Deep Learning Models for Computer Vision
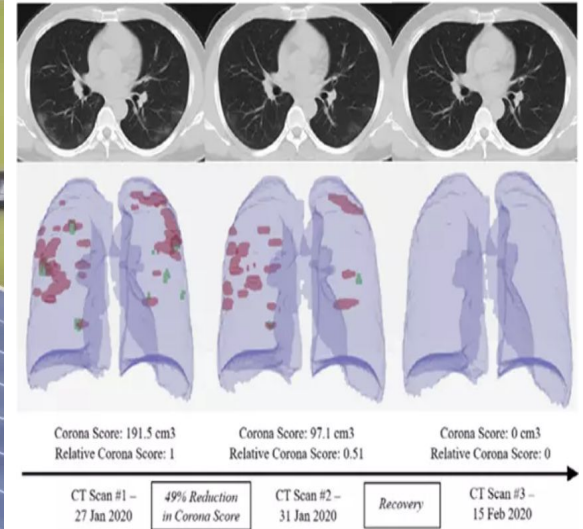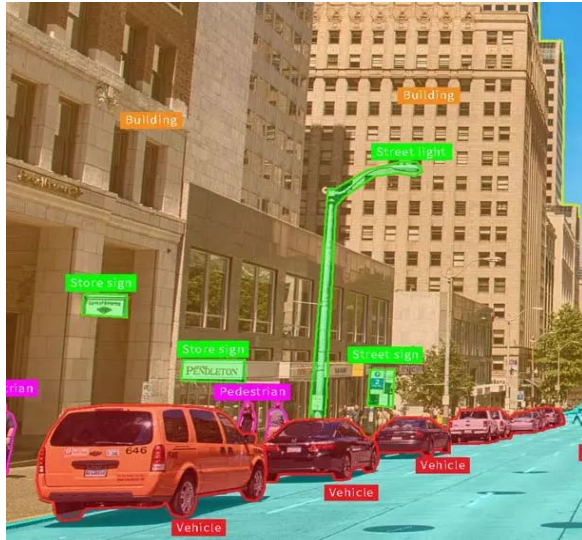
Shubham Agarwal
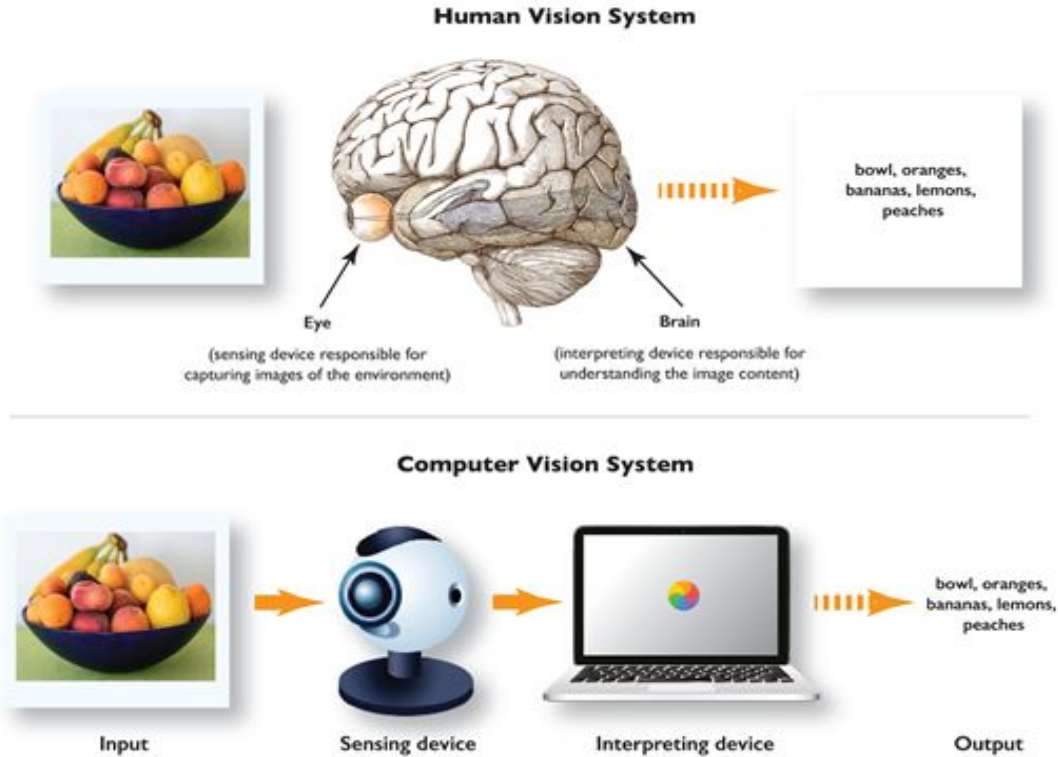agarwals@post.bgu.ac.il

Outline:

- What is computer vision?

