Intro to Computer Vision

$\underbrace{\text{Rectified Linear Unit}}$ — $f(x) \begin{cases} \to x & \text{if } x > 0 \\ \to 0 & \text{if } x \le 0 \end{cases}$

Images

Agenda → Images

→ History

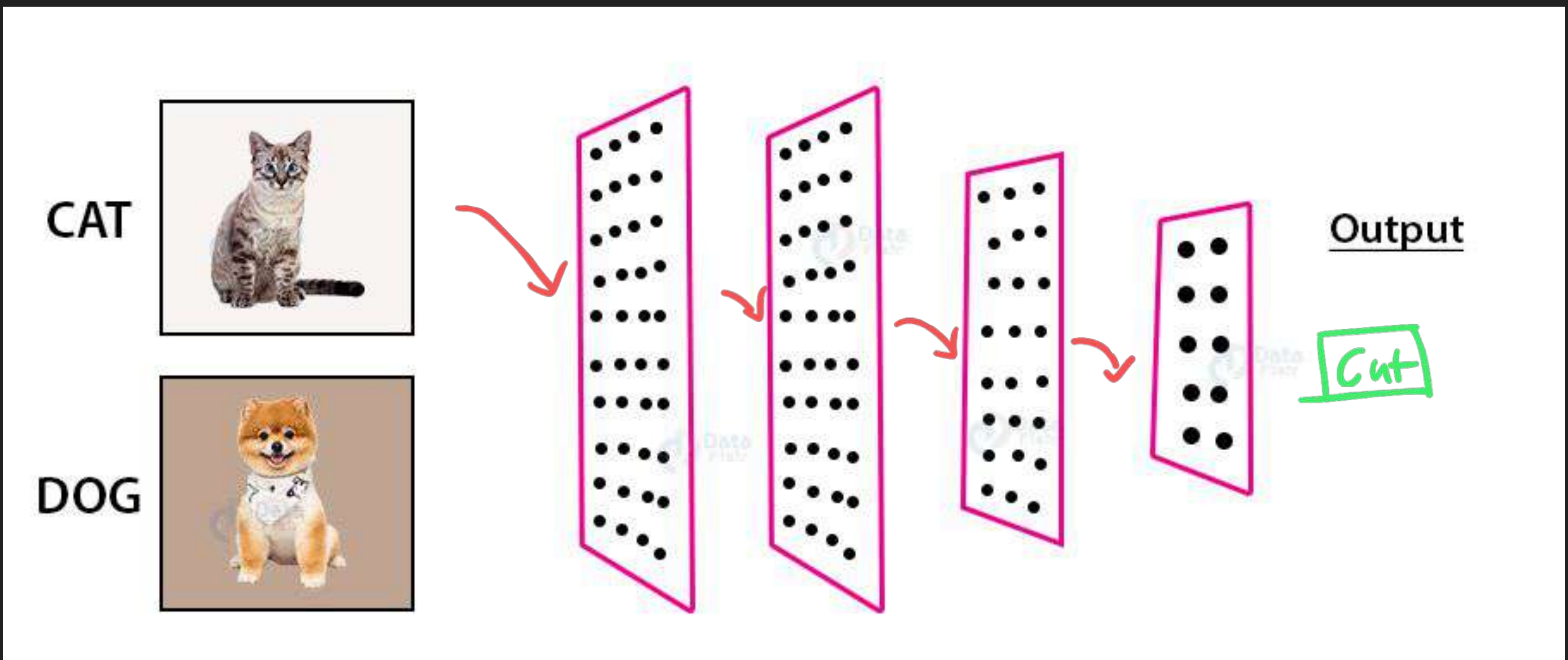
→ Cat Experiment } ← Inspiration

→ Challenges

→ Simple Neural Nets / Multi-Layer Perceptron
↳ Can we fit a simple NN on images

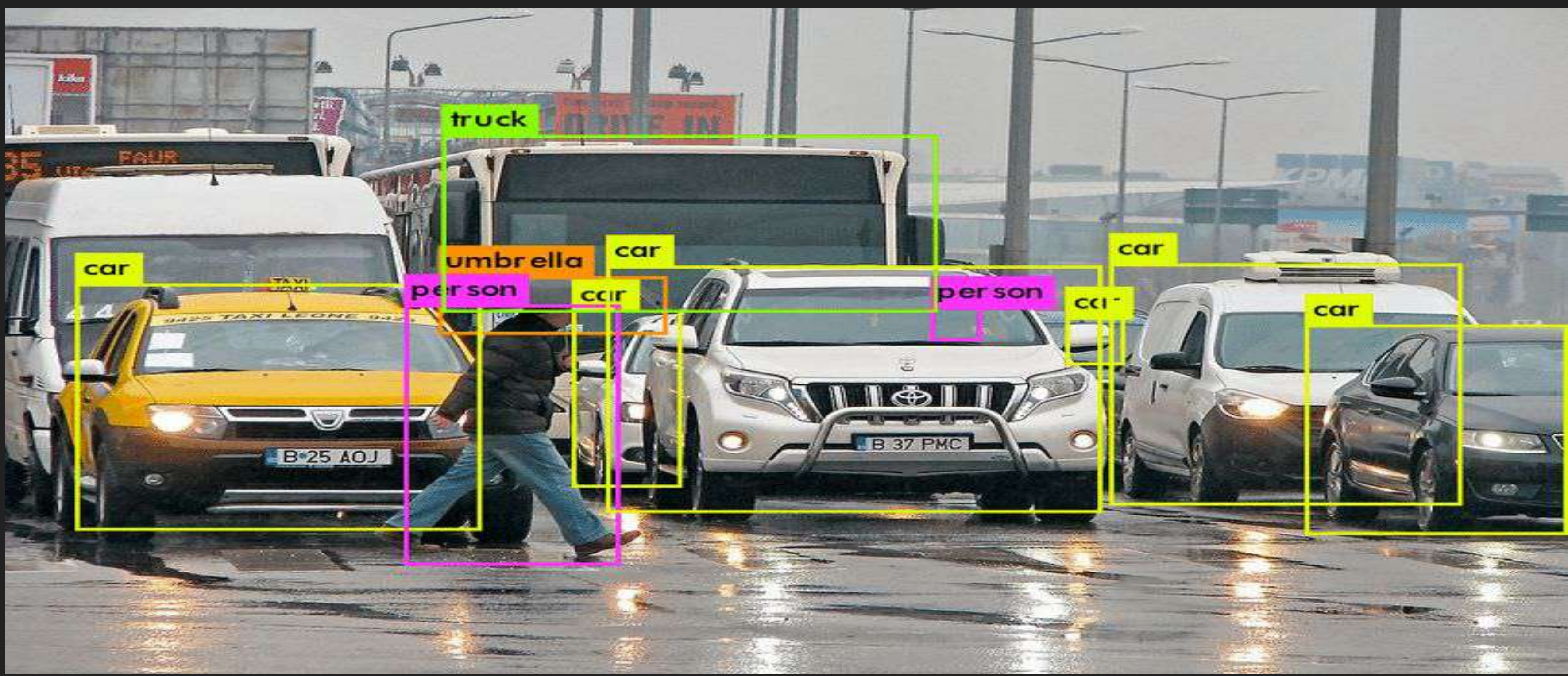
→ CNNs } next few class → Convolutional Neural Networks

