A Project Report

on

OBJECT DETECTION USING OPENCV AND DEEP LEARNING

Submitted in partial fulfillment of the
requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

Under The Supervision of
Dr. Sanjay Kumar
Professor
Department of Computer Science and Engineering

Submitted By

TUSHAR - 18SCSE1010487
Kundan Kumar - 18SCSE1010236

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA
MAY, 2022

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

1. CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project entitled **"Object Detection using OpenCV and Deep Learning"** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** submitted in the **School of Computing Science and Engineering** of Galgotias University, Greater Noida, is an original work carried out during the period of **JAN-2022** to **MAY-2022**, under the supervision of **Dr. Sanjay Kumar, Professor**, **Department of Computer Science and Engineering** of School of Computing Science and Engineering, Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

TUSHAR - 18SCSE101487

KUNDAN KUMAR - 18SCSE1010236

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Sanjay Kumar

Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **TUSHAR - 18SCSE1010487** and **KUNDAN KUMAR - 18SCSE1010236** has been held on **MAY 13, 2022** and their work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING.**

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: May 13, 2022

Place: Galgotias University, Greater Noida, U.P.

Abstract

The most often adopted methodologies for contemporary machine learning techniques to execute a variety of responsibilities on embedded devices are mobile networks and multimodal neural networks. In this research, we propose a method for identifying an item that takes into account the learning - based pre-trained system MobileNetV3 for an SSD (Single Shot Multibox Detector). This set of rules is utilized for constant identification, as well as for camera broadcasts to find an explanation camera which recognizes the thing for a clip transfer. Consequently, I utilize an item identification module which could find what's withinside the video transfer. To place into impact the module, we integrate the MobileNetV3 and the SSD system for a quick and green profound getting to know-primarily based totally approach of item detection. The primary goal of our study is to investigate the effectiveness of an object identification technique SSD and the significance of MobileNetV3. The testing findings demonstrate that the Average Closeness of the set of rules to discover one of a kind training as car, person and cat is 97.45%, 95.67% and 85.59%, respectively. It enhances the accuracy of content detection at the processing speed necessary for practical identification and the needs of regular progress indoors and outdoors.

Keywords : MobileNetV3, Computer vision, SSD(Single Shot Multibox Detector), OpenCV, Deep Learning Neural Network.

Table of Contents

List of Figures

CHAPTER 1

INTRODUCTION

Project Purpose:

The goal of object identification would be to recognise and locate any recognized items in a scenario. For mechanical control frameworks, recovering the posture of items in 3D space is ideal.

Providing machines intelligence and turning bots into a growing bunch of single and liberated humans has been a mechanical ambition for humanity. It is our ambition to empower robots to perform repetitive, tedious,or hazardous tasks in order for us to devote. Now is the moment for us to be more inventive. pursuits. Regrettably, the smart component appears to be trailing behind. In reality, we need programming that allows robots to autonomously complete tasks and act to achieve this goal, in addition to better equipment.

In a continuous circumstance, the gazing or acknowledgment procedure is incredibly difficult. Until now, no effective solution to this problem has been identified. Despite extensive research, the solutions developed thus far are ineffective, take a long time to prepare, are not suitable for continued use, and are not adaptable to a large number of classes. When the system is looking for a specific item to identify, object recognition is a little easier. Nonetheless, recognising all of the items necessitates the ability to distinguish one item from another, even if they are of the same type. Such a problem is extremely difficult for machines, especially when they are unfamiliar with the various possible outcomes.

Motivation:

Blind folks have their own way of doing things and lead normal lives. They do, however, confront difficulties due to inaccessible infrastructure and social issues. Navigating around locations is the most difficult task for a blind person, especially one who has lost all eyesight. Obviously, blind persons can navigate their homes without assistance because they are familiar with their surroundings. Blind persons have a difficult difficulty locating objects in their environment. As a result, we decided to create an OBJECT DETECTION SYSTEM IN REAL TIME. After reading a few publications in this field, This concept appeals to us. Consequently, we are extremely inspired to create a method in which we can recognise in real time objects.

CHAPTER 2

LITERATURE SURVEY

2.1 MobileNet-SSD

The MobileNet-SSD structure is used in our suggested model. It provides high item recognition rate and still being faster than other systems such as YOLO. This is true when trying to identify things in real time on low-powered devices like ours. By addressing the model using 8-bit integers rather than 32-bit floats, MobileNet-SSD allows for a faster detection time. The model's input was a 300 by 300 pixel picture, while the model's output addressed the bounding box's position as well as recognition confidentiality (between Zero to one) for every recognised item. To determine if the discovered item was real, a recognition certainty level of 0.5 was used.

2.2 OpenCV (Open-Source computer vision)

OpenCV is a programming library that is primarily focused on real-time computer vision. OpenCV is an open-source computer vision library that can be used for CCTV film analysis, video analysis, and picture analysis. It's a fantastic programme for image processing and computer vision jobs. OpenCV is a C++ library that contains over 2,500 efficient algorithms. [5] When we want to create computer vision apps but don't want to start from scratch, we can use this library to start focusing on real-world challenges. The cv2.VideoCapture function in OpenCV can read video (). By supplying 0 as a function parameter, we may access the webcam.

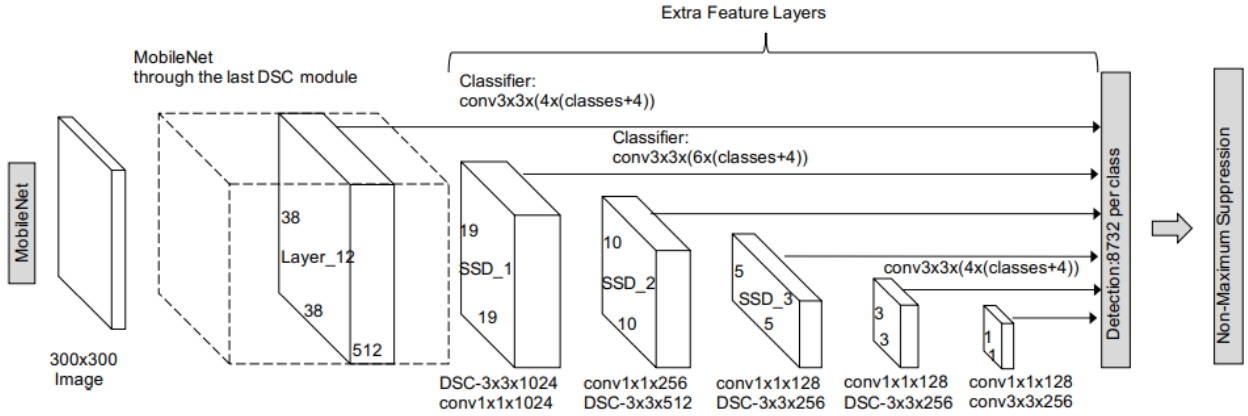

Fig: 2.1 MobileNet as a backbone for SSD-based detection.

CHAPTER 3

WORKING OF A PROJECT

3.1 Algorithms Development

Using the OpenCV library and deep learning pre-trained models, we attempt to recognise objects. It's almost like face recognition in real time. First, we train the system with known or facial references so that if that face is visible in either photos or clip feeds, It will be recognised by the system. We cannot forecast the quantity of objects or the items such as a car, people, or cats in this study, which is about object detection. If we have photos of a car to train a system with, the system will be able to anticipate these objects from the image or video. However, because there are so many objects around us, it is not realistic. Some pre-trained models were relayed. This project is made by Tushar only. Some third-party person has trained these pre-trained models. In these models, the majority of the items have already been pre-trained. Finally, the SSD technique allows System to recognise objects using pre-trained models. To implement the SSD approach in Python code, we employ pre-trained MobileNets. Using data for training and a collection of individual class bounding box colors, this model can identify labels. Resize each frame to a fixed scale (300x300) pixels and To make an input drop for a single frame, load the input video frame by frame.
The MobileNet approach is used to continuously improve the SSD algorithm and speed rating accuracy. To detect many objects, this method requires only one shot. For detecting purposes, the SSD is a neural network architecture design. The SSD methodology disconnects the bounding output space into a series of default boxes spanning distinct scales and fact ratios, whereas other methods such as the R-CNN series require two shots. The default box set's forbidden output space is revealed by SSD. The network quickly searches a default box for distinct object classes and combines the box to fit what's inside. This network also accommodates a variety of models with varying sizes of natural adhesives and resolutions. If no object is present, the background is used instead of the location.

