GOVERNMENT POLYTECHNIC, AURANGABAD

(An Autonomous Institute of Government of Maharashtra)

"PERSUIT FOR EXCELLENCE"

SEMINAR REPORT

ON

"CONVOLUTIONAL NEURAL NETWORK"

SUBMITED

BY

Mr. Waghmare Sagar Siddharth
(Enrolment No. 156213)

GUIDED

BY

Prof. P.S.Sadafule.

DEPARTMENT OF COMPUTER ENGINEERING

ACADEMIC YEAR 2018-19

DEPARTMENT OF COMPUTER ENGINEERING

GOVERNMENT POLYTECHNIC, AURANGABAD

(An Autonomous Institute of Government of Maharashtra)

CERTIFICATE

This is to certify that **Mr. Sagar.S.Waghmare** has successfully completed seminar work titled **"CONVOLUTIONAL NEURAL NETWORK"** during the academic year 2018-2019, in partial fulfillment of Diploma **in Computer Engineering** of Government Polytechnic, Aurangabad. To the best of my knowledge and belief this seminar work has not been submitted elsewhere.

Date: 26/09/2018

Prof. P. S. Sadafule
Seminar Guide

Prof. D. P. Sapkal
H.O.D, I. T/C.O

Prof. F. A. KHAN
PRINCIPAL

ACKNOWLEDGEMENT

We would like to express our profound thankfulness to the faculty members of Computer Engineering Department who has always been helpful and supportive. Much needed support and encouragement is provided on numerous occasions by the faculty members of our department. It gives us immense pleasure to express our sincere thanks to **Prof.P.S.SADAFULE** for sparing his valuable time in guiding us.

We also express our sincere thanks to our Head of Department **Prof.D.P.Sapkal** for his kind co-operation and encouragement without whose moral support and cheerful encouragement our seminar would not have been successful.

We take pleasure to convey our gratitude to **Prof. F. A. Khan**, Principal, Government Polytechnic, Aurangabad for permitting to do the seminar and encouragement.

Mr. SAGAR SIDDHARTH WAGHMARE
(Enrollment No : 156213)

Index

List of figures

ABSTRACT

Convolutional neural networks are a technology that combines artificial neural networks and recent deep learning methods. They have been applied to many image recognition tasks and have attracted the attention of the researchers of many countries in recent years. This paper summarizes the latest development of convolutional neural networks and expounds the relative research of image recognition technology and elaborates on the application of convolutional neural networks in handwritten numeral recognition.

Convolutional Neural Networks (CNN) is one kind of deep neural network. It can study concurrently. In this article, we gave a detailed analysis of the process of CNN algorithm both the forward process and back propagation. Then we applied the particular convolutional neural network to implement the typical face recognition problem by java. Then, a parallel strategy was proposed in section4. In addition, by measuring the actual time of forward and backward computing, we analyzed the maximal speed up and parallel efficiency theoretically.

1. INTRODUCTION

In machine learning, a convolutional neural network (CNN or ConvNet) is a class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery. CNNs use a variation of multilayer perceptions designed to require minimal preprocessing .They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

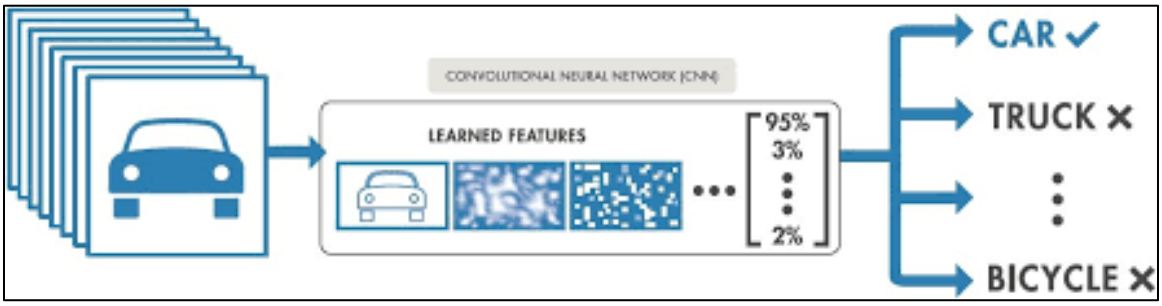

Fig.1.1. CNN Introduction

2. HISTORY

CNN design follows vision processing in living organisms.

2.1 RECEPTIVE FIELDS IN THE VISUAL CORTEX

Work by Hubel and Wiesel in the 1950s and 1960s showed that cat and monkey visual cortexes contain neurons that individually respond to small regions of the visual field. Provided the eyes are not moving, the region of visual space within which visual stimuli affect the firing of a single neuron is known as its **receptive field** [Neighboring cells have similar and overlapping receptive fields Receptive field size and location varies systematically across the cortex to form a complete map of visual space[The cortex in each hemisphere represents the contra lateral visual field

Their 1968 paper identified two basic visual cell types in the brain:

- simple cells, whose output is maximized by straight edges having particular orientations within their receptive field
- Complex cells, which have larger receptive fields, whose output is insensitive to the exact position of the edges in the field.

2.2. NEOCOGNITRONS

The Neocognitrons was introduced in 1980 the Neocognitrons does not require units located at multiple network positions to have the same trainable weights. This idea appears in 1986 in the book version of the original back propagation paper. Neocognitrons were developed in 1988 for temporal signals. Their design was improved in 1998, generalized in 2003 and simplified in the same year.

