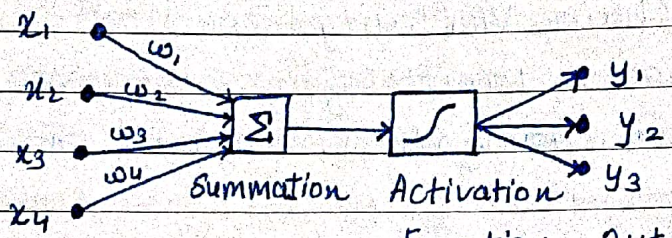
DATA SCIENCE HONOURS

~ENDSEM~

[Only major Points covered, extrapolate in exams]

UNIT 3 : Neural Networks for big data

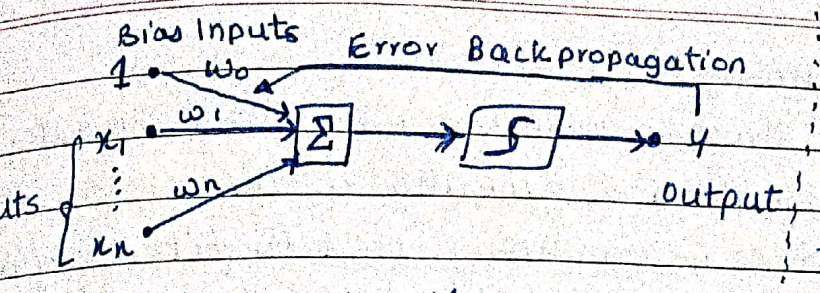
* Fundamental of neural networks & artificial neural networks
- Neural Network / Neural Nets are a system of interconnected processing units called neurons.
- Artificial neural networks is a computer system designed to simulate how human brain analyzes and processes information.

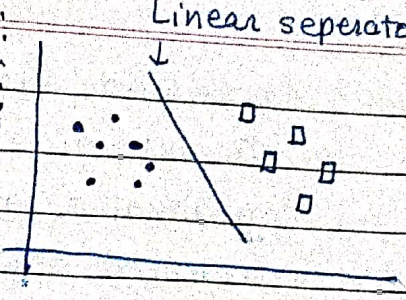


Artificial Neural Network.

- Comparision with biological neural network :-
  Dendrites - Inputs ; Cell Nucleus - Nodes
  Synapses - Weights ; Axon - Output
- Architecture of ANN consists of input layer, hidden layer and output layer. <Draw them>.
- Advantages of ANN : 1) Can process in parallel. 2) Stores data on entire network. 3) Can work with incomplete knowledge. 4) Has fault tolerance.
- Disadvantages :- 1) Unrecognized behaviour. 2) Hardware dependence. 3) Duration (no. of iterations) aren't known.

* Perceptron and linear models :-
- Perceptron is building block of ANN.
- It is linear supervised machine learning algorithm for various binary classifier.
- Perceptron is the simplest form of ANN with no-hidden layers and an single output layer, a single output node.





- It is a linear classifier meaning it can only classify points if these are linearly seperable.
- Multi-layer perceptrons are perceptrons that has 3 or more layers of perceptrons basically they must have a hidden layer.

• Linear Models :

- Describe a continuous response variable as a function of one or more predictor variables.
- Linear regression is a method to create linear model.
- It describes the relationship between one dependent variable y as a function of one or more independent variables X_i.

$$y = \underbrace{B_0}_{\text{Bias}} + \sum_{i=1}^{N} B_i X_i + \underbrace{\epsilon}_{\text{Error}}$$

- Linear models are analogus with linear regression.
- Types of linear regression :-
  1) Simple :- Only one predictor.
  2) Multiple :- Multiple predictors.
  3) Multivariate :- Multiple response.

* Non-linear model :-

- Describe non-linear relationships in experimental data.
- Generally these models are parametric and have non-linear equation [$ax^2 + by^2 = c$].
- Consists of dependent variable as a function of non-linear parameters and one or more independent variables.

$$y = f(X, \beta) + \epsilon$$

- Parameters can be exponential, trignometric or any other

non-linear function.
- To determine these non-linear parameter estimates an iterative algorithm is typically used.
- Gradient descent is commonly used to fit non-linear model.
- Uses of parametric non-linear regression :- Fit nonlinear models, generate predictions, evaluate goodness of fit.