3.2 There are three steps to detection

1. To load a pre-trained object identification network, use OpenCV's deep neural network (DNN) module.
2. Compute the forward pass for the input from a set of inputs to the network. The result is saved as Detections.
3. Then iterate through the Detection to figure out what and where the items in the photos are.

Aside from that, the location where a certain In the frame is an object, an individual and the location, and finally the detection accuracy. The pre-trained model already has a Person image and the label for that particular detection. A

label that refers to a person (see Figure 6) is followed by the label's accuracy. So these three processes were constrained and used a loop to create a bounding box around that specific object in the frame.

3.2.1 Model data

Our pre-trained models have two files: one for configuration and the other for weights. As a result, the model represents how neurons are organized in a neural network.
1- Configuration
2- Weight

3.3 PROPOSED SYSTEM

We will detect objects in real time using the Mobilenet-SSD model in the proposed system in a quick and efficient manner. We'll use OpenCV 3.4 to construct a Python script for object detection using deep neural networks.

The system operates as follows:

Input will be provided via real-time video from a camera or webcam, using the simplified MobileNet Architecture, which builds light weight deep neural networks using depth-wise separable convolutions. MobileNet layers receive the input video, which is separated into frames. Each feature value is calculated by subtracting the amount of pixel intensity in the bright region from the amount of pixel intensity in the dark region. To compute these elements, all of the image's available sizes and regions are used. An image may have few important qualities that can be utilized to detect the item and many irrelevant features.
The Layers of MobileNet job will be to convert pixels in the fill in the highlights that characterize the image's material. The bounding boxes and related class (label) of items are then determined using the MobileNet-SSD model. The only remaining step is to present or display the output.

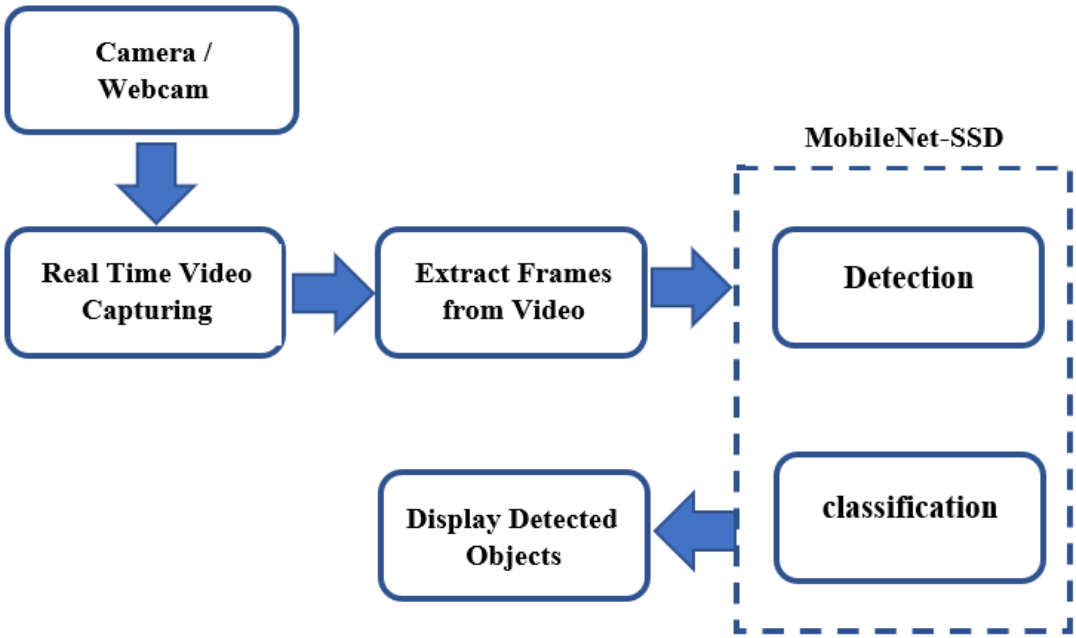

Fig: 3.1 Diagram of the Proposed System Architecture

CHAPTER 4

MODULE DESCRIPTION

4.1 OBJECT DETECTION

INTRODUCTION TO OBJECT DETECTION

The technique of extracting and detecting real-world item instances from images or videos, such as automobiles, motorcycles, televisions, flowers, and people, is known as object detection. Because it allows for the recognition, localisation, and detection of many objects within an image, an object detection technique allows you to grasp the nuances of an image or video.
Image retrieval, security, surveillance, and advanced driver assistance systems are all examples of areas where it is used (ADAS).

Object detection can be accomplished in a variety of ways:

- Detection of Objects Using Features
- Jones, Viola Object Recognition
- HOG Features in SVM Classifications
- Object detection with deep learning

Object recognition from video is an important task in video surveillance applications these days. The object detection approach is used to find and cluster pixels of needed objects in video sequences.
The detection of an item in a video sequence is critical in a variety of applications, including video surveillance.
Pre-processing, segmentation, foreground and background extraction, and feature extraction can all be used to find objects in a video stream.
Humans are quite good at detecting and identifying items in images. The visual processing system is quick and precise, and it can handle complicated activities such as detecting many items with a minimum of effort.

PROCESSING OF DIGITAL IMAGES

The need for extensive test work to build up the practicality of offered answers for a given issue characterizes computerized picture production. The massive amount of testing and trial that is typically necessary before touching base at a suitable arrangement is a key trademark hidden in the design of image preparation frameworks. The ability to plan approaches and fast model hopeful arrangements,

according to this trademark, has a significant role in reducing the cost and time commitment to arrive at suitable framework implementation.

WHAT EXACTLY IS DIP?

An image is described as a 2D potential f(x, y), whereas x and y are spatially coordinates and the dark level of the picture is determined by the adequacy of any combination of directions (x, y). A computerized picture is one in which x, y, and the abundance estimation of are all constrained discrete amounts. DIP refers to the process of creating enhanced images for digital computers. A complex image is made up of a small number of components, each of which has its own area and value. Pixels are the units of measurement.

Because The most evolved of our senses is eyesight, it's no surprise that The most significant part of human observation is visuals. Unlike people, though, imaging equipment that is confined to the visual band of the electromagnetic spectrum spans the entire electromagnetic spectrum, gamma rays to radio waves. It could also run with pictures created based on sources with which most people are unfamiliar.

There is no widespread agreement among designers about where picture handling ends and other linked areas, such as picture analysis and PC vision, begin. Occasionally, a distinction is established by describing picture handling as a teacher in which both the information and the output of an operation are images. This is a restricting and, to some extent, artificial limit. Picture inquiry falls somewhere between picture preparation and PC vision.
In the continuum from picture handling to finish vision on the opposite side, there are no visible bounds. In any event, considering three types of automated procedures in this continuum: low, mid, and abnormal state forms, is one helpful perspective. Primitive operations are included in low-level processes, for instance, image preparation to reduce commotion, differentiation update and image refinement. The fact that both its information sources and outputs are pictures defines a low-level process.

Assignments, for example, division, depiction of that Question, and characterization of individual pieces are part of the mid-level process on pictures.
The key information inputs to a mid-level process are visuals, words, and numbers, However, its outputs are unique from those images. Ultimately, more precise quantity handling involves "understanding an outlet of sensed things, such as in picture research, and engaging the intellectual skills normally related to human

sight at the extreme end of the continuum." As previously stated, modified image handling is used efficientlyin a broad spectrum of fields with high social and financial value.

WHAT EXACTLY IS AN IMAGE?

An image is described as a 2D potential f(x, y), whereas x and y are spatially coordinates and the dark level of the picture is determined by the adequacy of any combination of directions (x, y).

Fig. 4.1.1 Digital Photograph

Image manipulation:

Processing on photos can be divided into three categories. There are three levels: low, mid, and high.

1. **Processing at the Lowest level :**
    - Noise removal by preprocessing.
    - Enhancement of contrast.
    - Sharpening the image

2. **Processing at the Medium level::**
    - Categorization.
    - Detecting the edges
    - Extraction of objects

3. **Processing at the High level:**
    - Image evaluation
    - Interpreting the scene

Why are images processed?

The digital image must be prepared for display on one or more output devices because it is invisible (laser printer, monitor at). The appearance of the structures inside the digital image might be improved to make it more suitable for the application.

Image processing is divided into three categories. They are

- Image to Image conversion
- Transformations from image to data
- Transformation of data into images.

