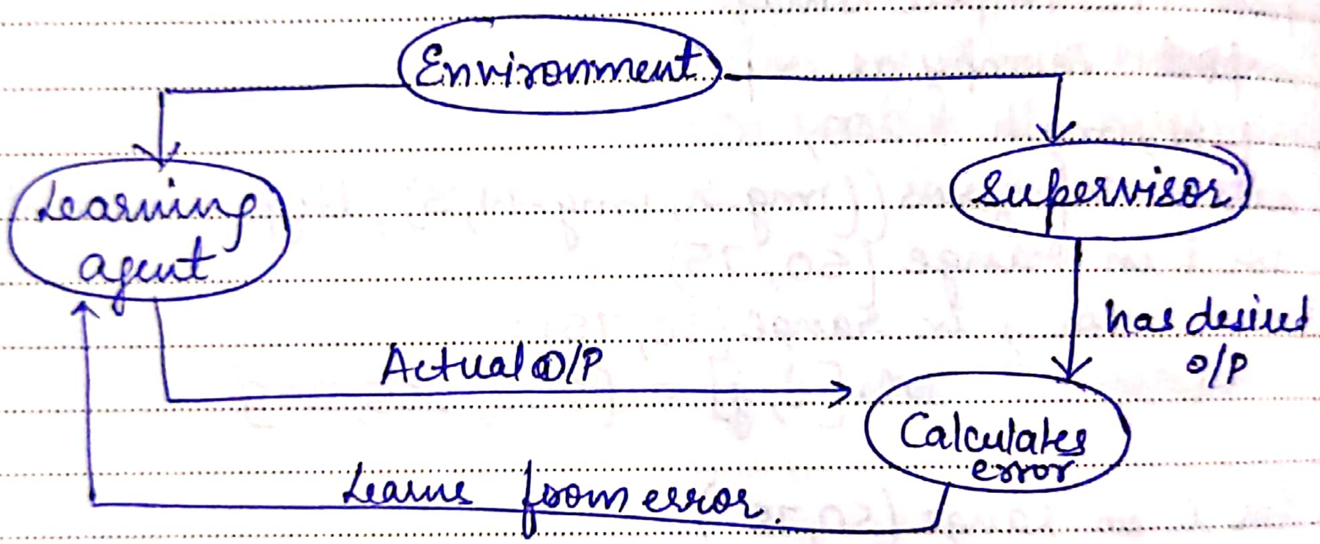
Day 2

Hierarchical feature learning

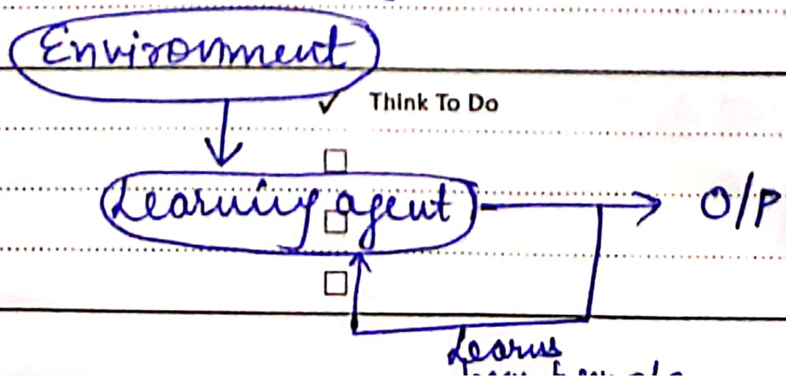
Supervised learning :-

It is the technique of accomplishing a task by providing training input and output patterns to the system. A learning agent is present who learns from environment & ~~Su~~ his errors. A supervisor is present who has the data of the desired/ideal O/P whereas learning agent only predicts O/P and on the basis of both, error is calculated.