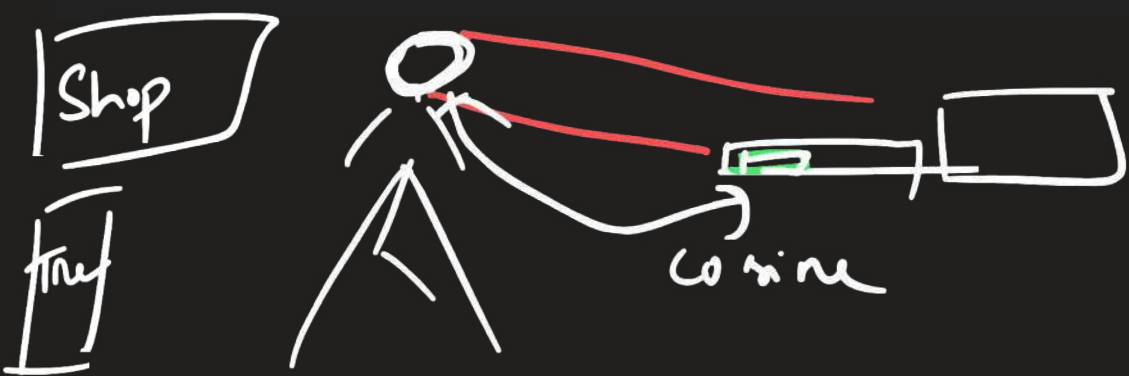
Can we understand images



Deep dive

Object Detection → Crucial part of classes





OCR → Optical Character Recognition

→ Name

→ Total

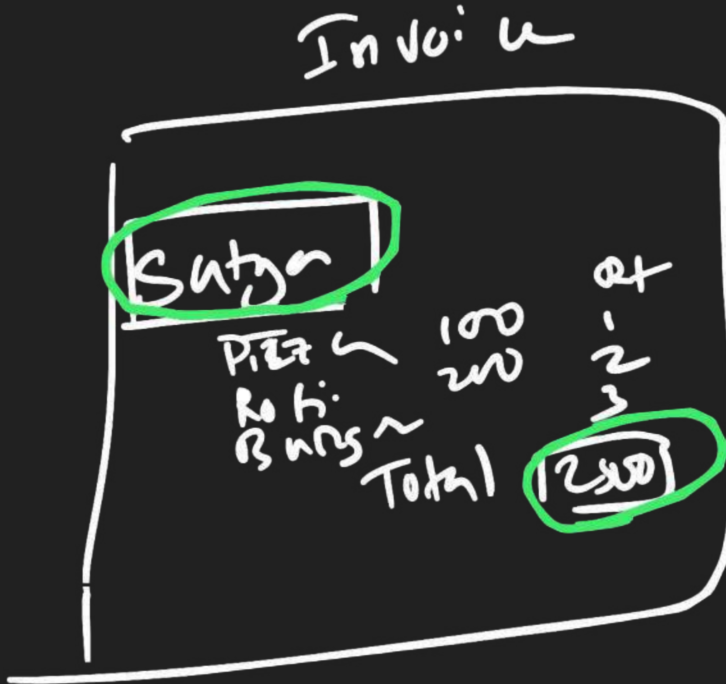


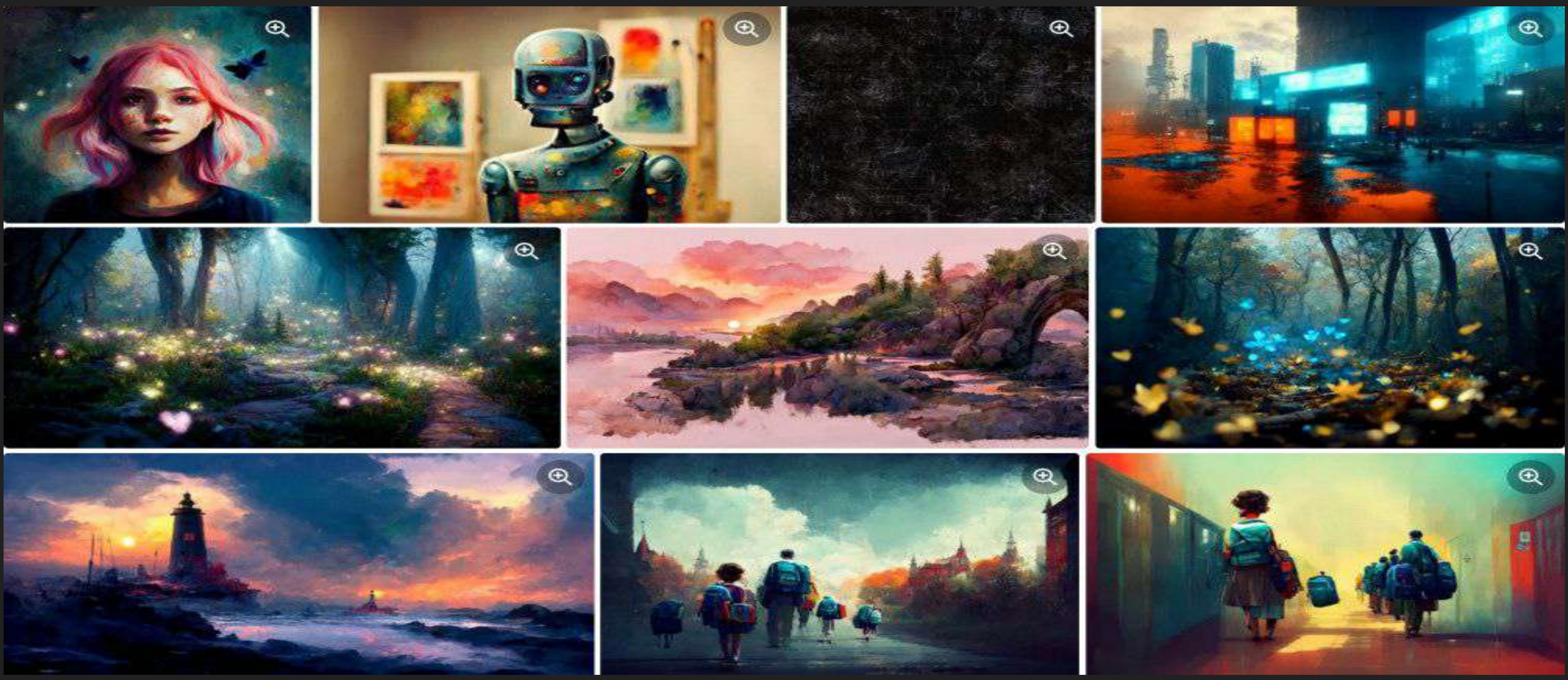
Image Captioning

Output

Input

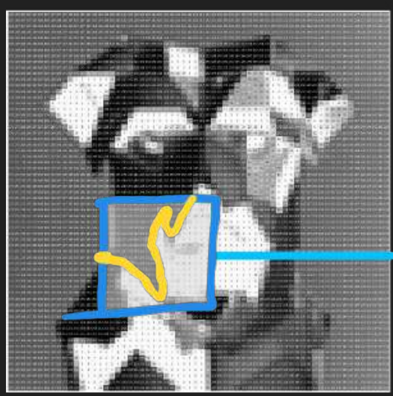
Generating Images

← Midjourney



What Computers Actually See?

Grayscale → B/W



>200

Height

158	176	246	246	251	241	235	242	254	249	244	253	248	255	127	0
159	172	243	247	249	239	240	251	255	185	220	255	249	244	27	4
160	168	239	248	247	250	253	252	246	109	247	250	255	160	4	28
161	164	237	248	248	249	249	255	199	15	234	255	254	97	27	3
162	163	235	250	248	249	246	255	122	0	188	255	195	24	0	4
162	162	233	252	249	249	251	250	44	0	139	255	62	0	8	6
163	158	228	254	249	246	255	188	0	0	93	185	0	0	0	0
161	165	236	252	249	246	255	190	0	0	38	68	13	50	78	87
160	224	253	247	249	248	249	251	58	0	12	25	55	86	100	67
207	255	251	249	255	247	247	255	189	0	8	32	0	0	0	0
255	251	255	145	144	255	244	248	253	58	0	7	12	12	9	5
255	248	251	46	0	192	255	241	255	112	0	3	1	3	3	3
248	255	205	3	0	22	229	250	255	167	0	8	1	4	3	2
243	255	154	0	12	0	66	251	253	209	5	12	10	5	5	3
245	255	182	16	0	7	0	116	255	232	30	0	3	5	1	5
250	252	227	155	25	7	2	0	169	255	57	8	34	4	1	4

White

Width

H×W

→ Each value in matrix – 0 to 255

Image → $\boxed{\text{Computer}}$

0 to 255

A computer sees numbers in an image

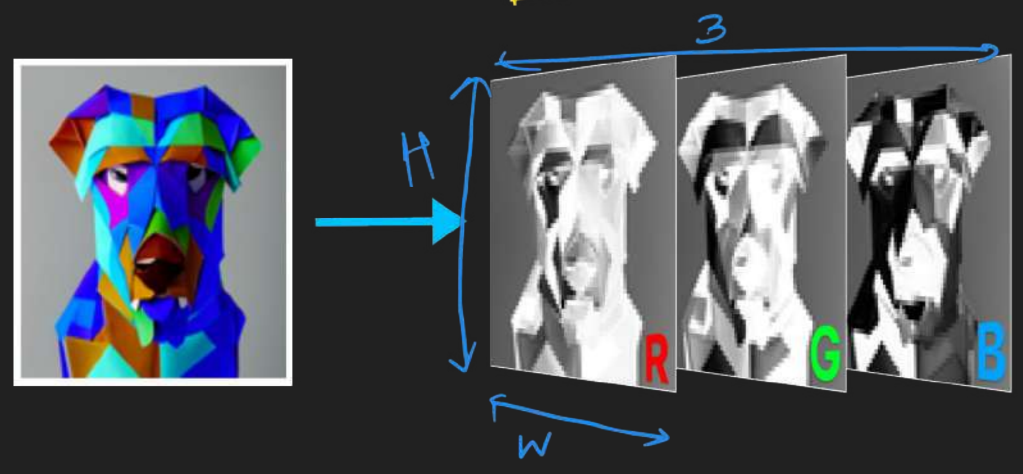
values of pixels

Any coloured image can be represented by combination of RGB

What Computers See?