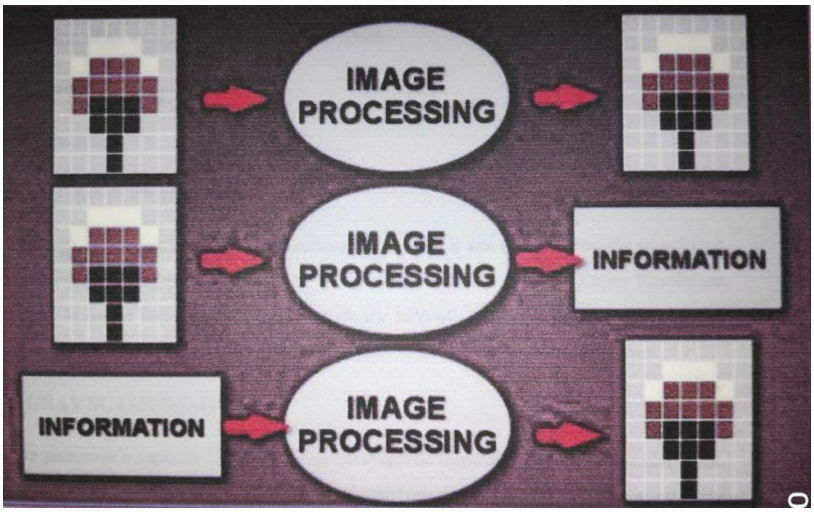

Fig. 4.1.2 Image Processing Types

Pixel :

The smallest constituent of an image is the pixel. Each pixel represents a single value. The value of a pixel in an 8-bit grayscale image is between 0 and 255. Each pixel holds a value proportionate to the intensity of light at that specific spot. The units of measurement are pixels per inch or dots per inch..

Resolution :

Resolution is described in a variety of ways. These include pixel density, positional accuracy, high temporal, and spectral information.. The total number of counts of pixels in a digital image is referred to as resolution in pixel resolution. The resolution of a picture with M rows and N columns, for example, is formmulated as M X N. The greater the image quality, the higher the pixel resolution.

In general, there are two types of image resolution:

- Image with low resolution
- Image with high resolution

Because high resolution is an expensive technique, It's not always possible to get high-resolution photos at a reasonable price. As a result, imaging is desirable. With the help of certain methods and algorithms, we may make high resolution photographs from low resolution images in Super Resolution imaging.

IMAGE IN GRAYSCALE

A grayscale image is a capacity I (xylem) of the picture plane's two spatial directions. At the location (x, y) on the image plane, the picture force is I. (x,y). Unless the image is restricted by a rectangle, I (xylem) take non-negative images.

IMAGE IN COLOR

Three capacities can communicate with it: The letters R, G, and B stand for red, green, and blue, respectively. An image can perseverance in terms of x, y facilitations, as well as in terms of adequacy. The instructions and adequacy must be digitized in order to convert such an image to advanced shape. Inspecting is the process of digitizing the facilitator's esteems. Quantization is the process of digitizing adequate esteems.

TECHNOLOGY IN CONNECTION:

R-CNN

R-CNN is a significant indication object identification system that combines bottom-up region suggestions with top-down region recommendations. with convolution neural network-generated rich alternatives.

R-CNN makes an advantage of region proposals methods to create possible boundary boxes in a photograph, which are then used to train a classifier.

SINGLE SIZE MULTIBOX DETECTOR

SSD uses different aspect ratios and scales to discretize the output space of bounding boxes into a collection of default boxes per feature map point. At the time of prediction, the network generates scores for the presence of each item type in each default box, as well as adjustments to the box to better reflect the object form.

The network integrates predictions from many feature maps with varying resolutions to handle objects of various sizes naturally.

ALEXNET

AlexNet is a classification convolutional neural network that contains 5 convolutional layers, 3 fully connected layers, and 1 softmax layer with 1000 outputs as its architecture.

YOLO

"You only live once," as the acronym suggests. Using a single neural network, it separates the picture into regions and predicts bounding boxes and possibilities for each region.

These bounding boxes are weighted based on projected probability. A single neural network predicts bounding boxes and class possibilities from complete pictures in a single assessment. Because the entire detection pipeline is a single network, it can be tuned from start to finish based on detection performance.

VGG

VGG network is another image categorization convolution neural network architecture.

MOBILENETS

Mobilenets are used to create lightweight deep neural networks. It uses depth-wise separable convolutions and is based on a streamlined design. MobileNet employs 33 depth-wise separable convolutions, which require up to 8 times less processing than ordinary convolution while achieving just a minor drop in precision. Object identification, fine grained categorization, facial characteristics, and large scale-localization are some of the applications and use cases.

TENSOR FLOW

Tensor flow is a high-performance numerical computation open source software package. Its adaptable design enables easy computing deployment across a variety of platforms (CPUs, GPUs, TPUs), from personal computers to server clusters to mobile and edge devices. Google Brain team academics and engineers at Google's AI company invented and produced Tensor flow. The adaptable numerical computation core is utilized across a range of scientific domains, and it features robust machine learning and deep learning support.

TensorFlow is used to build, train, and deploy Object Detection Models because it is simple and provides a library on the use of pre-trained detection models the The

Open Images dataset, the COCO dataset, and the Kitti dataset. The combination of Single Shot Detectors (SSDs) with Mobile Nets architecture is one of the many Detection Models that is rapid, efficient, and does not require a lot of computer power to achieve object detection.

OBJECT DETECTION IN PRACTICE

Object detection's main applications include :

FACIAL IDENTIFICATION

Deep Face is a facial recognition system that uses deep learning to recognise human faces in digital images. A group of Facebook researchers designed and built the app. Google Photographs also has its own facial recognition technology, which automatically divides all photos based on who is in the frame.
Facial Recognition involves several components, or one might say it concentrates on several characteristics such as the eyes, nose, mouth, and brows for recognising faces.

COUNTING PEOPLE

People counting is a type of object detection that can be used for a wide range of uses, including finding a person or a criminal, monitoring store performance, and computing crowd estimates during festivals. Because individuals move out of the picture fast, this is a challenging process.

CHECK FOR INDUSTRIAL QUALITY

Object detection is also useful in industrial processes for identifying and recognising products. Finding a specific article through visual examination is a common task in many industrial processes, including sorting, inventory management, machining, quality control, and packaging. Inventory management is difficult since goods are difficult to track in real time. Inventory accuracy can be improved by using automatic object counting and localisation.

CARS THAT DRIVE THEMSELVES

Self-driving cars are the most promising technology for the future, but their operation is complicated sinceRadar, laser light, GPS, odometer, and computer vision are some of the tools they utilize to sense their environment. In order for navigation methods, as well as obstacles and it, to operate, advanced control systems analyze sensory inputs. It is a significant step toward driverless cars because it occurs at a rapid rate.

SECURITY

Item Recognition is important in the field of security; it's employed in significant fields like Apple's Face ID and the retina scan seen in all sci-fi movies. Governments frequently utilize this application to access security feeds and compare them to their existing databases in order to identify criminals or detect things such as car numbers implicated in criminal activity. The possibilities are endless.

DETECTION OF OBJECTS EXTRACTION OF FEATURES AND WORKFLOW

Every Object Detection Algorithm works on the same concept; the only difference is how they function. They focus on extracting characteristics from photographs. that are provided as input, and then using these charatcterstics to establish the image's class.

4.2 DEEP LEARNING

INTRODUCTION TO DEEP LEARNING

Deep learning is a machine learning technique. It instructs a computer to learn how to predict and classify information by filtering inputs through layers. Images, writing, and music can all be used to express observations. Deep learning was inspired by the way the human brain filters information. Its goal is to generate some actual magic by simulating how the human brain functions. There are around 100 billion neurons in the human brain. Each neuron has approximately 100,000 neighbors. We're re-creating it, nevertheless, in a way and on a level that robots cannot understand. In human brains, a neuron has a body, dendrites, and an axon. A neuron's signal travels down the axon to the dendrites of the next neuron. A synapse is the connection through which the signal travels. Neurons are somewhat useless on their own. When there are a lot of them, however, they may create some major magic. That's how a deep learning algorithm works! You collect data from observations and combine it into a single layer. That layer generates an output, which is then used as an input by the next layer, and so on. This process continues till your ultimate output signal is reached! The neuron (node) receives one or more signals (input values) that travel across it. The output signal is delivered by that neuron.