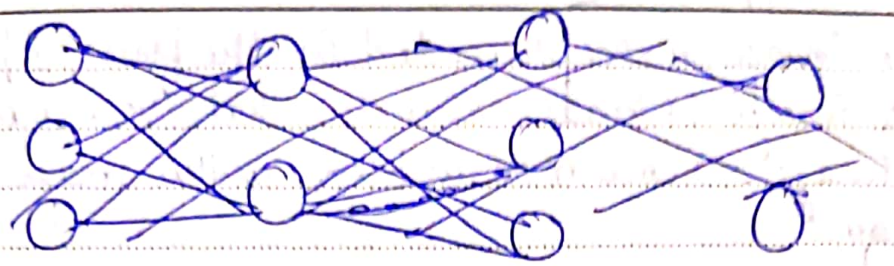

Unsupervised learning :-

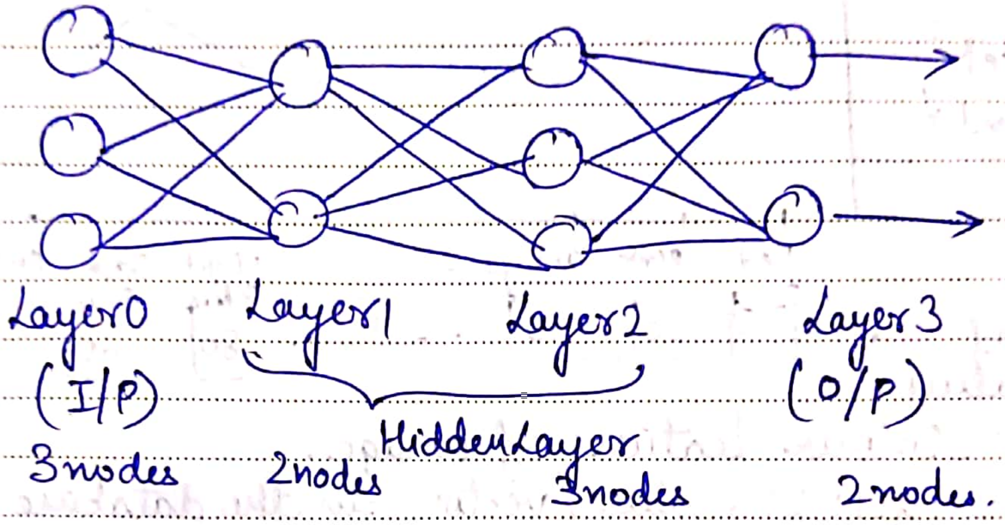
It is a self learning technique in which system has to discover the features of the input population by its own & no prior set of categories are used.
Here no supervisor and no desired O/P is present & learning agent learns by itself and keeps on ~~Envi~~ improving from the prev output.




Meeting ✓ Think To Do ✓ Important Calls

~~Hist~~





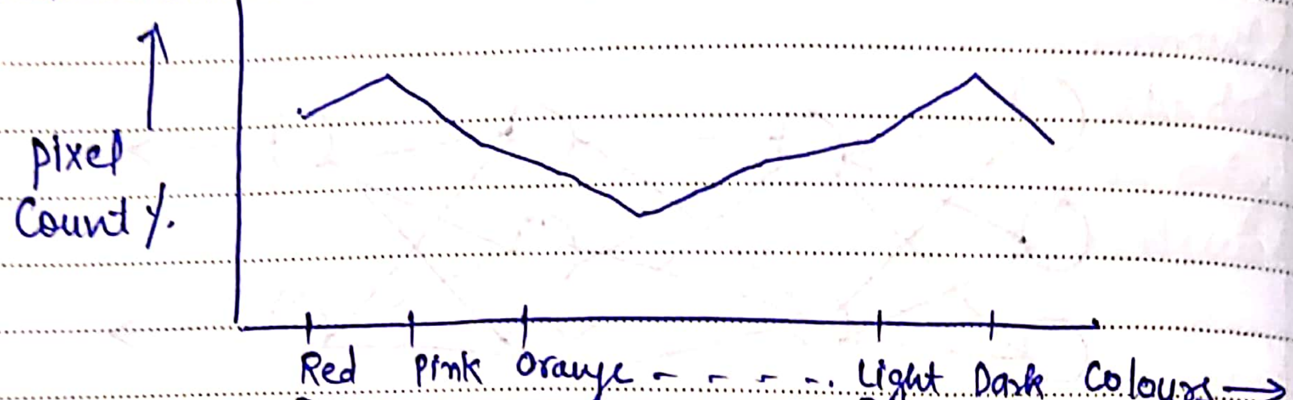

In the past, we used hand engineered features to quantify the contents of an image — we rarely used the raw pixel intensities as inputs to our ML models as we do these days. For each dataset's image, we perform feature extraction, or the process of taking an I/P image, quantifying it acc. to some algorithm (called as feature extractor/image descriptor), and then return a vector (a list of nos.) that aimed to quantify the contents of an image.

