

# UNIT-V

## Computer Vision

# CONTENTS

- ❖ Introduction
- ❖ Why Computer Vision ?
- ❖ Processing Components
  - ❖ Image Acquisition
  - ❖ Process
  - ❖ Analysis
- ❖ DL Architectures used in CV
- ❖ Real Time Example for Object Detection

# COMPUTER VISION

- **By definition**, computer vision mimics **natural processes**: It **retrieves visual information**, **handles it**, and **interprets it**. And state-of-the-art algorithms, so-called **neural nets used for computer vision tasks**, replicate natural neural networks.
- **Computer Vision** comprises of a **set of computational techniques** to **understand visual data** such as **images and videos**.
- Computer vision techniques are used for **image classification, motion tracking, image generation, colorizing black-and-white images, etc.**

- Recent **advanced applications** of Computer Vision techniques have a wide variety of applications - **form helping diagnose whether or not a patient has a tumor etc.**
- computer vision was mainly based with **image processing algorithms and methods.**
- The **main process** of computer vision was **extracting the features of the image.** ie, **Detecting the color, edges, corners and objects** were the **first step** to do when performing a computer vision task.



# WHY COMPUTER VISION ?

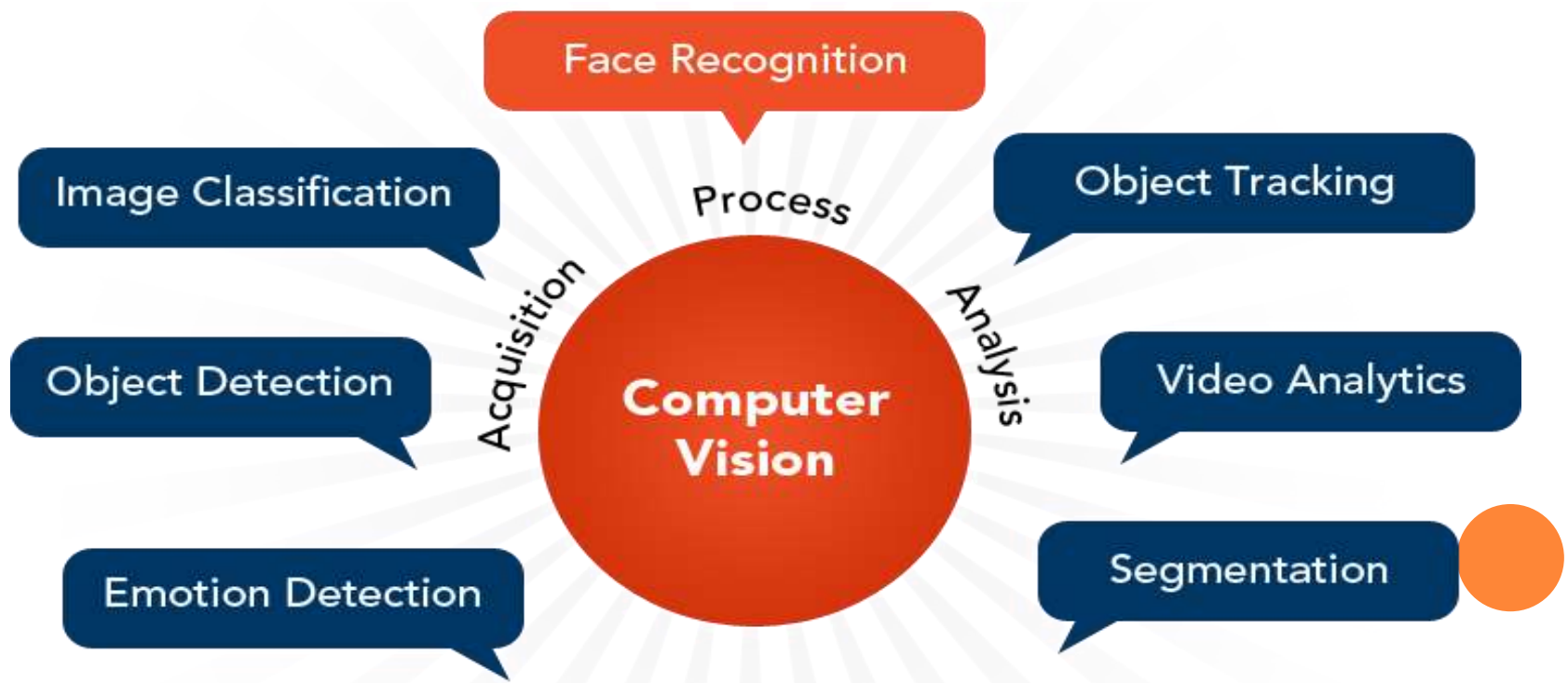
- Computer vision enables a wide range of technological innovation.
- It allows self-driving cars to safely steer through streets and highways
- It enables facial recognition tools to match images of people's faces to their identities and
- It enables augmented-reality applications to mix virtual objects with real-world images.
- Computer vision applications are used across industries to improve the consumer experience, reduce costs, and tighten security.

- **Manufacturers** use it to **spot defective products** on the assembly line and prevent them **from shipping to customers**.
- **Insurance adjusters** use it to assess **vehicle damage** and reduce fraud in the claims process.
- **Medical professionals** use it to **scan X-rays, MRIs, and ultrasounds to detect health problems**.
- **Banks** use it to **verify customers' identities** before conducting large transactions.

# PROCESSING COMPONENTS

## ○ Processing Components are:

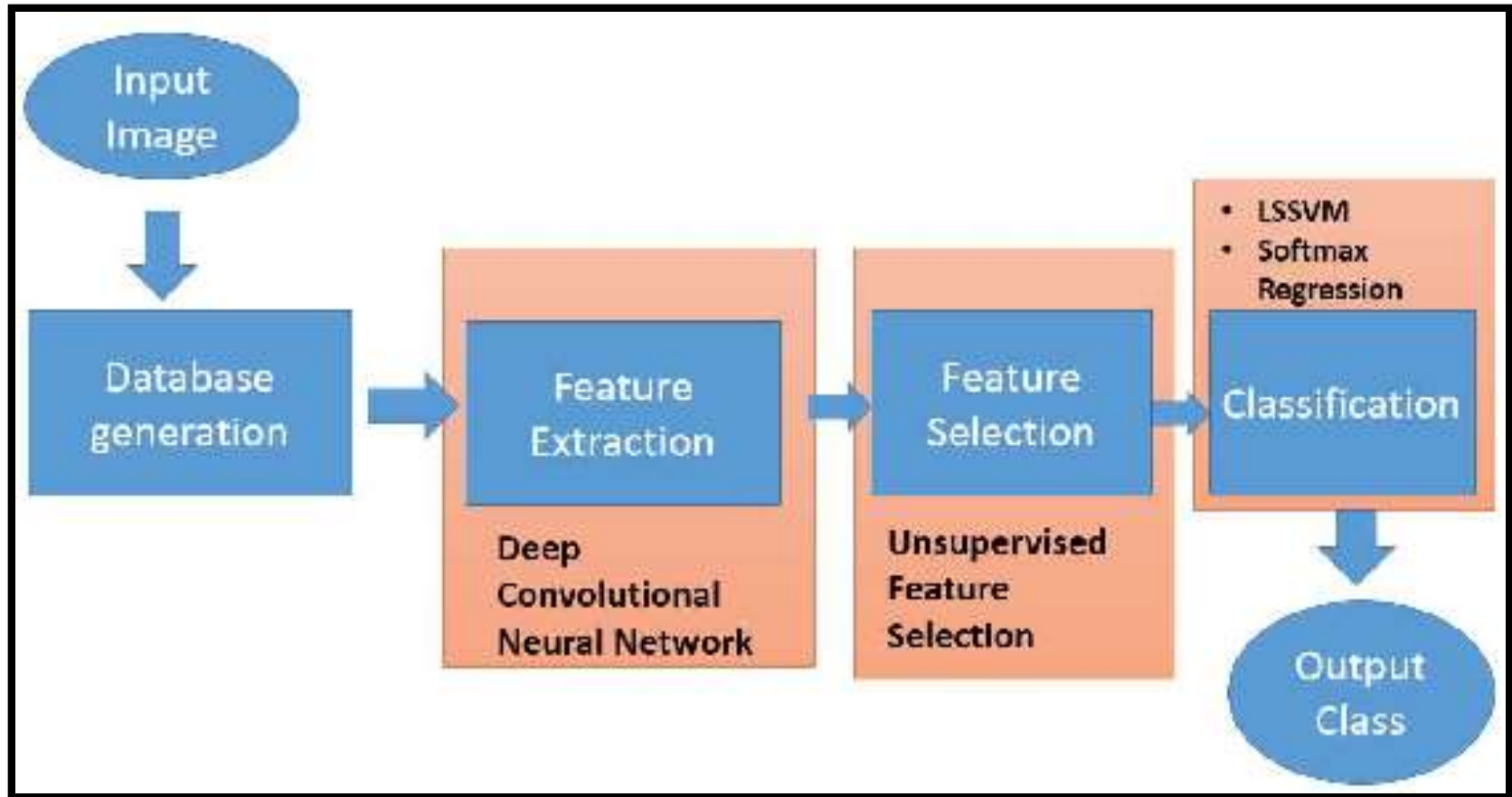
1. Image Acquisition
2. Image Processing
3. Image Analysis

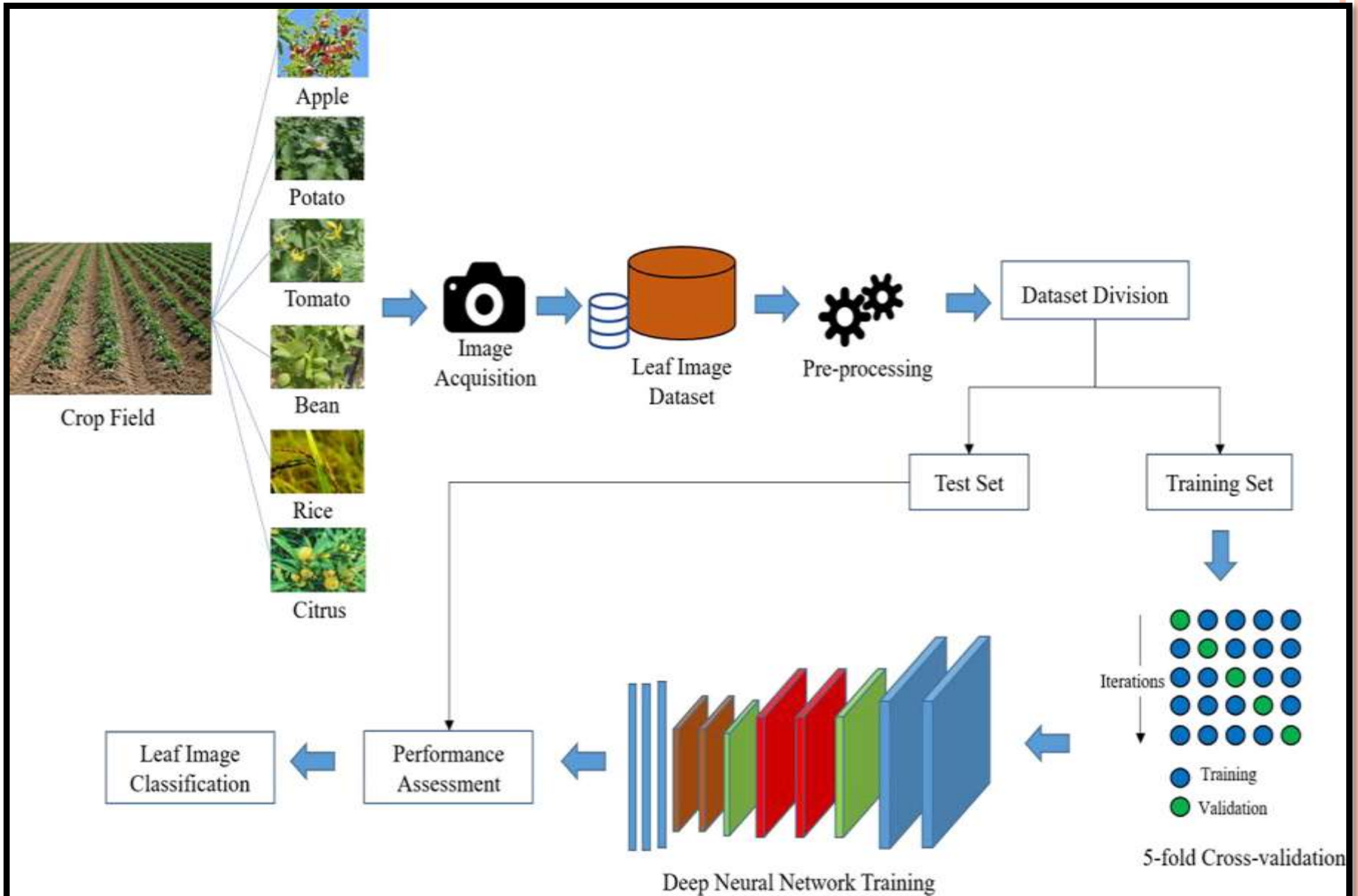


# 1. IMAGE ACQUISITION

- **1. Image Classification:** Image classification is where a computer can analyze an image and identify the image.
- Image classification involves **assigning a label** to an entire image or photograph.
- Image classification with deep learning most often involves convolutional neural networks, or CNNs.
- **Some examples of image classification include:**
  - 1. Labeling an x-ray as cancer or not (binary classification).
  - 2. Classifying a handwritten digit (multiclass classification).
  - 3. Assigning a name to a photograph of a face (multiclass classification).







## CONV LAYER 1

1st set of feature maps or "clues"



## CONV LAYER 2

2nd set of feature maps or "clues"

Shape:  $2 \times 2 \times 3 = 12$  pixels



## FLATTEN

Transform shape into single vector

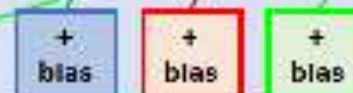
Shape:  $12 \times 1 = 12$  pixels




36 weights  
(12 pixels  $\times$  3 outputs)  
and 1 bias term

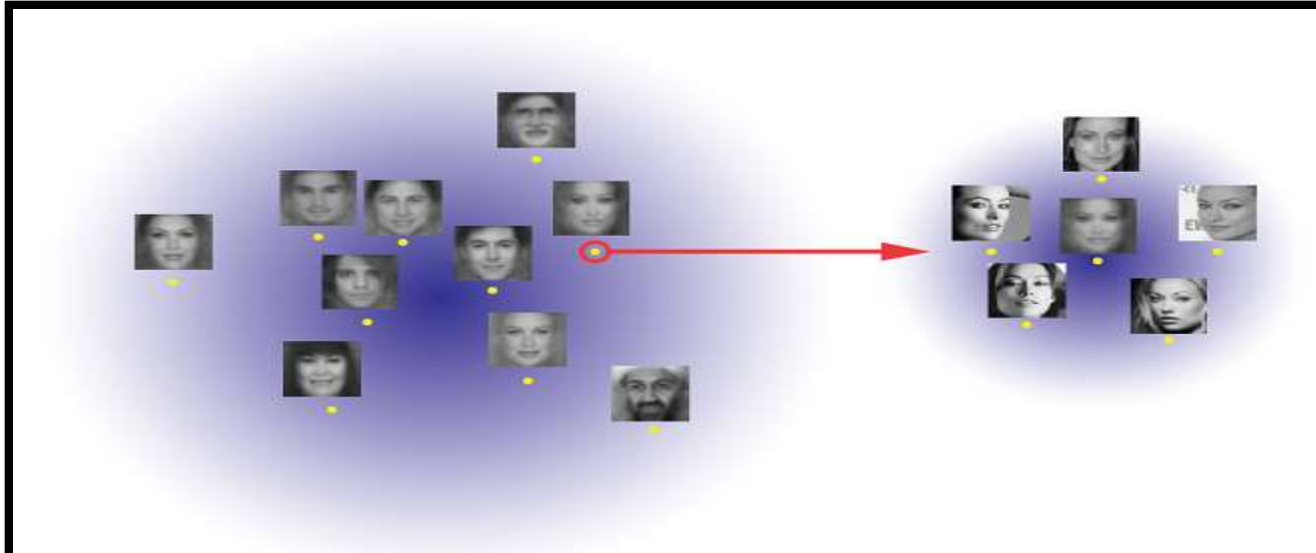
## FULLY CONNECTED LAYER

Connect the evidence to each suspect



**2. Object Detection:** It would be very useful if **robots** can **identify users** by their **faces as humans do**. Face verification system can be used very widely in tasks such as user verification for semi-personal devices, crime investigation, etc.

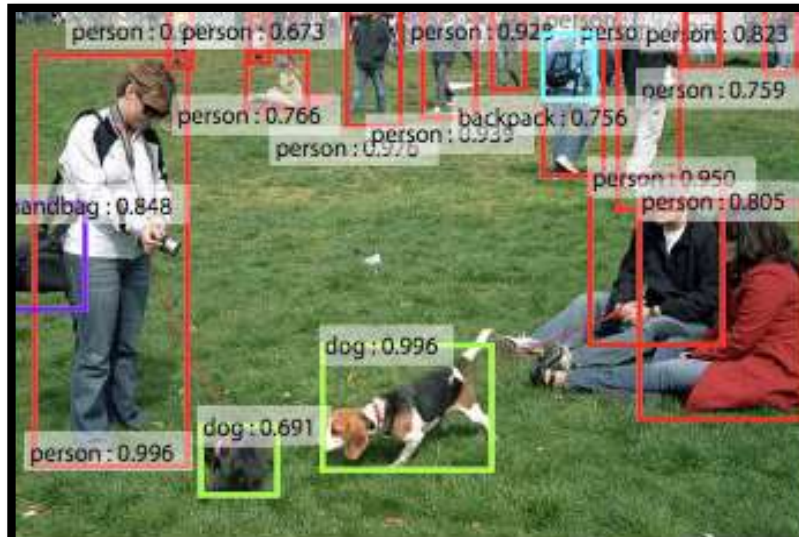
- **Convolutional Neural Network** is the most promising technique for **extracting features** and **identify the objects of the images**.
  - The most important thing in **face verification** is dealing with **similarity measurement** between the **input images** and **trained images**.
- 



## Some examples of object detection include:

- Drawing a bounding box and labeling each object in a street scene.
- Drawing a bounding box and labeling each object in an indoor photograph.
- Drawing a bounding box and labeling each object in a landscape.