- Brief introduction about neural networks

- Basic neural network architectures used in computer vision.

- Some more popular neural network architectures and their use cases.

- How can we start implementing these in code?

Computer Vision

Computer Vision is the branch of AI that enables machines to derive meaningful information from visual inputs like images and videos.

Image Source: https://www.thedatahunt.com/en-insight/computer-vision

How it works?

Human Vision System

Eye
(sensing device responsible for capturing images of the environment)

Brain
(interpreting device responsible for understanding the image content)

bowl, oranges, bananas, lemons, peaches

Computer Vision System

Input

Sensing device

Interpreting device

Output

bowl, oranges, bananas, lemons, peaches

Traditional way of Image Processing

Input   f(x,y)

$\frac{1}{9}$

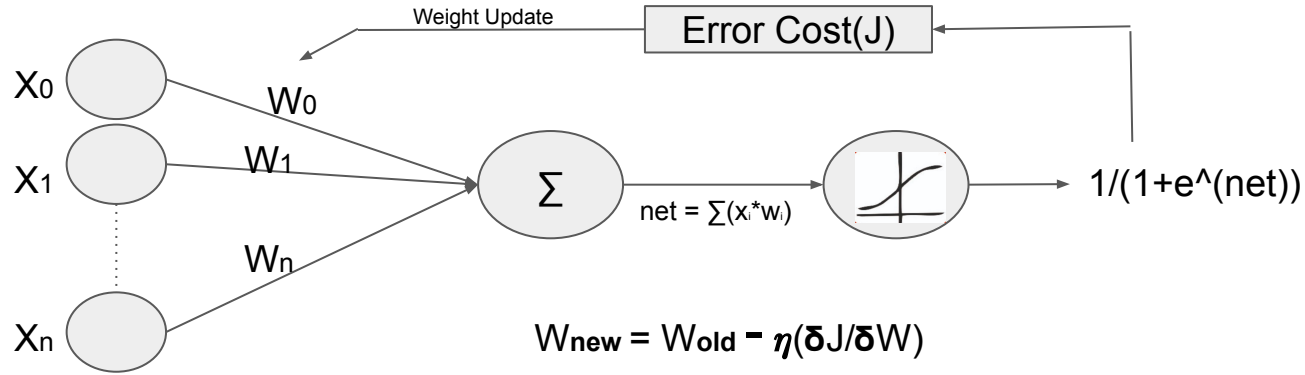| 1	1	1
1	1	1
1	1	1

Convolution operator, not multiplication!

Output g(x,y)

Neural Networks

Image Source: https://tikz.net/neural_networks/

Backpropagation

Weight Update

Error Cost(J)