## Feature Extraction in 2D colour images:-

Features of an image can be extracted by ~~the~~ its contents such as:- colour, texture, shape, position, dominant images items & regions, etc.

## Approach:-

Firstly, an image is represented in the form of pixels & then a histogram of image is made assuming that 256 colours are used in the image ⇒ Colour of an pixel in this image lies within these 256 colours.




| 75 | 80 | 70 | ...... | 80 | 65 |
|---|---|---|---|---|---|

Vector of 256 values ← 
↳ Colour features of image.

We store these 256 values / vector in the database. So, this process is repeated for all diff images.

# To search by image in the dataset:-

Use that query image & create its histogram & then find all the 256 pixel values.

After this, distance of this query image is calculated from all other images using a distance funcⁿ. ↓distance → ↑similarity.

Evening

There are 2 problems in above:-

a) Dimensionality curse:- Since each image is made of 256 various features. So, we try to reduce the no. of features. → more time

b) Cross talk:- Since the colours close to each other (like red, pink, orange) needs to be compared among each other. So, it takes more time of searching.

Meeting Think To Do Important Calls

## Approach 2: (Feature extraction)

Here we try to reduce the no. of features.
Initially, image is converted to pixels & then we use RGB concept.
Every pixel colour can be made by mixing % content of RGB colours.
Then we take the avg of all Red, Green & Blue colours Individually.

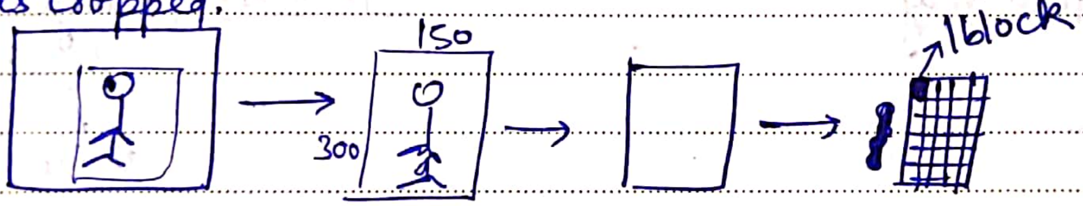
$$R_{avg} = \frac{1}{p}\sum_{p=1}^{p} R(p) \qquad G_{avg} = \frac{1}{p}\sum_{p=1}^{p} G(p) \qquad B_{avg} = \frac{1}{p}\sum_{p=1}^{p} B(p)$$

These 3 values are the (reduced) features values of our image. Then we search for the query image in the dataset as prev.

Histogram of Oriented gradients.

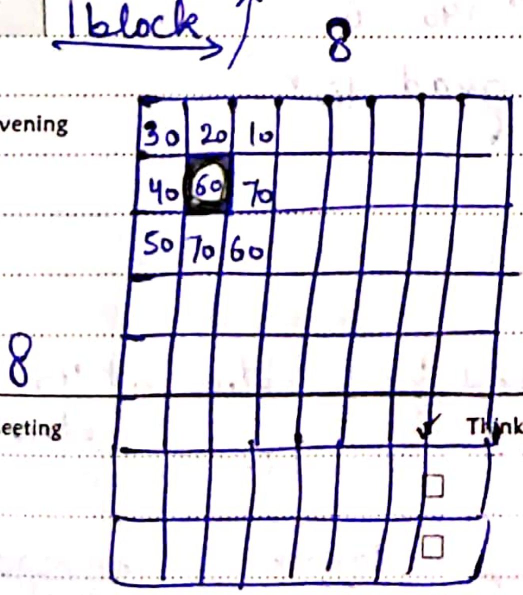
### HOG feature Vector Calculation :-

Any given image is cropped & a patch of it in the ratio 1:2 is cropped.