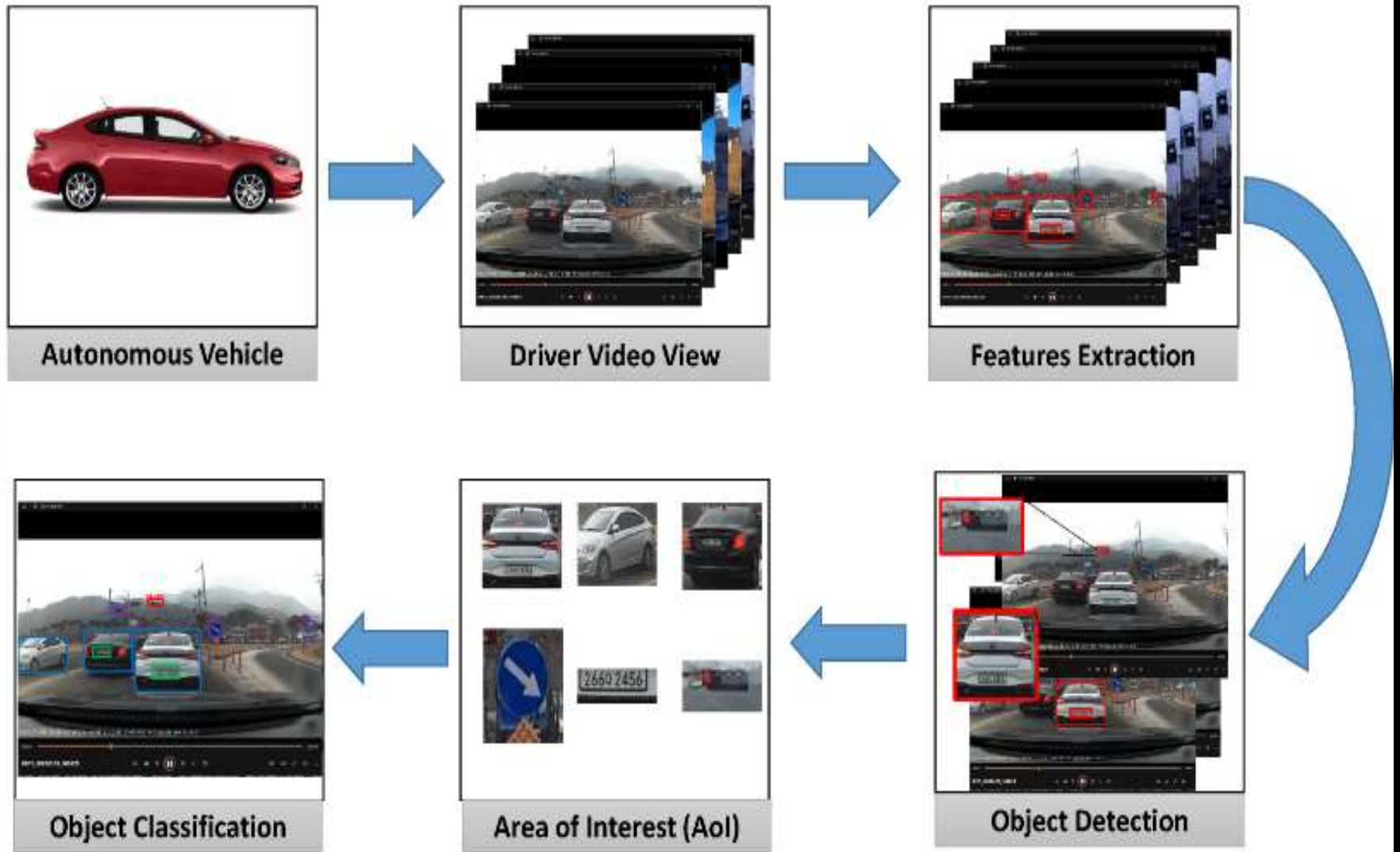


**Self Driving Cars:** Object detection for self-driving cars has to be more complex as they should be able to perform **localization, obstacle avoidance, and path planning**.

They should be **able to detect incoming collisions** and **alert the system** to **take the necessary action**.

This is a **complex problem** as CNN must both **classify the object and return the position of the bounding box of the object**. So the network designer must **avoid false alarms** and **need a huge volume of labeled training data**.

One source of such **labeled data comes from the Google Captcha system**, where users are asked to **categorize objects like traffic lights, cars, hydrants, and so on**.



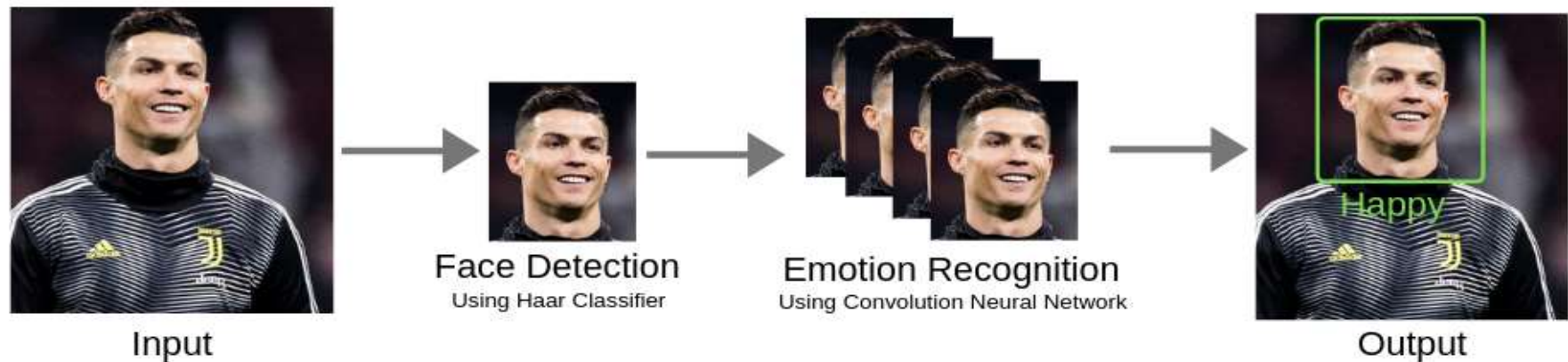
**Object Detection Procedure for Autonomous Vehicle**



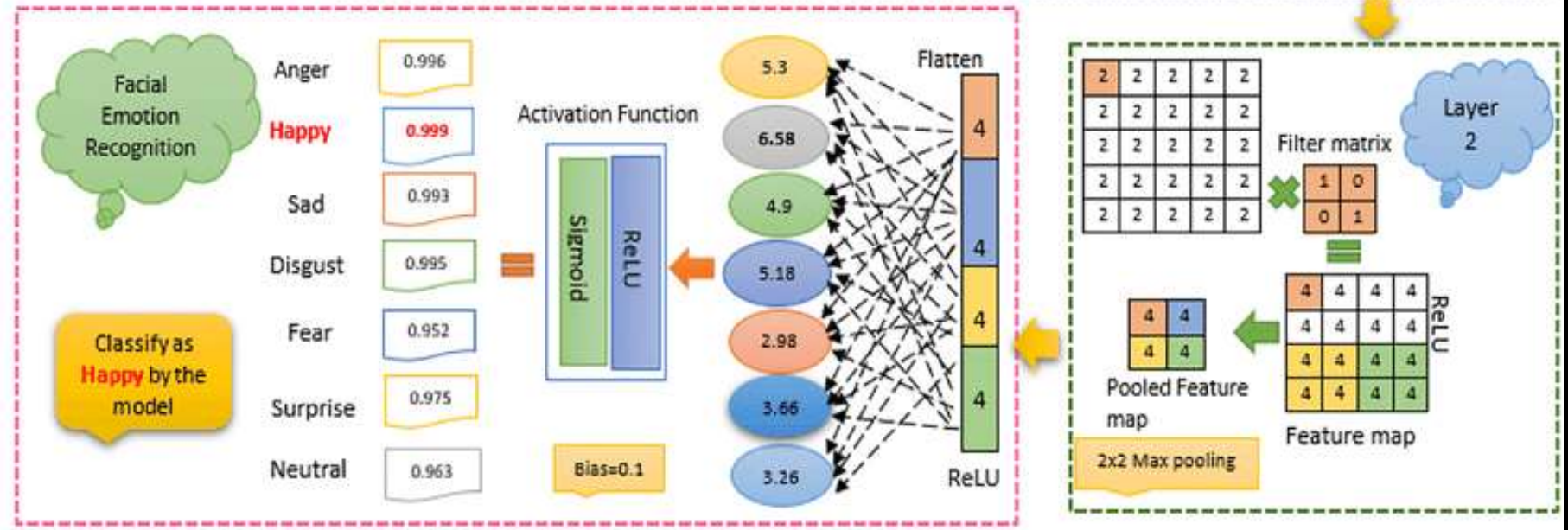
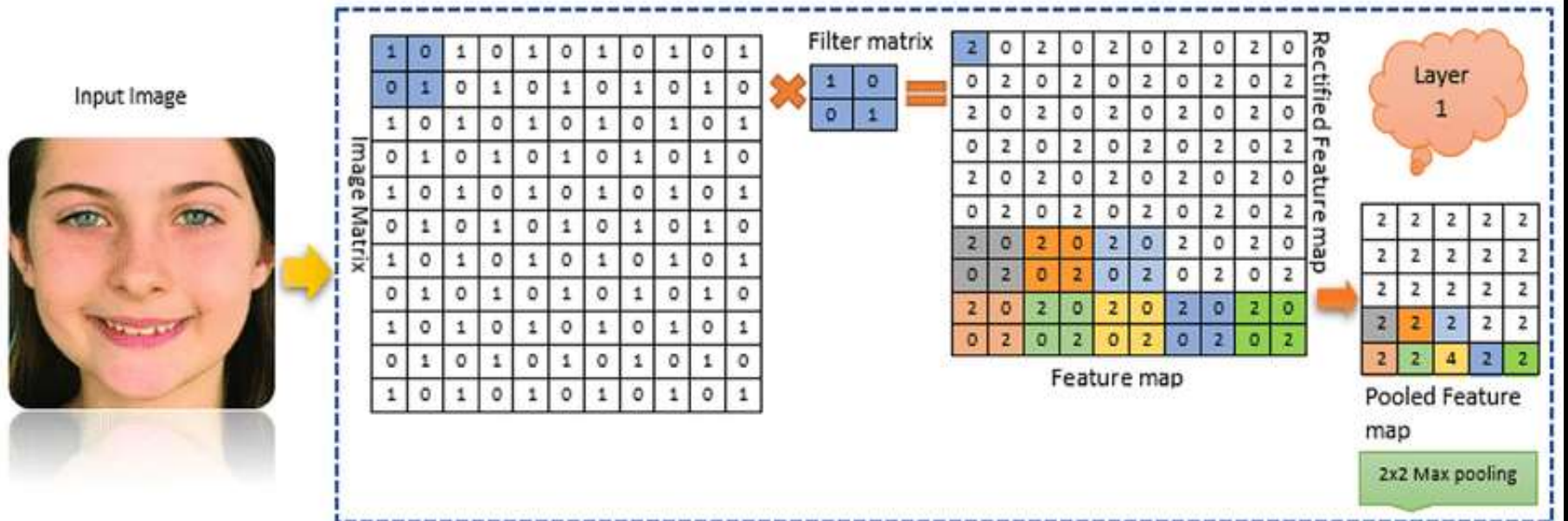
- **3. Object Segmentation:** Object segmentation, or semantic segmentation, is the task of object detection where a line is drawn around each object detected in the image. Image segmentation is a more general problem of spitting an image into segments.
- Object detection is also sometimes referred to as **object segmentation**.
- Unlike object detection that involves using a **bounding box to identify objects**, object segmentation identifies the **specific pixels in the image that belong to the object**.



- **4. Emotion Recognition:** In this Emotion Recognition we have to use the **purely Convolutional Neural Network to find out Emotion.**
- Here we have to give the input image of a person. By using the CNN we have to find out Emotions of that particular Image.







- **5. Image Reconstruction:** Image reconstruction and image inpainting is the task of **filling in missing or corrupt parts of an image.**
- Examples include reconstructing **old, damaged black and white photographs and movies (e.g. photo restoration).**
- Datasets often involve using **existing photo datasets** and creating corrupted versions of photos that models must learn to repair.



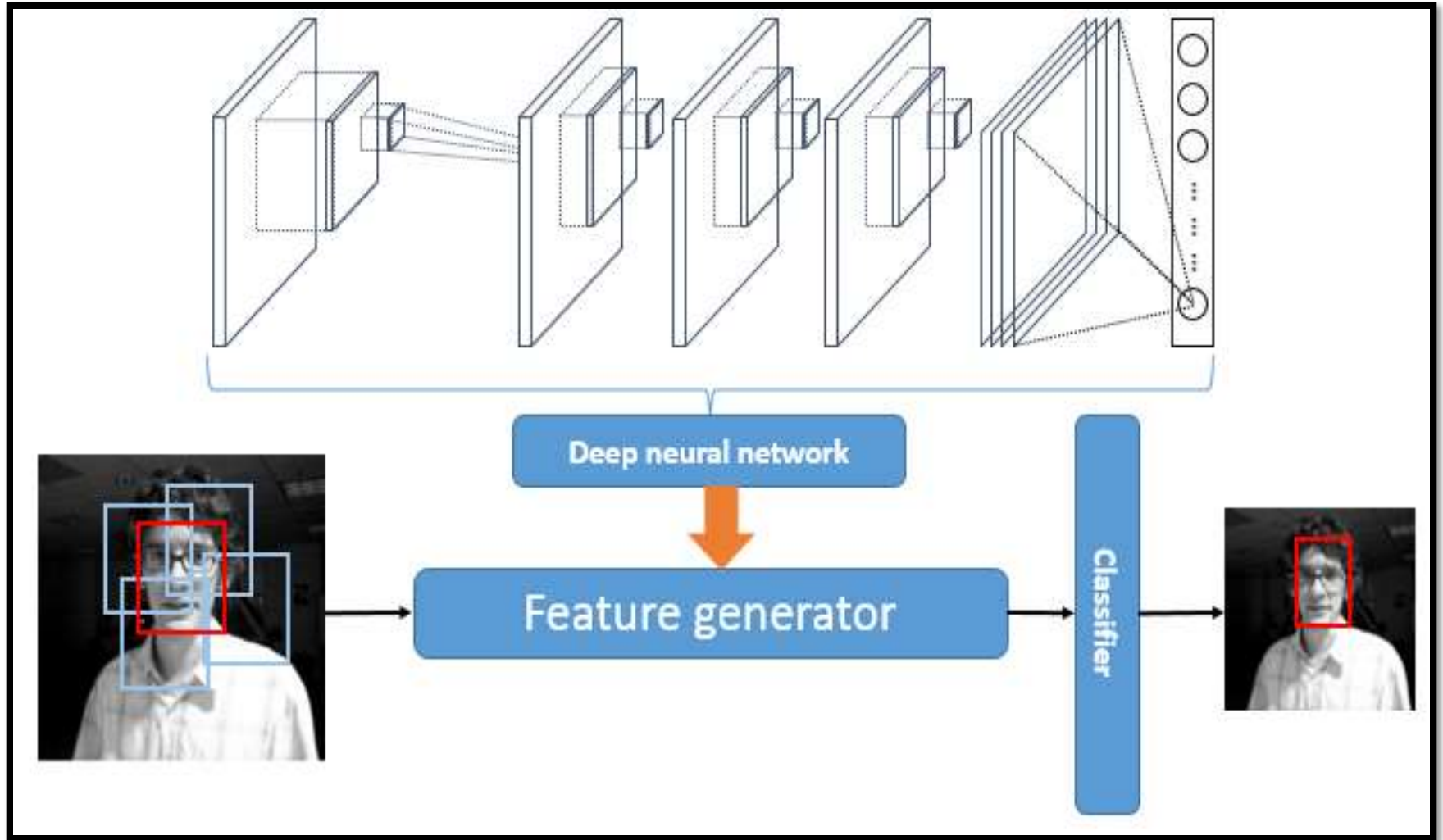
CONTD..