Consider your senses as the input layer: what you can see, smell, and feel. For a single observation, These are factors that are unrelated. This data is broken down into integers and binary bits that a computer can understand. To bring these variables into the same range, you'll need to standardize or normalize them. For feature extraction and transformation, they employ multiple layers of nonlinear processing units. The output of the previous layer is used as the input for the next layer. What they learn is organized into a hierarchy of ideas. Each level of this structure learns to adapt its input data into increasingly composite and abstract representations. This signifies the input of a picture, eg., might be a pixel matrix. The initial layer may encrypt the borders and pixel composition. The following layer could be a configuration of lines. A nose and eyes could be encoded in the next layer. The next layer may detect the presence of a face in the image, and so on.

What goes on within a neuron?

The input node receives data in numerical format. Each node is allocated a number, and the data is delivered as an activation value. The greater the number, the stronger the activation. Based on the connection strength (weights) and transfer function, the activation value is sent to the next node. Every node adds up activation values it gets (calculates the weighted sum) and adjusts it according to its transfer function. It then performs an activation function. A function that is applied to this specific neuron is called an activation function. The neuron then determines whether or not it needs to transmit a signal.

Weights are allocated to each synapse, which are vital in Artificial Neural Networks (ANNs). ANNs learn through weights. The ANN determines how far signals are carried along by altering the weights. You decide how the weights are modified when you're training your network.

The activation propagates throughout until it reaches the output nodes of the network. The information is subsequently presented to us in an intelligible manner via the output nodes. To compare the output with the actual expected output, your network will employ a cost function. The cost function is used to assess the model's performance. It is calculated as the difference between the actual and projected values.

You can use a variety of cost functions to determine what kind of network fault do you have? You are attempting to reduce the function of loss. (In other words, the lower the loss function, the closer you are to the desired output.) The data is returned, and the neural network proceeds to study with a purpose of cost function reduction by adjusting scales. Backpropagation is the term for this procedure.

In forward propagation, information is fed into the input layer and transmitted forward across the network to create our output values. We compare the values to what we predicted. We next calculate the errors and reverse the information. This allows us to update the weights and train the network. (We can alter all the weights at the same time via backpropagation.) Because of the way the algorithm is set up, you can alter all of the weights at the same time during this procedure. This helps to examine how each of your neural network weights contributes to the overall error.

You're ready to go on to the testing process once you've adjusted the weights to the ideal level.

What is the learning process of an artificial neural network?

There are two methods for getting a programme to do what you want. The first is a method that is carefully led and pre-programmed. You tell the application exactly what you want it to do. There are also fully connected layers to take into account. You tell your neural network which inputs to utilize and which outputs you want, and then you sit back and watch it learn on its own.

enabling our system to figure things out for itself eliminates the need to manually enter each and every rule. We can design the structure and after that learn from it. After it has been trained, you may feed it a new image and it will be able to distinguish between input and output.

Feedback and feedforward networks

Inputs, outputs, and hidden layers are all part of a feedforward network. The signals are only capable of traveling in one way (forward). Input data is passed to a layer that performs calculations. The weighted total of each processing element's inputs is used to compute. The new values are fed into the next layer as new input values (feed-forward). This is repeated throughout all layers and determines the final result. Data mining, for example, frequently employs feedforward networks.
There are feedback channels in a feedback network (such as a recurrent neural network). This means that they can use loops to send messages in both ways. All conceivable neuron connections are permitted. Because this form of network has loops, it creates a dynamic system that is non-linear that alters continually till the equilibrium is reached In optimization issues, evaluations are frequently used to find the optimal arrangement of interconnected components.

Weighted Average
A neuron's inputs might be either features from a training set or outputs from previous layer neurons. Each link between two neurons has its own synapse, which has its own weight. You must travel along the synapse and pay the "toll" to move from one neuron to the next (weight). The sum of the weighted inputs from each incoming synapse is then applied to an activation function by the neuron. It sends the result to all of the neurons in the following layer. We're talking about altering the weights on these synapses when we talk about updating weights in a network.

The sum of the weighted outputs from all the neurons in the previous layer is a neuron's input. The weight associated with the synapse connecting the input to the current neuron is multiplied by each input. Each neuron in the current layer will have three separate weights: one for each synapse, if the prior layer has three inputs or neurons.
The output of a node is determined by its activation function, in a nutshell.
Input signals are converted into output signals via the activation function (also known as the transfer function). It converts the output numbers to a range between 0 and 1 or -1 and 1. It's a representation of the cell's action potential firing rate as an abstraction. It's a number that indicates the probability of the cell firing. The function is binary at its most basic level: yes or no (the neuron fires) (the neuron does not).The output can be a number between 0 and 1 (on/off or yes/no) or any value in between. If you're estimating the chances that a picture is a cat using a method that converts a range between 0 and 1, an output of 0.9 means your image has a 90% chance of being a cat.

Activation feature
In a nutshell, a node's activation function determines the node's output.
Input signals are converted into output signals via the activation function (also known as the transfer function). It converts the output values into a 0 to 1 or -1 to 1 range. It's a simplified representation of the cell's action potential firing rate. It's a number that represents the probability of the cell firing. The function is binary at its most basic level: yes (the neuron fires) or no (the neuron does not fire). The output can be a single digit or a range of digits.
What are our options? There are several activation functions, but these four are the most commonly used.

Function of Threshold
A step function is what this is. If the input's total value exceeds a specific threshold, the function returns 0. It will pass on 1 if it is equal to or greater than zero. It's a yes-or-no function with very strict parameters.

Function sigmoid
In logistic regression, this function is utilized. It's a smooth, progressive development from 0 to 1, unlike the threshold function. It's useful at the output layer, and it's commonly employed in linear regression.

Tangent Hyperbolic Function
The sigmoid function is extremely close to this one. The value spans from -1 to 1 in contrast to the sigmoid function, which varies from 0 to 1. Even though this

function is not very close to what occurs in the brain, it gives better results when neural networks are trained. During sigmoid function training, neural networks can become "stuck." This occurs when a large amount of strongly negative input maintains the output near zero, causing the learning process to be disrupted.

Rectifier feature
In the world of neural networks, this may be the most prevalent activation function. It's the most effective and biologically sound option. Despite the fact that there is a kink at 0, it is slick and steady afterwards. For instance, your result may be "no" or a percentage of "yes." This function doesn't require any difficult computations or normalization.

When robots can execute jobs that would ordinarily need human intellect, artificial intelligence is required. It falls inside the machine learning layer, where robots may learn from past experience and develop skills without the involvement of humans. Artificial neural networks, algorithms inspired by the human brain, are used in deep learning to learn from vast volumes of data. Deep learning is built on human experiences; the deep learning algorithm will repeatedly do a task in order to improve the outcome. Learning is enabled by the many (deep) layers of neural networks. Any flaw that requires "thinking" to solve It's possible that deep learning can learn to fix this problem.

4.3 CONVOLUTIONAL NEURAL NETWORK

INTRODUCTION TO CONVOLUTIONAL NEURAL NETWORKS (CNN)

Neural Networks (Artificial)

The concept of artificial neural networks (ANNs) is founded on the premise that the human brain's functioning may be mimicked by employing silicon and wires as living neurons and dendrites.

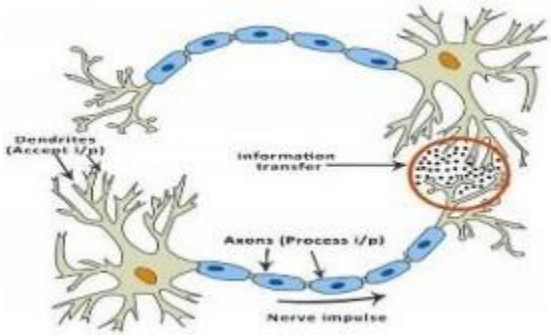

Fig. 4.3.1 Neurons

Neurons are 86 billion nerve cells that make up the human brain. Axons connect them to thousands of other cells. Dendrites accept stimuli from the outside world as well as sensory organ inputs. Electric impulses are generated by these inputs and travel fast across the neural network. The message can then be forwarded to another neuron to handle the problem, or it can be ignored.

ANNs are built up of many nodes that mimic brain neurons. The neurons are connected and communicate with one another. The nodes may take in data and perform simple actions on it. Other neurons get the outcome of these operations. The activation or node value of each node is the output. Each link has a weight connected with it. Learning is possible with ANNs, which is accomplished by changing weight values.