Neocognitrons A system to recognize hand-written zip code numbers involved convolutions in which the kernel coefficients had been laboriously hand designed

LeCun et al. in 1989, used back-propagation to learn the convolution kernel coefficients directly from images of hand-written numbers. Learning was thus fully automatic, performed better than manual coefficient design, and was suited to a broader range of image recognition problems and image types.

LeNet-5

LeNet-5, a pioneering 7-level convolutional network by LeCun et al. in 1998,that classifies digits, was applied by several banks to recognise hand-written numbers on checks (cheques) digitized in 32x32 pixel images. The ability to process higher resolution images requires larger and more layers of convolutional neural networks, so this technique is constrained by the availability of computing resources.

2.3. SHIFT-INVARIANT NEURAL NETWORK

Similarly, a shift invariant neural network was proposed for image character recognition in 1988.The architecture and training algorithm were modified in 1991 and applied for medical image processing and automatic detection of breast cancer in mammograms.

A different convolution-based design was proposed in 1988 for application to decomposition of one-dimensional electromyography convolved signals via de-convolution. This design was modified in 1989 to other de-convolution-based designs.

2.4. NEURAL ABSTRACTION PYRAMID

The feed-forward architecture of convolutional neural networks was extended in the neural abstraction pyramid by lateral and feedback connections. The resulting recurrent convolutional network allows for the flexible incorporation of contextual information to iteratively resolve local ambiguities. In contrast to previous models, image-like outputs at the highest resolution were generated.

2.5. GPU IMPLEMENTATIONS

Following the 2005 paper that established the value of GPGPU for machine learning, several publications described more efficient ways to train convolutional neural networks using GPUs.In 2011, they were refined and implemented on a GPU, with impressive results. In 2012, Ceresin et al. significantly improved on the best performance in the literature for multiple image databases, including the MNIST database, the NORB database, the HWDB1.0 dataset (Chinese characters), the CIFAR10 dataset (dataset of 60000 32x32 labeled RGB images), and the Image Net dataset.

3. ARTIFICAL NEURAL NETWORK

Artificial neural networks (ANN) or connectionist systems are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge about cats, e.g., that they have fur, tails, whiskers and cat-like faces. Instead, they automatically generate identifying characteristics from the learning material that they process.

An ANN is based on a collection of connected units or nodes called artificial neurons which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it.

In common ANN implementations, the signal at a connection between artificial neurons are a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The connections between artificial neurons are called 'edges'.

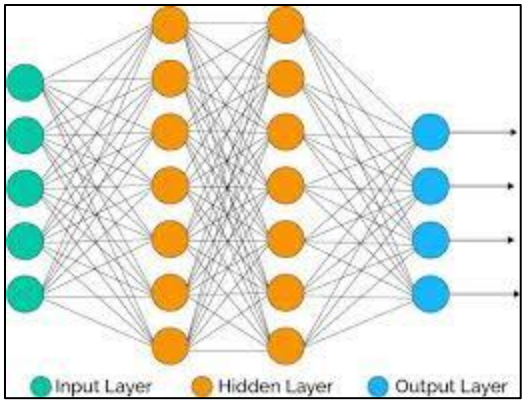

Fig.3.1. Hidden Layers in ANN

Artificial neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis.

4. DESIGN

A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers

Description of the process as a convolution in neural networks is by convention. Mathematically it is a cross-correlation rather than a convolution. This only has significance for the indices in the matrix, and thus which weights are placed at which index.

4.1. CONVOLUTIONAL

Convolutional layers apply a convolution operation to the input, passing the result to the next layer. The convolution emulates the response of an individual neuron to visual stimuli. Each convolutional neuron processes data only for its receptive field.

Although fully connected feed forward neural networks can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary, even in shallow (opposite of deep) architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10000 weights for *each* neuron in the second layer. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. For instance, regardless of image size, tiling regions of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters. In this way, it resolves the vanishing or exploding gradients problem in training traditional multi-layer neural networks with many layers by using back propagation

4.2. POOLING

Convolutional networks may include local or global pooling layers, which combine the outputs of neuron clusters at one layer into a single neuron in the next layer. For example, *max pooling* uses the maximum value from each of a cluster of neurons at

the prior layer. Another example is *average pooling*, which uses the average value from each of a cluster of neurons at the prior layer.

4.3. FULLY CONNECTED

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perception neural network (MLP).

4.4. RECEPTIVE FIELD

In neural networks, each neuron receives input from some number of locations in the previous layer. In a fully connected layer, each neuron receives input from *every* element of the previous layer. In a convolutional layer, neurons receive input from only a restricted subarea of the previous layer. Typically the subarea is of a square shape (e.g., size 5 by 5). The input area of a neuron is called its *receptive field*. So, in a fully connected layer, the receptive field is the entire previous layer. In a convolutional layer, the receptive area is smaller than the entire previous layer.

4.5. WEIGHTS

Each neuron in a neural network computes an output value by applying some function to the input values coming from the receptive field in the previous layer. The function that is applied to the input values is specified by a vector of weights and a bias (typically real numbers). Learning in a neural network progresses by making incremental adjustments to the biases and weights. The vector of weights and the bias are called a *filter* and represent some feature of the input (e.g., a particular shape). A distinguishing feature of CNNs is that many neurons share the same filter. This reduces memory footprint because a single bias and a single vector of weights is used across all receptive fields sharing that filter, rather than each receptive field having its own bias and vector of weights.

5. LAYERS IN CONVOLUTIONAL NEURAL NETW.

A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers. Description of the process as a convolution in neural networks is by convention.

5.1 CONVOLUTION LAYER

What is convolution?