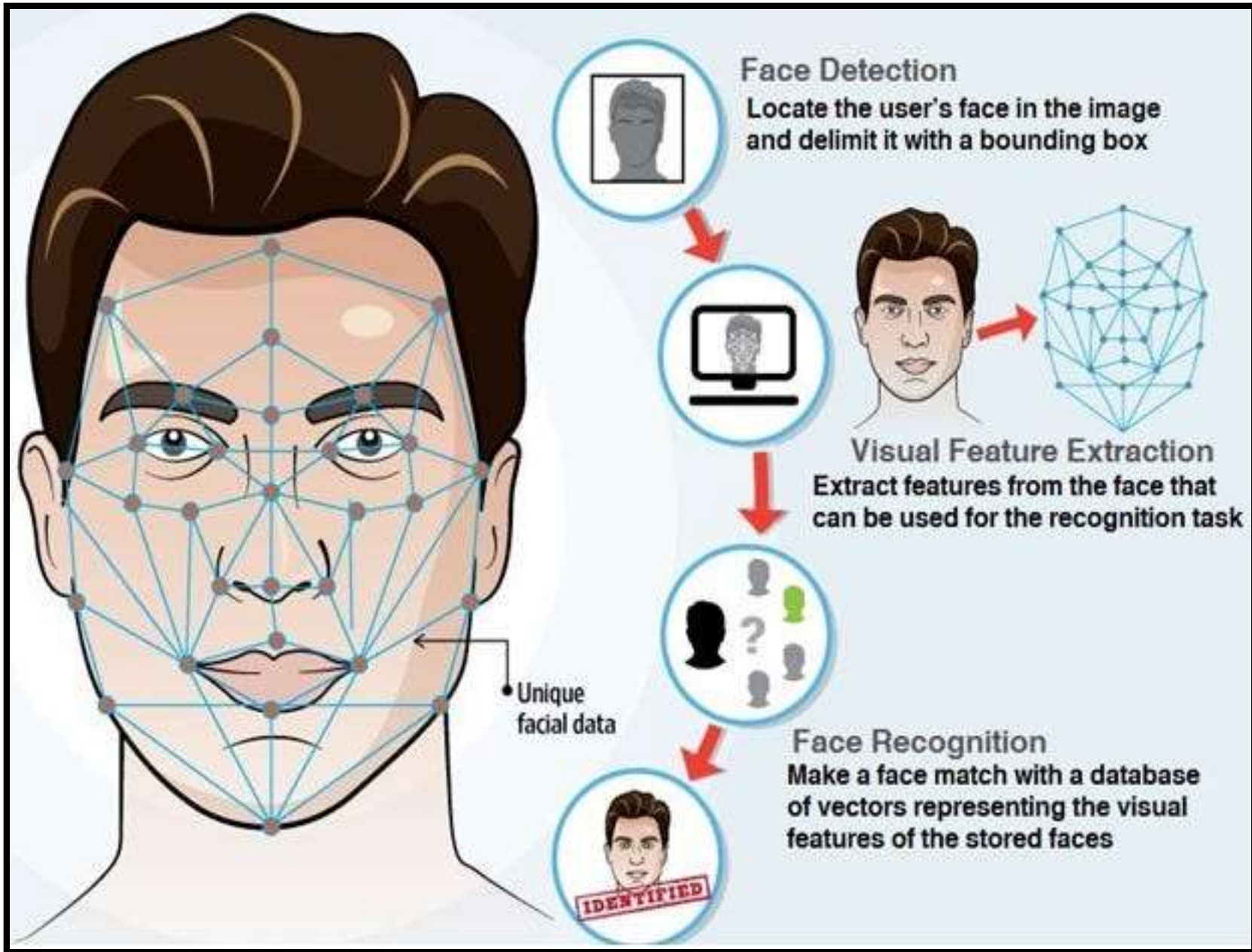


## 2. PROCESS

- **Facial Recognition:** In Facial Recognition we have to find out the **visual Tracking's of the particular Face.**
- Visual object tracking is one of the traditional problems of computer vision and comes to the important core issues with wide-ranging applications **including self-driving car and robot-vision interaction, etc.**
- First, they can consider the **tracking problem** as **classifying each video and frame by learning all dataset.**
- Second, use the deep neural network **as feature generator and use other classifiers** for using their features such as **Artificial Neural Network(ANN).**

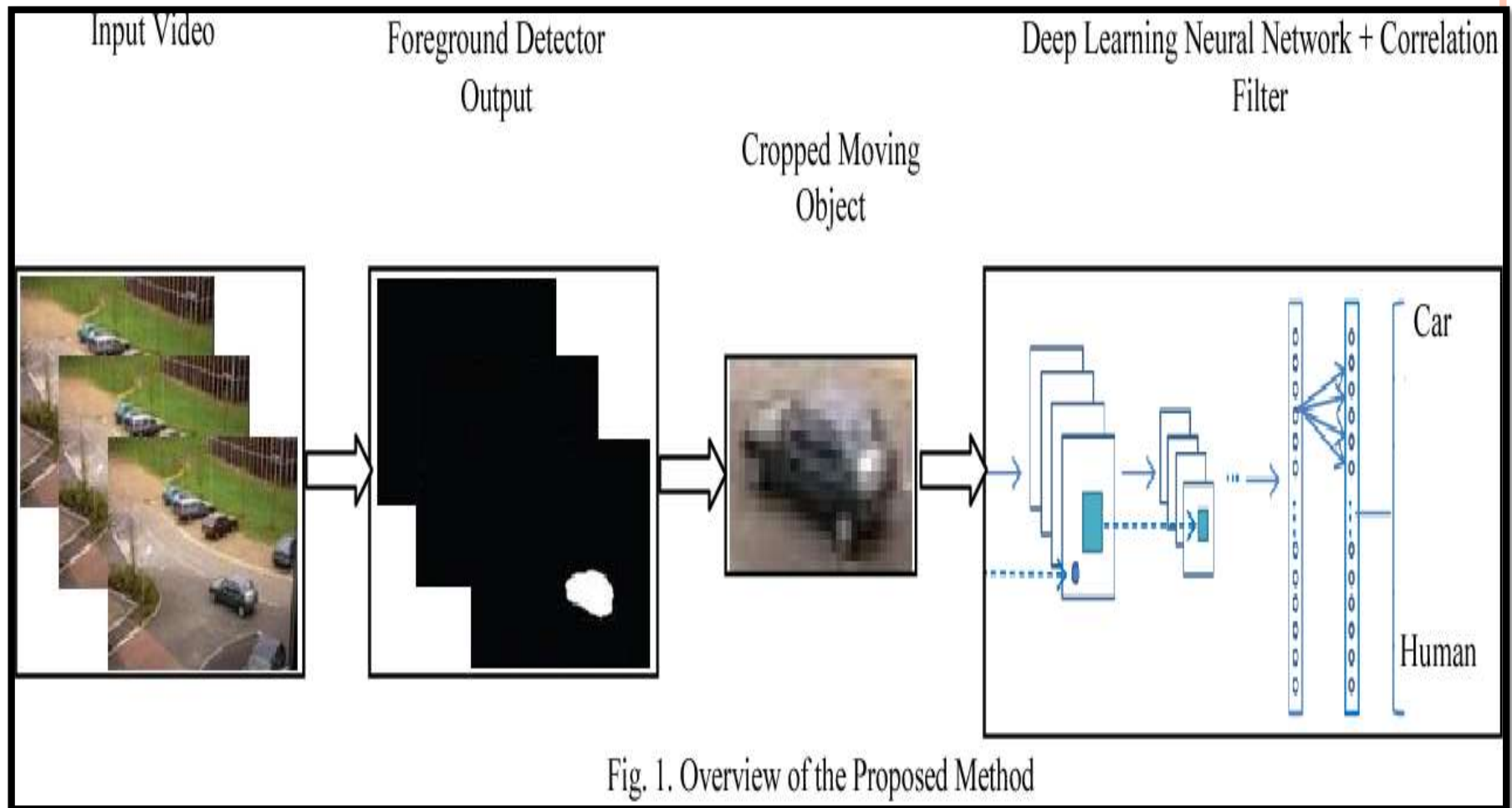






### 3. ANALYSIS

- **1. Object Tracking:** Object tracking is a key step in computer vision for **video surveillance, public safety, and traffic analysis.**
- Object detection and tracking are the two correlated **components of Video Surveillance.**
- Object detection in videos is the first step before performing **complicated tasks such as tracking.**
- Deep learning neural networks is a powerful programming paradigm which learns multiple levels **of representation and abstraction of data such as images, sound, and text.**



# DL ARCHITECTURES USED IN CV

1. **Convolutional Neural Networks:** In computer vision deep learning, Convolutional Neural Networks (CNNs) are pivotal, employing convolutional, pooling, and fully connected layers to **analyze visual data**. CNNs progressively **developed and understanding of input images**.
  - While **pooling layers increase efficiency by lowering spatial dimensions**,
  - **convolutional layers highlight picture features using trainable filters**.
  - **The fully connected layers** are used to merge spatial features and be **fed to the target task like classification**.

**2. Region-based Convolutional Neural Networks:** CNNs use **spatial information by using convolution layers**. R-CNNs solve this by using CNNs **on 'proposal areas'** for **object detection**.

Evolving versions like **Fast R-CNN, quicker R-CNN, and Mask R-CNN** (for pixel-level segmentation) enhance efficiency and effectiveness in addressing this limitation.

**3. Generative Adversarial Networks:** GANs, distinct from discriminative tasks, excel in **generating tasks**. Comprising a **discriminator and a generator**, GANs play a **minimax game** where the networks are **trained simultaneously**.



# REAL TIME EXAMPLE FOR OBJECT DETECTION

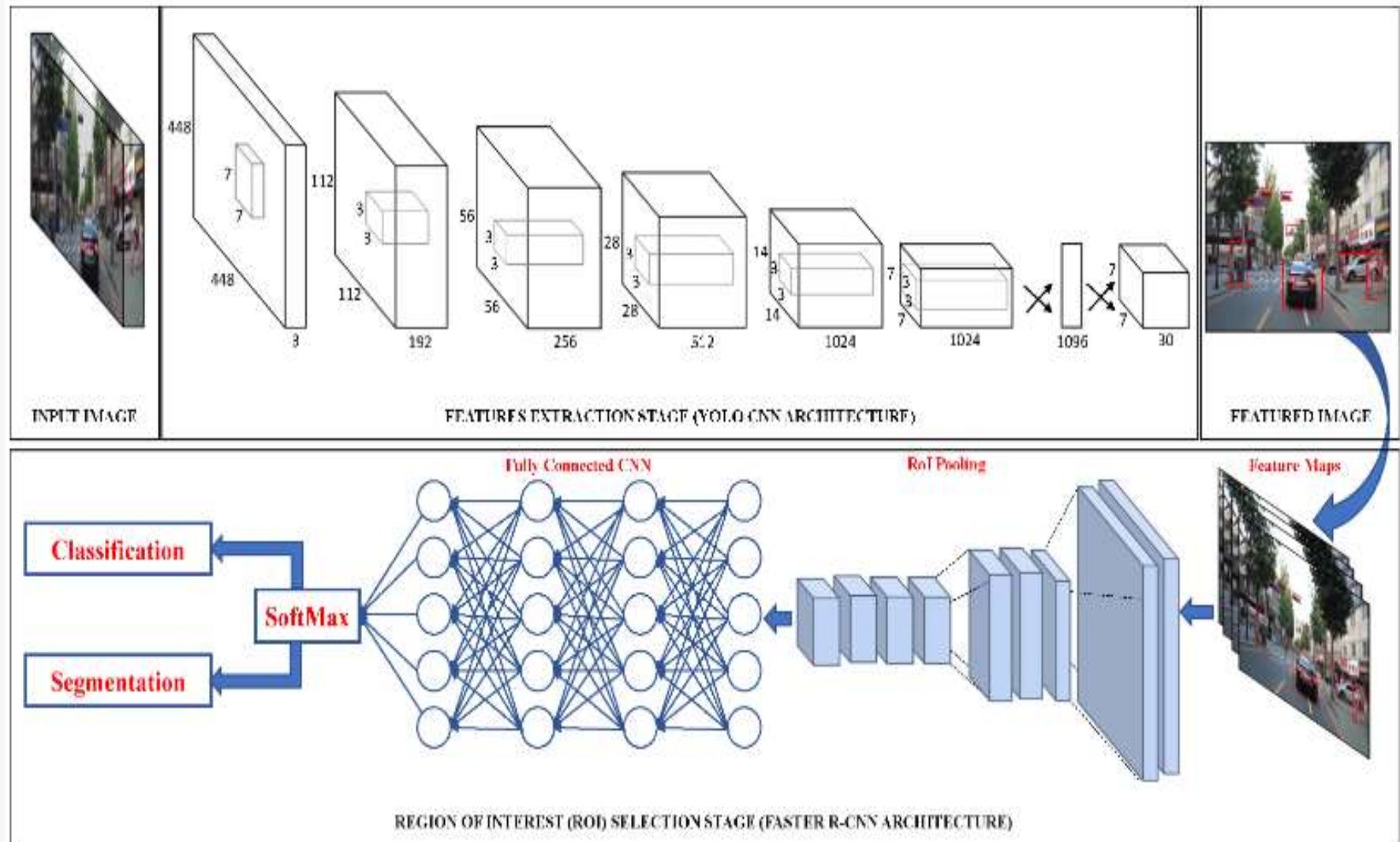
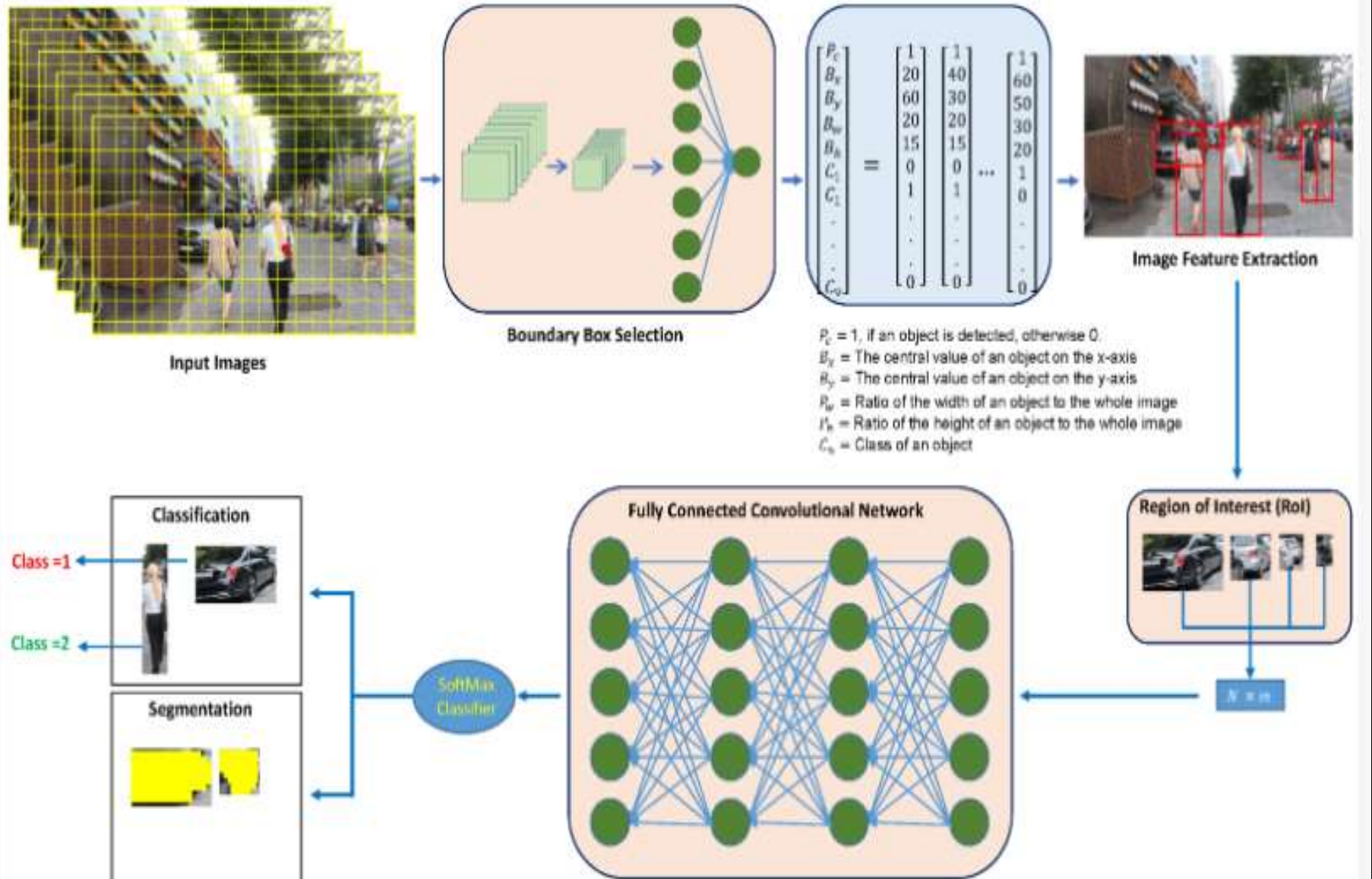


Fig: Network Structure Diagram for Object Detection

- The **first step is data collection**, where the data are collected from **driving scenarios** to **train the object detection model**.
- The **collected data are then preprocessed** to make them suitable **for training**.
- This includes **resizing images, normalizing pixel values**, and **dividing the dataset into training and validation sets** to **evaluate the model's performance**.
- Here It combines the **features of the YOLO and Faster R-CNN architectures**.

# REAL TIME EXAMPLE FOR OBJECT DETECTION





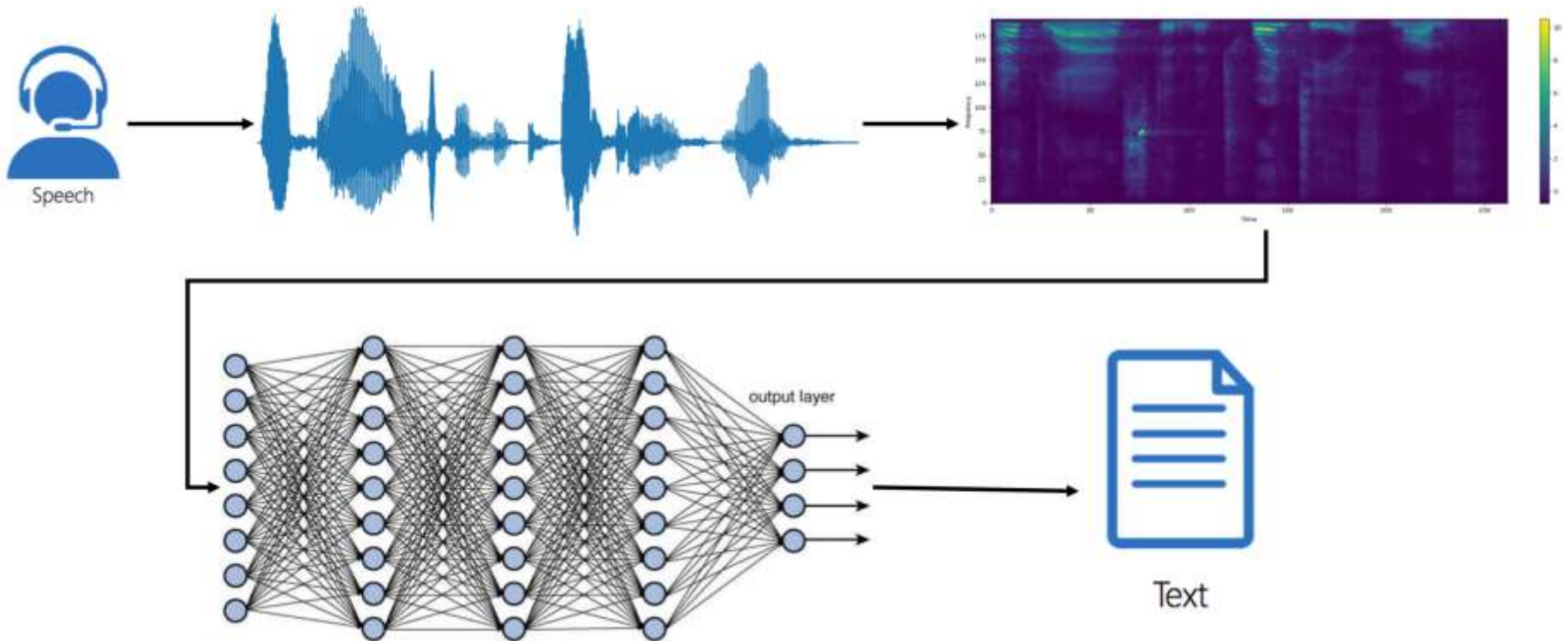


**THANK YOU**



# UNIT-V

## Speech Recognition



# CONTENTS


- ❖ Introduction
- ❖ Challenges in Speech Recognition
- ❖ Speech Recognition
- ❖ How does Speech Recognition Works
- ❖ Automatic Speech Recognition
- ❖ Where it can be Used?
- ❖ How does it Works?
- ❖ Speech Recognition Algorithms
- ❖ Applications

# WHAT IS SPEECH ?

- What is Speech: The act of **expressing or describing thoughts, feelings,** or perceptions by the articulation of words. i.e., A form of **communication in spoken language,** made by a **speaker before an audience.**
- The development of speech recognition technology dates back to **the 1940s** when it was used for **military communication and air traffic control systems.**
- **In the 1950s,** researchers developed the first commercial speech recognition system to **recognize digits spoken into a telephone.** This system was limited to identifying **only numbers, not full words or sentences.**

- In between Researchers developed so **many speech recognition tools**.
- Recently, In the 2000s, the development of **speech recognition technology** continued to advance with the development of **more accurate models** and the incorporation of **acoustic models**. This led to the development of **virtual assistant devices** such as **Google Home** and **Amazon Alexa**.
- **In the 2010s**, the development of **deep learning algorithms** further **improved the accuracy** of speech recognition models.