* Feed-Forward neural networks :-
- Basic type of neural network in which input is processed only in one direction, data never flows backwards/ opposite.
- In the most basic form it is a single layer perceptron.
- Here nodes never form a cycle, it is the opposite of recurrent networks.
- CNN [convolutional Neural Networks] are a type of feed-forward neural networks.
- <Draw NN diagram> <Explain architecture>

* Gradient descent and backpropagation :-
- Gradient descent is the process of using gradients to find the minimum cost function.
- Backpropagation is calculating gradients by moving in the backward direction.
- Gradient descent relies on back propagation.
- Backpropagation :-
- <Diagram of network; on previous page>.
- Backpropagation simply adjusts weights based on error
- Algorithm :-

1) Traverse the network by computing hidden layer's and output layer's output. [Feedforward step].

2) In output layer calculate derivateive of cost function with respect to hidden and input layers.
3) Repeatedly update weights until they converge or model has gone undergone enough iterations.

- Equation :-

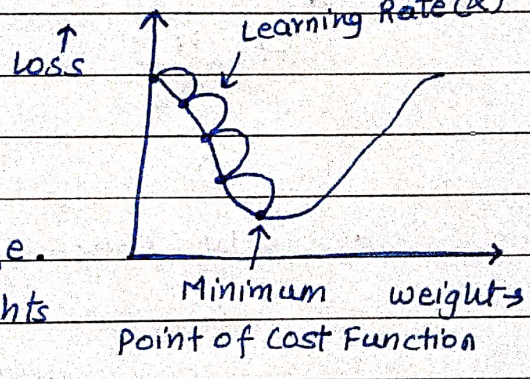
$$\theta^{t+1} = \theta^{t} - \alpha \frac{\partial E(X, \theta^t)}{\partial \theta}$$

(θ^{t+1}: parameter; α: learning rate; $E(X, \theta^t)$: cost function)

$$E(X, \theta^t) = \frac{1}{2N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$$

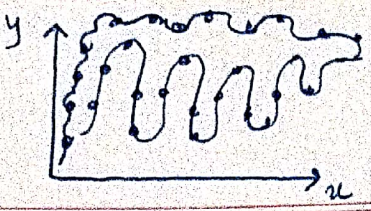
- Gradient Descent :-
- Optimization algorithm, helps find ~~minimum~~ cost weights that minimize the cost function.
- Equation:

$$W_{new} = W_{old} - \alpha \frac{\partial I}{\partial w}$$



- Learning Rate (α) determines step size.
- We go on finding loss for new weights until minimum loss is reached.

* Overfitting :-
- An undesirable machine learning behaviour.
- Occurs when ML model gives accurate predictions for training data but not for new data.
- Overfit model gives inaccurate predictions and does not work well for all types of new data.
- It occurs when :-
  1) Training data is too small.
  2) Training data contains irrelevant data, called noise.
  3) Model trains too long on single sample set of data.
  4) Model complexity is too high.



Overfitted Model.

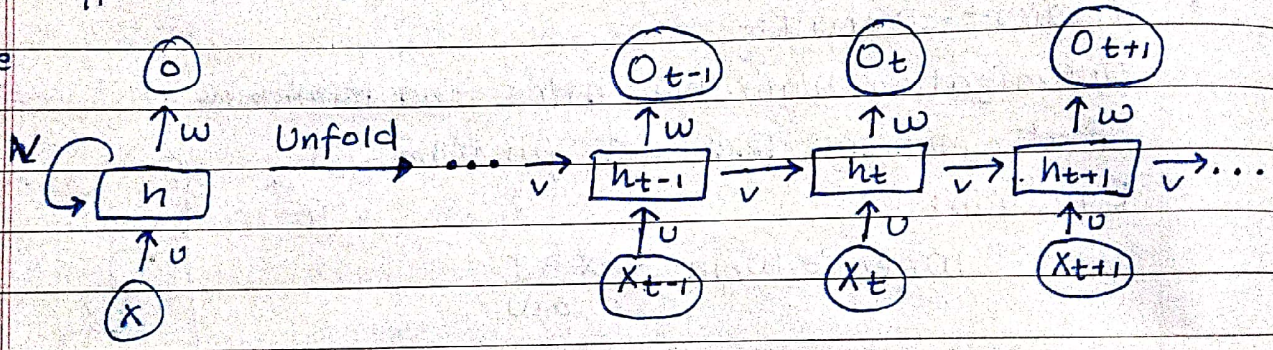
- Prevent overfitting:-

1) Early stopping : Stop before model trains on noisy data.
2) Pruning: Eliminate irrelevant features.
3) Regularization :- Add penalty/relevance value to each feature.
4) Ensembling :- Combine several predictions.
5) Data augmentation :- Change data for every training iteration.