Channels

Red Green Blue



Vector → $[\quad]_{1\times N}$ $[\quad]_{N\times 1}$

Matrix $[\quad]_{N\times M}$

Tensor — any D

→ Q - color image $\underbrace{[\quad]_{H\times W\times C}}_{\text{Tensor}}$

→ Q - grayscale - $[\quad]_{H\times W}$

$[\quad]_{30\times 30\times 3}$ → $\boxed{\text{DTree}}$

Tensor ← Vector

Tensor ← Matrix

Tensor

HxWxC

30x30x3 → $\boxed{\quad}$ 2700

+ Dtree

$\begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \end{bmatrix}$

Vector

[1, 2, 3]

Matrix

[[1, 2, 3]
[3, 4, 6]]

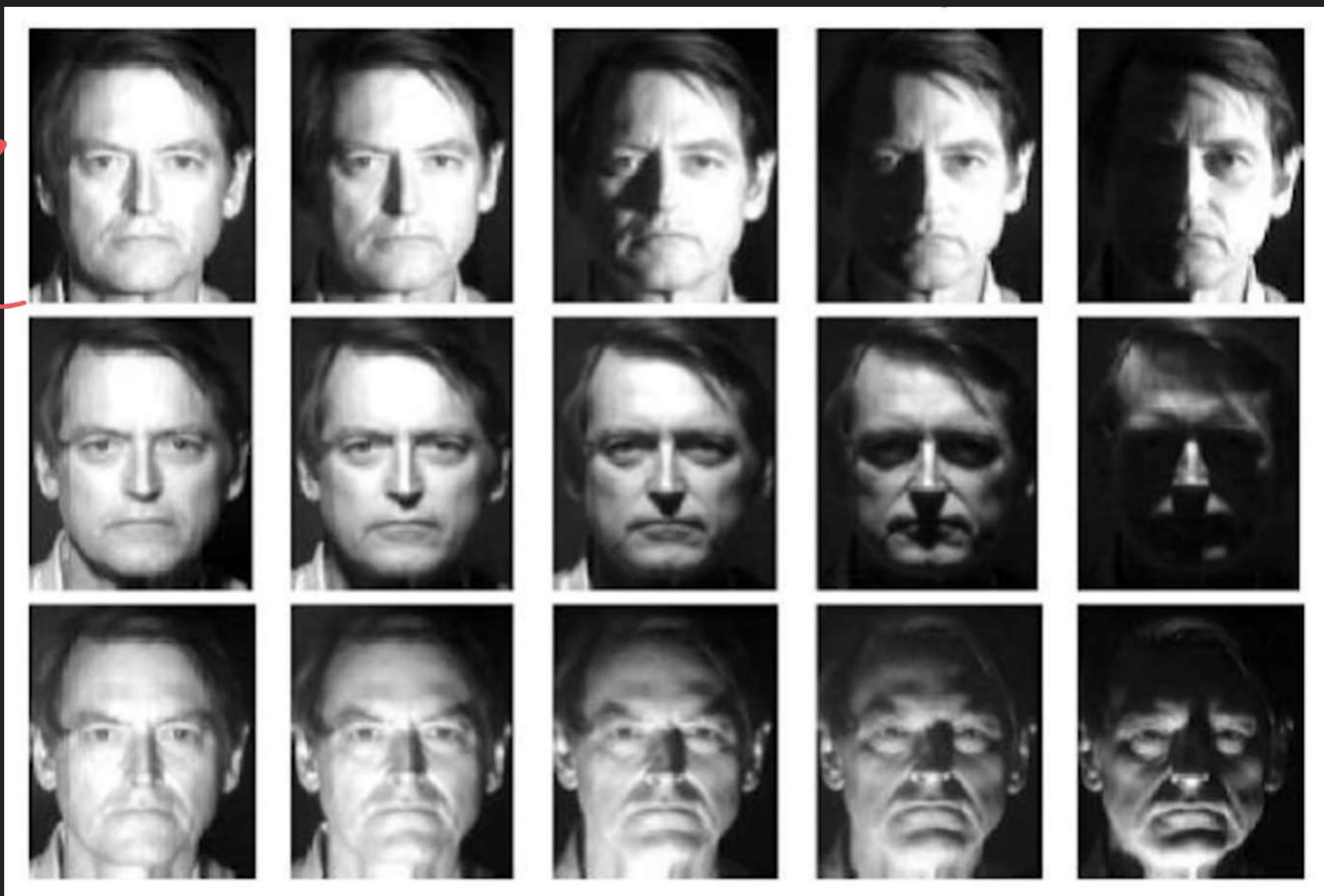
Tensor

[[[1 2 3]
[3 4 6]], [[6 7 8]
[9 10 11]]]

Is CV Easy?

Occlusion

Illumination variable



light
smile

Pose variability

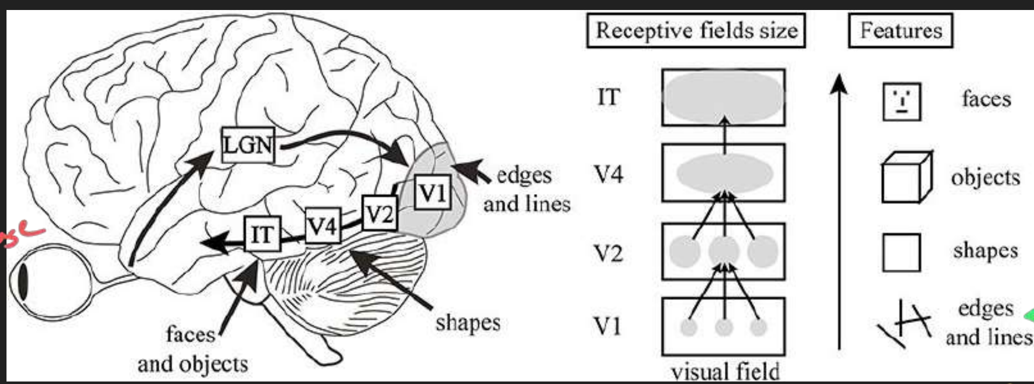
Figure 1. **The deformable and truncated cat.** Cats exhibit (al-

Text

Characters/Alphabets → Words → Text/Sentence → Paragraphs → Book

Why Deep Neural Nets for Images?