# CHALLENGES IN SPEECH RECOGNITION

- One of **speech recognition's main challenges** is **dealing with human speech variability**. People may have **different accents, pronunciations, and speech patterns**, and the model **must recognize this variability accurately**. Additionally, **background noises** and other **environmental factors** can interfere with the model's accuracy.
  - **Another challenge** is dealing with **words that sound similar**.
  - **For example**, the words **"to" and "too"** may sound similar but have different meanings. The model must distinguish between these words to **generate an accurate output**.
- 

- Similarly, words with multiple meanings can be difficult for the model to interpret accurately.
- Finally, the model must be able to recognize and process multiple languages. Different languages have different phonetic and grammatical rules, and the model must recognize these differences to generate an accurate output.
- All of these challenges make speech recognition a difficult task. However, with the advancement of deep learning and natural language processing, speech recognition has become more accurate and efficient.



# SPEECH RECOGNITION

- **Speech Recognition:** Speech Recognition (SR) is the ability to **translate a dictation or spoken word to text.**
- Speech recognition, also known as **automatic speech recognition (ASR)** also known as **speech-to-text, speech recognition, or voice recognition**, is a technology that converts **spoken language into written text.**
- A **text-to-speech (TTS) system**, also known as **speech synthesis**. This turns a **text into a verbal, audio form.**

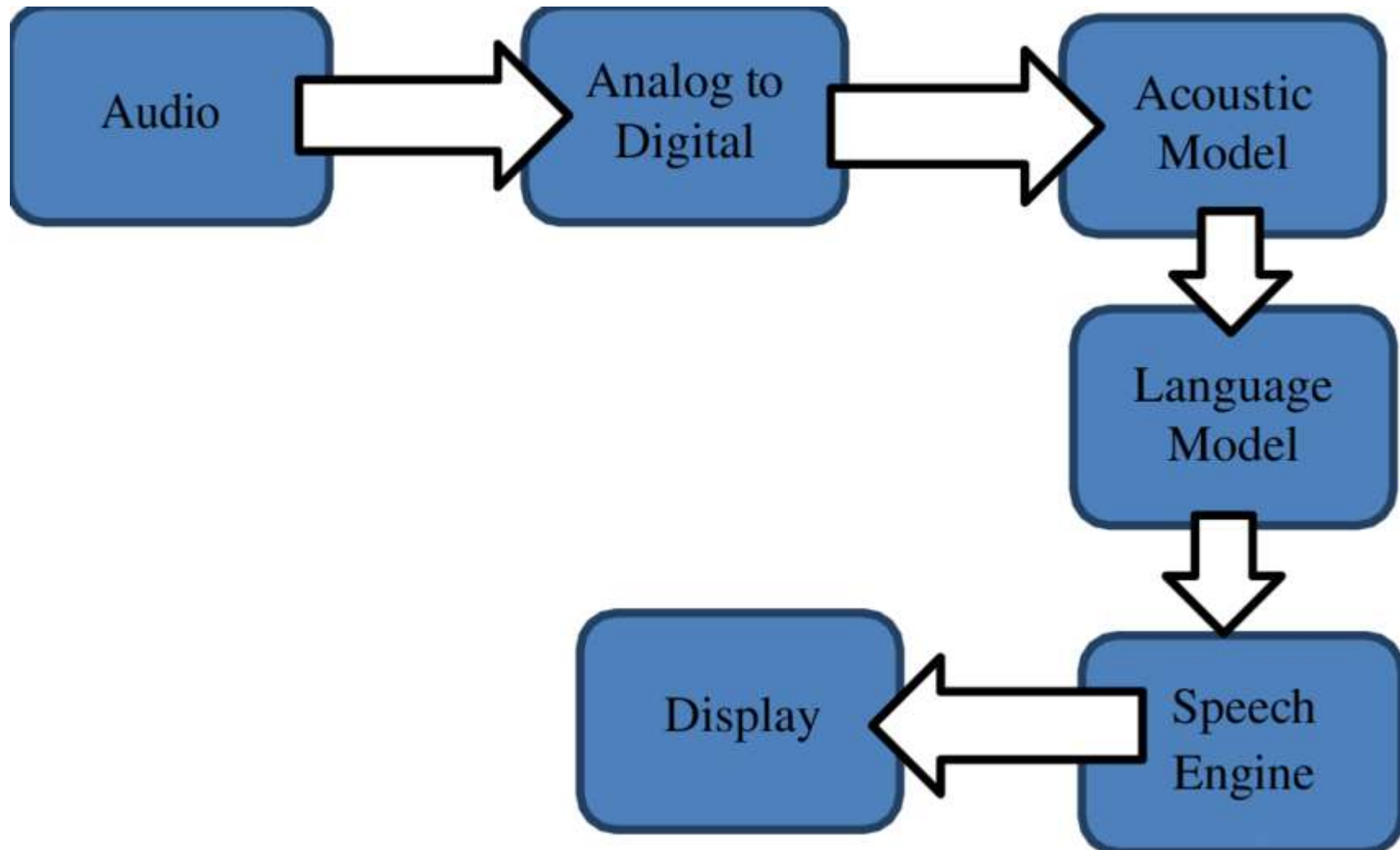




- The **primary goal of speech recognition systems** is to **accurately and efficiently transcribe spoken words** into a **format that can be processed, stored, or used for various applications**. This technology relies on sophisticated algorithms and **Deep Learning techniques** to **interpret and understand human speech patterns**.
- **Automatic speech recognition (ASR)** refers to the **task of recognizing human speech** and **translating it into text**.

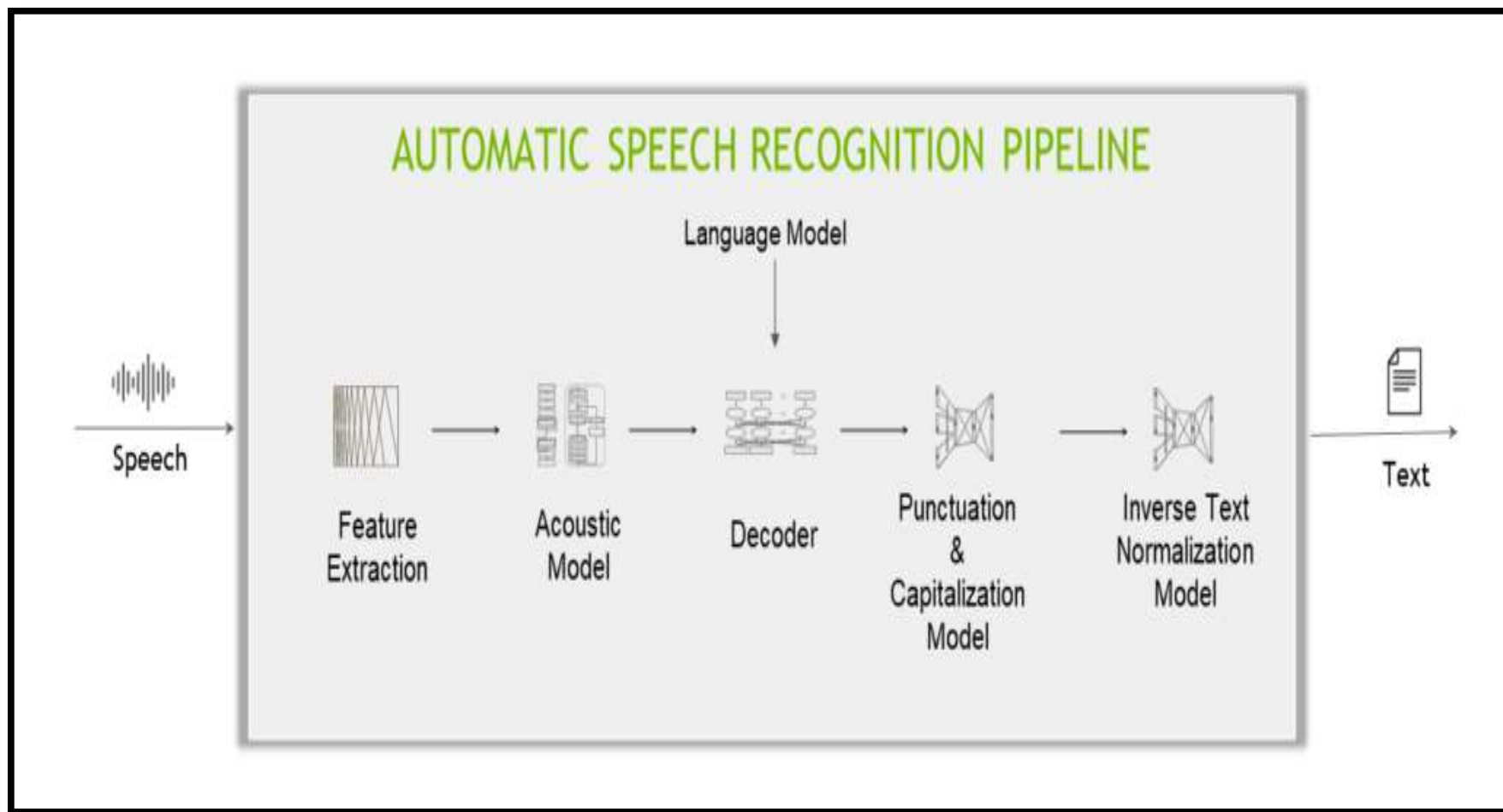


# HOW DOES SPEECH RECOGNITION WORKS?



# AUTOMATIC SPEECH RECOGNITION

- A typical deep learning-based **ASR pipeline** includes **five main components**.



## Feature extractor:

**Spectral Analysis:** The digital signal undergoes spectral analysis to **extract relevant features**. This involves **breaking down the signal into frequency components**, revealing patterns representing speech sound characteristics.

**Pitch and Intensity Analysis:** Additional features, such as pitch (frequency of the speech) and intensity (loudness), are extracted to capture more **nuances of the spoken language**.



**2. Acoustic Signal Processing:** This Deep learning model (usually a **multi-layer deep neural network**) predicts the probabilities over characters at each time step of the audio data. Here, we have to work, **Acoustic Signal Processing. In the we used,**

**Capture the Analog Signal:** The process begins with a **microphone capturing the analogue signal of spoken words**. This analogue signal is a continuous waveform representing the variations in air pressure caused by speech.

**Analogue-to-Digital Conversion:** The **captured analogue signal is converted into a digital format** through **analog-to-digital conversion**. This digital signal becomes the **input for further processing**.

## Acoustic Model:

**Statistical Models:** An acoustic model is a **statistical model trained on vast datasets of audio recordings**. It learns to associate acoustic features with phonemes, the smallest sound units in a language.

**Hidden Markov Models (HMMs)** or **Deep Neural Networks (DNNs)**: Traditional models like HMMs or more modern approaches like DNNs represent the probabilities of transitioning between different phonemes over time.






3. Decoder and language model: A decoder converts the **matrix of probabilities** given by the **acoustic model** into a **sequence of characters**, which in turn **make words and sentences**.

**In Language Model, we have to use:**


**Context and Structure:** A language model complements the acoustic model by providing information about the structure and context of the **spoken language**. It helps the system distinguish between **words that may sound similar but have different meanings**.



3. Decoder and language model: A decoder converts the **matrix of probabilities** given by the **acoustic model** into a **sequence of characters**, which in turn **make words and sentences**.

**In Language Model, we have to use:**

**Context and Structure:** A language model complements the acoustic model by providing information about the structure and context of the **spoken language**. It helps the system distinguish between **words that may sound similar but have different meanings**.




**Grammar and Syntax:** The language model incorporates grammar and syntax rules, enabling the system to generate more accurate transcriptions based on the context of the spoken words.

**4. Punctuation and capitalization model:** The punctuation and capitalization model adds **punctuations and capitalizes the decoder-produced text.**

**5. Inverse text normalization model:** Lastly, inverse text normalization (ITN) rules are applied to **transform the text in verbal format** into a **desired written format.**

**for example,** “ten o’clock” to “10:00,” or “ten dollars” to “\$10”.

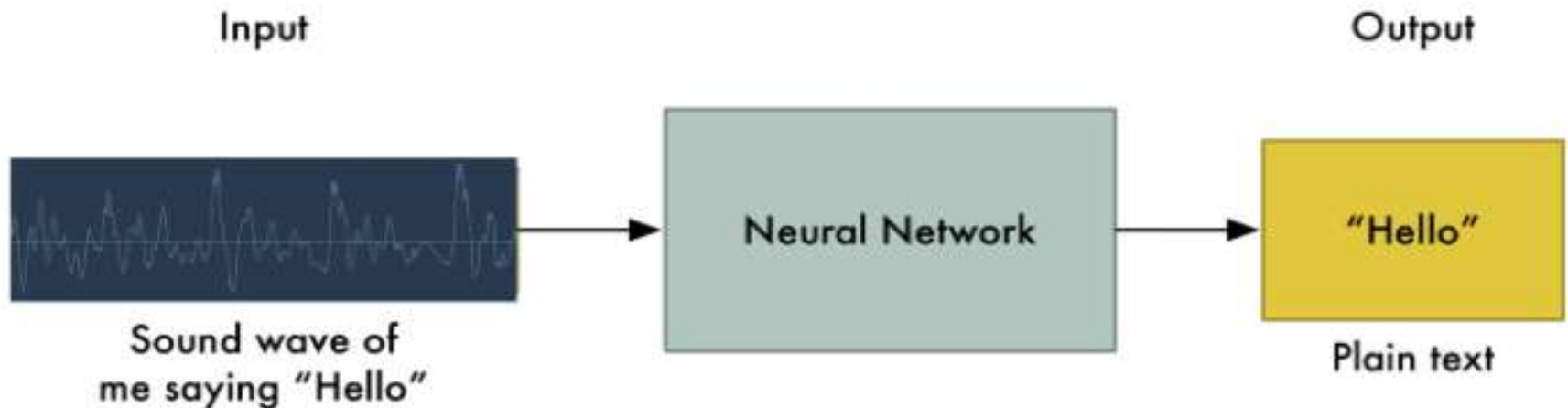
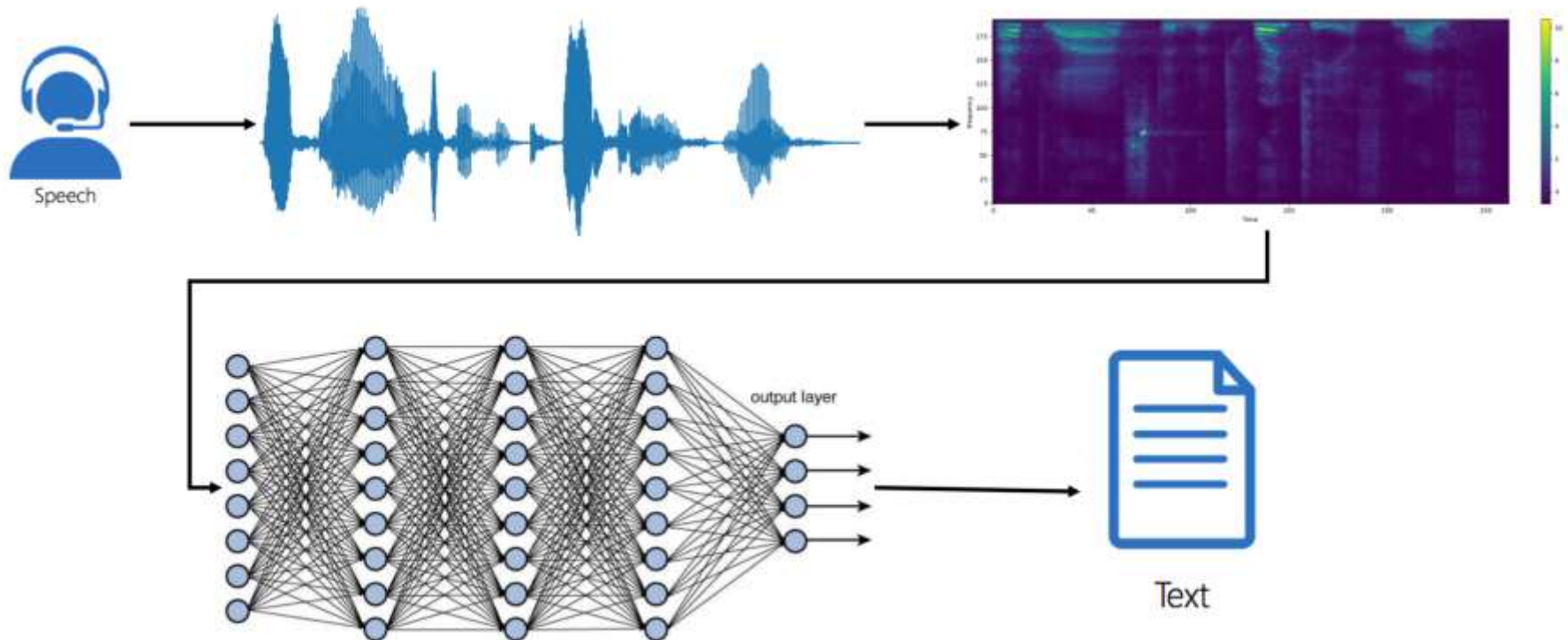


# WHERE IT CAN BE USED?

- Dictation
- System control/navigation
- Commercial/Industrial applications
- Personal Computers
- Health Care
- Telephony - Smart-phones - Customer Helpline Services



# HOW DOES IT WORKS?



- The big problem is that **speech varies in speed.**
- One person might say **“hello!”** **very quickly** and another person might say **“heeeelllllllllllooooo!”** **very slowly**, producing a much **longer sound file with much more data.**
- Both both sound files should be recognized as exactly the **same text – “hello!”**
- Automatically aligning **audio files of various lengths** to a fixed-length piece of text turns out to be pretty hard.