Convolution is a Mathematical Operation in which a matrix is applied on image and on basis of RGB values of the pixel the value of each box is selected.

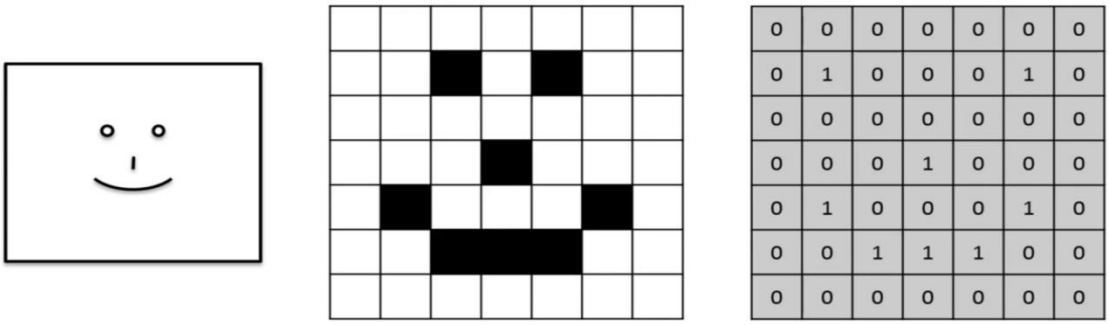

Fig.5.1. Convolutional of image

See figure 5.1. this is an example of simple black and white image the image on left side is an input image, when we apply convolution on this image we get image like middle image in above figure, let suppose the RGB value of white pixel is 0 and the RGB value of white pixel is 1, then what happen here is that where we found black pixel the value of that box become 1 and where we found black pixel the value of that box become 1 and after this we get matrix of input image for further processing.

The objective of a Conv layer is to extract features of the input volume. A part of the image is connected to the next Conv layer because if all the pixels of the input is connected to the Conv layer, It will be too computationally expensive. So we are going to apply dot products between a receptive field and a filter on all the dimensions. The

outcome of this operation is a single integer of the output volume (feature map**)**. Then we slide the filter over the next receptive field of the same input image by a Stride and compute again the dot products between the new receptive field and the same filter. We repeat this process until we go through the entire input image. The output is going to be the input for the next layer.

Filter, Kernel, or Feature Detector is a small matrix used for features detection. A typical filter on the first layer of a Convolutional Network might have a size [5x5x3].Convolved Feature**,** Activation Map or Feature Map is the output volume formed by sliding the filter over the image and computing the dot product. Receptive field is a local region of the input volume that has the same size as the filter. Depth is the number of filters. Depth column (or fiber) is the set of neurons that are all pointing to the same receptive field. Stride has the objective of producing smaller output volumes spatially. For example, if a stride=2, the filter will shift by the amount of 2 pixels as it convolves around the input volume. Normally, we set the stride in a way that the output volume is an integer and not a fraction. Common stride: 1 or 2 (Smaller strides work better in practice), uncommon stride: 3 or more. Zero-padding adds zeros around the outside of the input volume so that the convolutions end up with the same number of outputs as inputs. If we don't use padding the information at the borders will be lost after each Conv layer, which will reduce the size of the volumes as well as the performance.

The convolution operation extracts different features of the input. The first convolution layer extracts low-level features like edges, lines, and corners. Higher-level layers extract higher-level features. Figure 6 illustrates the process of 3D convolution used in cnns. The input is of size N x N x D and is convolved with H kernels, each of size k x k x D separately. Convolution of an input with one kernel produces one output feature, and with H kernels independently produces H features. Starting from top-left corner of the input, each kernel is moved from left to right, one element at a time. Once the top-right corner is reached, the kernel is moved one element in a downward direction, and again the kernel is moved from left to right, one element at a time. This process is repeated until the kernel reaches the bottom-right corner. For the case when N = 32 and k = 5 , there are 28 unique positions from left to right and 28 unique positions from top to

bottom that the kernel can take. Corresponding to these positions, each feature in the output will contain 28x28 (i.e., (N-k+1) x (N-k+1)) elements. For each position of the kernel in a sliding window process, k x k x D elements of input and k x k x D elements of kernel are element-by element multiplied and accumulated. So to create one element of one output feature, k x k x D multiply-accumulate operations are required.