Neural network :

In a more contemporary definition, a neural network is a network or circuit of neurons, or an artificial neural network made up of artificial neurons or nodes. As a result, a neural network can be either biological (made up of real biological neurons) or artificial (made up of artificial neurons). The biological neuron's connections are modeled as weights. An excitatory link has a positive weight, while inhibitory connections have a negative weight. All inputs are given a weight and then added together. A linear combination is the name for this action. Finally, the output's amplitude is controlled by an activation function. For instance, an acceptable output range is normally between zero and one, although it might also be between -1 and 1.

These artificial networks might be utilized in predictive modeling, adaptive control, and other applications that need a training dataset. Self-learning based on experience can take place neural networks, which can derive conclusions from a complex and apparently disconnected set of data.

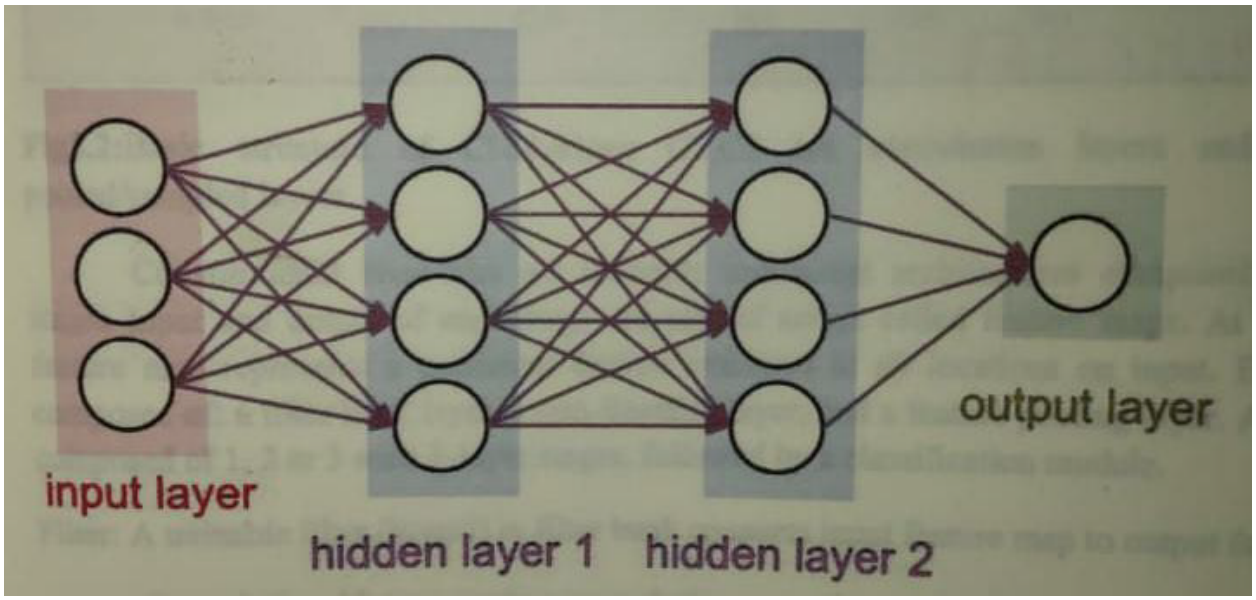

Fig. 4.3.2 A straightforward neural network

A deep neural network is an artificial neural network (ANN) with several layers between the input and output layers (DNN). The DNN identifies the necessary mathematical adjustment to transform the input into the output, whether it's a linear or non-linear connection.

NEURAL NETWORKS IN CONVOLUTION :

Convolutional Neural Networks, like conventional Neural Networks, are made up of neurons with learnable weights and biases. Each neuron accepts a set of inputs, performs a dot product, and optionally adds nonlinearity to the outcome. Convolutional Neural Networks (CNNs) are made up of neurons that learn to optimize themselves, comparable to conventional artificial neural networks (ANNs). Each neuron will still receive input and conduct an operation (such as a scalar product followed by a nonlinear function) - the foundation of numerous artificial neural networks. The entire network will still express a single perceptual scoring function out of the raw data picture vectors to the class score's final output (the weight). All of the normal ANN recommendations and methods will apply to the final layer, which will include the loss functions associated with the classes.

The main significant distinction among CNNs and standard CNNs are the ANNs. are mostly utilized in picture pattern recognition.This enables us to embed image-specific features into the architecture, improving the network's suitability for image-focused tasks while simultaneously reducing the number of parameters required to build up the model. The computational complexity required to compute picture data is one of the most significant drawbacks of older types of ANN.

Due to its relatively tiny picture dimensions of only 28 x 28, Most versions of ANN may use typical machine learning benchmarking datasets like the MNIST database of handwritten digits. A single neuron in the first hidden layer will contain 784 weights (28281, where 1 is the number of black and white values in MNIST), which is possible for most types of ANN using this dataset. When you look at a 64 64 coloured visual input, the amount of weights on a single first-layer neuron skyrockets to 12, 288. Assume that the network used to recognise color-normalized MNIST digits will need to be much larger to deal with this volume of data., and you will see why adopting such models has its drawbacks.

CNN STRUCTURE :

CNNs are feedforward networks, which means that information flows in just one direction, from inputs to outputs. CNNs are biologically inspired, just like artificial neural networks (ANN). Their architecture is inspired by the The brain's visual cortex, which is made up of layers of basic and complicated cells alternate.

CNN designs occur in a variety of shapes and sizes, but they all have convolutional and pooling (or subsampling) layers that are organized into modules. These modules are followed by one or more fully linked layers, similar to a normal feedforward neural network. To create a deep model, modules are frequently stacked on top of one another. For a toy image classification problem, it shows a conventional CNN design. The network receives an image directly and performs multiple steps of pooling and convolution. These operations' output is subsequently fed into one or more fully linked layers.

Ultimately, the name of the class is output by the last layer that is totally linked. Despite the fact that this is the most widely used in the literature base architecture, various architecture modifications have just been presented with the goal of enhancing picture categorization precision or lowering costs of computation. We only mention typical CNN architecture briefly in the rest of this section.

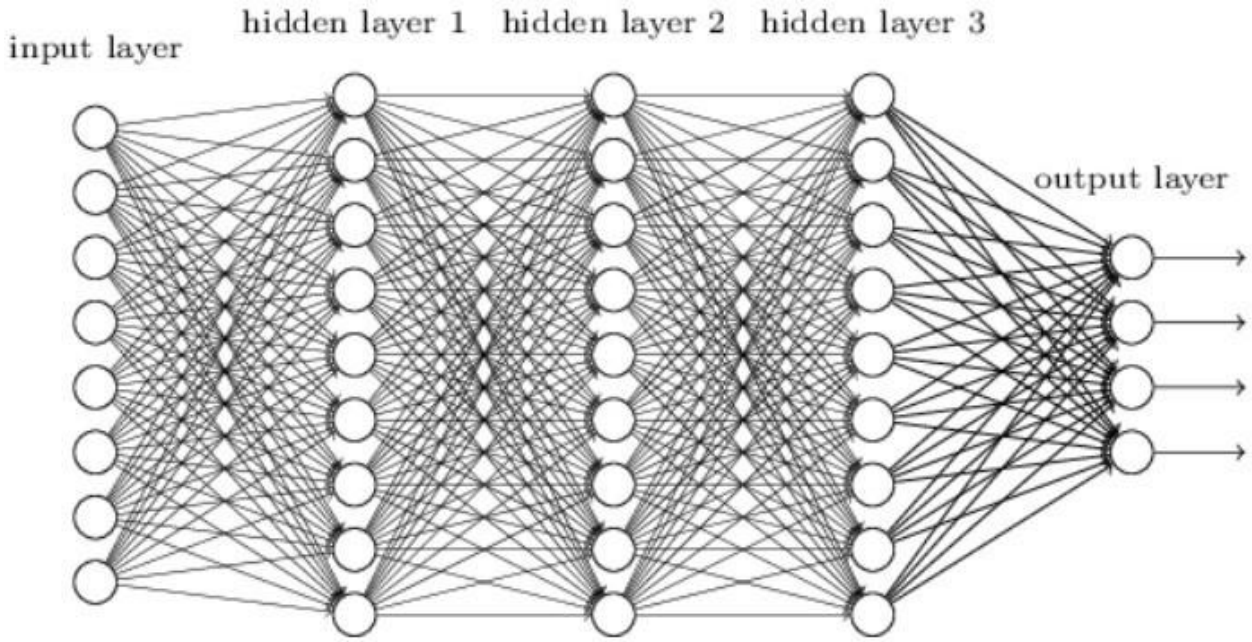

Fig: 4.3.3 CNN's structure

ARCHITECTURE IN GENERAL:

CNNs are made up of three layers. Convolutional layers, pooling layers, and fully-connected layers are the three types. A CNN architecture is generated when these layers are stacked. Figure 2 shows a simplified CNN architecture for MNIST classification. convolution with ReLu pooling input 0 9 output fully-connected with ReLu fully-connected... Fig. 2: A basic CNN design with only five layers
The core functionality of the CNN in the example above may be divided into four categories.