Image →



1

V1 → Gradients/Edges → } Alphabets

V2 → Shapes → } Words

V3 → Objects → } Sentences

V4 → Faces / Car parts / Animal leg

V5 → Human Being / Car / Animal

I/P

Line/Edge: ML1, 3-4 layers →

Shapes: ML2, 5-6

Eye/Face/Car Window: ML3

Human/Car: ML4

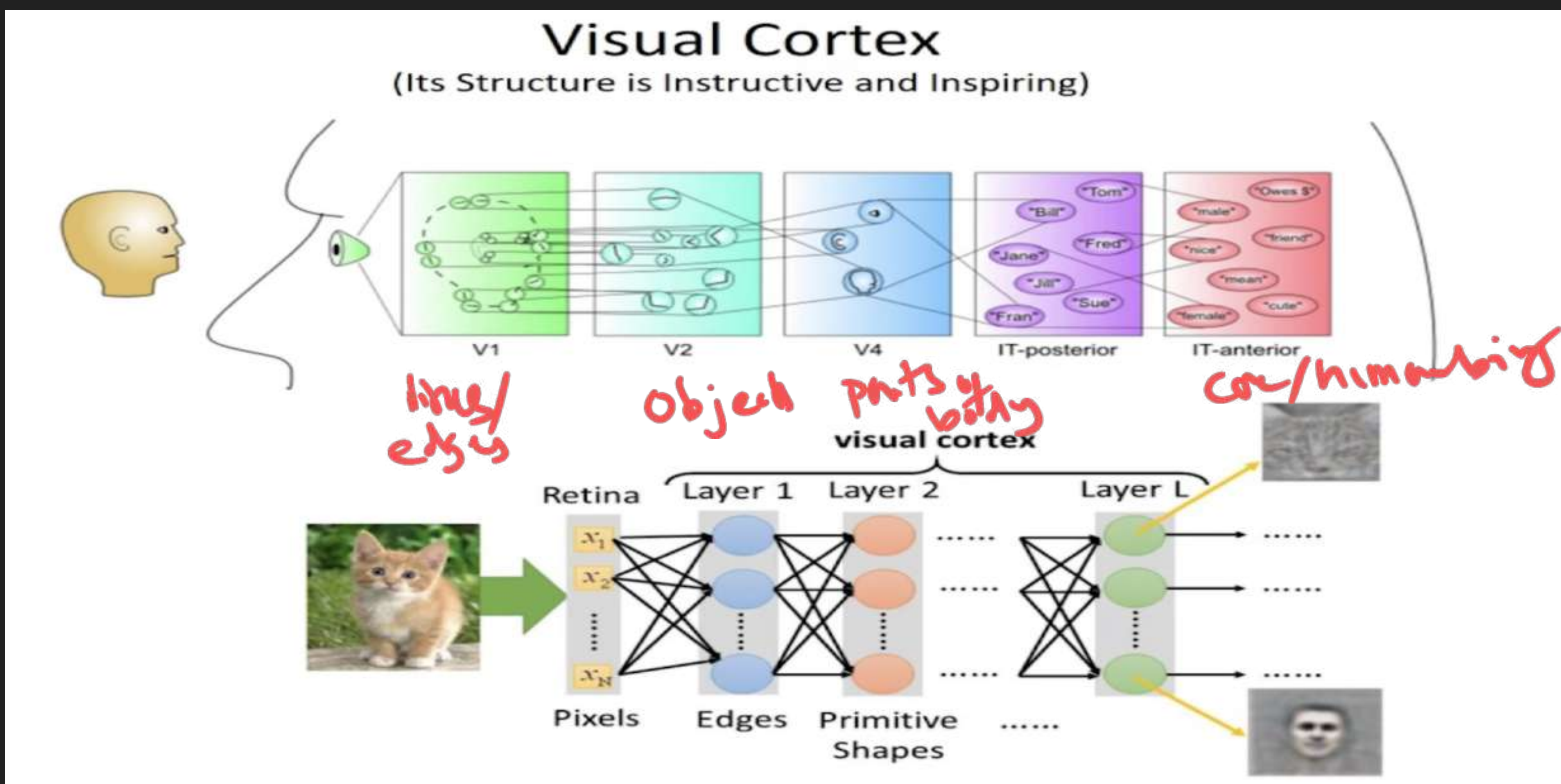
O/P

1000s of layers

8 bits ← pixel value

2^8 values → 0 to 255

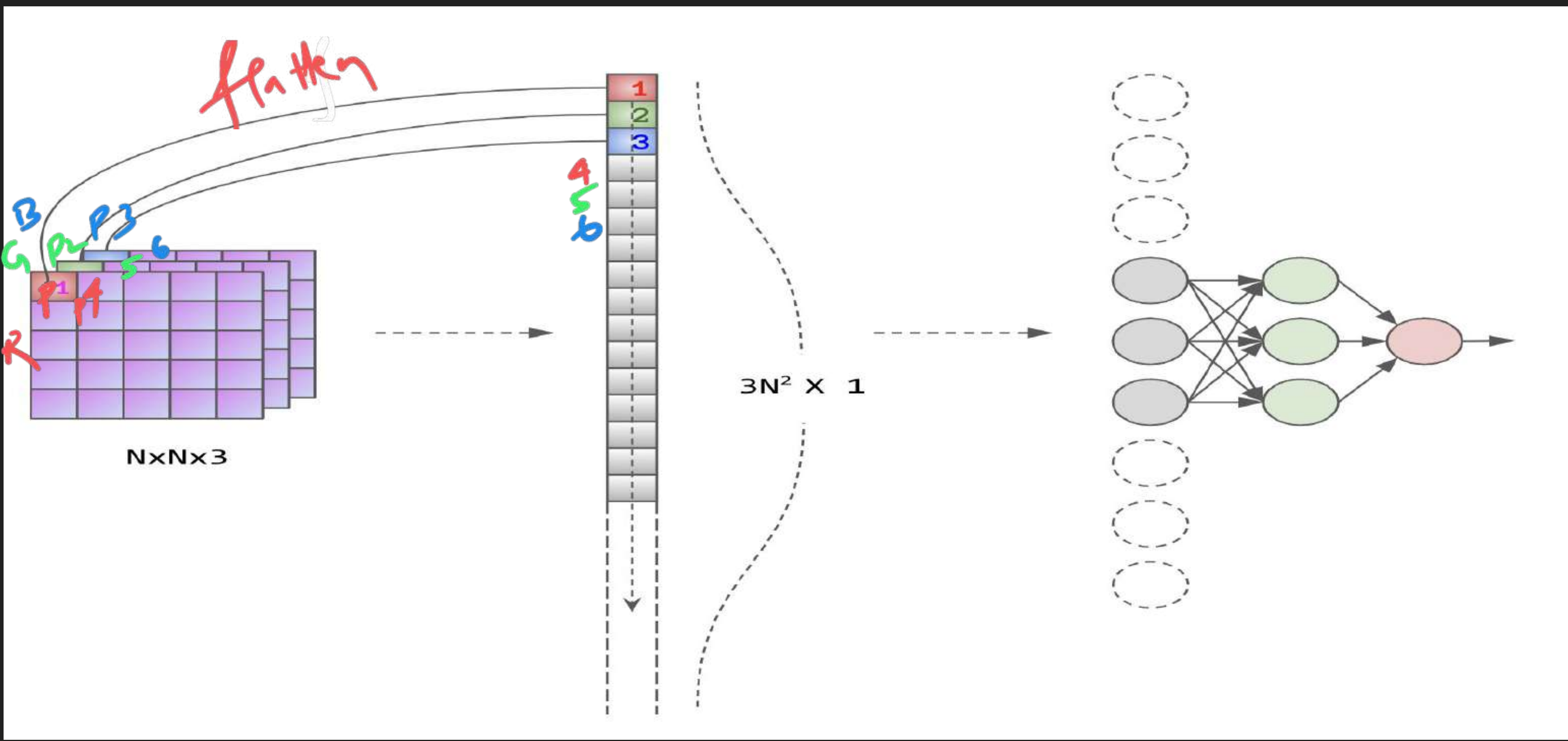
0 to 255 | 0



Visual Cortex
(Its Structure is Instructive and Inspiring)
"Tom"
"Bill"
"Fred"
"Jane"
"Jill"
"Sue"
"Fran"
"Owes $"
"male"
"friend"
"nice"
"mean"
"cute"
"female"
V1
V2
V4
IT-posterior
IT-anterior
lines/ edges
objects
parts of body
cat/human being
visual cortex
Retina
Layer 1
Layer 2
Layer L
Pixels
Edges
Primitive Shapes

$30 \times 30 \times 3 \rightarrow$ flat vector $\rightarrow 2700$
Tensor $\qquad\qquad 3*(30)^2$

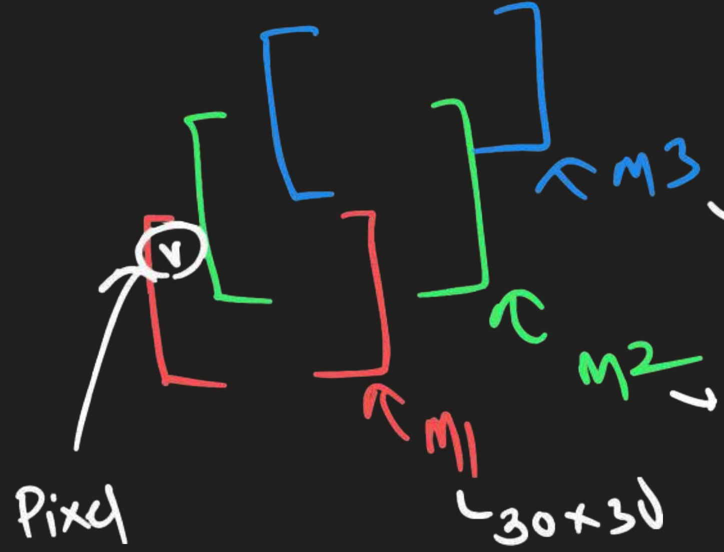
Preprocessing



1) Flatten ($N \times N \times 3 \rightarrow 3N^2 \times 1$)
2) Dense + Relu
3) Dense + ReLU
4) Dense + softmax $\qquad \leftarrow$ Output

Train data $\rightarrow$ 1000

$1000 \times 30 \times 30 \times 3 \xrightarrow{\text{flatten}} 1000 \times 2700$