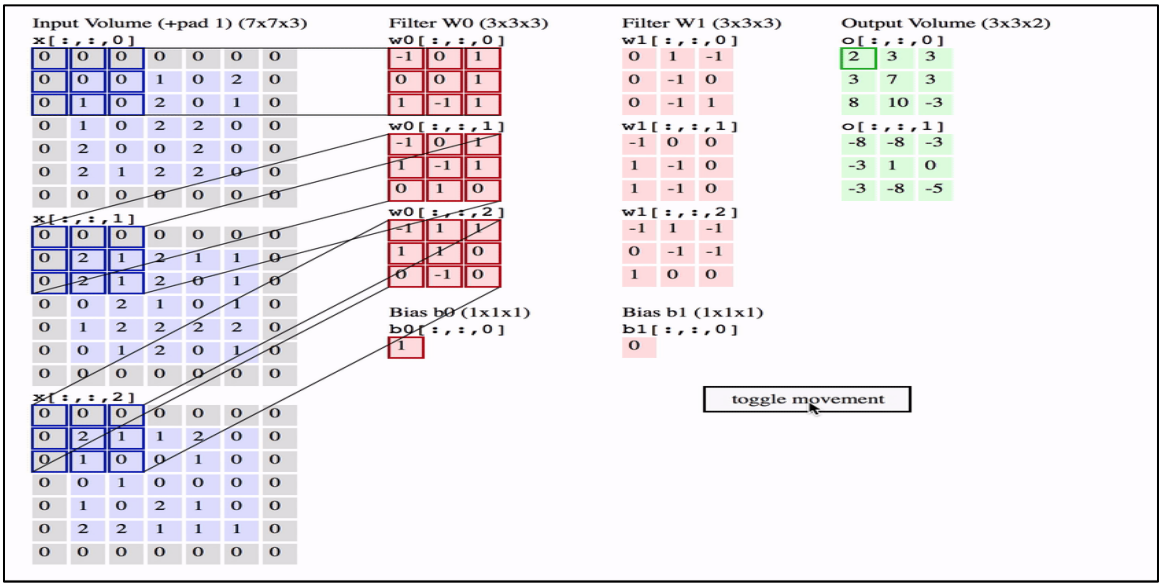

Fig.5.2. Convolutional layer

Example:-

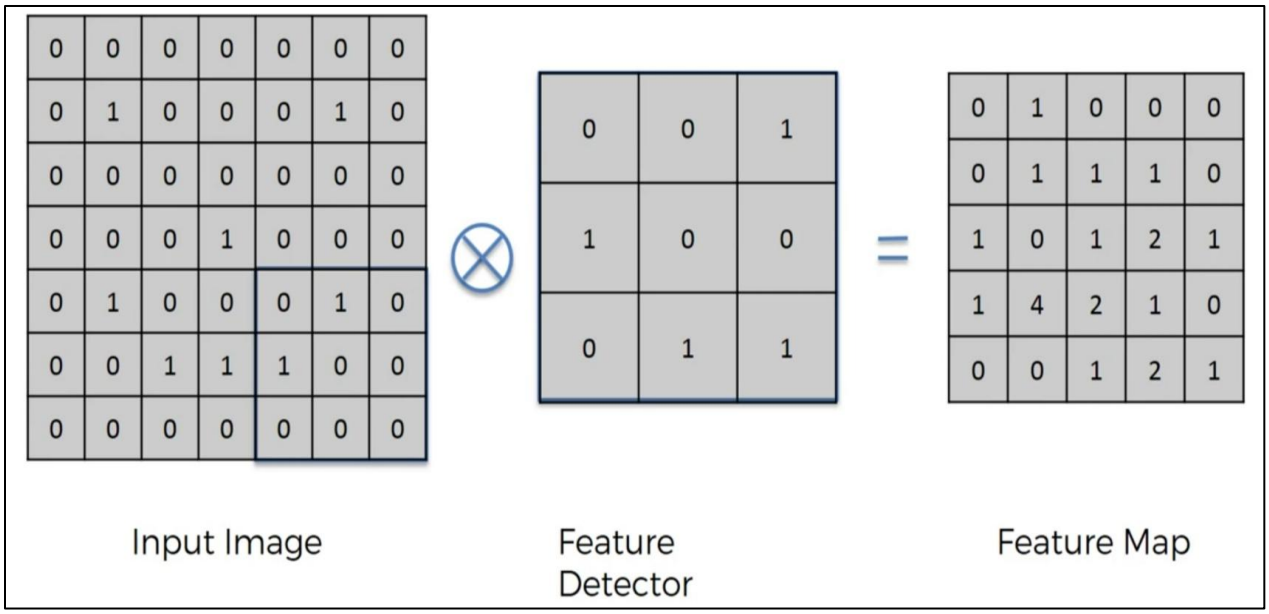

Fig.5.3. Convolutional layer Example

Here on left we have input image as we discussed that how we going to look images just in 0 and 1's format to simplify things and you can see smiley face there. Convolution operation is denoted by X, in a circle as you see in above figure then we featured detector feature detector is 3*3 matrix it doesn't have to be 3*3 it may have of any size like 7*7,5*5 featured detectors but usually they are 3*3 featured detectors ,featured detector is also called as filters .

So let see what operation is performed on this layer, take the featured detector or filter and put it on your image like you cover the 9 pixel in top left corner and basically multiply each value by each value basically position no 1by position no 1 ,position no 2 by position no 2 and so on so. Just it is element wise multiplication of the matrices then add up the result and record it then slide the featured detector on the image by selecting stride here we are selecting stride of 1 pixel and repeat the process.

So we created feature map in this layer feature map is also called as convolved feature. As you see here we reduces the image size and that is very important feature of feature detector in whole convolutional operation is to make image smaller because, it is easier to process smaller image as compared bigger image.

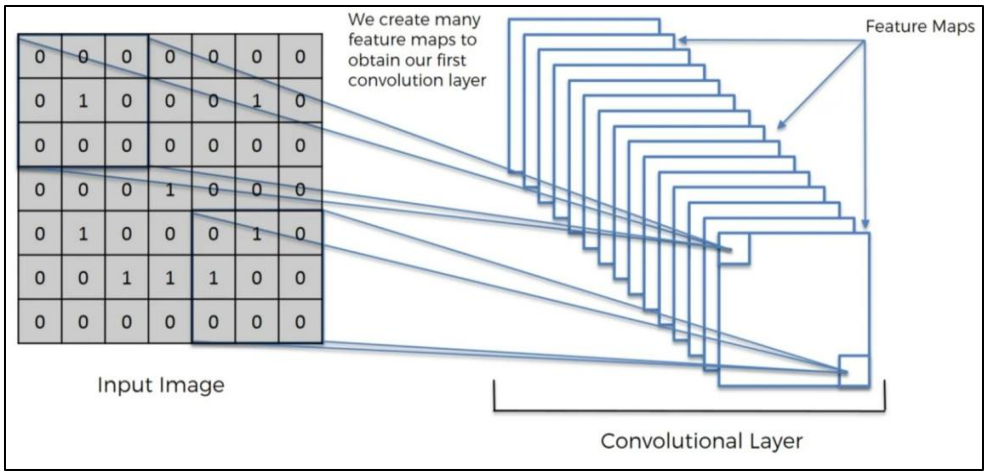

Fig.5.4. Feature map

We created feature map as you see in figure Let say front feature map we created but how come there is m any of them, but we created multiple featured maps because we use different features and that's another way that we preserve lots of information. So without having 1 feature map we look several features and then network decides through its training which features are more important for certain types and categorized it and look for it and therefore we have different filters.