150X300 1:2 → 64X128 again resized → 8X16 grid again resized.
↓
Each block in the grid will be 8X8 pixels
8 8 ↳ 1block

1block → pixel values




To calculate gradient magnitude & direction in the 60 block :-

| | | |
|---|---|---|
| | 20 | |
| 40 | 60 | 70 |
| | 70 | |

We need to calculate pixel intensity diff in both x & y dir$^n$.

Evening
Meeting ✓ Think To Do ✓ Important Calls ✓

In x dir$^n$ → $|40-70| = 30$

In y dir$^n$ ↓ $|20-70| = 50$

$$\text{Gradient magnitude} = \sqrt{\begin{pmatrix}\text{diff in}\\ x\ \text{dir}^n\end{pmatrix}^2 + \begin{pmatrix}\text{diff in}\\ y\ \text{dir}^n\end{pmatrix}^2}$$

$$= \sqrt{(30)^2 + (50)^2} = 58$$

$$\text{Gradient dir}^n = \tan^{-1}\left(\frac{\text{diff in } x\ \text{dir}^n}{\text{diff in } y\ \text{dir}^n}\right) = \tan^{-1}\left(\frac{30}{50}\right) = 30^\circ$$

Calculate above 2 for all pixel values.

**Ex:-**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 50 | | 60 | | 60 | | 60 | |

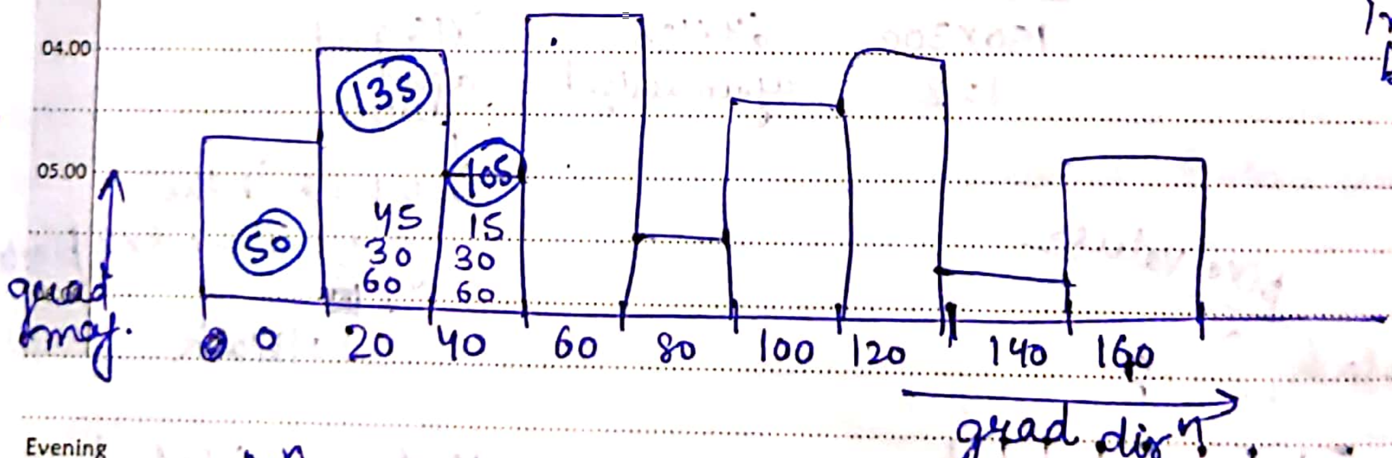
Grad mag.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 20 | | 40 | | 30 | | 25 | |

Grad dir$^n$ (Range = $0^\circ$–$180^\circ$)

↓

divide $180^\circ$ into 9 blocks.




| dir$^n$ | mag | |
|---|---|---|
| for $20^\circ \longrightarrow$ | 60 | put in 20 block |
| $40^\circ \longrightarrow$ | 60 | " " 40 " |