* Recurrent neural networks :-

- Class of ANN where connection between nodes can create a cycle, allowing output from some nodes to affect subsequent input to same nodes.

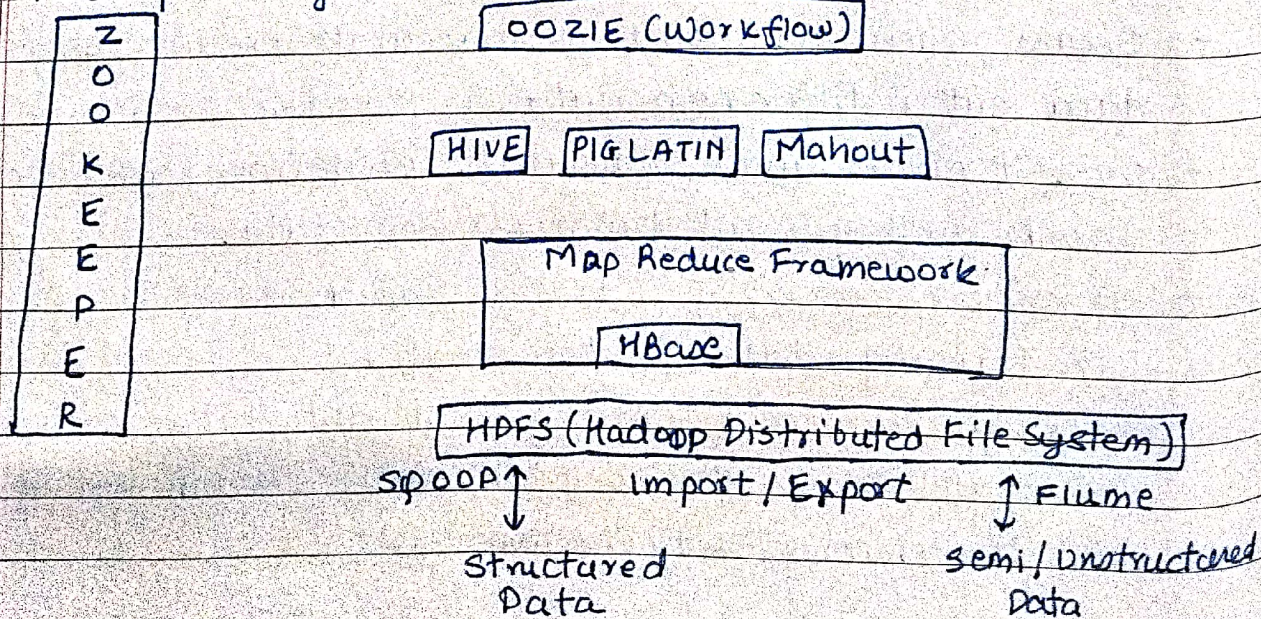
V, W, U are weights.



- <DL UNIT 3>

UNIT IV: Big data analytics using Hadoop-I

* Hadoop Ecosystem :-



• Components:-

1) SQOOP [SQL + Hadoop] :- After processing data using HDFS we export it to RDBS and store it using SQOOP.
2) HDFS :- Main component of Hadoop. Uses a technique to store data in a distributed manner for fast computations. Saves Data in a block of 64 or 128 MB. Information is in data node and meta data in name node.
3) MapReduce Framework :- MapReduce program can be written in any languages C, C++, Java, etc. "Map" maps logic onto data and after computation, reducer collects results of map to generate final output of MapReduce.
4) HBase :- NoSQL DB, created for large tables. Provides fault tolerance & Horizontal scalibility.
5) Hive :- Deals with structured data using SQL. In context of Hive we call it HQL (Hive Query Language).
6) Pig :- Deals with structured data using Pig Latin. Helps generate workflows for Hadoop operations.
7) Mahout :- Open source ML library written in Java.
8) Oozie :- Workflow scheduler, manages hadoop jobs.
9) Zookeeper :- Distributed / Centralized service that provides working service for a hadoop cluster.