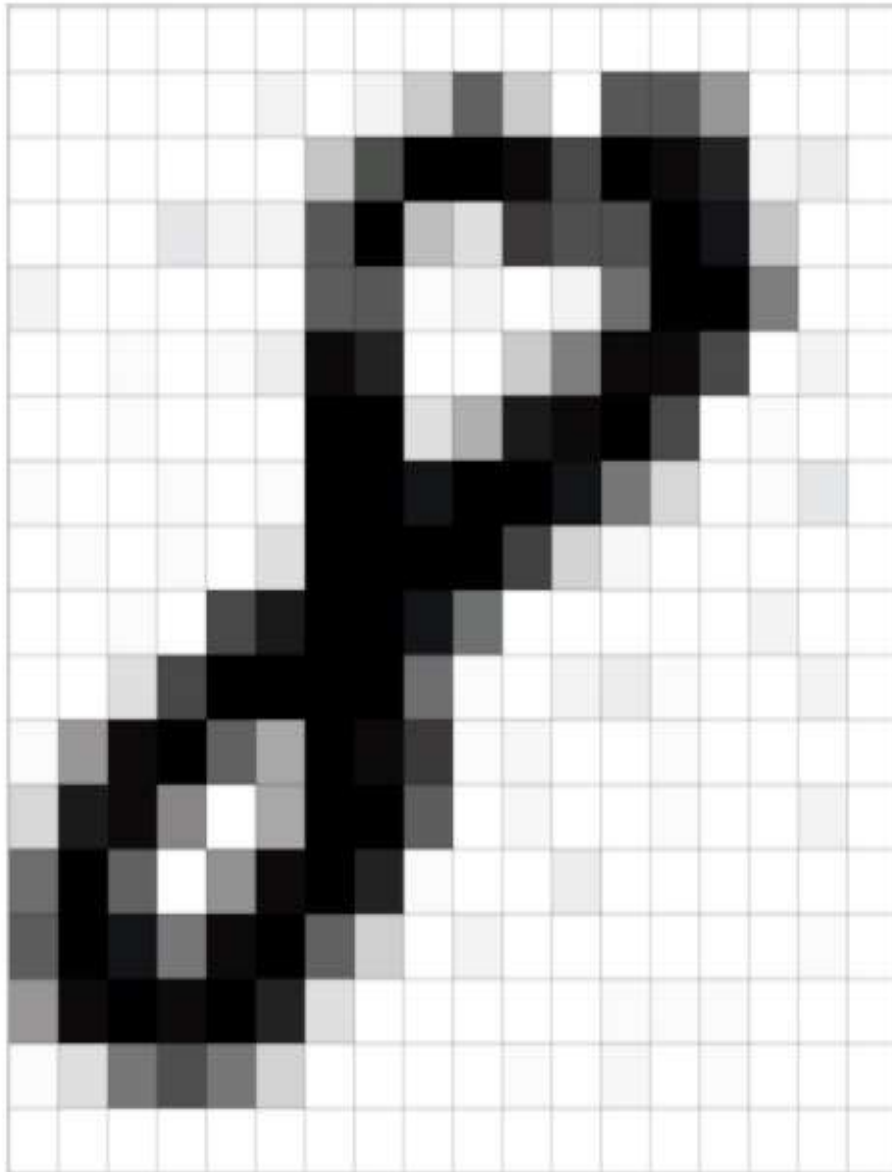




- **Step-1: Turning Sounds into Bits:** The first step in speech recognition is obvious — we need to feed sound waves into a computer.
- **For Eg:** how to take an image and treat it as an array of numbers so that we can feed directly into a neural network for image recognition:



# CONTD..

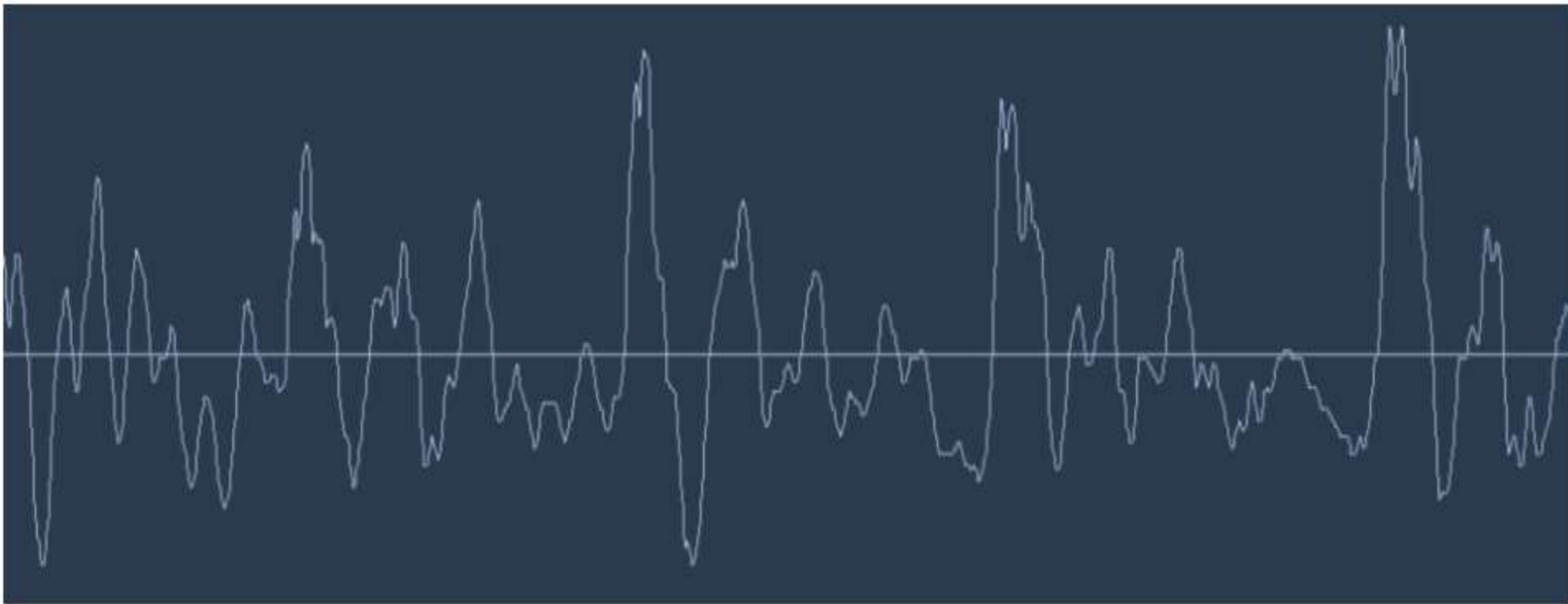


Images are just arrays of numbers that encode the intensity of each pixel

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	12	0	11	39	137	37	0	152	147	84	0	0	0	0
0	0	1	0	0	0	41	160	250	255	235	162	255	238	206	11	13	0	0
0	0	0	16	9	9	150	251	45	21	184	159	154	255	233	40	0	0	0
10	0	0	0	0	0	145	146	3	10	0	11	124	253	255	107	0	0	0
0	0	3	0	4	15	236	216	0	0	38	109	247	240	169	0	11	0	0
1	0	2	0	0	0	253	253	23	62	224	241	255	164	0	5	0	0	0
6	0	0	4	0	3	252	250	228	255	255	234	112	28	0	2	17	0	0
0	2	1	4	0	21	255	253	251	255	172	31	8	0	1	0	0	0	0
0	0	4	0	163	225	251	255	229	120	0	0	0	0	0	11	0	0	0
0	0	21	162	255	255	254	255	126	6	0	10	14	6	0	0	9	0	0
3	79	242	255	141	66	255	245	189	7	8	0	0	5	0	0	0	0	0
26	221	237	98	0	67	251	255	144	0	8	0	0	7	0	0	11	0	0
125	255	141	0	87	244	255	208	3	0	0	13	0	1	0	1	0	0	0
145	248	228	116	235	255	141	34	0	11	0	1	0	0	0	1	3	0	0
85	237	253	246	255	210	21	1	0	1	0	0	6	2	4	0	0	0	0
6	23	112	157	114	32	0	0	0	0	2	0	8	0	7	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

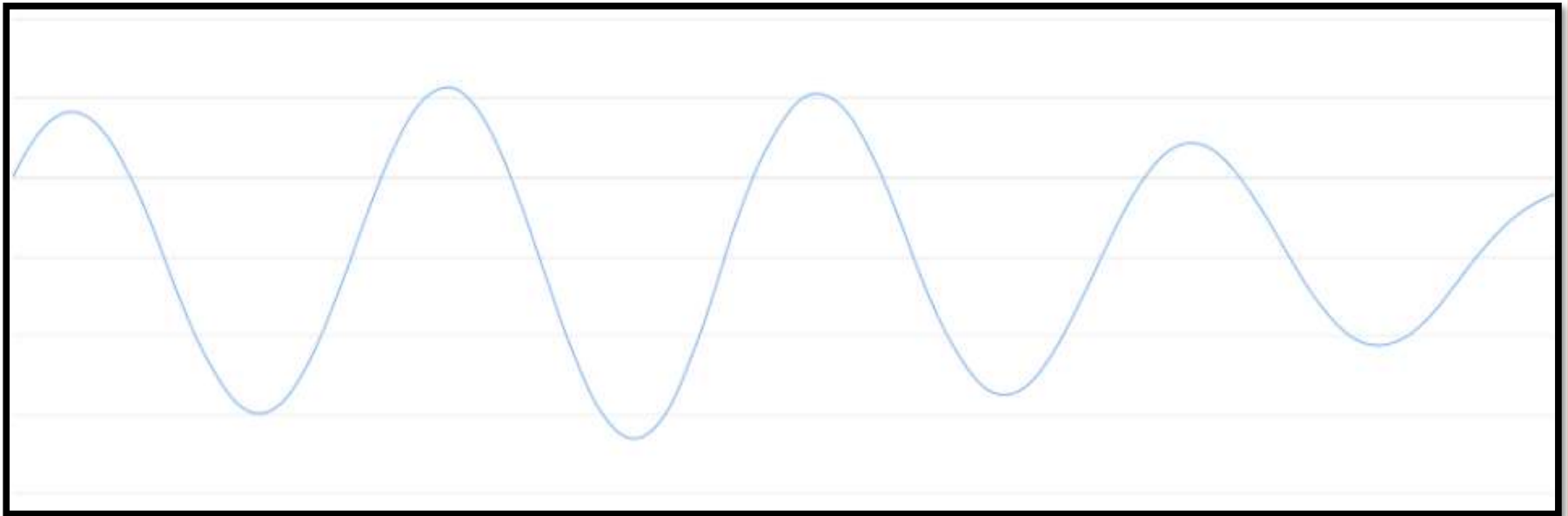
Images are just arrays of numbers that encode the intensity of each pixel

- But sound is transmitted as **waves**. How do we turn sound waves into numbers? Let's use this sound clip of me saying **"Hello"**:

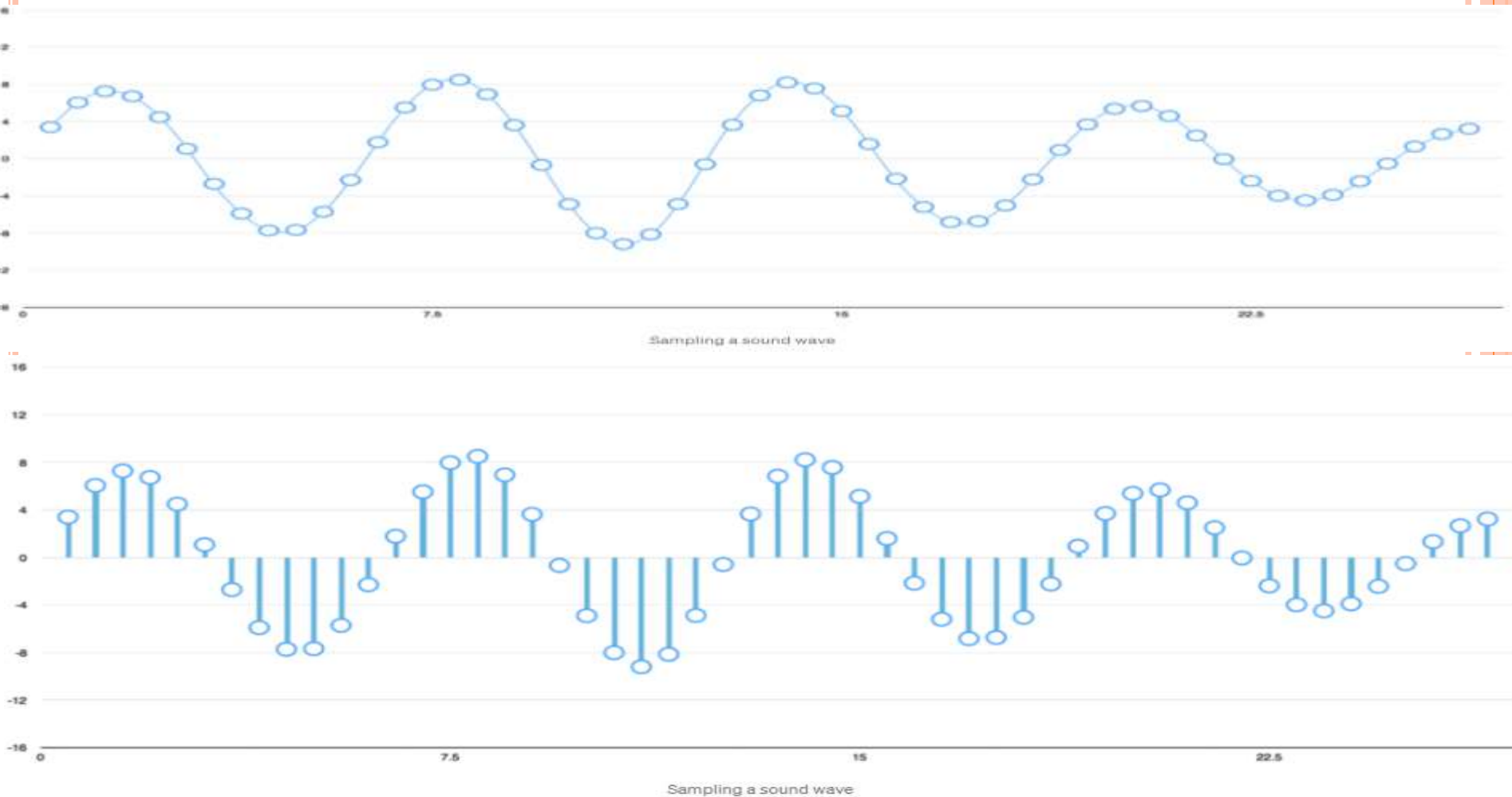


A waveform of me saying "Hello"

- Sound waves are **one-dimensional**. At every moment in time, they have a **single value based on the height of the wave**. Let's zoom in on one tiny part of the sound wave and take a look( **These are Analog Signals**)



- So first converted Analog to Digital, To **turn this sound wave into numbers**, we just **record of the height of the wave at equally-spaced points**: we are **saved the file name .wav file**



- By using the speech recognition, **a sampling rate of 16khz** (16,000 samples per second) is enough to cover the frequency range of human speech.
- Lets sample our “Hello” sound wave 16,000 times per second. Here’s the first 100 samples:

```
[-1274, -1252, -1160, -986, -792, -692, -614, -429, -286, -134, -57, -41, -169, -456, -450, -541, -761, -1067, -1231, -1047, -952, -645, -489, -448, -397, -212, 193, 114, -17, -110, 128, 261, 198, 390, 461, 772, 948, 1451, 1974, 2624, 3793, 4968, 5939, 6057, 6581, 7302, 7640, 7223, 6119, 5461, 4820, 4353, 3611, 2740, 2004, 1349, 1178, 1085, 901, 301, -262, -499, -488, -707, -1406, -1997, -2377, -2494, -2605, -2675, -2627, -2500, -2148, -1648, -970, -364, 13, 260, 494, 788, 1011, 938, 717, 507, 323, 324, 325, 350, 103, -113, 64, 176, 93, -249, -461, -606, -909, -1159, -1307, -1544]
```

Each number represents the amplitude of the sound wave at 1/16000th of a second intervals



## ➤ Step-2: Pre-processing the Sample sound data

- We could feed these numbers right **into a neural network**. But trying to recognize speech patterns by processing these samples **directly is difficult**. Instead, we can make the problem easier by doing some **pre-processing on the audio data**.
- To make this data easier for a neural network to process, we are going to **break apart this complex sound wave into it's component parts**. We'll **break out the low-pitched parts, the next-lowest-pitched-parts, and so on**. Then by adding up how much energy is in each of those frequency bands (from low to high),