5.2 ReLU LAYER

After convolution layer there is ReLU layer ReLU means rectifier linear unit. So we have our input image we convolutional layer which we discussed then we are going to apply rectifier function on it reason we want to apply rectifier function is that we want to increase non linearity in image and rectifier function act as a function or filter which breaks up linearity .Reason why we want to increase nonlinearity is that because images themselves contain highly nonlinear elements specially for recognizing different objects

We have input image when we apply any filter in that image, we get a image like middle image in above so you see here black is negative while white is positive. Here you see that image is highly linear so rectifier function does that it removes linearity from image that is remove all black to make it non linear .

Original image Linear image Non linear image

Fig.5.5. ReLU Layer

5.3 POOLING LAYER

In pooling there are several types of pooling like min pooling ,max pooling and average pooling here, we are applying max pooling.

In pooling we take smaller block from convolution layer. Subsample it to produce single output from that block max pooling layer takes maximum output from that block. The pooling/sub sampling layer reduces the resolution of the features. It makes the features robust against noise and distortion. There are two ways to do pooling: max pooling and average pooling. In both cases, the input is divided into non-overlapping two-dimensional spaces. For example, in Figure 4, layer 2 is the pooling layer. Each input feature is 28x28 and is divided into 14x14 regions of size 2x2. For average pooling, the average of the four values in the region are calculated. For max pooling, the maximum value of the four values is selected. Figure elaborates the pooling process further. The input is of size 4x4. For 2x2 sub sampling, a 4x4 image is divided into four non-overlapping matrices of size 2x2. In the case of max pooling, the maximum value of the

four values in the 2x2 matrix is the output. In case of average pooling, the average of the four values is the output. Please note that for the output with index (2,2), the result of averaging is a fraction that has been rounded to nearest integer

So let's see how as per figure 5.3.1 we have prepare feature maps now, we are going to apply max pooling on that. So take a box of 2*2 pixel it doesn't have to 2*2 it may of any size and place it on top left corner of feature map.

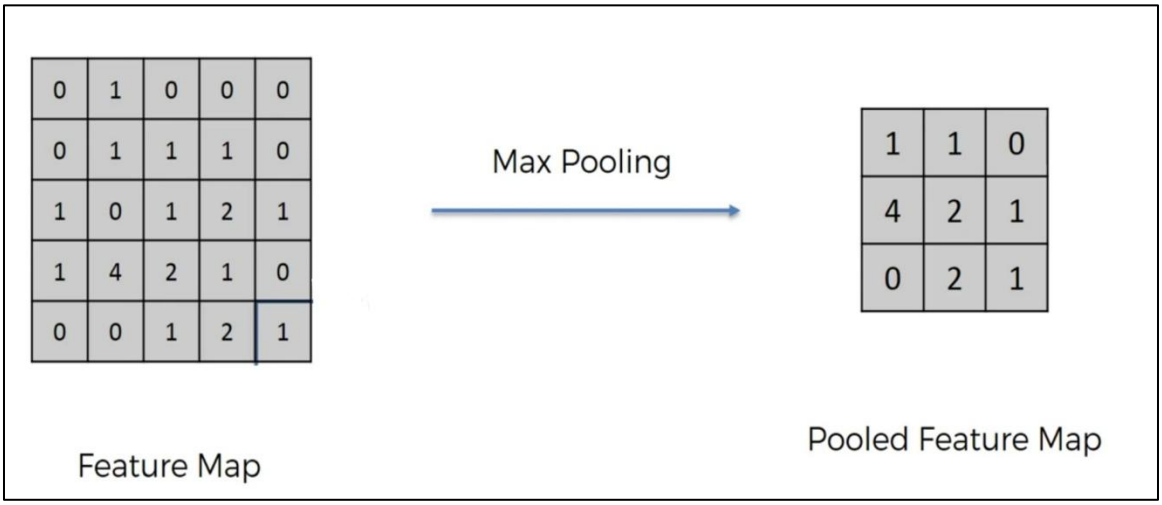

Fig.5.6. Max pooling

Then find the maximum value from that box and record it then move the box towards right side by selecting stride you can select stride of 1, 2,3 or any overlapping and then repeat the process

So as you can see we here also reduces image size but we are still able to preserve features. The maximum number represent features by polling features this features we get rid of 75% information, that is not features it is not important and another benefit of it is we reduces image size by 75% which really help us for processing.