* HDFS :-

- Hadoop Distributed File System.
- Data is distributed over several machines and replicated.
- Ensures durability to failure and high availability.

• HDFS used for :-

1) very large files. 2) Streaming Data Access. 3) Low cost hardware.

• HDFS concepts :-

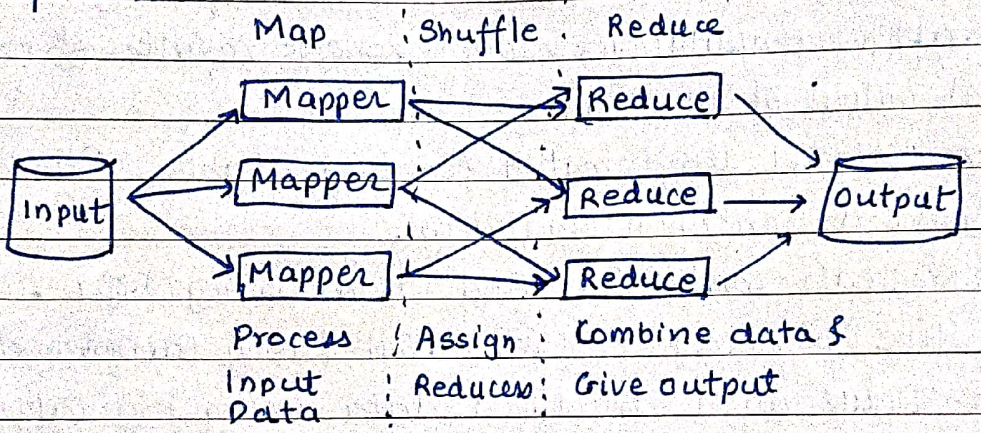
1) Blocks :

- Minimum amount of data that can be read or written.
- Each block is an independent unit.

2) Name Node:-
- Name node acts as master, controls & manages HDFS.
- Knows status & metadata of all files in HDFS.

3) Data Node:-
- Store and retrieve blocks when told to i ~~stores~~ slaves.
- Report to name node periodically.
- start-dfs.sh to start hadoop.

* MapReduce:



* Python and hadoop streaming :-
- Hadoop streaming allows writing programs for MapReduce in languages other than to Java, such as Python, and run them on hadoop cluster.
- It uses standard input/output [stdin & stdout] for streaming.
- How to use it? :-

1) Write MapReduce program in Python.
2) Prepare input in a format as required in HDFS.
3) Upload data to HDFS.
4) Run the streaming job.
5) Retrieve output.

* Apache Spark Tutorial :-
- Open-source distributed computing system.
- Designed for processing and analyzing large-scale datasets.

- Provides high-level API to perform data processing tasks in a distributed and parallel manner.
- Basics:

1) Resilient Distributed Datasets:-
- Immutable distributed collection of objects.
- Created from data stored in HDFS.
- RDDs support transformations [MapReduce] & actions [collect, save].

2) DataFrames and Datasets:-
- Higher level abstractions built on top of RDDs.
- Provide schema and structured based API similar to database table or dataframes.

3) Spark Context:-
- Makes the connection to spark cluster; an entrypoint.
- Helps create RDDs, control Job execution & make system configuration.

4) Spark Architecture:- Follows master/slave architecture.

5) Spark Libraries:- SQL, Streaming [Real-time data stream], MLlib [ML], GraphX [Graphic processing]

* PySpark:-
- Apache spark is written in Scala.
- Pyspark provides a Python API for spark i.e we can use python instead of scala.
- Some Libraries compatible with PySpark are:-
  - PySparkSQL:- SQL Queries on Apache Spark.
  - MLlib:- Classification, regression, clustering etc.
  - Graphframes:- Graph processing library.