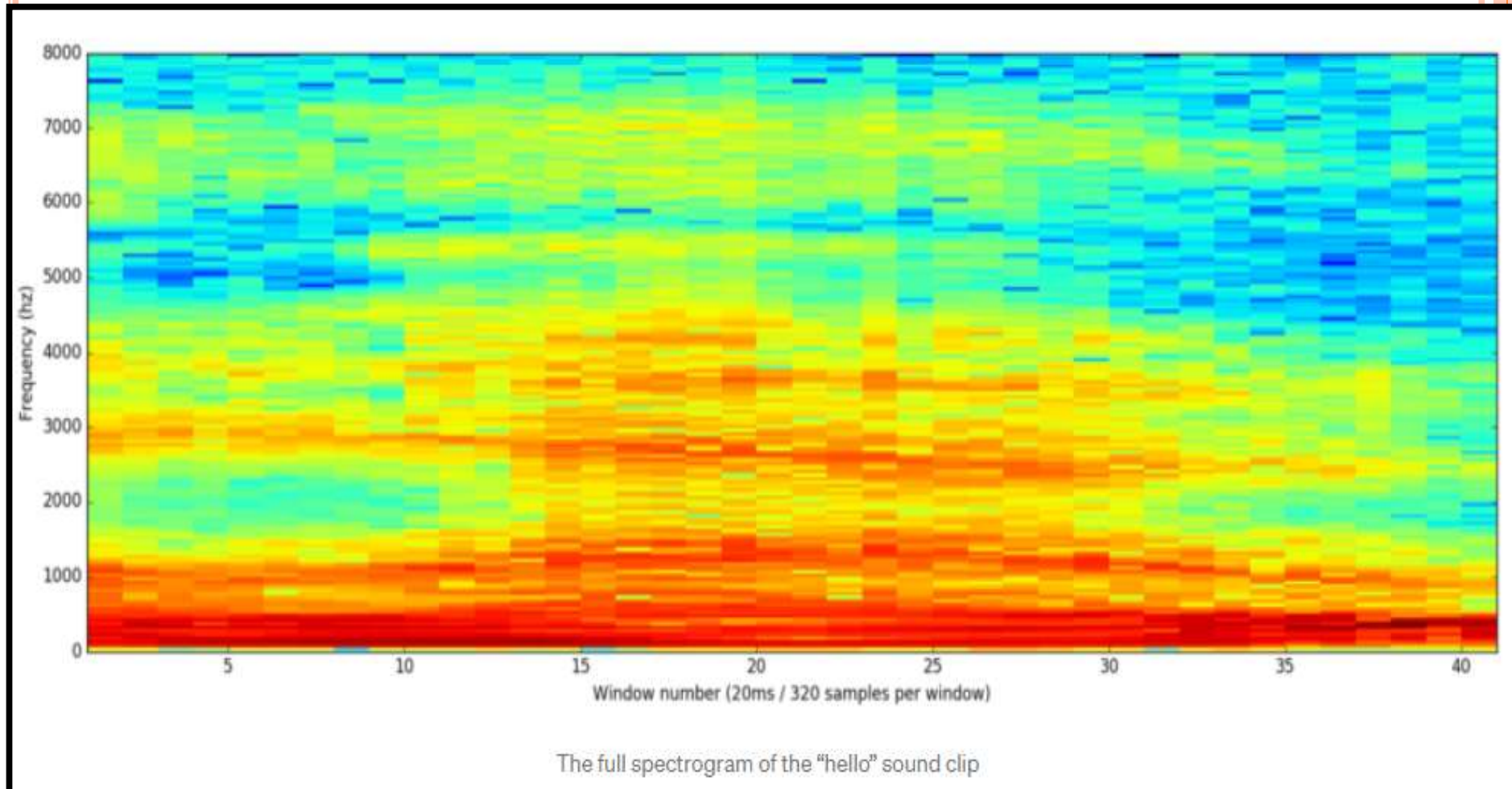


175.09309469913353, 180.0168691095916, 176.00619977472167, 179.79737781786582, 173.53025213548219, 176.87177119846058, 170.426847328531, 154.34196586290136, 151.46179061113972, 152.99674239973979, 143.98878156117371, 156.6033737693738, 155.78237530428544, 157.17930941017, 147.23266451005145, 133.26597973863801, 116.5170100028831, 116.85501120577126, 115.40519005123537, 120.85619013711488, 112.48406123161, 95.673146446282971, 90.391748128064208, 79.355818055314899, 86.080143147713926, 84.748200268709567, 83.050569583779065, 86.2071802622, 96, 90.746777849123049, 86.726552726337033, 85.709412745066928, 95.938840816664865, 99.09254575917069, 96.632437741434885, 103.23961231, 996, 129.20890691592615, 130.43460361780441, 138.15581799444712, 128.25056761852832, 138.14492240466387, 140.0352714810314, 128.1513813, 2509, 114.23027889344033, 119.1717342154997, 101.02560719093093, 110.91192243698025, 106.04872005953503, 100.86977927980999, 92.1233019, 77845, 110.24526597732718, 113.72249347908021, 120.63960942628063, 122.06482553759932, 117.96716716036715, 120.87682744817975, 125.0609, 0130265, 114.40659619324526, 79.869543980883975, 104.83111191845597, 104.66218602004588, 104.91691734582642, 97.143620527536072, 78.434, 52360313, 74.100307226086798, 64.861423011415653, 59.167561212002269, 62.479712687304911, 63.568362396107467, 55.906096471453267, 42.79, 1220671298, 51.062413666348945, 58.493563858289065, 53.081835042922769, 73.060663128159547, 68.21625202122361, 66.7701034934517, 59.760, 45346381, 44.910670465379937, 59.282513769840705, 69.241393677323856, 81.778634874076346, 88.409923803546008, 94.688033733251245, 96.64, 0315589074, 97.899164767741183, 75.176507616277235, 80.947474423758905, 71.859103451990862, 93.863684037461738, 96.757146539348298, 96, 596663023235, 101.78493139911082, 103.7883358299547, 99.915220403870748, 107.43478470929935, 104.46449552620618, 105.70789868195298, 10, 7278943060093, 90.936627732905492, 71.134275744339803, 72.504304977841457, 76.233185506299705, 63.281284410272761, 45.380164336858961, 586423555688746, -4.4730776113028883, 50.833000650183408, 51.003802143009629, 39.577356593427531, 47.096919248906332, 55.44219717566438

## Data Preprocessing

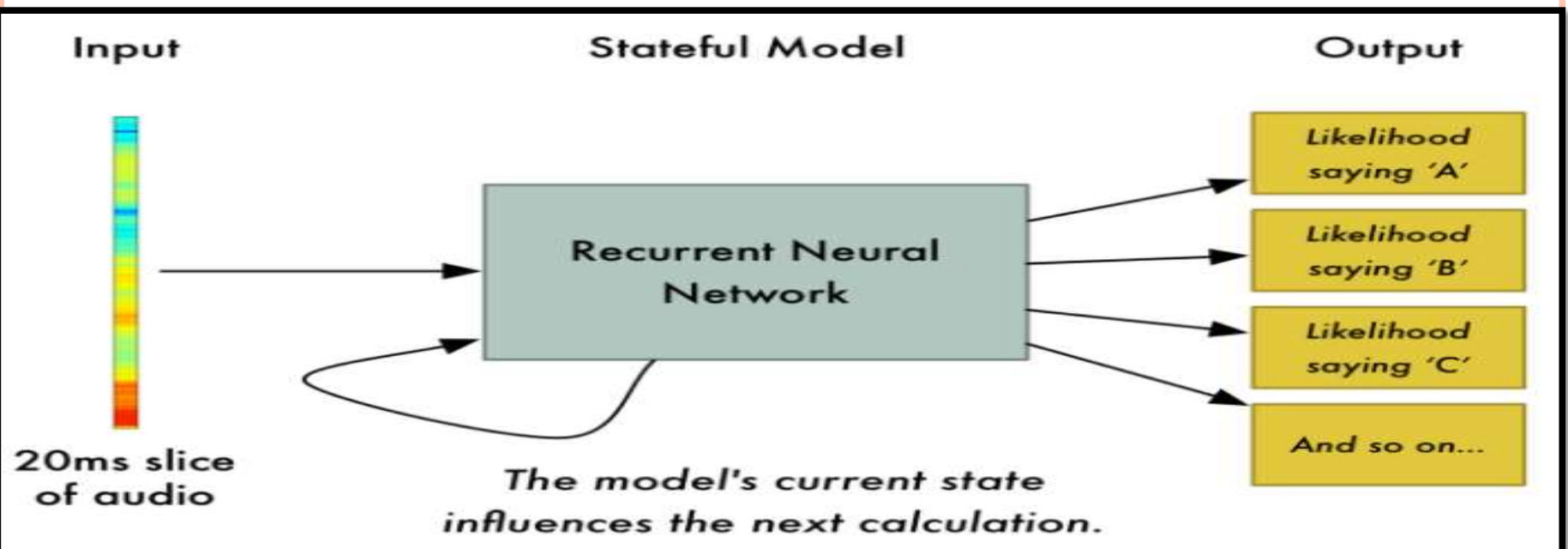


- But this is a lot easier to see when you draw this as a chart:



### ➤ Step-3: Recognizing Characters from Short Sounds

- Here we have to **apply deep neural network**. The **input to the neural network** will be **20 millisecond audio chunks**. For each little audio slice, it will try to figure out the letter that corresponds the sound currently being spoken.



- We'll use a recurrent neural network — that is, a neural network that has a **memory that influences future predictions**.
- That's because each letter it predicts should affect the likelihood of the **next letter it will predict too**.
- **For example**, if we have said **"HEL"** so far, it's very likely we will say **"LO"** next to finish out the word **"Hello"**.



- After we run our **entire audio clip** through the **neural network (one chunk at a time)**, we'll end up with a mapping of each audio chunk to the letters most likely spoken during that chunk. Here's what that mapping looks like for me saying **"Hello"**:





# CONTD..



Most likely letter:  
(per 20 milliseconds)

□ □ H H H E E \_ L L \_ \_ L L L O O O □ □ □ □

- Our neural net is predicting that one likely thing I said was “HHHEE\_LL\_LLLOOO”. But it also thinks that it was possible that I said “HHHUU\_LL\_LLLOOO” or even “AAAUU\_LL\_LLLOOO”.
- We have some steps we follow to clean up this output. First, we'll replace any repeated characters a single character:

**HHHEE\_LL\_LLLOOO becomes HE\_L\_LO**

**HHHUU\_LL\_LLLOOO becomes HU\_L\_LO**

**AAAUU\_LL\_LLLOOO becomes AU\_L\_LO**

- Then we'll remove any blanks:

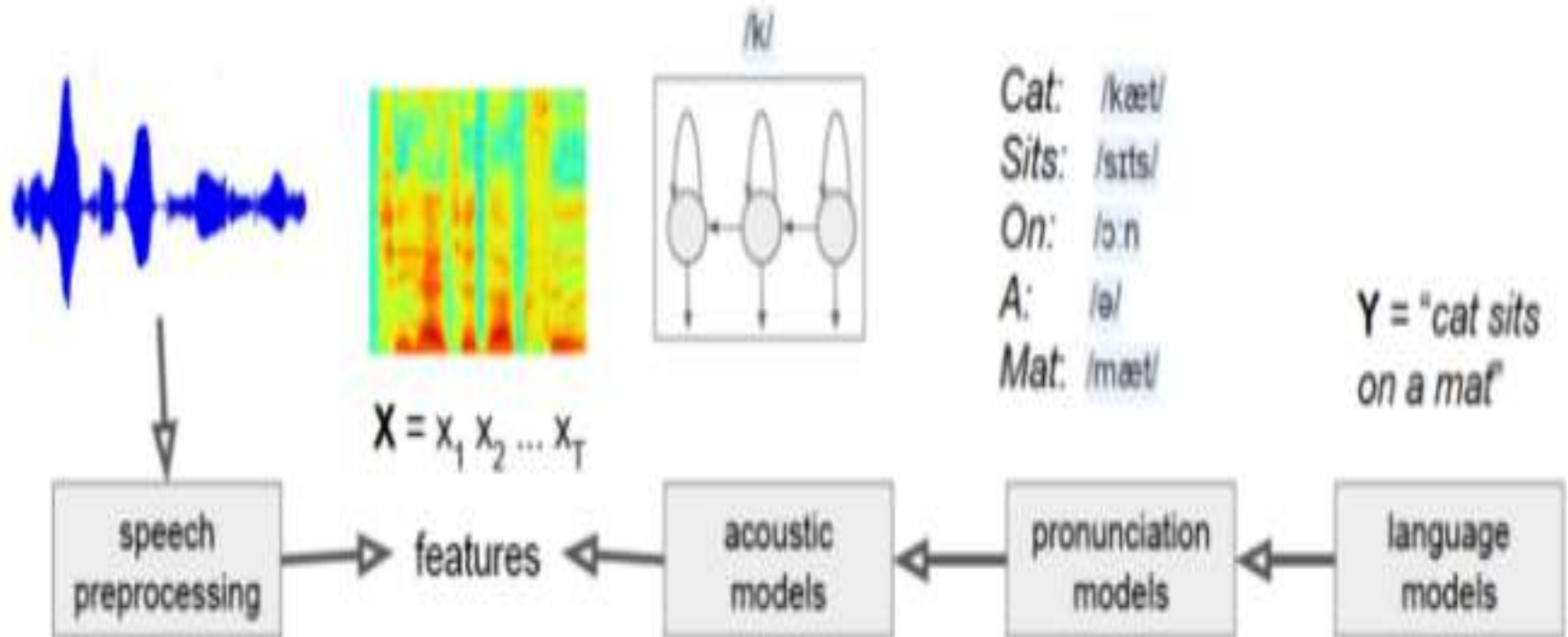
**HE\_L\_LO becomes HELLO**

**HU\_L\_LO becomes HULLO**

**AU\_L\_LO becomes AULLO**



## Finally the Speech Recognition In Deep Learning is :




# SPEECH RECOGNITION ALGORITHMS

- Speech recognition uses **various algorithms** and computation techniques to **convert spoken language into written language**.
- The following are some of the most commonly used **speech recognition methods**:
- **1. Hidden Markov Models (HMMs)**: Hidden Markov model is a **statistical Markov model** commonly used in **traditional speech recognition systems**. HMMs capture the relationship between the **acoustic features** and **model the temporal dynamics of speech signals**.



2. Natural language processing (NLP): NLP is a subfield of artificial intelligence that focuses on the interaction between **humans and machines through natural language**. Some of the key roles of NLP in speech recognition systems:

- Estimate the **probability of word sequences** in the recognized text
- **Convert colloquial expressions** and **abbreviations** in a spoken language into a standard written form
- **Map phonetic units** obtained from **acoustic models** to their corresponding words in the target language. 

3. Deep neural Networks: Neural networks process and transform input data by simulating the non-linear frequency perception of the human auditory system.

4. Connectionist Temporal Classification (CTC): It is a training objective introduced by Alex Graves in 2006.

CTC is especially useful for sequence labeling tasks and end-to-end speech recognition systems. It allows the neural network to discover the relationship between input frames and align input frames with output labels.





# APPLICATIONS

## Applications of Speech Recognition



**Customer  
Service**



**Virtual  
Assistant**



**Transcription  
Services**



**Accessibility  
Features**



**Smart  
Homes**

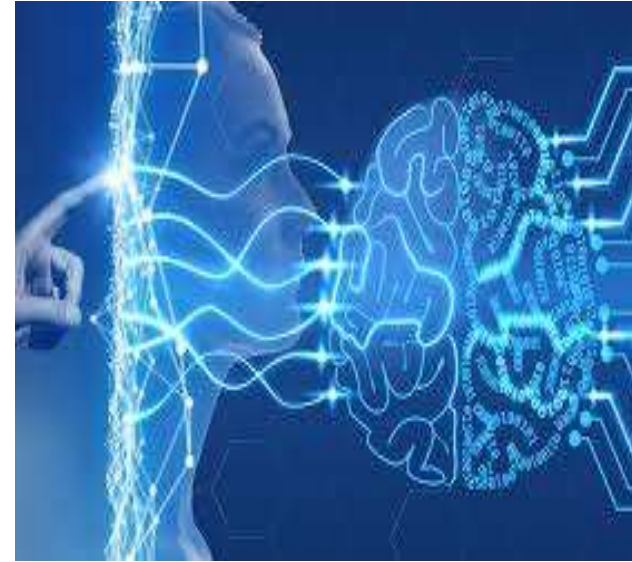


**THANK YOU**



# UNIT-V

## (Deep Learning Applications)



# Deep Learning Applications

## Topics :

- What is NLP
- How does it works

# What is NLP?

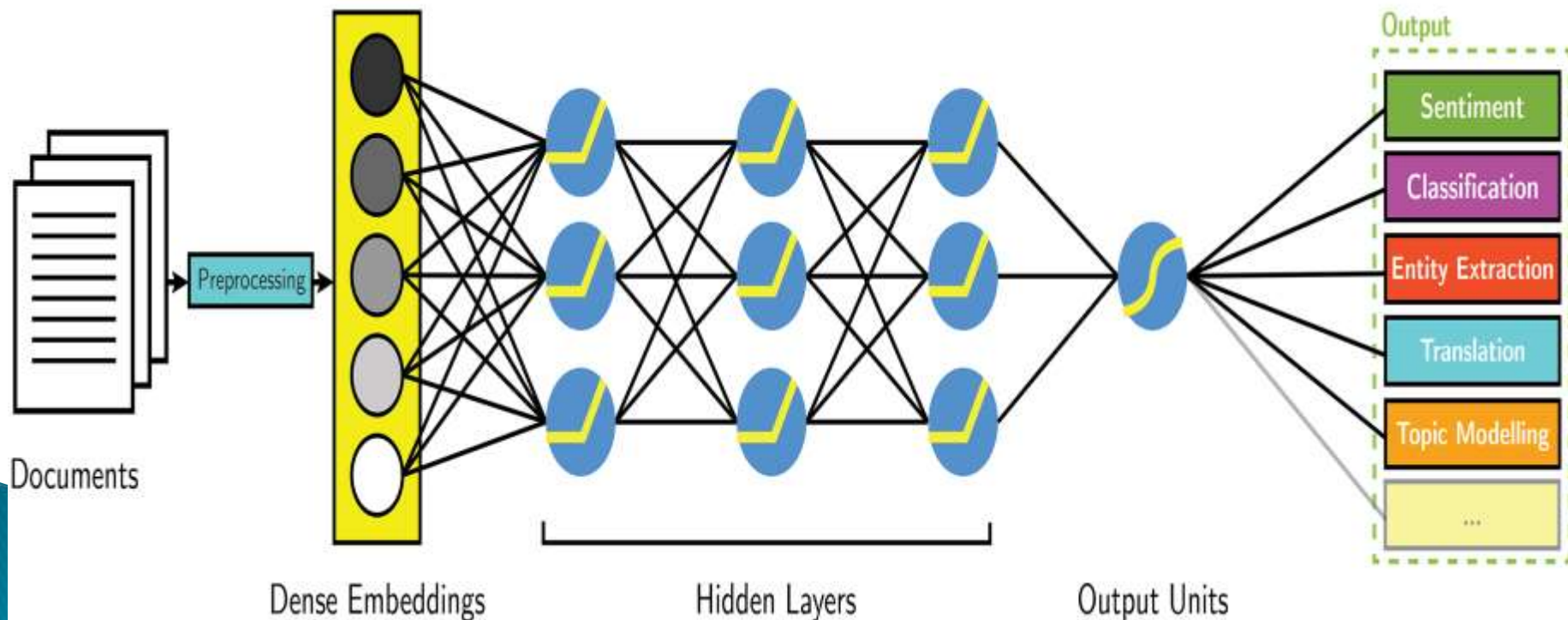
- **NLP stands for Natural Language Processing**
- Natural Language Processing (NLP) is a branch of AI that helps computers to **understand, interpret and manipulate human language.**
- NLP is mainly used to **interact between Human Languages and Computers .**
- **Examples** include English, French, and Spanish.
- Early computers were designed **to solve equations and process numbers.** They were not meant to understand natural languages.
- Computers have their own programming languages (C, Java, Python) and communication protocols (TCP/IP, HTTP, MQTT).