$30^\circ \longrightarrow 60 \longrightarrow$ this is divided $\frac{1}{2}$ – $\frac{1}{2}$ b/w 20 & 40 block
(lies in b/w 20 & 40 bin) ~~30~~ i.e. 30 – 30

$25^\circ \longrightarrow 60$
(lies in 20 & 40 bin)

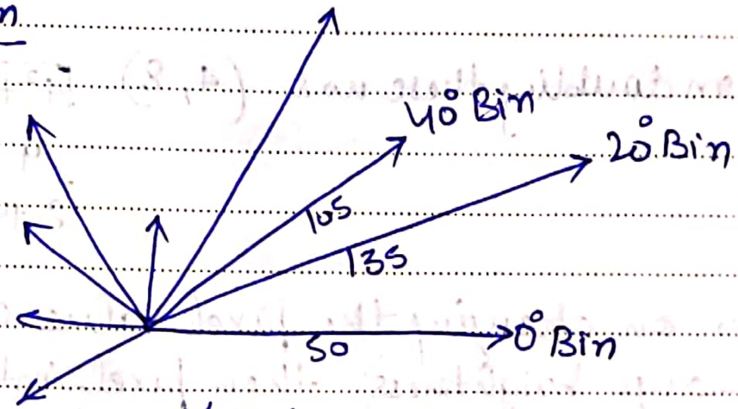
20 block $\frac{40-25}{20} = \frac{3}{4}$  40 block $\frac{25-20}{20} = \frac{1}{4}$  So 60 gets ~~divi~~

Since 25 lies close to 20, so it gets major part ~~40-25~~ ~~25~~

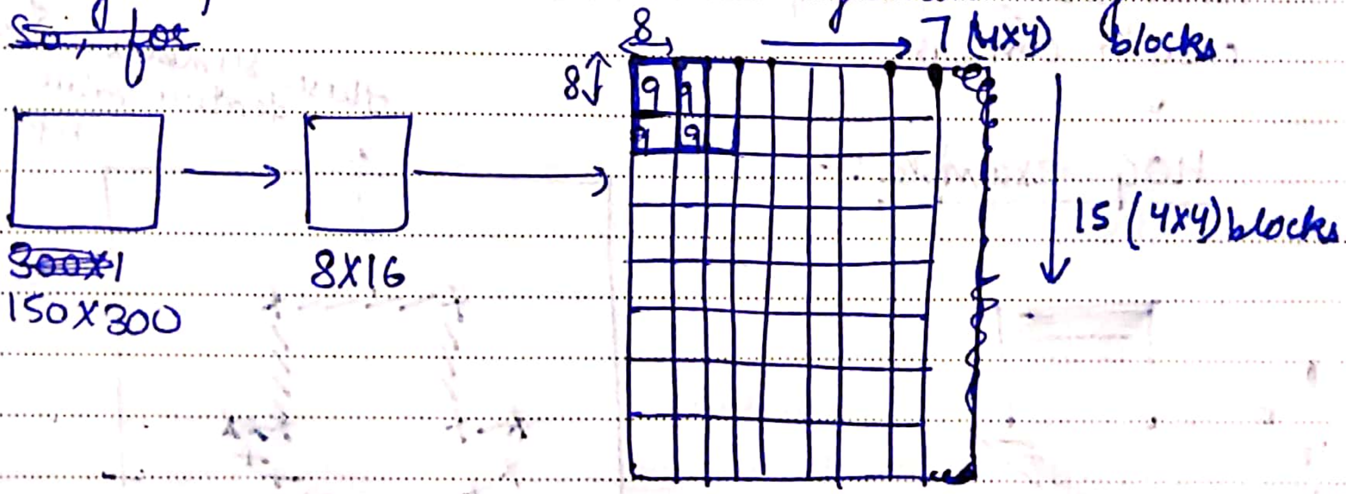
So, 60 gets divided in $\frac{3}{4} : \frac{1}{4}$ ∴ So put 45 in 20 block & 15 in 40 block.