Img

$30 \times 30 \times 3$

2700 features

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}_R \quad \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}_G \quad \begin{bmatrix} 9 & 10 \\ 11 & 12 \end{bmatrix}_B$ ← Image

flatten

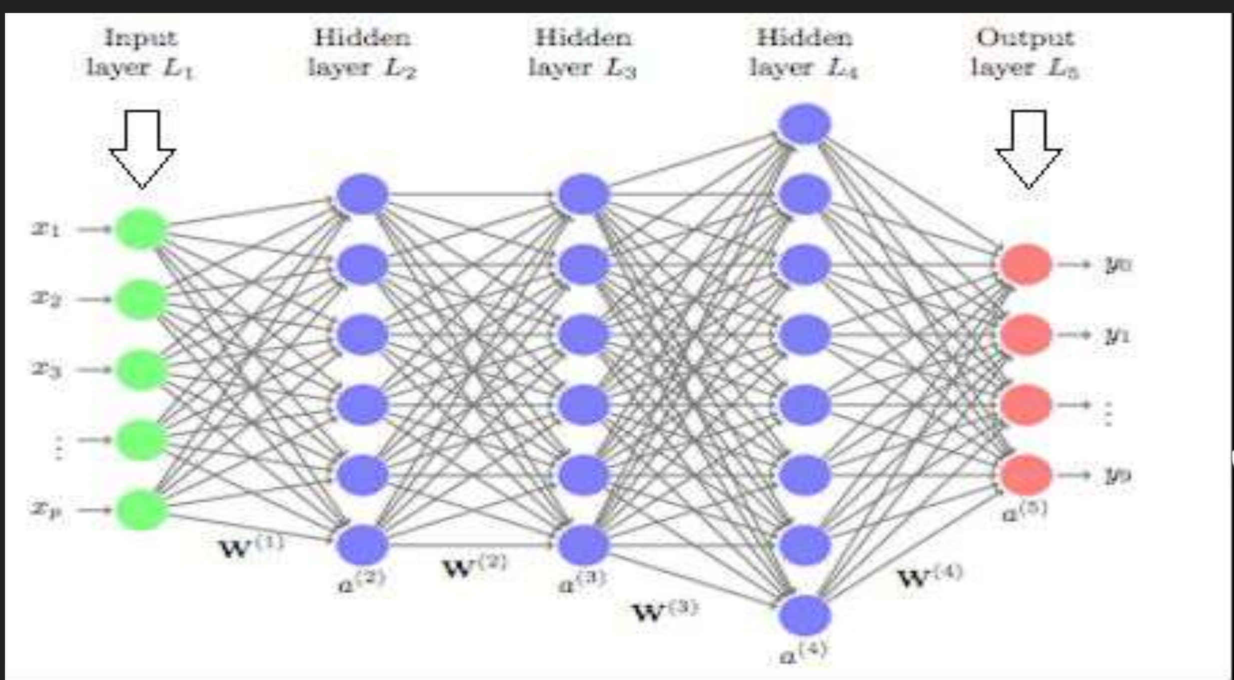
$[\,1 \;\; 5 \;\; 9 \;\; 2 \;\; 6 \;\; 10\,]$

NN

10 classes

$8 \times 2700 \rightarrow$ NN $\rightarrow$ 1 2 … 10

Fitting a Simple NN



$8N^2 \times 1$

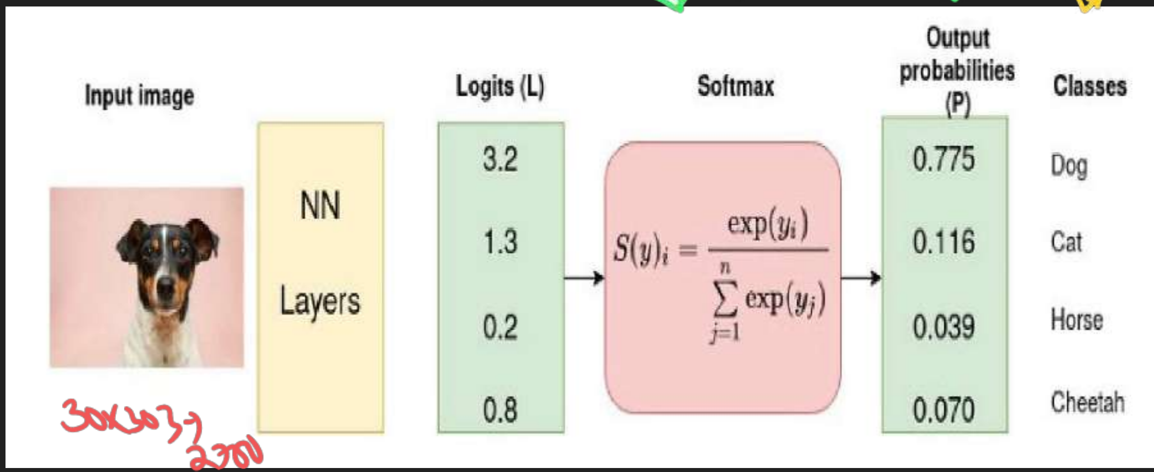
10 classes

Cross Entropy loss ← Fight

Model

Ground truth

NN Loss



Input

Output

→ Pant

30x30x3 → 2700

Batch Size - 4

$4 \times 30 \times 30 \times 3$ → 4 x 1 [Pant, Shirt, Tshirt, Sweater]

Batch

Channel

Each image → 30x30x3 ← Pant

30x30x3 ← Shirt

30x30x3 Input → Model →

Shirt	Pant	Shirt
0.2	0.8	0.2

BS = 8   # of featn = 10

Batch → [ ] 8x10

Each Image   30x30x3

Batch → 8 x 30 x 30 x 3

Batch Size

Total 1000

↓

Train

$\underbrace{1000}_{\text{no of data point}} \times \underbrace{30 \times 30 \times 3}_{\text{image each}}$

→ 30×30×3

→ 2700

→ NN — L1 + Relu
L2 + Relu
O + softmax

→ CE(Softmax, GT)

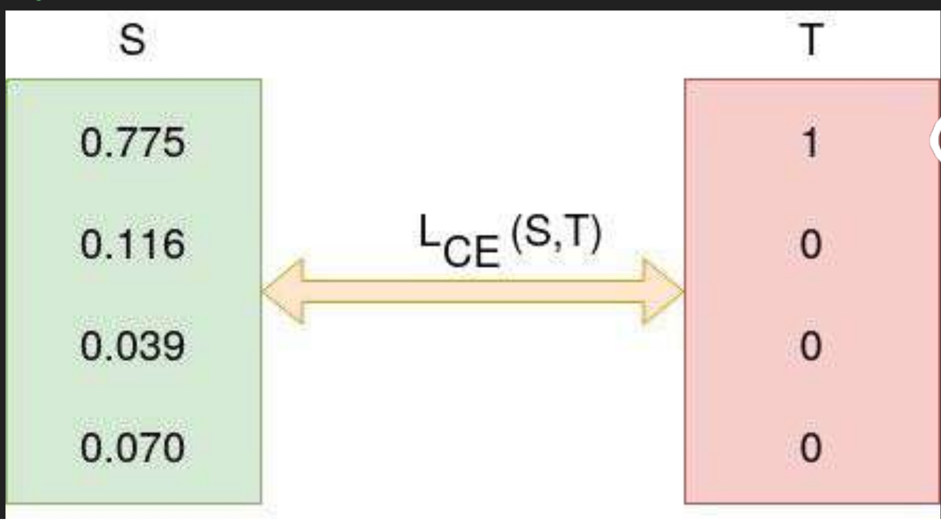
↑

Cross Entropy

Each Bat → 8 diff DP
8 diff images

NN Loss Modell

c1
c2
c3
c4



← C1 ✓

$$
\begin{aligned}
L_{CE} &= -\sum_{i=1} T_i \log(S_i) \\
&= -[1\log_2(0.775) + 0\log_2(0.126) + 0\log_2(0.039) + 0\log_2(0.070)] \\
&= -\log_2(0.775) \\
&= 0.3677
\end{aligned}
$$

