy,et al [16] implemented the concept of contribution of depth in improving performance in convolutional networks, with their architecture named 'Inception' which was made out to be deeper as well as sparser than other CNNs. The network was 22 layers deep(considering the layers with parameters) and was found to have improved performance and decreased error rates over previous shallower implementations.

In 2015,Simonyan et al [15] explicitly studied the direct effect of increased depth on performance. The network (containing 19 weight layers) outperformed the previous ILVSRC winners by achieving lower error rates than them. The network still had the classical architecture of Lenet-5 but was improved substantially just by creating a deeper network.

However, it was found that increasing the network depth (i.e, the no. of layers) after a certain point leads to a degradation problem. As the depth increases, the accuracy of the network gets saturated and then starts decreasing quite rapidly. This error, however, is not caused by overfitting but by stacking more layers as verified by experiments[56].

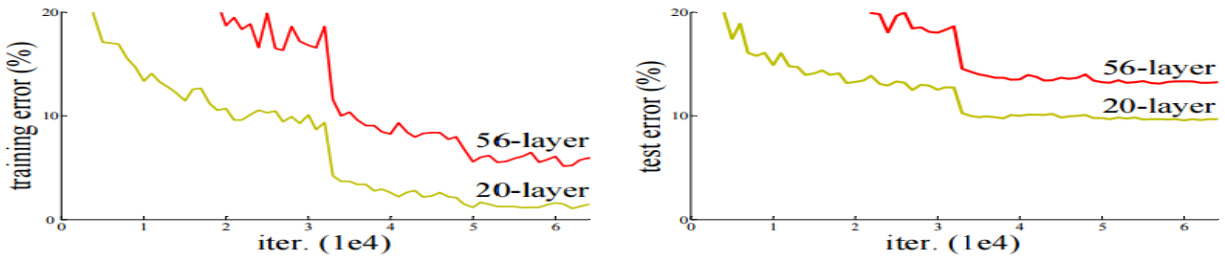

Fig 4(a)(source:[60])

The figure above clearly depicts training error(left) and test error (right).As we can observe the error rates are higher for the deeper network, i.e. the 56 layer network.

Hence, Kaiming He ,et al,in 2015 proposed a deep residual neural network inspired by the principle that introducing identity mapping in a network would decrease degradation. A residual mapping was introduced in this framework.

Here, the underlying desired mapping is denoted by H(x) ,while another mapping fitting the non-linear layers that are stacked together is denoted by F(x), such that

$$F(x) = (H(x) - x).$$

The original mapping is, as a result, remodeled into $F(x) + x$ since it was found that optimization of residual mapping is easier as well as more efficient compared to the original mapping optimization. Shortcut connections in the network refer to the connections formed by skipping one or more layers and the F(x) + x is formulated using these shortcut connections.

However, in 2017, Fractal Nets were introduced as an alternative to Residual networks. It was proposed that the key to ultra-deep networks was the transition from shallower to deeper networks during training and that the existence of

residual networks was in fact,incidental. The networks were found to match the performance of residual networks on Image Net and CIFAR datasets[57].

Other than image analysis,[53] showed that deeper CNNs help in achieving better performance for sentence classification in text analysis. The network performed substantially better than its shallower counterparts in text analysis which usually contained one or two convolutional layers for 17 and 29 layers. However, when the network depth was increased to 49 layers, the performance started degrading. The use of shortcut connections somewhat decreased the degradation, but state-of-the-art results could not be obtained.

CONCLUSION

Deep convolutional neural networks have been applied successfully to various applications. Besides providing breakthrough results with image recognition and classification tasks, they have also been applied to speech recognition, video analysis, natural language processing, etc., receiving positive results in terms of accuracy and cost They comprise of various components, with new components continuing to be added for the accomplishment of various tasks. Studies are constantly being done to find ways to improve their performance and widening their use to new tasks. Advances in GPU computing have helped CNNs to be used with a huge datasets and provide way faster computing which has eventually led to better generalization capacity by the networks.

Depth has played a major factor in increasing efficiency of the network with various experiments continously being done in order to achieve the most optimal performance with varying the number of hidden layers constantly.

However, with advancements in the architectures of convolutional neural networks, more advances in technology for better software and hardware representations too are needed. Also, though methods such as deconvolution have been proposed to recognize the intuitive theory behind the working of CNNs, the solid theory behind how CNNs work and why they provide such results with visual tasks is still not formulated. The number of parameters to be reduced in order to obtain higher accuracy is constantly being experimented on, with different regularization techniques being introduced .The advent of fully convolutional neural networks, also has reduced the number of parameters used by the network, by completely eliminating the fully connected layers.