# How Does NLP Works in DL

- Deep learning for NLP is the part of Artificial Intelligence which is used to help the computer to understand, manipulating and interpreting the human language.
- NLP-based systems have enabled a **wide range of applications** such as **Google's powerful search engine, and more recently, Amazon's voice assistant named Alexa.** NLP is also useful to teach machines the ability to perform complex natural language related tasks such as machine translation and dialogue generation.



- Deep Learning is the concept of **neural networks**.
- Deep learning methods are helping to solve problems of Natural Language Processing (NLP) which couldn't be solved using machine learning algorithms.



- It solves **non-linear problems** such as **processing text and words**.
- Before the arrival of deep learning, representation of text was built on a basic idea which we called **One Hot Word encodings** like shown in the below images:
- In fact, since there can be hundreds of thousands of words in a given language, representing and storing words as one-hots can be extremely expensive.

Words	The	Dog	Is	Barking	In	Street
The	1	0	0	0	0	0
Dog	0	1	0	0	0	0
Street	0	0	0	0	0	1
Is	0	0	1	0	0	0
Barking	0	0	0	1	0	0
In	0	0	0	0	1	0

**Table: One Hot Encoding**

# Example is: Text Classification

- Here Text is Sequence of
  - Characters
  - Words
  - Phrases and Named Entities
  - Sentences
  - Paragraphs etc.

There are several types to **convert word into numbers.**  
**Different methods to take the input into different ways.**

# Contd..

- 1. One Hot Encoding:
- One hot encoded vector that is **huge vector of zeros** that has **only one non zero value** which is in the **column corresponding to that particular word**
- Here we have to calculate Sparse Matrix (ie, converted into **one hot encoding format**)
- So in this example we have **very good and movie** and all of them are **vectorised independently**.

# Bag of Words (Sparse)

~100k columns

	good	movie	very	a	did	like
very →	0	0	<b>1</b>	0	0	0
good →	<b>1</b>	0	0	0	0	0
movie →	0	<b>1</b>	0	0	0	0



- But in real life we have **hundreds of thousands of columns** and how do we get to bag of words representation.
- In this case we can **sum up all those values all those vectors** and we come up with bag of words vectorization now corresponds to very good movie and apply one hot encoding

## Bag of words way (sparse)

~100k columns

	good	movie	very	a	did	like
very →	0	0	1	0	0	0
			+			
good →	1	0	0	0	0	0
			+			
movie →	0	1	0	0	0	0
			=			
very good movie →	1	1	1	0	0	0

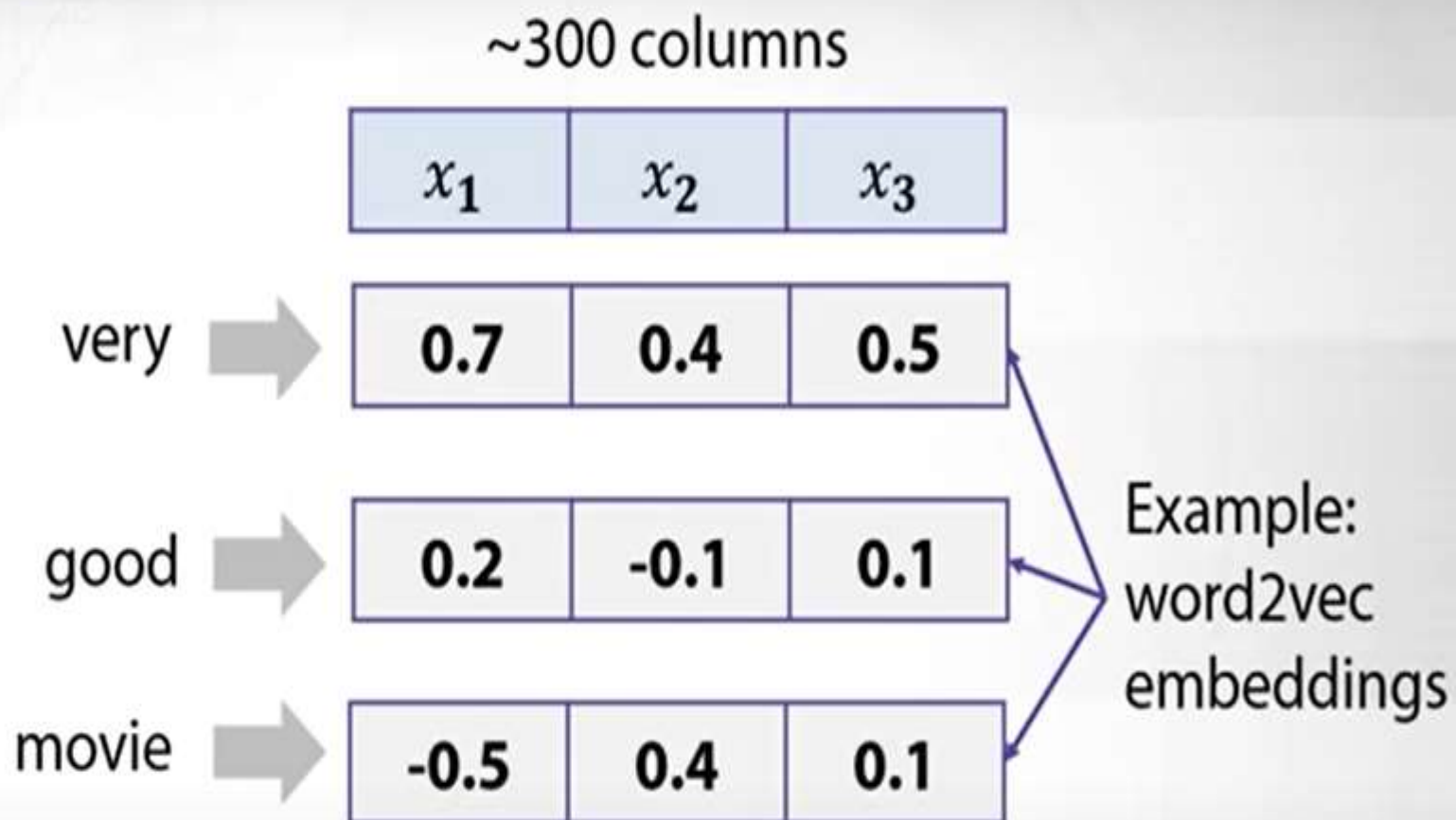
Bag of words representation  
is a sum of sparse one-hot-encoded vectors



## 2. Word Embedding:

- Now Next move to the **Narrow Network Way**(Dense Matrix)and it is opposite to the Sparse Matrix.
- Ie, here we have seen bag of words in **neural networks like dense representation.**
- That is we should replace each word by **Dense Vector**. Ie, here we have taken the real values. And trained these values using **Unsupervised Learning.**

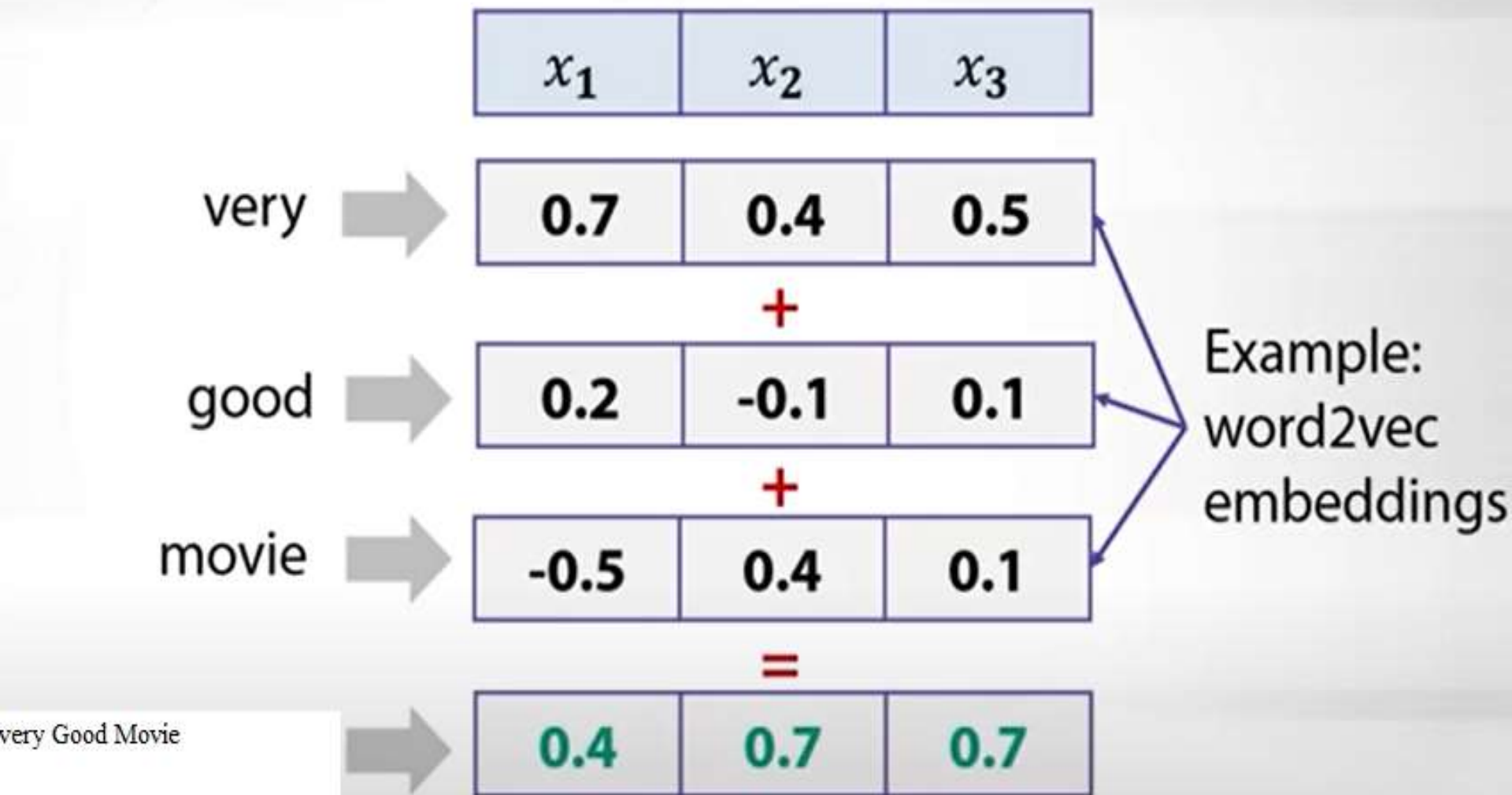
# Neural way (dense)



## Word2vec property:

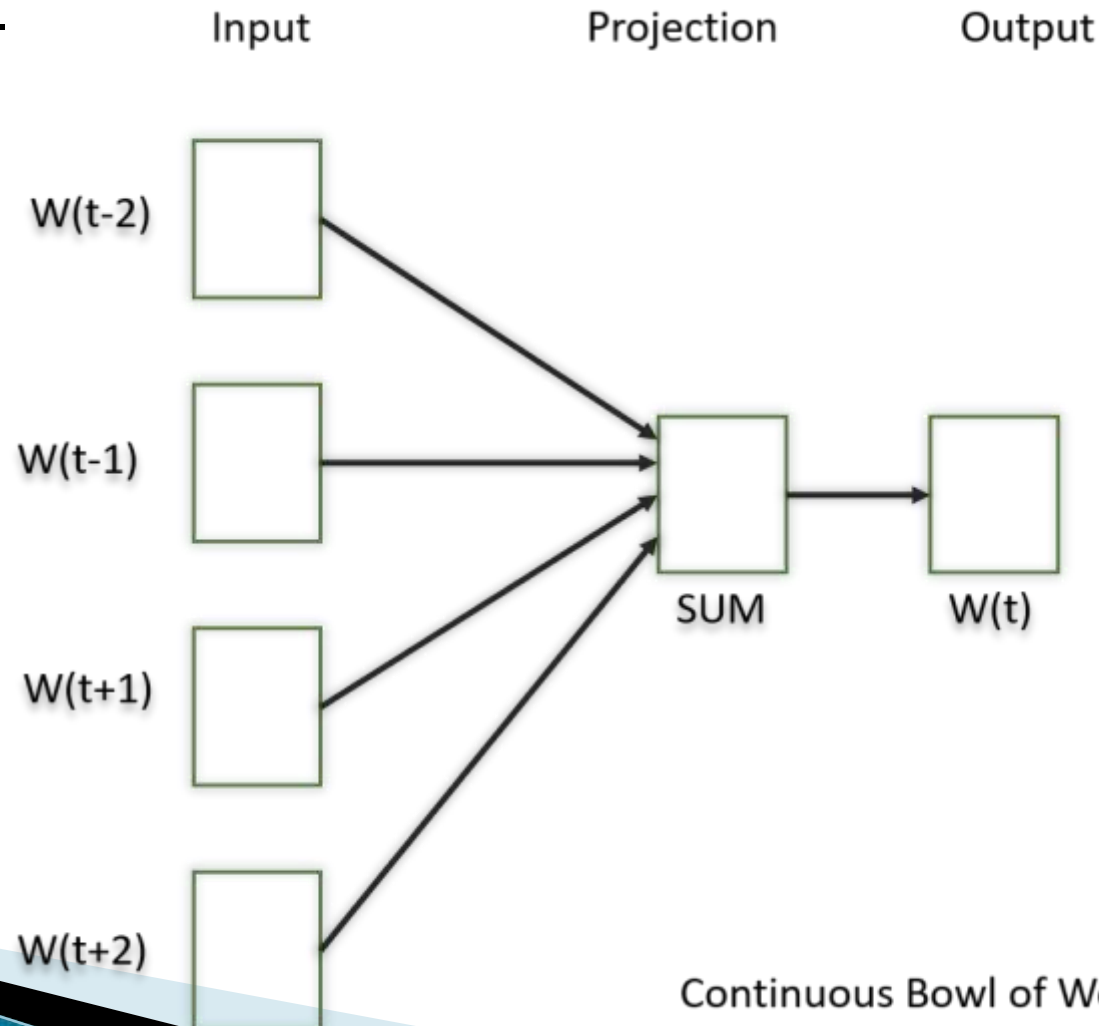
Words that have similar context tend to have collinear vectors

~300 columns



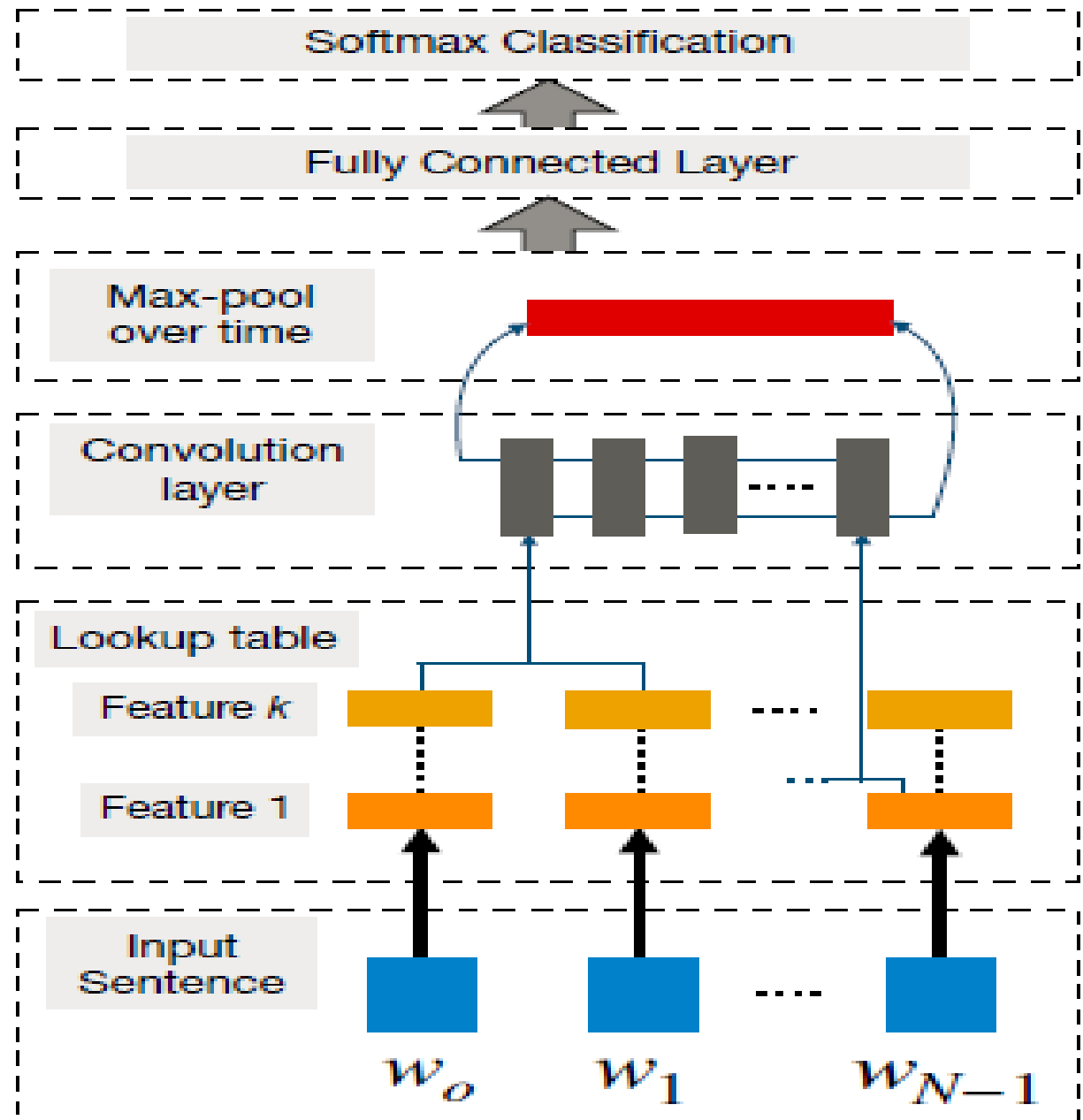
Sum of word2vec vectors

- **3. Continuous Bowl of Words(CBOW):** In this model what we do is we try to fit the neighboring words in the window to the central word.