1. The input layer, as with other types of ANN, will store the image's pixel values.
2. Calculating the scalar product between their weights and the area associated to the input volume in the convolutional layer will determine the output of neurons connected to particular regions of the input. The ReLu aims to apply a 'system is crucial' activation function like sigmoid to the output of the preceding layer's activation function..
3. The max pooling executes levels that can indicate along the input's spatial dimensions, reducing the number of factors inside that activation even further.
4. The completely interconnected layers will then Make an effort to generate class scores from the activations, which will be utilized for categorization, in the same way that normal ANNs do. It's also possible to employ ReLu

between these layers to boost performance. It can use convolutional and downsampling techniques to change the original input layer by layer to get class scores for classification and regression. using this simple method of transformation. It is important to remember, however, that simply grasping the overall design of a CNN architecture is insufficient. It can take a long time to create and optimize these models, and it might be confusing. We'll now look at the various layers in further depth, including their connectivities and hyperparameters

LAYERS OF CONVOLUTION:

It learns the feature representations of the pictures they're given and operate as feature extractors.. The convolutional layers' neurons are organized into feature maps. Each neuron in a feature map has a receptive field, which is linked to a neighborhood of neurons in the previous layer by a filter bank of trainable weights. To create a new feature map, inputs are convolved with learned weights, and the convolved outputs are provided via a nonlinear activation function.

The values among all cells in a feature map must be equal; however, different feature maps within the same convolutional layer might have different weights, enabling numerous features to be extracted..

The convolutional layer, as its name suggests, is critical to how CNNs work. The learnable kernels are the focus of the layers parameters.
The spatial dimensionality of these kernels is usually low, yet they cover the entire depth of the input.As data travels through a convolutional layer, each filter is convolved across the spatial dimensions of the input to form a 2D activation map.. These maps of activation can be seen.
The scalar product is calculated for each value in that kernel as we progress through the input. As a result of this, the network will learn kernels that "fire" when they perceive a given characteristic at a specific spatial point in the input. This is referred to as activations.

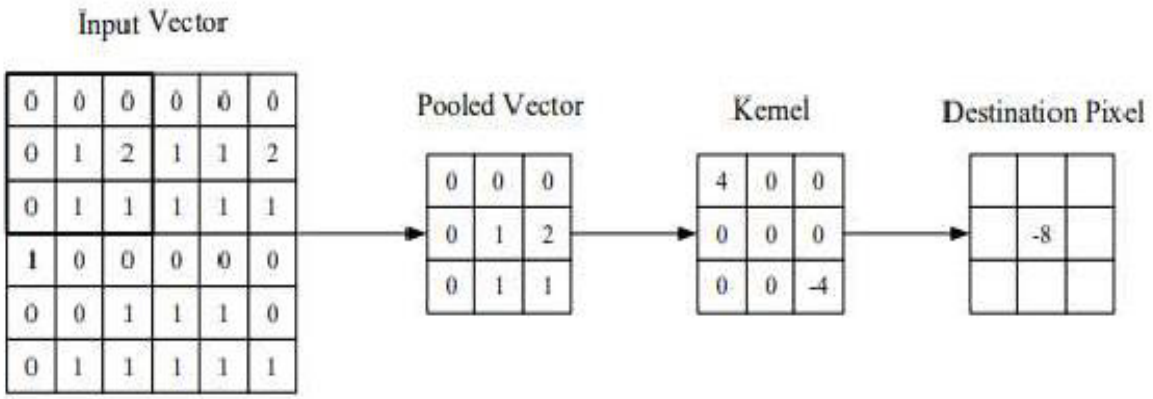

Fig: 4.3.4 A convolutional layer's visual representation

The input vector is placed over the kernel's center element, After that, a weighted sum of itself and any nearby pixels is calculated and substituted.
Each kernel will have its own activation map, which will be layered along the depth dimension to form the whole output volume of the convolutional layer.

As we mentioned before, training ANNs on image inputs produces models that are too large to train efficiently. This is due to the fully linked nature of traditional ANN neurons, hence To mitigate this, each neuron in a convolutional layer is only linked to a tiny fraction of the input volume. The receptive field size of the neuron is frequently also known as the dimensionality of this region. The depth of the input is virtually always equal to the magnitude of the connectivity through the depth.

If the network's input is a 64 * 64 * 3 picture (aRGB color picture with 64 X 64 dimensions and a 6 * 6 receptive field size, each neuron in the convolutional layer will have a total of 108 weights. (6 * 6 * 3), where 3 denotes strength of connectedness over the volume's density. A typical neuron in other forms of ANN might have 12, 288 weights, to put this into perspective.

It can also greatly decrease the model's intricacy by optimizing its output. The depth, stride, and establishing zero-padding are the three hyperparameters that are optimized
.
The number of neurons within the layer can be manually set to the same region of the input to set the depth of the output volume created by the convolutional layers. Other forms of ANNs, where all of the neurons in the hidden layer were previously directly linked to each other, may exhibit this behaviour. Reducing this

hyperparameter reduces the total number of neurons in the network, but it also reduces the model's pattern recognition skills.

We can also select the surrounding spatial depth dimensions of theinput for receptive field placement. If we set the stride to 1, for example, we will get a massively overlapped receptive field with extraordinarily big activations. Alternatively, increasing the stride will limit the amount of overlapping and result in a lower spatial dimension output.

Zero-padding is a basic operation that involves padding the input's boundary, and it's a good way to manage the output volumes' dimensions.

It's crucial to realize that by employing these strategies, we'll change the spatial dimensionality of the convolutional layers' output.
Using an image input of any actual dimensionality will still result in our models becoming gigantic, despite our best efforts thus far. However, techniques have been devised to drastically reduce the amount of parameters in the convolutional layer.

The notion behind parameter sharing is that if one area characteristic is beneficial to calculate in one spatial region, it will almost certainly be useful in another. By restricting each individual activation map inside the output volume to the same weights and bias, we may dramatically minimize the amount of parameters created.

As a result, when the backpropagation step occurs, each neuron in the output will reflect the overall gradient, which may be totalled throughout the depth, instead of updating all of the weights.

Stacking Layers

The feature maps' spatial resolution is reduced by the pooling layers, resulting in spatial invariance to input distortions and translations. To transport the average of all the input values of a local neighborhood of an image to the next layer, use average pooling aggregation layers. was typical practice at first. Max pooling aggregation layers, on the other hand, propagate the maximum value inside a receptive field to the next layer in more modern models.

Pooling layers attempt to gradually lower the representation's dimensionality, reducing the number of parameters and the model's computing complexity.

The "MAX" function is used by the pooling layer to scale the dimensionality of each activation map in the input. Most CNNs employ max-pooling layers with dimensionality 2 2 kernels and a stride of 2 along the input's spatial dimensions. The activation map is reduced to 25% of its original size, while the depth volume is kept at its full size.

There are only two commonly observed techniques of max-pooling due to the destructive nature of the pooling layer. The pooling layer's stride and filters are generally both set to 2 2, allowing the layer to extend across the input's full spatial dimensions. Another option is Overlapping pooling, with the stride set to 2 and the 3 is the kernel size. Because of Pooling is detrimental, and having a kernel size bigger than 3 will typically result in a significant reduction in model performance.

It's also vital to note that, in addition to max-pooling, CNN designs can also include general-pooling. Pooling neurons in general pooling layers are capable of performing a variety of common operations such as L1/L2-normalization and average pooling. However, the focus of this lesson will be on using max-pooling.