X_0

W_0

X_1

W_1

Σ

$net = \sum(x_i \cdot w_i)$

$1/(1+e^\wedge(net))$

W_n

X_n

$$W_{new} = W_{old} - \eta(\delta J/\delta W)$$

$W^{[1]}$ $\frac{\partial Z^{[1]}}{\partial W^{[1]}}$

$W^{[2]}$ $\frac{\partial Z^{[2]}}{\partial W^{[2]}}$

X

$\frac{\partial A^{[1]}}{\partial Z^{[1]}}$

$Z^{[1]}$

$A^{[1]}$

$\frac{\partial Z^{[2]}}{\partial A^{[1]}}$

$Z^{[2]}$

$\frac{\partial A^{[2]}}{\partial Z^{[2]}}$

$A^{[2]} = \hat{y}$

$\frac{\partial J}{\partial A^{[2]}}$

COST

$b^{[1]}$ $\frac{\partial Z^{[1]}}{\partial b^{[1]}}$

$b^{[2]}$ $\frac{\partial Z^{[2]}}{\partial b^{[2]}}$
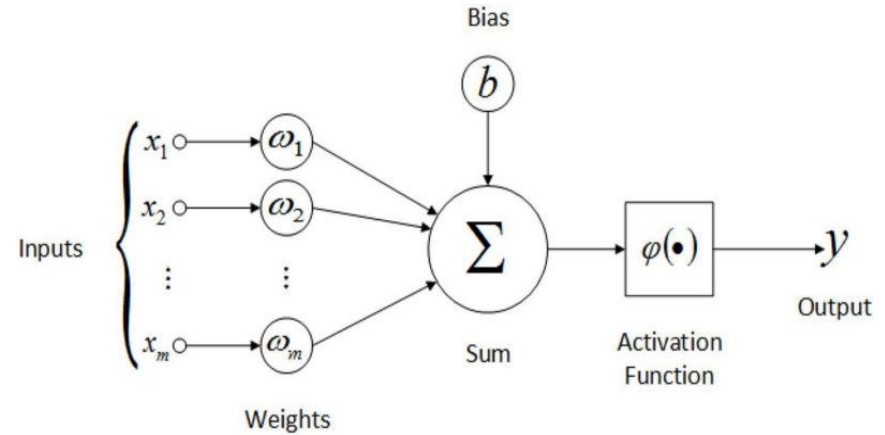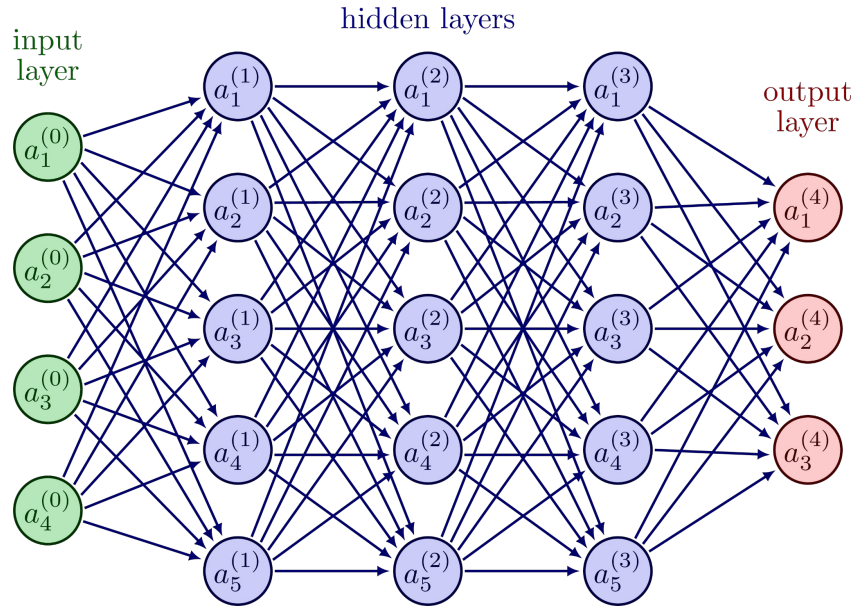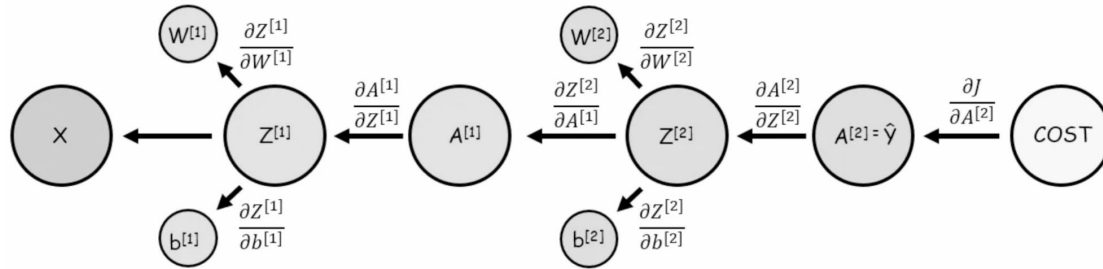
Feed Forward Neural Network