# CNN using DL



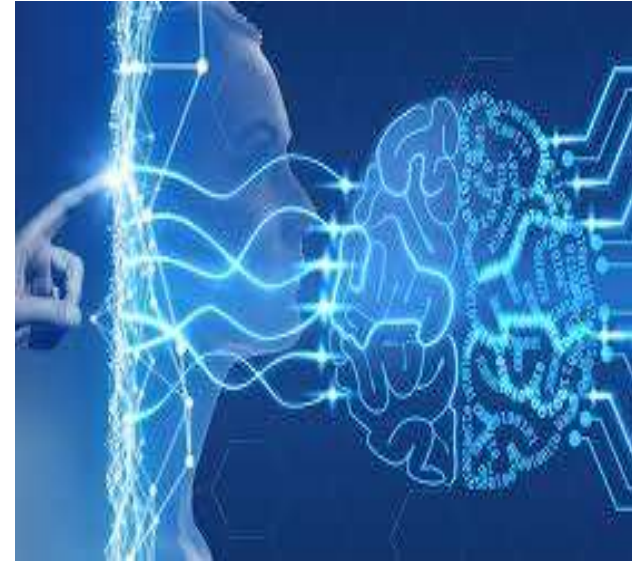
## ➤ Applications in NLP:

- 1. Machine translation: Machine translation is the automatic translation of speech or text into another language. E.g., translating a text document from French to English to the central word.
- 2. Question answering: This system helps in answering user queries that involve noun phrases—such as ‘Who is the president of the USA?’, ‘what is deep learning?’, ‘restaurants near me,’ and so on—and questions regarding medical records, news articles, best tutorials, etc.

**THANK YOU**

# UNIT-V

## (Deep Learning Applications)



# Deep Learning Applications

## Topics :

- What is Decision Making in DL
- Example : Self Driving Car

# What is Decision Making?

- Decision Making is the **process of making choices** by identifying a **decision, gathering information and assessing alternative solutions.**
- Decision Making applied in Deep Learning by using **CNN,RNN** etc.
- Decision making is applied in several areas
  - **Self Driving Car**
  - **Facial Recognition**
  - **Handwritten Digit Recognition**
  - **Smart Home**
  - **Healthcare etc.**

# Example : Self Driving Car

- Self-driving vehicles are cars or trucks in which **human drivers are never required** to take control to safely operate the vehicle. Also known as autonomous or “driverless” cars, they **combine sensors and software to control, navigate, and drive the vehicle**
- The Autonomous Driving System can be generally divided in to **4 blocks**, **Sensors, Perception subsystem, planning subsystem, and control of the vehicle**



# Example : Self Driving Car

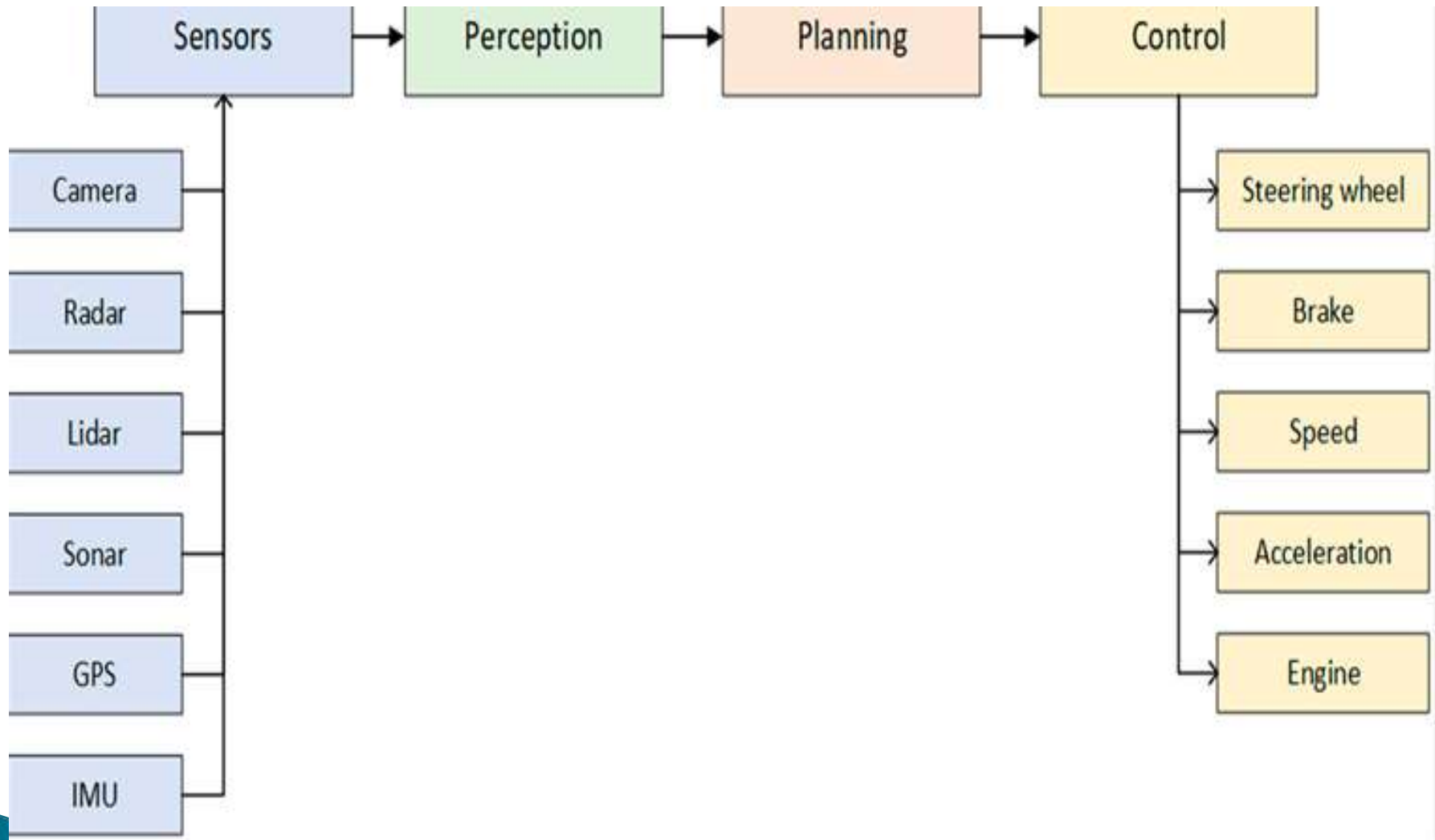


Figure: Autonomous vehicle system block diagram.

# Contd..

- The vehicle is sensing the world using many different sensors mounted on the vehicle. The information from the sensors is **processed in a perception block**, whose components combine sensor data into meaningful information.
- The **planning subsystem** uses the **output from the perception block** for behavior planning and for both **short- and long-range path planning**.
- The **control module** ensures that the **vehicle follows the path provided by the planning subsystem** and **sends control commands to the vehicle**.

# Contd..

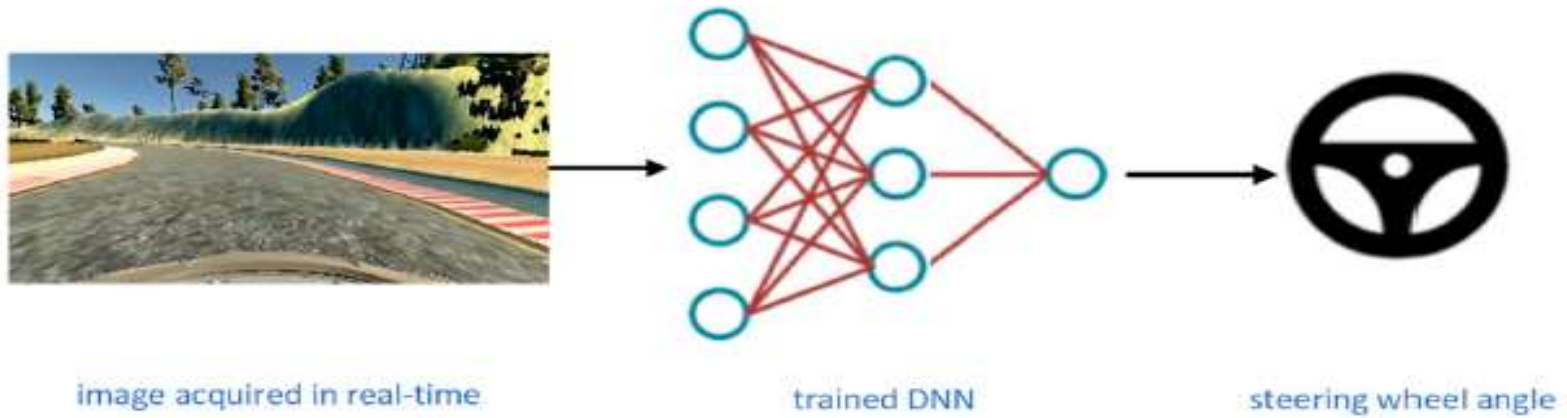
- An end-to-end deep neural network we designed for autonomous driving uses **camera images** as an **input**, which is a **raw signal (i.e., pixel)**, and **steering angle predictions** as an **output** to control the vehicle.



Figure . Block diagram of an end-to-end autonomous driving system.

# Contd..

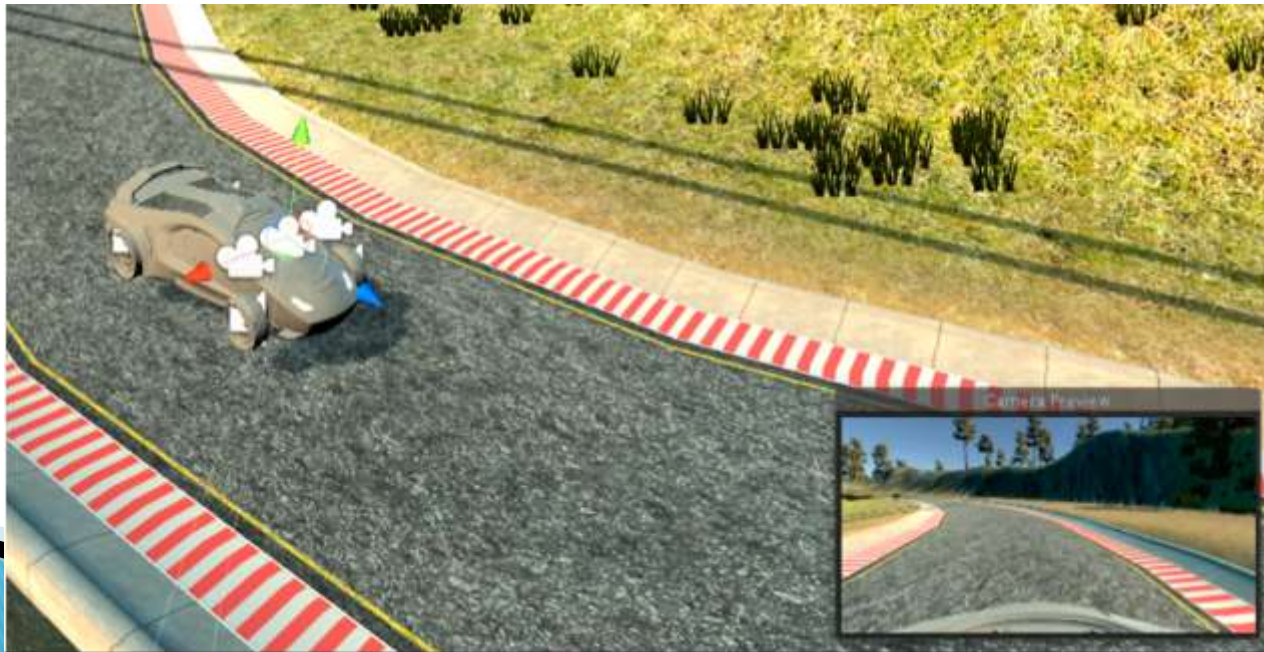
- The purpose of end to end learning is the system **automatically learns internal representations** of the necessary processing steps, such as such as **detection of useful road characteristics**, based only on the **input signal**.
- Here The input in our autonomous driving system is only the **camera image, the raw pixel**. Output is **steering angle prediction**.



**Figure** Real-time autonomous driving—the image acquired from the central camera is fed to the trained deep neural network (DNN) model and the output of this model is the steering angle prediction that controls the vehicle.

# Contd..

- **1. Data Collection:** Here first we have to **collect the data**. The same representative track was used for driving in the autonomous driving mode. where the decision about steering angle was made by a **real-time image from the camera mounted on the vehicle**.





# Contd..

- For Example: The model has to learn how to **handle sharp turns, different textures, different borders of the road.**
- Examples of images recorded with the central camera in different frames, which is presented in **road map characteristics.**



(a)



(b)



(c)



(d)

# Contd..

- In order to have good quality samples, the slight difference in the field of view per each central, left and right camera leads to better generalization of the model.

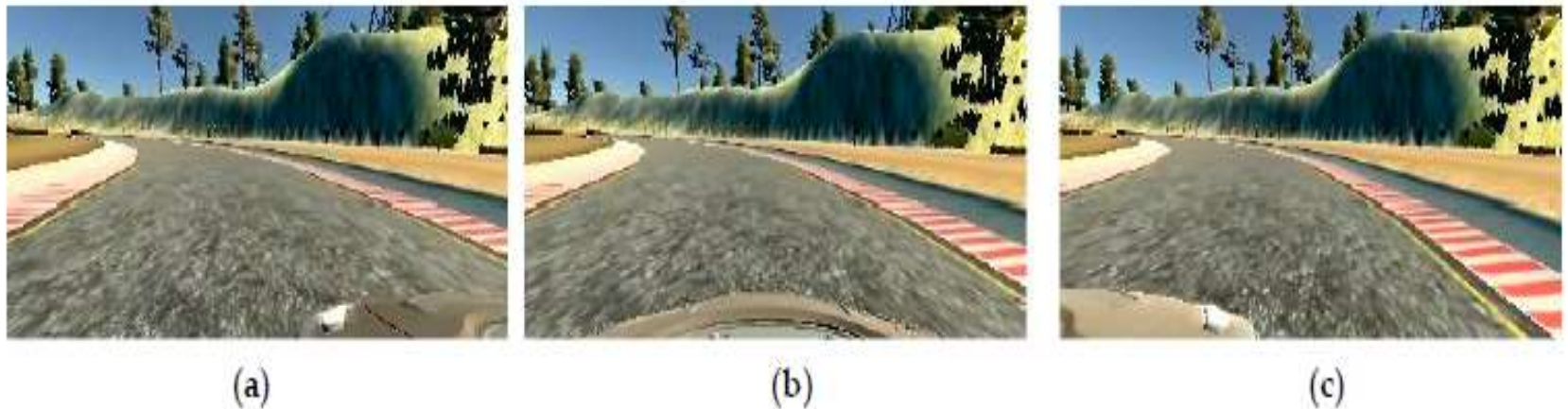


Figure 1. Example of data collection taken simultaneously using the three cameras mounted on the vehicle: (a) left; (b) center; (c) right camera. This is an example of images captured in the very first frame.



# Contd..

- **2. Data Preprocessing:** For training **Deep Neural Networks**, all images are captured for training CNN.
- In data Preprocessing, Here **Cropping was used** in order to remove the parts of the image that do not have valuable information for autonomous driving, so as to remove the **sky trees and hills on the top of image and the part of the hood of the vehicle on the image bottom. An example of the image after cropping is presented**

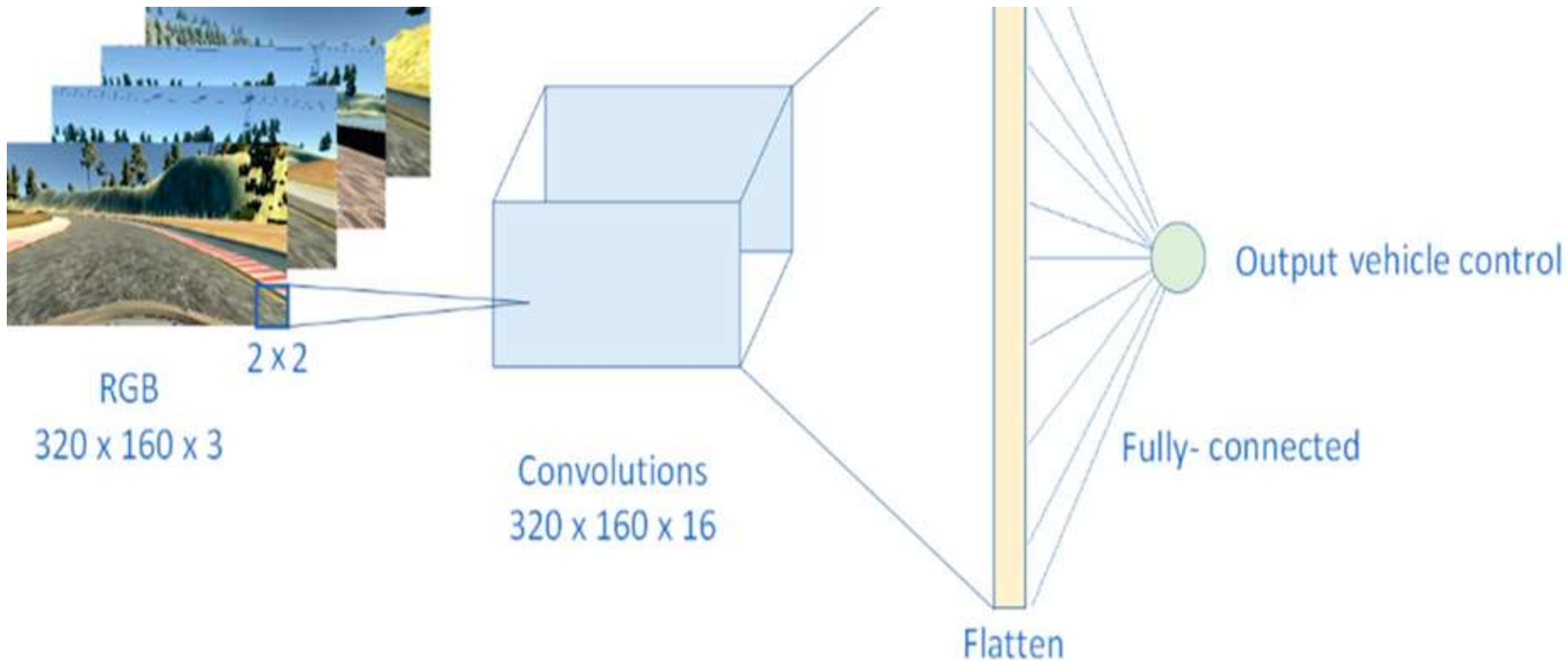


Figure . Input image after the cropping.

# Contd..

- The images from dataset were normalized by dividing each pixel of the image by 255, which is the maximum value of an image pixel. Once the image was normalized to a range **between 0 and 1..**

## ➤ 3. Using Convolutional Neural Network:



- By taking the quality images, Using CNN it takes the decision where we need to go to different roots.

Contd..

**THANK YOU**