Fully Connected Layers

When traveling through the network, To extract more abstract feature representations, numerous convolutional and pooling layers are commonly layered on top of each other. Following these layers are fully linked layers that read these feature representations and do high-level reasoning. The softmax operator is commonly used on top of a DCNN for classification challenges. The classifier on top of the convolutional towers discovered that using a support vector machine (SVM) instead of the softmax operator improved classification accuracy, leading to early success.

Neurons in the fully-connected layer are directly linked to neurons in the two adjoining layers, but not to neurons in any of the layers inside them.

Despite the fact that a CNN only requires a few layers, there is no standard technique to construct a CNN design. However, merely slapping a few layers together and expecting it to work would be foolish. CNNs, like other types of ANNs, tend to follow a similar design, as seen by reading relevant literature. Figure 2 shows a standard design in which convolutional layers are layered, then pooling layers are applied repeatedly preceding advancing to fully linked layers.

CNN varies from other types of Artificial Neural Networks in that they use information about a single sort of input rather than focusing on the full issue domain. As a result, a considerably simpler network design may be created.

The basic ideas of Convolutional Neural Networks have been presented in this paper, along with the layers necessary to construct one and how to arrange the network for most image processing applications.

In recent years, progress in the field of image processing utilizing neural networks has stagnated. This is due in part to a misunderstanding about the amount of intricacy and expertise required to get started modeling these incredibly powerful algorithms for machine learning. I believe that this work has helped to clear up any misunderstandings and made the area more approachable to newcomers.

Training

In order to get the intended network output, CNNs and ANNs in general require learning algorithms to alter their free parameters. Backpropagation is the most often used algorithm for this purpose. Backpropagation determines how to alter a network's parameters to reduce mistakes that impact performance by computing the gradient of an objective function. Overfitting, or Once the network has been trained on a small or even big training set, poor performance on a held-out test set, is a typical problem with training CNNs, particularly DCNNs. This has an impact on the system capacity to be general to new information and a significant difficulty for Regularization can be used to overcome DCNNs.

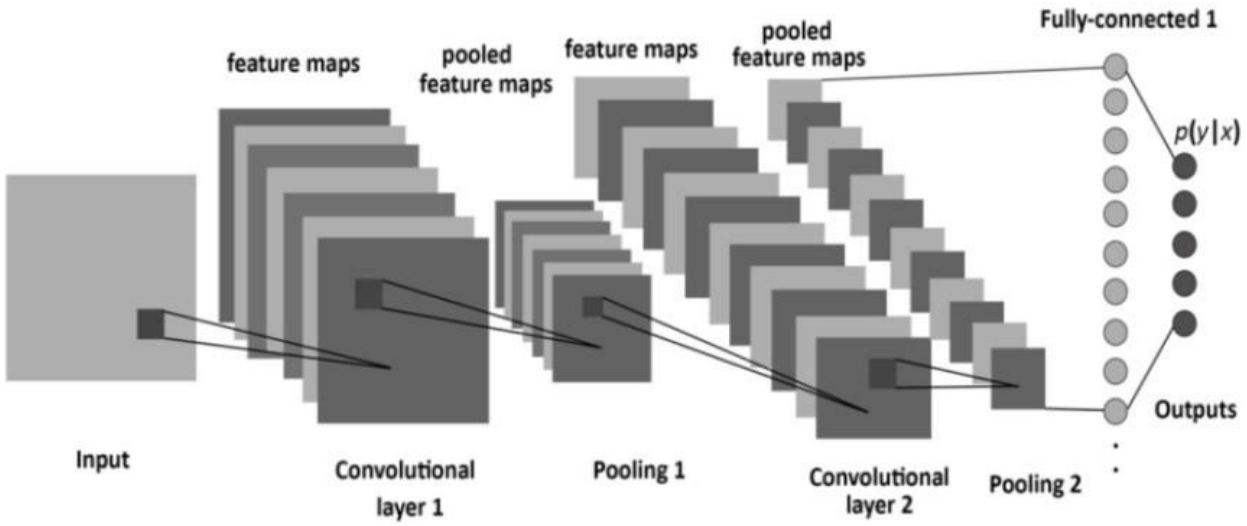

Fig: 4.3.5 CNN Model Training

Caffe Model

Caffe is defined as a Deep Learning structure that was created to build and access the following features in an object detection system.

• Expression: Unlike other models that employ codes, the caffe model defines models and optimizations as plaintext schemas.

• Speed is critical for state-of-the-art models and huge data in both research and industry.

• Modularity: For new jobs and diverse circumstances, flexibility and expansion are essential.

• Openness: The core needs of scientific and applied advancement are common code, reference models, and reproducibility.

HIERARCHY OF LEARNING FEATURES:

From pixel classifiers to hierarchy, learn everything. One layer collects features from the preceding layer's output, and all layers are trained together.

1. **Zero-One Loss**
   These deep learning courses' models are mostly used for categorization. The main goal of classifier training is to decrease the number of mistakes (zero-one loss) on unknown samples.

2. **Loss of Negative Log-Likelihood**
   Because the zero-one loss isn't differentiable, optimizing it for big models (thousands or millions of parameters) is prohibitively costly

(computationally). Given all of the labels in a training set, the classifier's log-likelihood is maximized. The number of correct predictions and the likelihood of the suitable class are not equal., but from the perspective of a randomly initialized classifier, they are quite close. Although the probability and zero-one loss are distinct goals, we should constantly keep in mind that they are associated with the validation set. However, one may grow while the other lowers, or vice versa.

3. **Descent using Stochastic Gradients**
   Ordinary gradient descent is a simple rule in which we generate little steps downward on an error surface specified by a loss function of certain parameters over and over again. The training data is incorporated into the loss function, which we take into consideration for the purpose of normal gradient descent. The pseudo code for this method is as follows: Stochastic gradient descent (SGD) works in the same way as random gradient descent. It works rapidly by calculating the gradient using a few examples at a time rather than the whole training set. To estimate the gradient, we utilize just one sample at a time in its purest form.

Caffe is a machine learning platform (or framework) focused on speed and modularity of expression. It was created by young King Gia at Berkeley artificial intelligence research. Tensorflow, Tiano, Charis, and SVM are just a few of the deep learning or machine learning frameworks for computer vision. However, the expressive architecture of the edition café is the reason for its implementation. We can simply swap between CPU and GPU while training and optimizing GPU machine modules. Configuration, rather than hard coding, defines our problem. Cafes are open source libraries, therefore they support expandable programming. It has grown to over 20,000 developers and, since its inception, has provided development platforms in extensible languages such as Python and C++.

The third argument is that when it comes to training neural networks, speed is the most important factor. With a conventional media GPU that processes photos in milliseconds per image, Caffe can process over a million images in a single day. Because it is an open source library, Tiana and Kara's Caffe is the fastest convolution neural network in the current community. A large amount of studies projects are driven by cafes. Every day, something new emerges from it.

4.4 OPENCV

INTRODUCTION TO OPENCV

OpenCV (Open Computer Vision Library) is a free software library for computer vision and machine learning.The purpose of OpenCV was to develop a standard infrastructure for computer vision applications and to speed up the adoption of machine perception in business goods. Because The software OpenCV is BSD-licensed, it is fairly simple for companies to use and change the code. It's a very good package with 2500 optimized algorithms that cover both classic and sophisticated machine vision and deep learning approaches. These algorithms are utilized for a variety of tasks, including face recognition and discovery. Human acts are classified by identifying objects. Track camera motions and moving objects in movies. Remove red eyes from photographs shot with the flash, detect eye movements, recognise scenery, and overlay it with augmented reality.

OpenCV was first started in 1999 as a part of a series of tests that included real-time ray tracing and 3D display walls, an Intel Research initiative to build CPU-intensive applications. Intel Russia's optimization expertise, as well as Intel's Performance Library Team, were major contributors to the project. The project's aims were characterized as follows in the early days of OpenCV:

- Advance vision research by making fundamental vision infrastructure code not just available but also optimized. No need to reinvent the wheel.
- Distribute vision information by offering a standard framework on which developers may build, making code more clear and transferrable.
- Make portable, performance-optimized code available for free — with a license that doesn't need the source to be open or free.

In 2000, during the IEEE Conference on Computer Vision and Pattern Recognition, the first alpha version of OpenCV was given to the public, followed by five beta versions between the years 2001 and 2005. In 2006, the initial 1.0 version was launched. In October 2008, A "pre-release" of version 1.1 was made available.