REFERENCES

1.D. H. Hubel and T. N. Wiesel,"Receptive fields and functional architecture of monkey striate cortex",The Journal of physiology,1968,Vol.195,pp. 215-43

2.Yann Lecun,Leon Bottou,Yoshua Bengio,Patrick Haffner,"Gradient based learning applied to document recognition",Proceedings of the IEEE,1998,Vol. 86, pp. 2278 - 2324

3.Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton,"ImageNet Classification with Deep Convolutional Networks",Neural Information Processing Systems,2012, Vol 1,pp. 1097-1105

4.Matthew Zeiler and Rob Fergus,"Visualizing and Understanding Convolutional Neural Network"s,European Conference on Computer Vision,2013,pp 818-833

5.Gu, Jiuxiang & Wang, Zhenhua & Kuen, Jason & Ma, Lianyang & Shahroudy, Amir & Shuai, Bing & Liu, Ting & Wang, Xingxing & Wang, Gang,"Recent Advances in Convolutional Neural Networks", Pattern Recognition,2015

6.M. D. Zeiler, R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks",International Conference on Learning Representations, 2013

7.D. Yu, H. Wang, P. Chen, Z. Wei, "Mixed pooling for convolutional neural networks", Rough Sets and Knowledge Technology,2014,pp. 364-375

8.V. Nair, G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines", International Machine Learning Society, 2010,Vol. 27, pp.807-814

9. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors",Computing Research Repository, 2012.

10.L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, R. Fergus, "Regularization of neural networks using dropconnect",International Machine Learning Society,2013.,Vol. 30, pp.1058-1066

11.H. S. Baird,"Document Image Defect Models", Structured Document Image Analysis,IEEE Computer Society Press, 1995

12.Y. Le Cun, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel,"Handwritten digit recognition with a back-propagation network",Neural Information Processing Systems,1990

13.F. J. Huang, Y. LeCun, "Large-scale learning with svm and convolutional for generic object categorization", Conference on Computer Vision and Pattern Recognition,2006,Vol. 1, pp. 284- 291

14.D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification", International Joint Conference on Artificial Intelligence, 2011,pp. 1237-1242

15.K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition",International Conference on Learning Representations, 2015.

16.C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions", Conference on Computer Vision and Pattern Recognition, 2015,pp. 1-9

17.Jonathan Krause,Hailin Jin,Jianchao Yang, Li Fei-Fei,"Fine-Grained Recognition without Part Annotations,Conference on Computer Vision and Pattern Recognition" 2015,pp 5546-5555,

18.Thomas Berg Peter N. Belhumeur,"Part-Based One-vs-One Features for Fine-Grained Categorization,ace Verification, and Attribute Estimation",Conference on Computer Vision and Pattern Recognition,2013,pp 955-962

19.Steve Branson,Grant Von Horn,Serge Belongie,Pietro Perona,"Bird Species Categorization Using Pose Normalized Deep Convolutional Nets",British Machine Vision Conference,2014

20.T.-Y. Lin, A. RoyChowdhury, S. Maji, "Bilinear cnn models for fine-grained visual recognition", International Conference on Computer Vision, 2015,pp.1449-1457

21.Bolei Zhou , Agata Lapedriza1, Jianxiong Xiao , Antonio Torralba1 , and Aude Oliva1,"Learning Deep Features for Scene Recognition using Places Database",Neural Information Processing Systems,2014,Vol 1,pp. 487-495

22.C. Farabet, C. Couprie, L. Najman, Y. LeCun, "Learning hierarchical features for scene labeling",Pattern Analysis and Machine Intelligence, 2013, Vol 35, Isuue 8,pp. 1915-1929

23.Dumitru Erhan,Christian Szegedy,Alexander Toshev,Dragomir Anguelov,"Scalable Object Detection using Deep Neural Networks",Conference on Computer Vision and Pattern Recognition,2013,pp. 2155-2162

24.Christian Szegedy Alexander Toshev Dumitru Erhan,"Deep Neural Networks for Object Detection",Neural Information Processing Systems,,2013,Vol 2,pp.2553-2561

25.R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation",Conference on Computer Vision and Pattern Recognition,2013,pp. 580-587

26.Jonathan Krause,Timnit Gebru,Jia Deung,Li-Jia-Li,Li-Fei-Fei,"Learning Features and Parts for Fine-Grained Recognition", International Conference on Pattern Recognition,2014,pp. 26-33

27.Henry A. Rowley,Shumeet Baluja,Takeo Kanade,"Neural Network-Based Face Detection",IEEE Transactions on Pattern Analysis and Machine Intelligence,1996,Vol.20,pp.203-208

28.Christophe Garcia and Manolis Delakis,A Neural Architecture for Fast and Robust Face Detection,IEEE Transactions on Pattern Analysis and Machine Intelligence,2004,Vol.26 pp.1408 - 1423

29.C. Garcia and G. Tziritas, "Face Detection Using Quantized Skin Color Region Merging and Wavelet Packet Analysis", IEEE Trans. Multimedia,1998,Vol. 1,pp. 264-277

30.E. Osuna, R. Freund, F. Girosi." Training Support Vector Machines: an application to face detection", Conference on Computer Vision and Pattern Recognition, 1997,pp.130-136

31.Haoxiang Li ,Zhe Lin ,Xiaohui Shen ,Jonathan Brandt ,Gang Hua,"A convolutional neural network cascade for face detection", Conference on Computer Vision and Pattern Recognition,2015,pp. 5325-5334

32.Sachin Sudhakar Farfade, Mohammad Saberian, Li-Jia Li ,"Multi-view Face Detection Using Deep Convolutional Neural Networks",International Conference on Multimedia Retrieval,2015,pp. 643-650