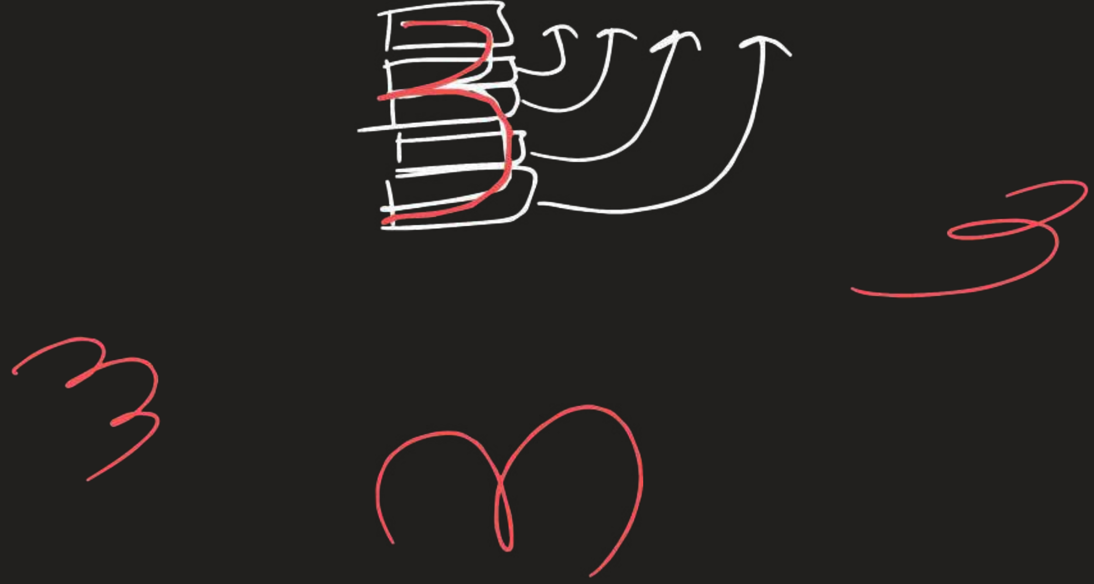
10 classes

8× 30×30×3 → 8×2700 → NN → 8×10

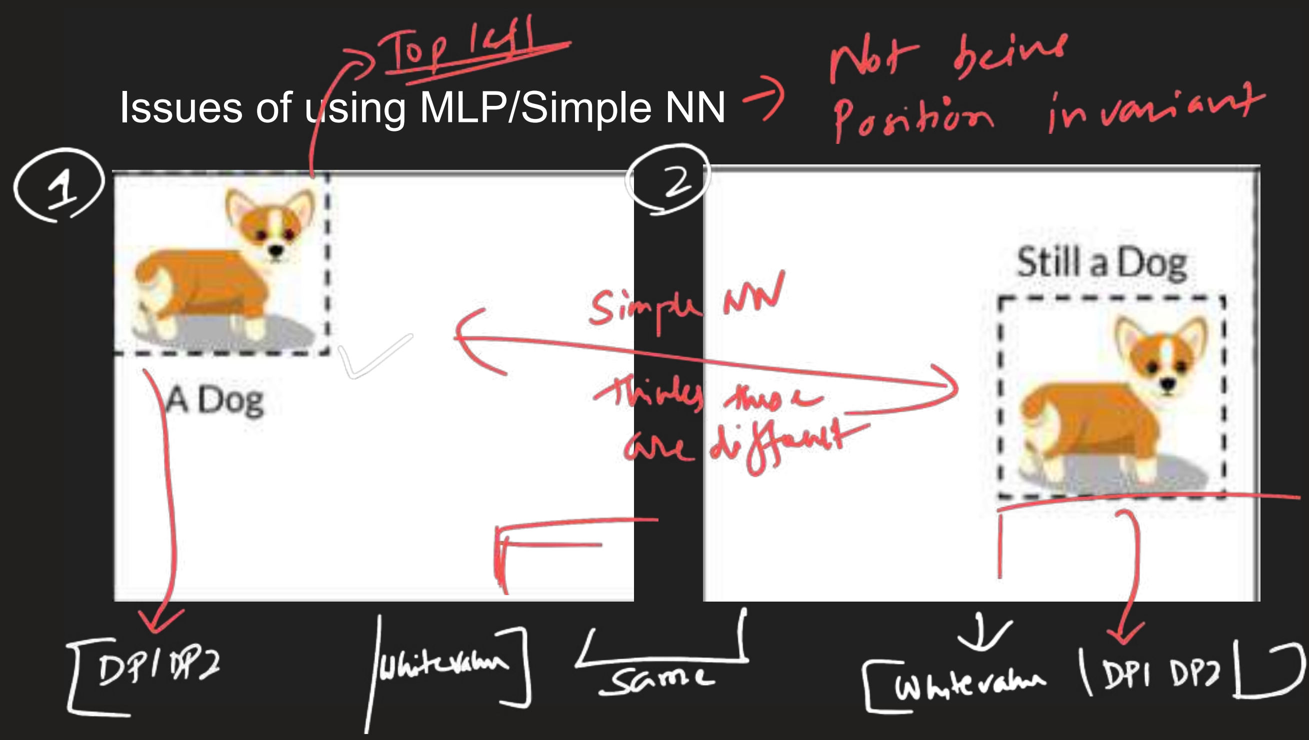
GT
8×1

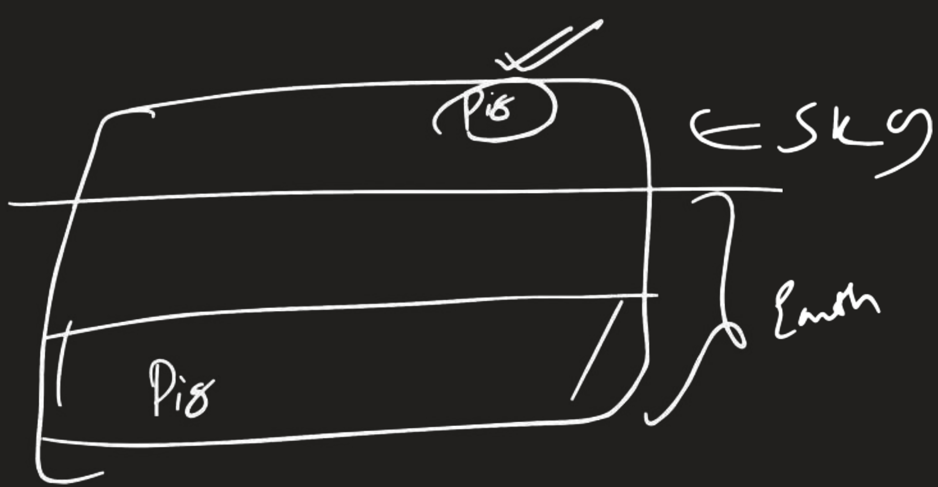
↑

Sparse categorical Cross Entropy



Issue → Spatial info completely lost



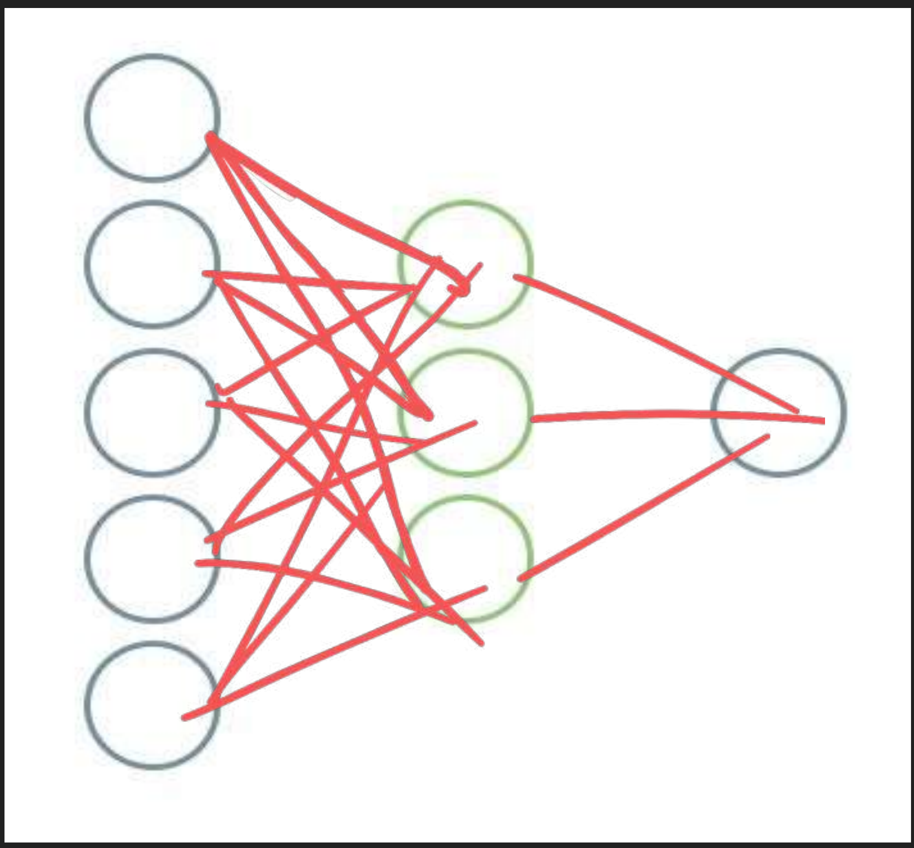