5.4 FLATTENING

We have pooled feature map now, we are flatten them into column, basically take number row by row and put in column and reason for that is because, we want to later input that into artificial neural network for further processing

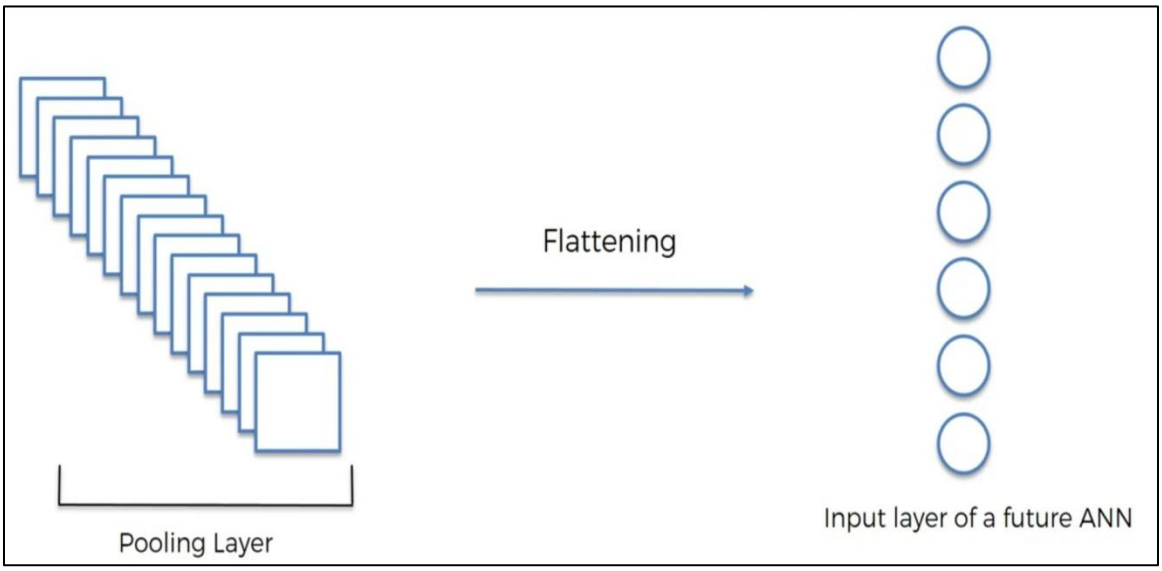

Fig.5.7. Flattening

Fig 5.7 shows. We have many pooling layer of many pooled future maps and then flattened them, so put them into the column sequentially one after other and make vector of input for artificial neural network.

5.5 FULL CONNECTION

In this layer we are adding whole artificial neural network to our convolutional neural network. So after flattening, we got input layer we pass it to artificial neural network, this layer in artificial neural network are also called hidden layers but here we are calling it fully connected layer because, they are hidden layer but at same time they are more specific type of hidden layers that they are fully connected.

In artificial neural network this hidden layers doesn't have to be fully connected layer but in convolutional neural network hidden layers are must be fully connected

.we have whole column or vector of input which we get after flattening we are passing it into the input layer of artificial neural network

The Main benefit of artificial neural network is to combine the features in to more attributes. That predict the classes even better. In fully connected layer every neuron is Connected to every neuron in another layer in neural layer. Each neuron receives input from some number of location in previous layer.

In fully connected layer each neuron receives input from every element of previous layer each neuron in a neural network compute an output value coming from previous layer called Weight.

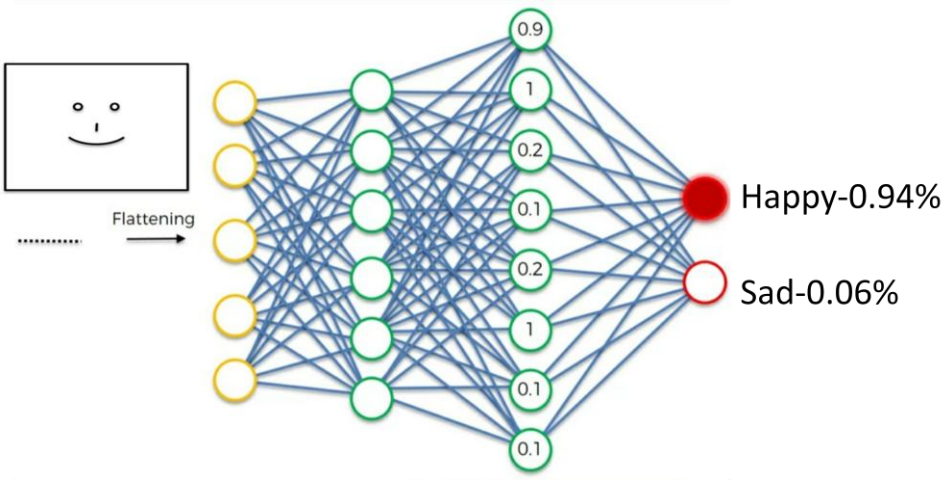

Fig.5.8. Full connection Example

Learning in neural network progresses by making incremental adjustment of weight. Weight is a output value computed by neuron and on the basis of maximum weight we get it detect what object it is. As you see in fig 5.8. maximum weight is shears to emotion happy and it detect the object.

6. APPLICATIONS

Convolutional Neural Networks (CNN) are everywhere. It is arguably the most popular deep learning architecture.CNN is also computationally efficient. It uses special convolution and pooling operations and performs parameter sharing.

6.1 IMAGE RECOGNITION

CNNs are often used in image recognition systems. In 2012 an error rate of 0.23 percent on the MNIST database was reported. Another paper on using CNN for image classification reported that the learning process was "surprisingly fast"; in the same paper, the best published results as of 2011 were achieved in the MNIST database and the NORB database.

When applied to facial recognition, CNNs achieved a large decrease in error rate. Another paper reported a 97.6 percent recognition rate on "5,600 still images of more than 10 subjects". CNNs were used to assess video quality in an objective way after manual training; the resulting system had a very low root mean square error.