UNIT V: Big data analytics using Hadoop II

* Data warehousing and mining:-

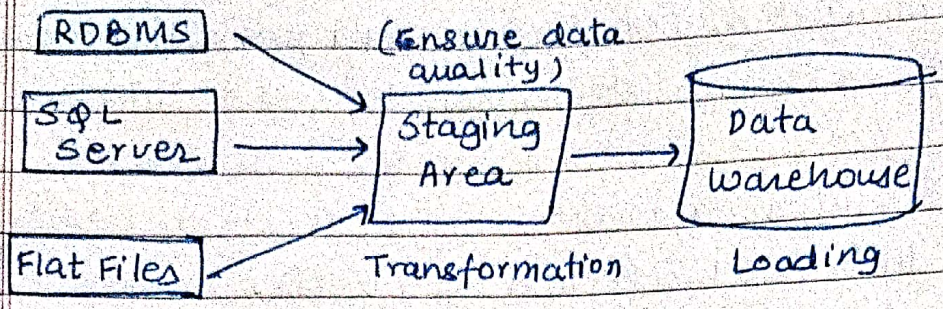


Fig: Data warehouse is a method of organizing and compiling data into one database.

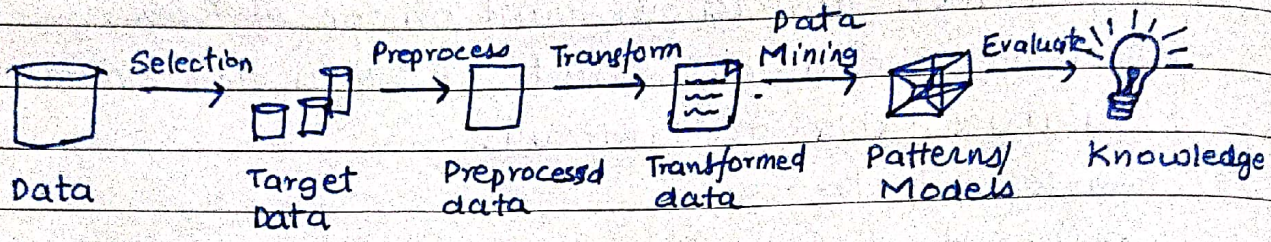


Fig:- Data is extracted and analyzed using statistics & ML system to find patterns in data mining.

* Data analysis using Hive:-
- Hive is a datawarehouse system used to analyze structured data.
- Built on top of Hadoop, developed by facebook.
- Provides functionality of reading, writing, managing large datasets residing in distributed storage.
- SQL like queries i.e HQL is used which gets internally converted to MapReduce jobs.
- Hive supports DDL (Data defn), DML (Data Manipulation) and UDF (User defined function)
- Hive can operate on compressed data & uses indexing.
- However it cannot handle real-time data, faces latency and cannot be used for OLTP.

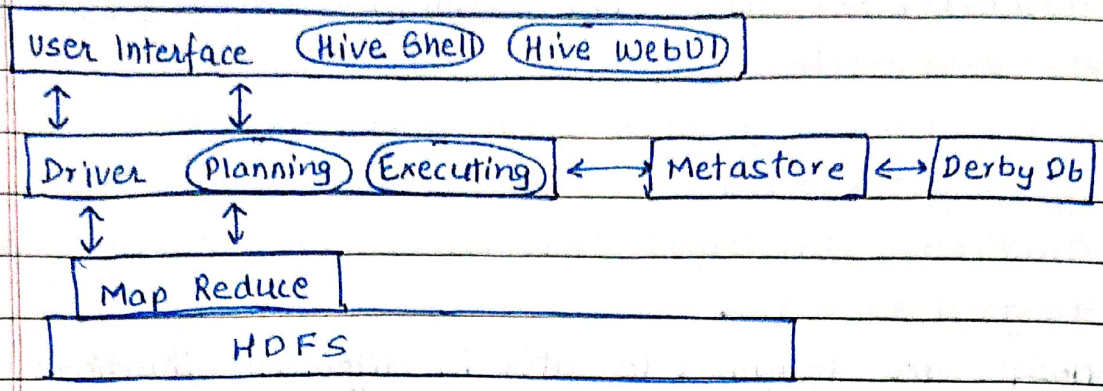


Fig: Hive Architecture

- User Interface: Inteface between HDFS & User to run HQL queries.
- Meta Store: Schema, table metadata, HDFS location stored here.
- HiveQL Process Engine :- Communicates with metastore to run HQL.
- Execution Engine :- Processes HQL queries to convert to MapReduce jobs.

* Data ingestion :-
- Process of loading and importing data into a system.
- Critical step in data analytics workflow.
- A company ingests data from various sources such as email marketing platforms, CRM systems, financial system, social media platforms and so on.
- Data ingestion is typically done without any changes to data and is ~~hence~~ different~~ent~~ than ETL.
- It describes moving data from one location to another, i.e one DB to another DB.
- Types of Ingestions :-

1) Real-time :- Real time ingestion using some cloud based service.
2) Batch :- Collecting large amount of raw data & processing it later.