In Oct 2009, OpenCV released its second major version. OpenCV 2 makes significant modifications to the C++ interface, aiming for simpler, more type-safe patterns, new methods, and higher performance implementations for existing ones.

Official releases are now released every six months, and development is being done by an independent Russian team with the assistance of commercial firms.

OpenCV support was taken up by a non-profit organization, OpenCV.org, in August 2012, which maintains a development and user site.
In May 2016, Intel announced the acquisition of ITSEEZ, a prominent OpenCV developer.

OpenCV (Open source computer vision) is a programming library primarily designed for real-time computer vision. Itseez and Willow Garage supported it after Intel established it (which was later acquired by Intel).The library is cross-platform and free to use thanks to the open-source BSD license.

It supports Windows, Linux, Android, and Mac OS and offers C++, Python, Java, and MATLAB interfaces. When MMX and SSE instructions are available, OpenCV leans heavily toward real-time vision applications. A full-featured CUDAandOpenCL interface is currently being developed.

Over 500 algorithms exist, having 10 times the number of functions that make up or support them. OpenCV is written fully in C++ and has a user interface that is templated.

Among the applications of OpenCV are :

- Toolkits for 2D and 3D features
- Estimation of egomotion
- Face recognition software
- Recognition of gestures
- Computer–human interaction (HCI)
- Robotics on wheels
- Understanding motion
- Object recognition
- Recognition and segmentation
- Depth perception from two cameras in stereopsis stereo vision
- Motion creates structure (SFM)
- Motion detection
- Virtual reality

OpenCV has a statistical machine learning library to support some of the aforementioned areas:

- Increasing the effectiveness of decision-tree learning
- Trees that increase gradients
- k-nearest neighbor algorithm Expectation-maximization algorithm
- The classifier Naive Bayes
- Networks of artificial neurons
- Forest at random
- Forest at random
- Vector support machine (SVM)
- Deep neural networks (DNNs) are a kind of artificial intelligence (DNN)

Numpy libraries in OpenCV :

The term NumPy stands for "Numeric Python" or "Numerical Python." It's a Python extension module that provides quick precompiled functions for mathematical and numerical tasks. NumPy also adds strong data structures to the Python programming language, allowing for faster multidimensional arrays and matrix computation. Even large matrices and arrays are targeted by the approach. The module also includes a large library of high-level mathematical functions for working with these matrices and arrays.

It is the most important Python module for scientific computing. It has a number of characteristics, including the following :

- A numpy array is a grid of identical-type items; a tuple of nonnegative integers serves as the index.
- The array's rank is the number of dimensions;
- The shape is a collection of numbers indicating the array's size along each dimension.

Array in Numpy:

A numpy array is a grid of identical-type items indexed by a tuple of nonnegative integers. The array's rank is the number of dimensions; the shape is a tuple of numbers indicating the array's size along each dimension.

SciPy:

SciPy is frequently discussed alongside NumPy. SciPy enhances NumPy's capabilities by adding new functions for minimization, regression, Fourier

processing, and more. NumPy is built on top of two previous Python array modules. Numeric is one among them. Numeric is similar to NumPy, a Python package for high-performance numeric processing that is no longer supported. Numarray, which is a full rebuild of Numeric but is also obsolete, is another NumPy ancestor. NumPy is a combination of the two, since it is based on Numeric's code and Numarray's capabilities.

Python as a Matlab Replacement:

Python may be used to replace MATLAB when combined with Numpy, Scipy, and Matplotlib. NumPy, SciPy, and Matplotlib work together to provide a free (as in "free beer") alternative to MATLAB. Despite the fact that MATLAB offers a large number of supplementary toolboxes, NumPy has the benefit of being a more current and comprehensive programming language that is also open source. SciPy extends Python's capabilities to include additional MATLAB-like features. With the Matplotlib package, which provides MATLAB-like charting features, Python is rounded out in the direction of MATLAB.

CHAPTER 5

RESULTS AND DISCUSSIONS

RESULTS

This object identification method can recognise objects at up to 14 frames per second, therefore even low-quality cameras at any frame rate may yield decent results. In this scenario, a 6 frames per second camera is used. The SSD method exhibited interior and outdoor feed video frames through camera in our testing, although the location of the objects differed between two adjacent frames. The webcam's footage and the algorithm reduce the size of a single frame to 300 x 300 pixels. With the precision of a class label, SSD can recognise items frame by frame and create a bounding box around the discovered object.

The following figures demonstrate the outcomes of this process on a picture. With a confidence level of 98.87 percent and a level of 100 percent (i.e., probability), CNN has a system for identifying human traits that is extremely accurate. By employing a higher proportion of default boxes, which can have a bigger effect, and using various boxes for each location, the SSD can create several bounding boxes for distinct classes with varied confidence levels. Frame difference is the basis for this suggested single-shot multibox detection technique. The suggested approach was evaluated using frames.

INPUT **OUTPUT**

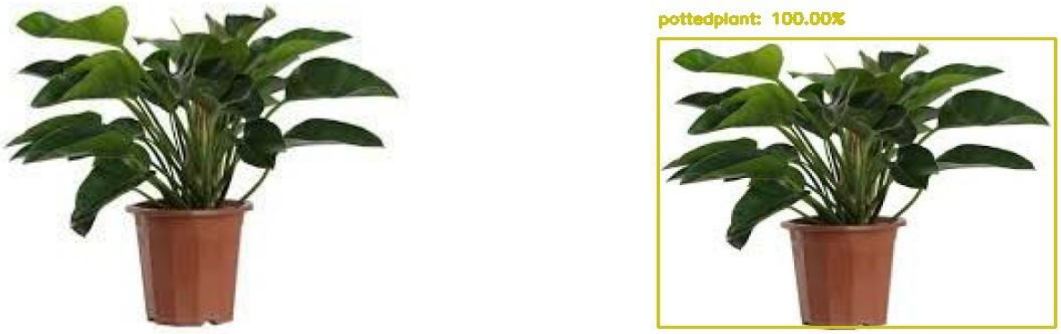

Fig.5.1. Detection of Potted plant with confidence level of 100%

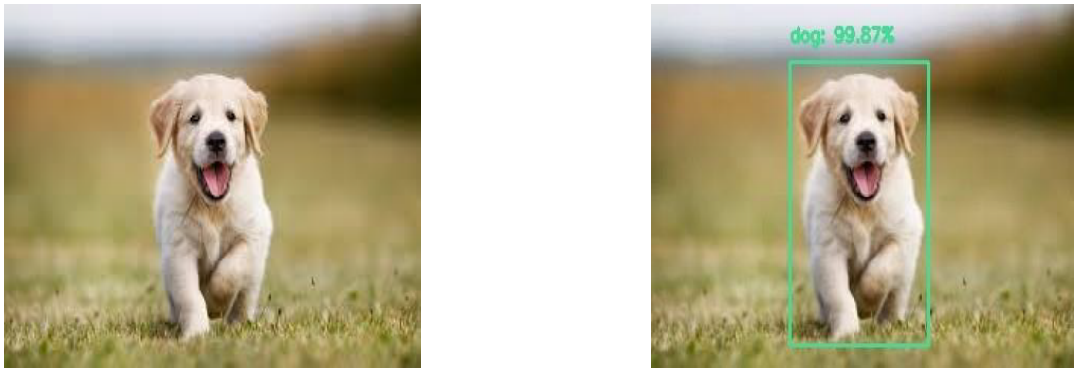

Fig.5.2. Detection of Dog with confidence level of 99.87%

Fig.5.3. Detection of Train with confidence level of 100%

Fig.5.4. Detection of Car, Train and Airplane with confidence level of 99.90%, 54.40% and 47.45% respectively.

CHAPTER 6

CONCLUSION

CONCLUSION

A high accuracy object detection solution has been devised using the MobileNet and the SSD detector for object detection, which can push processing rates to 14 fps and make it efficient to all cameras that can only process at 6 fps. This system can recognise things in its dataset such as a vehicle, bicycle, bottle, chair, and so on. The dataset may be enlarged indefinitely by employing deep learning technologies to add an endless number of objects. For the SSD algorithm experiment, we utilized Windows 10, Pycharm, OpenCV 3.4.2, and the Python programming language. The purpose of this study is to create an autonomous system in which object and scene identification aids the community in making the system more engaging and appealing. This study will be used mostly in the future to identify items with superior characteristics in the external world.

THANK YOU!!!