The Image Net Large Scale Visual Recognition Challenge is a benchmark in object classification and detection, with millions of images and hundreds of object classes. In the ILSVRC 2014, a large-scale visual recognition challenge, almost every highly ranked team used CNN as their basic framework. The winner GoogLeNet (the foundation of Deep Dream) increased the mean average precision of object detection to 0.439329, and reduced classification error to 0.06656, the best result to date. Its network applied more than 30 layers. That performance of convolutional neural networks on the Image Net tests was close to that of humans. The best algorithms still struggle with objects that are small or thin, such as a small ant on a stem of a flower or a person holding a quill in their hand. They also have trouble with images that have been distorted with filters, an increasingly common phenomenon with modern digital cameras. By contrast, those kinds of images rarely trouble humans. Humans, however, tend to have trouble with other issues. For example, they are not good at classifying objects into fine-grained categories such as the particular breed of dog or species of bird, whereas convolutional neural networks handle this.

In 2015 a many-layered CNN demonstrated the ability to spot faces from a wide range of angles, including upside down, even when partially occluded, with competitive performance. The network was trained on a database of 200,000 images that included faces at various angles and orientations and a further 20 million images without faces. They used batches of 128 images over 50,000 iterations.

6.2. VIDEO ANALYSIS

Compared to image data domains, there is relatively little work on applying CNNs to video classification. Video is more complex than images since it has another (temporal) dimension. However, some extensions of CNNs into the video domain have been explored. One approach is to treat space and time as equivalent dimensions of the input and perform convolutions in both time and space. Another way is to fuse the features of two convolutional neural networks, one for the spatial and one for the temporal stream. LSTM units are typically incorporated.

Fig.6.1. Video analysis

After the CNN to account for inter-frame or inter-clip dependencies. Unsupervised learning schemes for training spatio-temporal features have been introduced, based on Convolutional Gated Restricted Boltzmann Machines and Independent Subspace Analysis.

6.3. NATURAL LANGUAGE PROCESSING

CNNs have also explored natural language processing. CNN models are effective for various NLP problems and achieved excellent results in semantic parsing, search query retrieval, sentence modeling, classification, prediction and other traditional NLP tasks.

6.4. HEALTH RISK ASSESSMENT AND BIOMARKERS OF AGING DISCOVERY

CNNs can be naturally tailored to analyze a sufficiently large collection of time series representing one week long human physical activity streams augmented by the rich clinical data (including the death register, as provided by, e.g., the NHANES study). A simple CNN was combined with Cox-Gompertz proportional hazards model and used to produce a proof-of-concept example of digital biomarkers of aging in the form of all-causes-mortality predictor.

6.5. FACE RECOGNITION

Face recognition constitutes a series of related problems-

- Identifying all the faces in the picture
- Focusing on each face despite bad lighting or different pose
- Identifying unique features

Comparing identified features to existing database and determining the person's name Faces represent a complex, multi-dimensional, visual Stimulus which was earlier presented using a hybrid neural Network combining local image sampling, a self-organizing Map neural network and a convolutional neural network. The results were presented using Karhunen-Loe`ve Transform in place of the self-organizing map which Performed almost as well (5.3% error versus 3.8%) and a Multi-layer perception which performed poorly (40% error Versus 3.8%) Face Recognition: Face recognition constitutes a Series of related problems

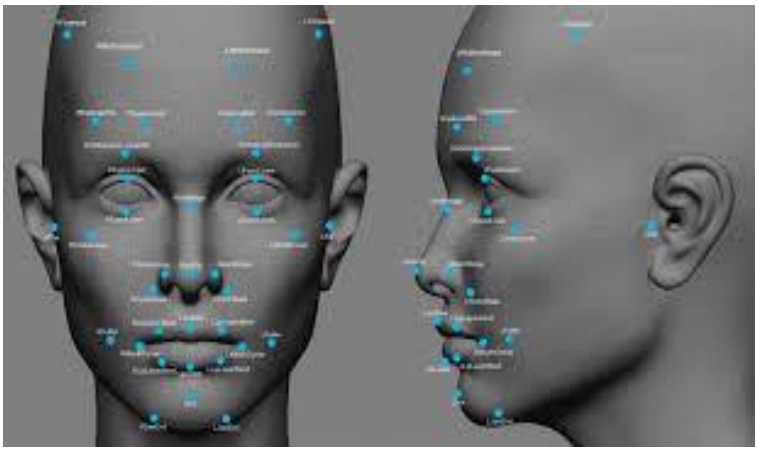

Fig.6.2. Face Recognition

6.6. SCENE LABELING:

Each pixel is labeled with the category of the object it belongs to in scene labeling. Clement Farebeat et al proposed a method using a multistage convolutional network that yielded record accuracies on the Sift Flow Dataset (33 classes) and the Barcelona Dataset (170 classes) and near-record accuracy on Stanford Background Dataset (8 classes). Their method produced 320 X 240 image labeling in under a second including feature extraction.

Recurrent architecture for convolutional neural network suggests a sequential series of networks sharing the same set of parameters. The network automatically learns to smooth its own predicted labels. As the context size increases with the built-in recurrence, the system identifies and corrects its own errors. A simple and scalable detection algorithm that improves mean average precision (map) by more than 30% relative to the previous best result on VOC 2012—achieving a map of 53.3% was suggested by researchers at UCB and ICSI. It was called as R-CNN: Regions with CNN features as it combined region proposals with CNN features.

7. ADVANTAGES

- The usage of CNNs are motivated by the fact that they can are able to learn relevant features from an image /video (sorry I don't know about speech / audio) at different levels similar to a human brain.