- Data ingestion Framework (DIF) is a set of services that allow us to ingest data into our database.

* Scalable machine learning using Spark :-
- As the era of internet started after 2000's data explosion took.

- To work with such data building scalable systems was the only solution.
- We majorly try to solve 2 types of tasks:-
  (i) Compute-heavy. (ii) Data-heavy.
- Traditional databases such as MySQL, Oracle were not designed to scale.
- NoSQL are designed to cater in different situations.
- ML & deep learning algos are both compute & data heavy, hence we want a solution that is scalable to both
- Spark is good for both tasks:-
  1) Data-heavy :- Uses HDFS
  2) Compute-heavy :- Uses RAM instead of disks.
- As spark utilizes RAM it is an efficient solution for iterative tasks in Machine learning like Stochastic Gradient Descent (SGD).
- For these reasons, Spark MLlib can be used to perform scalable ML.

UNIT VI : Applications

* Natural language processing steps:-
  - Text Preprocessing :-
    1) Tokenization:- Convert sentence to individual words/tokens.
    2) Lowercasing:- To maintain case-sensitivity.
    3) Stop word removal :- Removal of common words ['and', 'the', 'a']
    4) Noise removal :- Special characters punctuation removal.
    5) Lemmatization/Stemming :- Conversion to base word.
    6) Spell Correction:- Correct spellings.
  - Feature Extraction :-
    1) Bag-of-words : Collection of word frequencies.
    2) TF-IDF :- Term frequency & Inverse term frequency.
    3) Word embeddings :- Represent words as dense vectors.

4) N-grams :- Capture sequential patterns of N-words.
5) Topic modelling :- Extract topics using LDA, NMF, LSA.

- Applying NLP techniques :-

1) Sentiment analysis :- Determine sentiment/polarity of a text.
2) Named Entity Recognition :- Extract named entities [mostly nouns].
3) Parts of speech tagging :- Assign grammatical tags.
4) Text Classification, Machine Translation, Dialogue system, Question Answering etc.

* Sentiment Analysis :-

- Classify if block of text is positive, neutral or negative.
- Our goal is to analyze people's opinions.
- Sentiment Analysis is an efficient & quick way of analyzing large corpus of responses, tweets and more.

- Types of sentiment analysis :-

1) Fine-grained :- Rating out of 5 star [3 star, 4 star, 1 star]
2) Emotion detection :- Happy, sad, angry, upset.
3) Aspect-based :- Focuses on particular aspect [Iphones have great camera but come with a high price tag.]
4) Multilingual : Consists of different languages [I love saitama because of "Shumède hiro wo yatteru mono da" phrase.]

- Building sentiment analysis system :-

Rule-based; ML Based; Neural Networks; Hybrid.

* Computer vision :-

- It involves processing and analyzing images to extract meaningful information.

- Image Pre-processing :-

1. Image resizing & enhancement.
2. Noise removal.
3. Color space conversion :- Converting RGB to grayscale.

- Feature extraction :-

1) Edge & corner detection.
2) Blob detection :- Identify regions of interest.
3) Texture analysis :- Extract patterns using local binary patterns.
4) Scale-Invariant Feature Transform :- Detecting local features that are invariant to scale, rotation & affine transformations.

- Applying machine learning algorithms :-

1) Classification :- Classify into labels.
2) Object Detection :- Using YOLO (You look only once).
3) Image Segmentation :- Segment image using CNN.
4) Image captioning :- Generate textual descriptions.
5) Image Generation :- Generative Adversarial networks (GANs).

* Application in object detection :-

- Object detection involves identifying and localizing objects of interest within an image or video. Applications :-

1) Autonomous driving :- Detect pedestrian, vehicles, signals.
2) Surveillance :- Detect & track suspicious activities.
3) Robotics :- To perceive & interact with surroundings.
4) Healthcare :- Detect abnormalities in medical scans.
5) Augmented Reality (AR) :- Accurately overlay virtual objects.
6) Environment Monitoring :- Detect & track wildlife, enviroment-change
7) Industrial Automation :- Quality Checking & Assuarance.