Suppose 50 is present in 0 block & then sum all these values.
Feature Vector → array of all these values i.e.
[50, 135, 105 ....] → Size = 9.
↳ matrix of 9x1

Pictorial representation




Length of these arrows ∝ Total gradient mag.
~~So, for~~




Every block is converted to ~~an~~ ~~image~~ a feature vector of size = 9. After this we will combine 4 such feature (f.v.) vectors of size 9 so as to create a concatinated f.v. of size = 36. Then we will move this window 1 block to right & in a row, we will get 7 such blocks.

Evening

Meeting Think To Do Important Calls

In row → 7 In column → 15

Resultant = 7X ~~8~~ 15X36 = 3780 } = HOG feature will be of length 3780.
↳ size

Here we have normalised the feature vector.

Ex:- Normalising 2 nos. (2 & 4) :- $\sqrt{2^2+4^2} = 4.472$

$$\frac{2}{4.472}, \frac{4}{4.472} = 0.44, 0.89$$

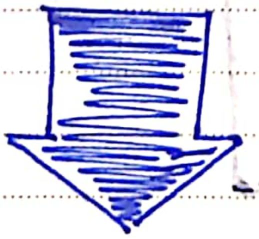
on doubling these nos (4, 8) $\sqrt{4^2+8^2} = 8.944$

$$\frac{4}{8.944}, \frac{8}{8.944} = 0.4472, 0.8944$$

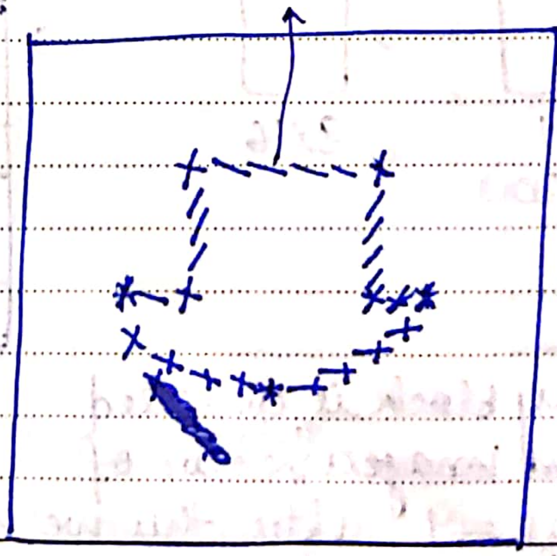
So, on changing the pixel values of an image (~~Either~~ to change brightness) then pixel intensity will change. But, on normalising it, our feature vector values will not change & easy to compare.

HOG example :-

These symbols are HOG feature values (pictorial representation)



I/P image

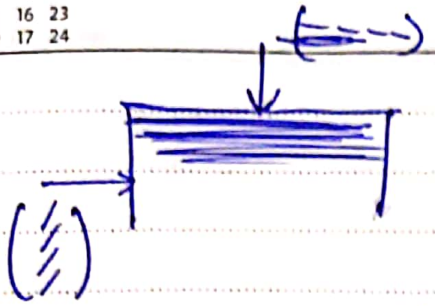


Histogram

Evening

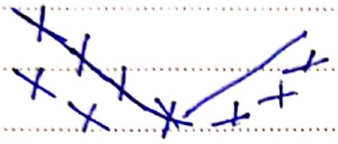
Here no change in ~~cont~~ contrast, so no change in HOG value.

Meeting ✓ Think To Do ✓ Important Calls



Here change in grad mag. is along →

Along diff. corners, these feature values have diff shapes



## Object detection with HOG – SVM :-

Image → HOG features → Human Model → o/p of convolving with human model $\xrightarrow{SVM}$ Human OR NOT

(HOG features → Human Model:) Convolving of features with human model is done.

* In deep learning, the features are learned from training process instead of hand defining algorithms & rules to extract these features from an image.

* In CNN, a series of hidden layers are used to extract features from image & they build in a hierarchal fashion. First edge → then corners → then other parts are formed.

* In hierarchical learning: each layer in the network uses the o/p of prev. layer (building blocks). to inc. concepts ~~which a~~ & these layers are learned automatically. This CNN focusses on process of training our network to learn these filters ~~instead of~~ & allows to skip the feature extraction step.

Evening

Meeting Think to Do Important Calls