- Another main feature of CNNs is **weight sharing**. Let's take an example to explain this. Say you have a one layered CNN with 10 filters of size 5x5. Now you can simply calculate parameters of such a CNN, it would be 5*5*10 weights and 10 biases i.e. **5* 5*10 + 10 = 260 parameters**. Now let's take a simple one layered NN with 250 neurons, here the number of weight parameters depending on the size of images is '250 x K' where size of the image is P X M and K = (P *M). Additionally, you need 'M' biases. For the MNIST data as input to such a NN we will have (**250*784+1 = 19601) parameters.** Clearly, CNN is more efficient in terms of memory and complexity. Imagine NNs and CNNs with billions of neurons, then CNNs would be less complex and saves memory compared to the NN.

- In terms of performance, CNNs outperform NNs on conventional image recognition tasks and many other tasks. Look at the Inception model, Resnet50 and many others for instance.

- For a completely new task / problem CNNs are very good **feature extractors.** This means that you can extract useful attributes from an already trained CNN with its trained weights by feeding your data on each level and tune the CNN a bit for the specific task. Eg : Add a classifier after the last layer with labels specific to the task. This is also called **pre-training** and CNNs are very efficient in such tasks compared to NNs. Another advantage of this pre-training is we avoid training of CNN and save memory, time. The only thing you have to train is the classifier at the end for your labels.

- Convolutional Neural Networks take advantage of *local spatial coherence* in the input (often images), which allow them to have fewer weights as some parameters are shared. This process, taking the form of ***convolutions***, makes them especially well suited to extract relevant information at a low computational cost.

- You could try to use CNNs on data with no local coherence, such as shuffled images, but that would give poor results as the network wouldn't be able to identify spatial *patterns*.

8. DISADVANTAGES

- The scale of a net's weights (and of the weight updates) is very important for performance. When the features are of the same type (pixels, word counts, etc), this is not a problem. However, when the features are heterogeneous--like in many datasets--your weights and updates will all be on different scales (so you need to standardize your inputs in some way).

- They completely lose all their internal data about the pose and the orientation of the object and they route all the information to the same neurons that may not be able to deal with this kind of information.

- A CNN makes predictions by looking at an image and then checking to see if certain components are present in that image or not. If they are, then it classifies that image accordingly.

- In a CNN, all low-level details are sent to all the higher level neurons. These neurons then perform further convolutions to check whether certain features are present. This is done by striding the receptive field and then replicating the knowledge across all the different neurons

- It require large dataset

- A convolution is a significantly slower operation than, say maxpool, both forward and backward. If the network is pretty deep, each training step is going to take much longer.

- CNN completely loss internal data of object, because it look for only feathers.
- Checks for component only on basis of stored features.

6.CONCLUSION

Convolutional Neural Net is a popular deep learning technique for current visual recognition tasks. Like all deep learning techniques, CNN is very dependent on the size and quality of the training data. Given a well prepared dataset, CNNs are capable of surpassing humans at visual recognition tasks. However, they are still not robust to visual artifacts such as glare and noise, which humans are able to cope. The theory of CNN is still being developed and researchers are working to endow it with properties such as active attention and online memory, allowing CNNs to evaluate new items that are vastly different from what they were trained on. This better emulates the mammalian visual system, thus moving towards a smarter artificial visual recognition system.

Comparative study with other traditional methods suggest that CNN gives better accuracy and boosts the performance of the system due to unique features like shared weights.

CNN is better than other algorithm for applications pertaining to computer vision and natural language processing because it mitigates most of the traditional problems.

QUESTION-ANSWERS

Q.1. what is difference between ANN and CNN?

Answer: They are not different terms. That is, a convolutional neural network is a specialized artificial neural network.

Q.2. What is fully connected layer?

Answer: In a fully connected layer each neuron is connected to every neuron in the previous layer, and each connection has its own weight. This is a totally general purpose connection pattern and makes no assumptions about the features in the data. It's also very expensive in terms of memory (weights) and computation (connections).

Q.3. In which language we can implement it?

Answer: You can implement it in any language, but met labs is suitable because it consist inbuilt library of CNN.

Q.4.who can change the features?

Answer: Developer can change the features.

Q.5. What is filtering?

Answer: Filtering is a technique for modifying or enhancing an image. For example, you can filter an image to emphasize certain features or remove other features. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement.

Q.6. What is size of filter?

Answer: It may have of any size like 3*3, 5*5, 3*3 etc, but generally it is 3*3.

Q.7. On which basic value of matrix is decided for each pixel?

Answer: On the basic of RGB value of each pixel the value is selected.

Q.8.why convolutional neural network is better than other methods?

Answer: Because, convolutional neural network required less computational power because it focus on limited features only.

Q.9. What is that symbol in figure?

Answer: This is symbol of convolution operation,

Q.10. What is its response time?

Answer: It depends on which type of input is it/picture quality of input.

REFERENCES

[1] Y. LeCun, O. Matan, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, H. S. Baird. *Handwritten zip code recognition with multilayer networks*. Proceedings of the International Conference on Pattern Recognition, pages 35 - 40, Atlantic City, 1990.

[2] David Stutz. *Understanding Convolutional Neural Networks*. Seminar on *Current Topics in Computer Vision and Machine Learning*, 2014

[3] **https://deeplearning4j.org**

[4] **https://www.researchgate.net**

[5] **https://www.coursera.org**

[6] **https://towardsdatascience.com**

[7] **http://scs.ryerson.ca/~aharley/vis/conv/flat.html**