33.P. Jonathon Phillips ,"A Cross Benchmark Assessment of A Deep Convolutional Neural Network for Face Recognition",IEEE International Conference on Automatic Face & Gesture Recognition,2017

34.Hurieh Khalajzadeh, Mohammad Mansouri and Mohammad Teshnehlab,"Face Recognition using Convolutional Neural Network and Simple Logistic Classifier", Advances in Intelligent Systems and Computing,2017,Vol. 223,pp. 197-207

35.Guosheng Hu et al,"When Face Recognition Meets with Deep Learning: an Evaluation of Convolutional Neural Networks for Face Recognition",International Conference on Computer Vision Workshop,2015

36.Ivan Laptev and Patrick Perez,"Retrieving actions in movies",International Conference on Computer Vision,2017,pp. 1-8

37.Karen Simonyan Andrew Zisserman,"Two-Stream Convolutional Networks for Action Recognition in Videos",Neural Information Processing Systems,2014,Vol 1,pp. 568-576

38.Andrej Karpathy et al,"Large-scale Video Classification with Convolutional Neural Networks",Conference on Computer Vision and Pattern Recognition,2014,pp. 1725-1732

39.Shuiwang Ji,Wei Xu,Ming Yang,Kai Yu, "3D Convolutional Neural Networks for Human Action Recognition",Pattern Analysis and Machine Intelligence,2013,Vol 35, Issue 1,pp. 221-231

40.Lixin Deng, & Patrick Kenny, Matthew Lennig, Vishwa Gupta, F Seitz,& Mermelstein,"Phonemic Hidden Markov Models with Continuous Mixture Output Densities for Large Vocabulary Word Recognition",IEEE Transactions On Signal Processing,1991,Vol 39,pp. 1677 - 1681

41.Dong Yu, Michael L. Seltzer, Jinyu Li , Jui-Ting Huang, Frank Seide ,"Feature Learning in Deep Neural Networks – Studies on Speech Recognition Tasks",International Conference on Learning Representations,2013

42.Y. LeCun and Y. Bengio, "Convolutional networks for images,speech, and time-series," in The Handbook of Brain Theory and Neural Networks, M. A., Ed. MIT Press, 1995,pp. 255-258

43.H. Lee, P. Pham, Y. Largman, and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in Advances in Neural Information Processing Systems, 2009,Vol 22, pp. 1096–1104

44.Ossama Abdel-Hamid, Li Deng, Dong Yu,"Exploring Convolutional Neural Network Structures and Optimization Techniques for Speech Recognition",InterSpeech,2013

45.O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn," Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition", International Conference on Acoustics, Speech, and Signal Processing,2012,pp. 4277 –4280

46.N. Sainath, Tara & Mohamed, Abdel-rahman & Kingsbury, Brian & Ramabhadran, Bhuvana,"Deep Convolutional Neural Networks for LVCSR",International Conference on Acoustics, Speech, and Signal Processing,2013,pp. 8614-8618

47.T. Sercu, V. Goel, "Advances in very deep convolutional neural networks for LVSCR", Interspeech, 2016

48.Wootaek Lim, Daeyoung Jang and Taejin Lee ,"Speech Emotion Recognition using Convolutional and Recurrent Neural Networks",Asia-Pacific Signal and Information Processing Association ,2016,pp. 1-4.

49.Yoon Kim,"Convolutional Neural Networks for Sentence Classification",Conference on Empirical Methods in Natural Language Processing,2014,pp. 1746-1751

50.Wenpeng Yin and Hinrich Schutze,"Multichannel Variable-Size Convolution for Sentence Classification",Conference on Natural Language Processing ,2016

51.Max Jaderberg,Karen Simonyan, Andrea Vedaldi,Andrew Zisserman,"Deep structured output learning for unconstrained text recognition",International Conference on Learning Representations,2015

52.Tao Wang,David J. Wu,Adam Coates,Andrew Y. Ng,"End-to-End Text Recognition with Convolutional Neural Networks",International Conference onPattern Recognition,2012,pp. 410-414

53.Alexis Conneau,Holger Schwenk,Yann Le Cun,"Very Deep Convolutional Networks for Text Classification,European Chapter of the Association for Computational Linguistics",2016,Vol 1,pp. 1107-1116

54.Luiz G. Hafemann,Robert Sabourin , Luiz S. Oliveira,"Learning Features for Offline Handwritten Signature Verification using Deep Convolutional Neural Networks",Asia-Pacific Conference on Simulated Evolution and Learning,2017,pp 310-319

55.Phạm Việt Dũng,"Multiple Convolution Neural Networks for an Online Handwriting Recognition System", IInternational Conference on Simulated Evolution And Learning,2012, pp 310-319

56.Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian SunDeep,"Residual Learning for Image Recognition",Conference on Computer Vision and Pattern Recognition,2015,pp. 770-778

57.Gustav Larsson, Michael Maire, Gregory Shakhnarovich,"FractalNet: Ultra-Deep Neural Networks without Residuals",International Conference on Learning Representations,2017