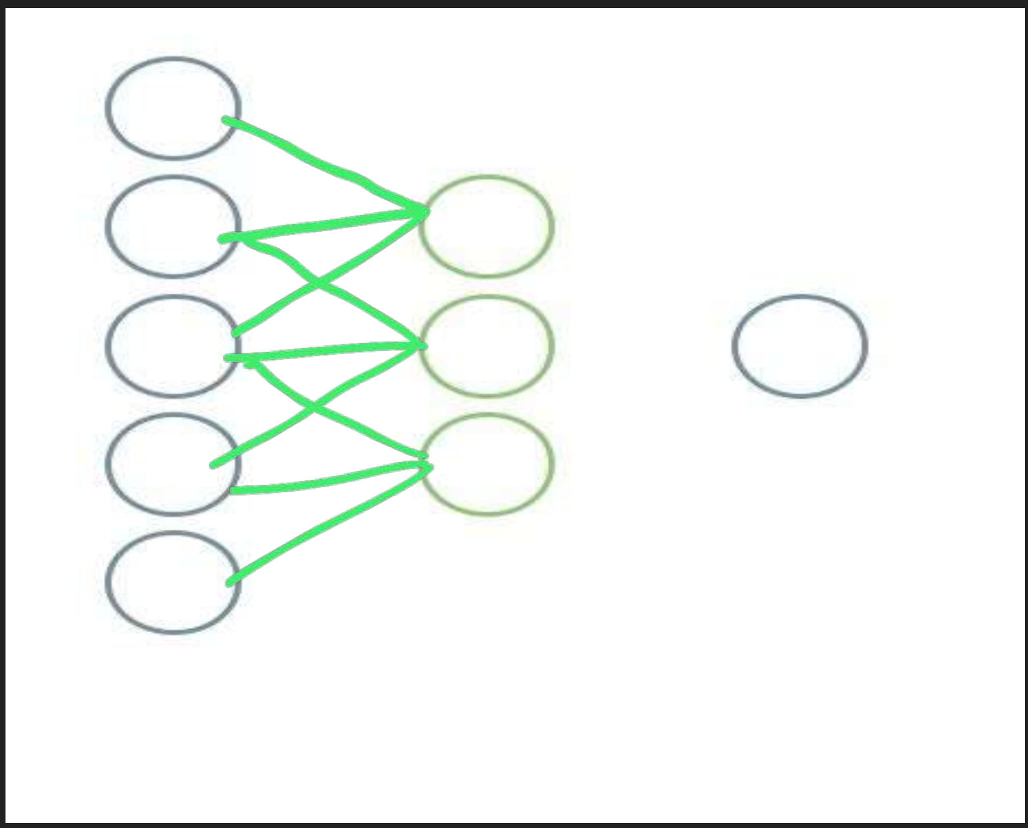
Issues

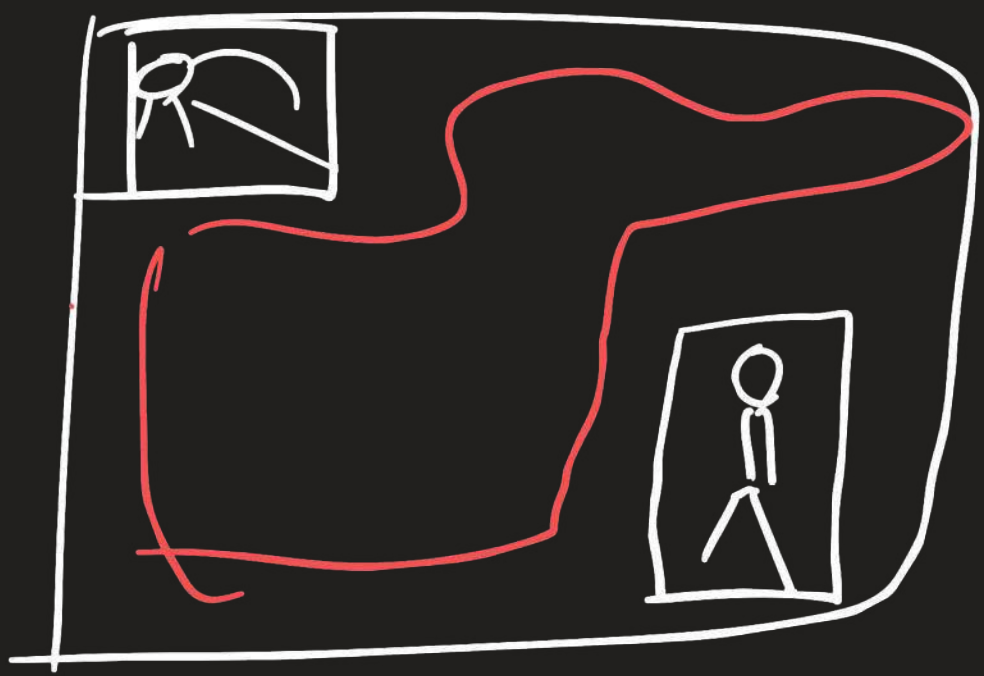
→ Not translation invariant

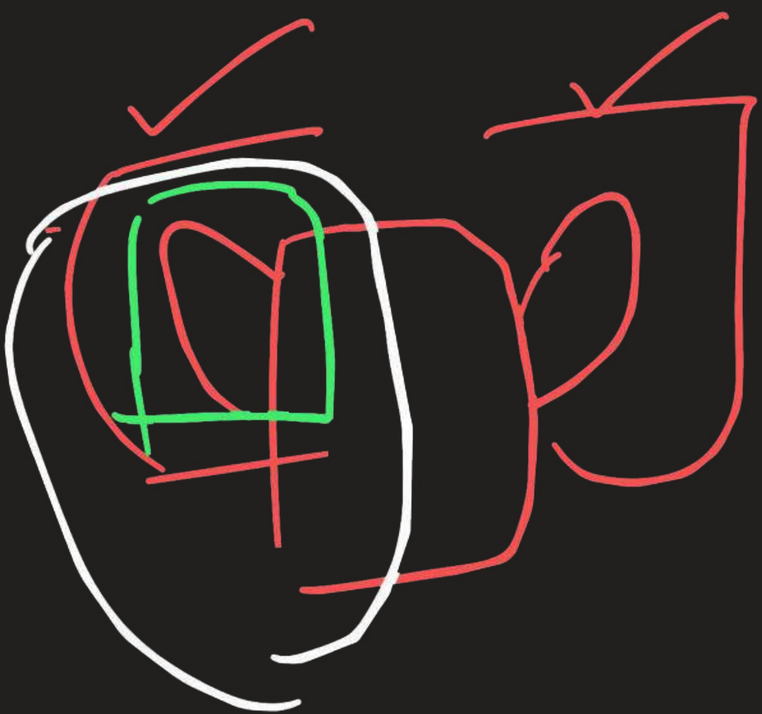
→ Too ~~many pa~~ many parameters } $

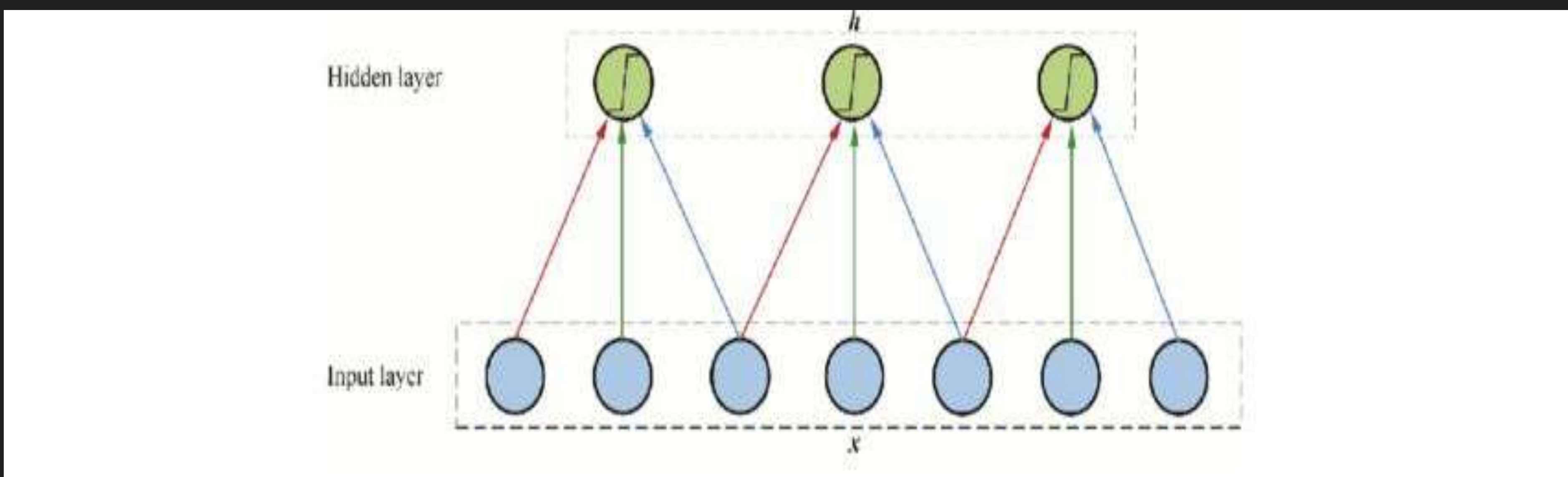
Improving MLPs → Enforcing locality

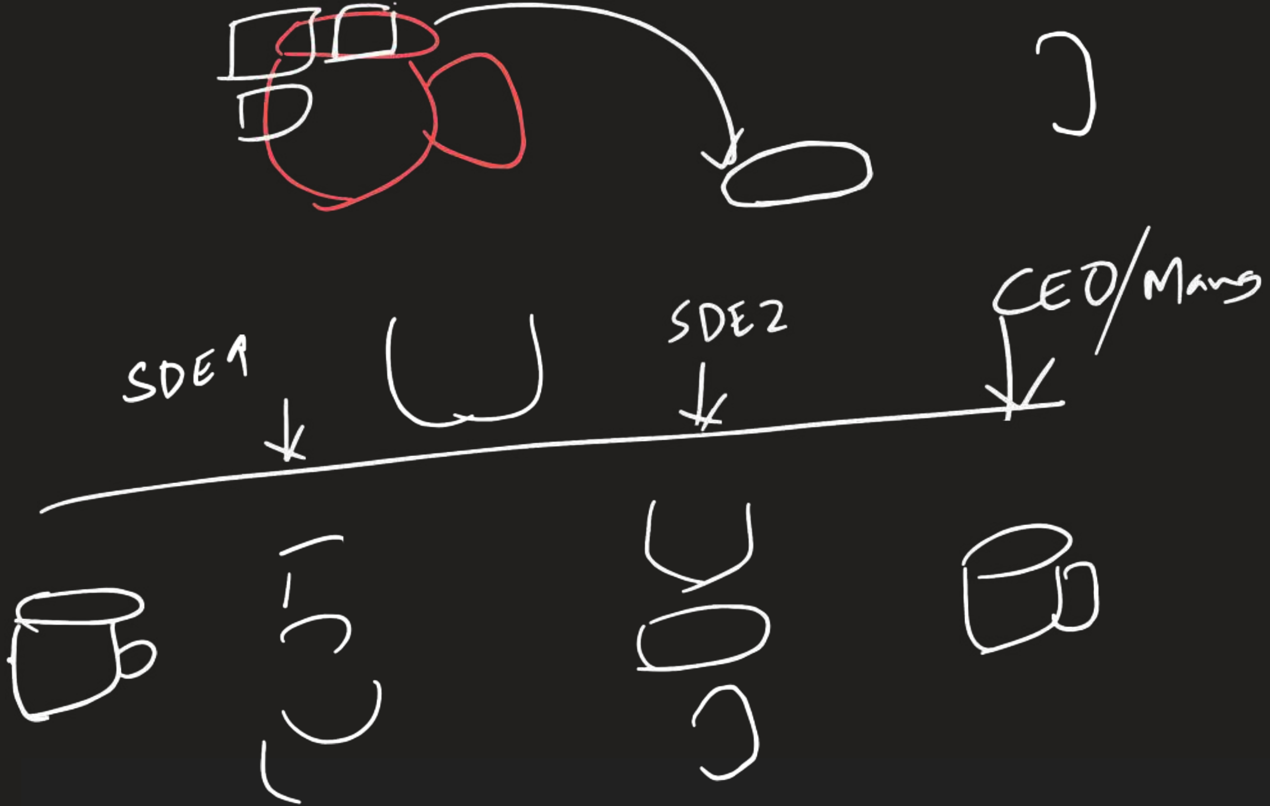




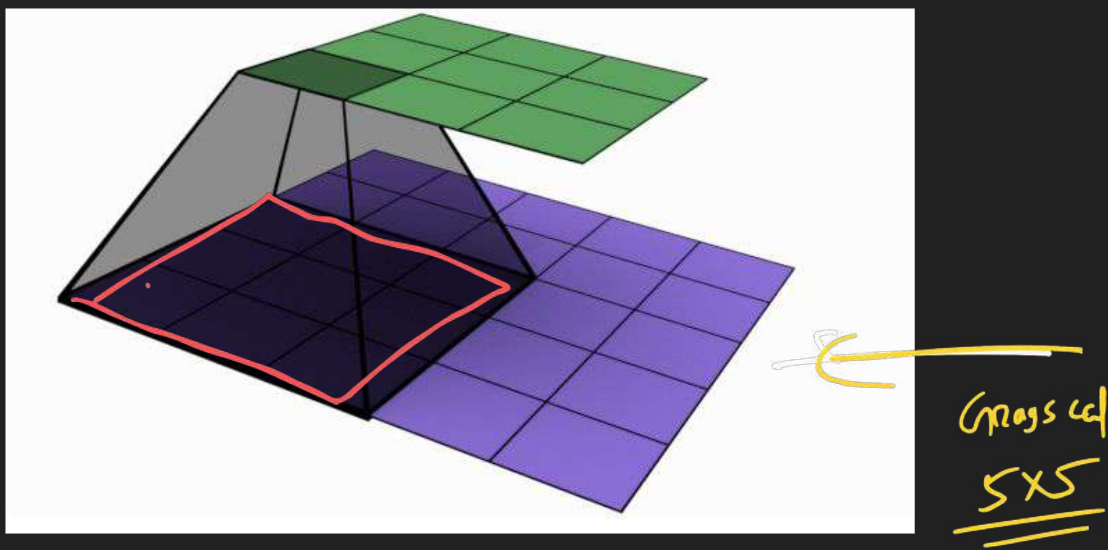


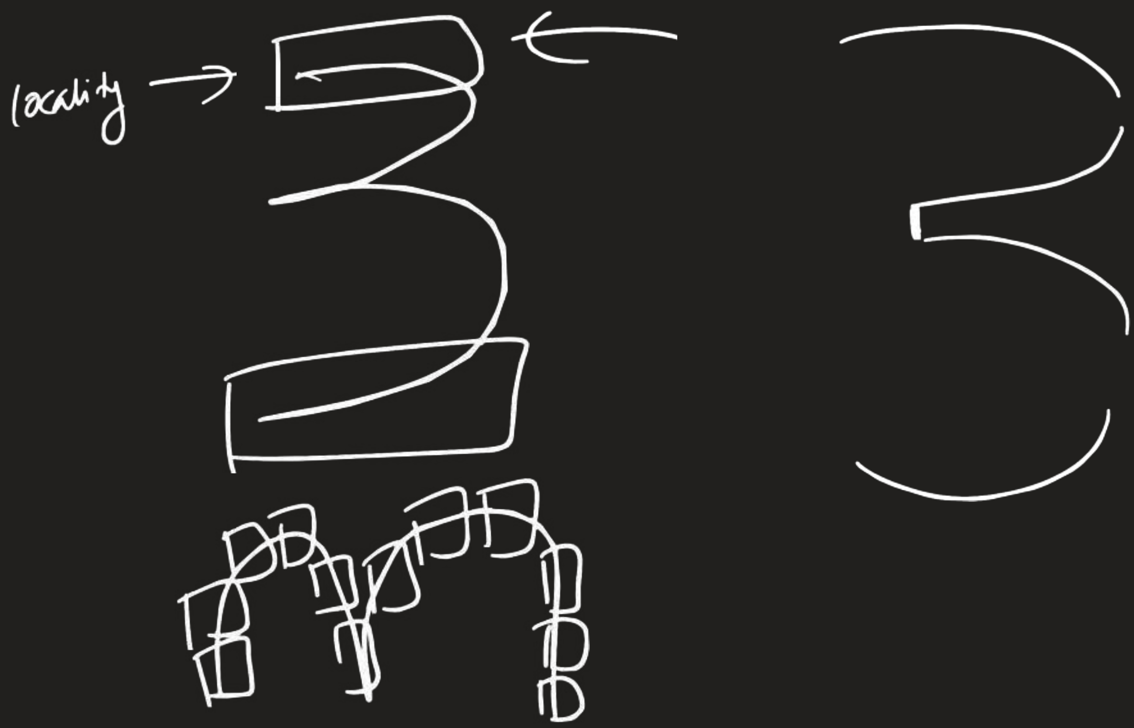


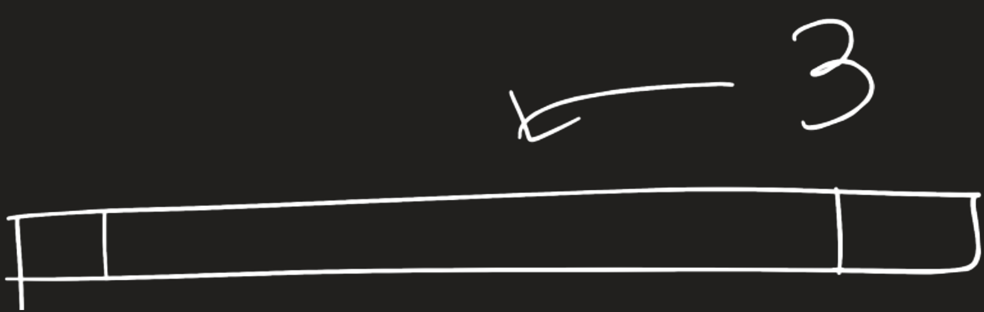

h
Hidden layer
Input layer
x



Convolutions







- → Images ← Dataset
- → Neural Nets ← images
- → Limitations