Each neuron input is a weighted combination of the outputs from all the neurons from the previous layer.
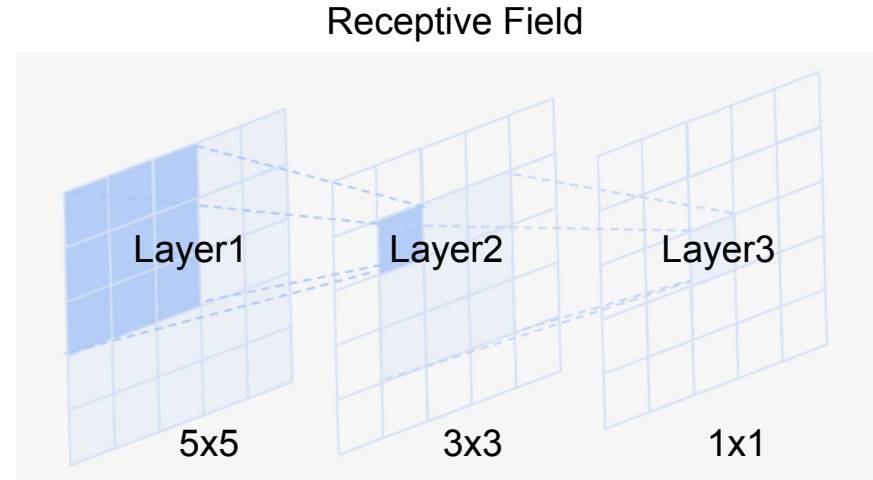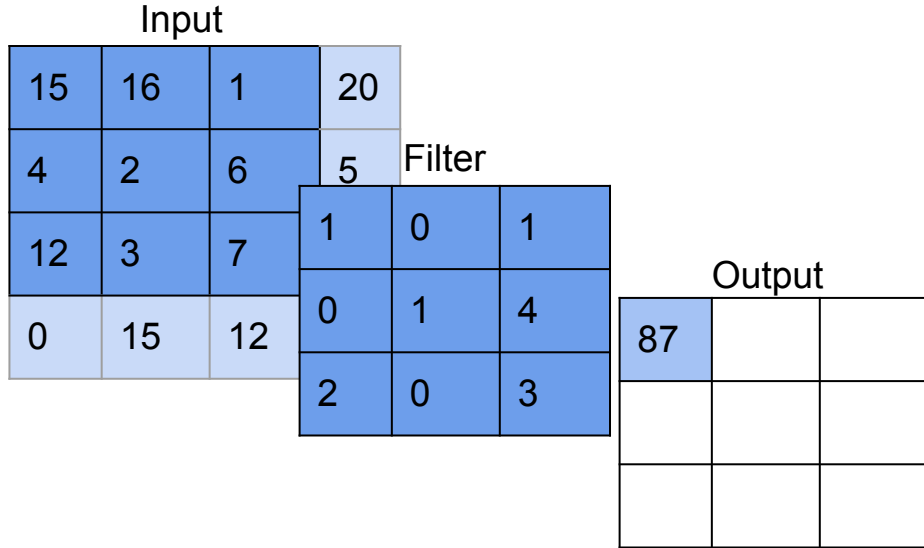
Advantage:

- Every neuron has a wholistic information from the input.
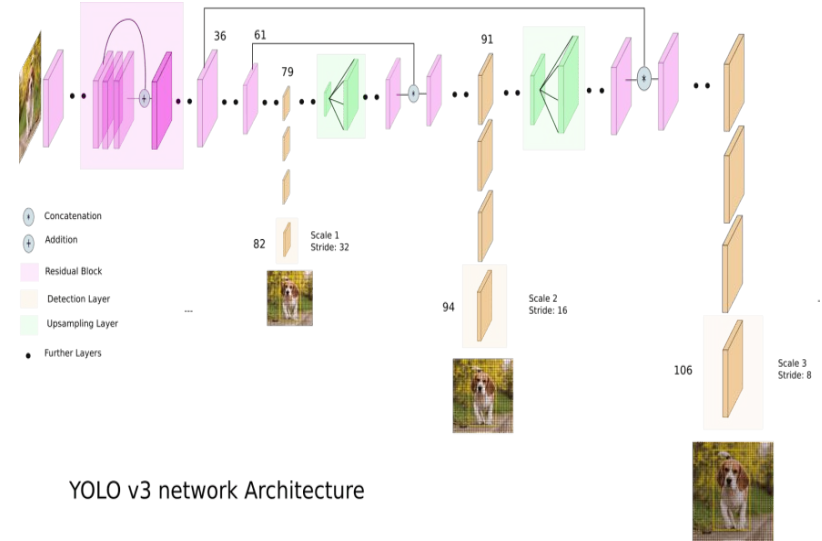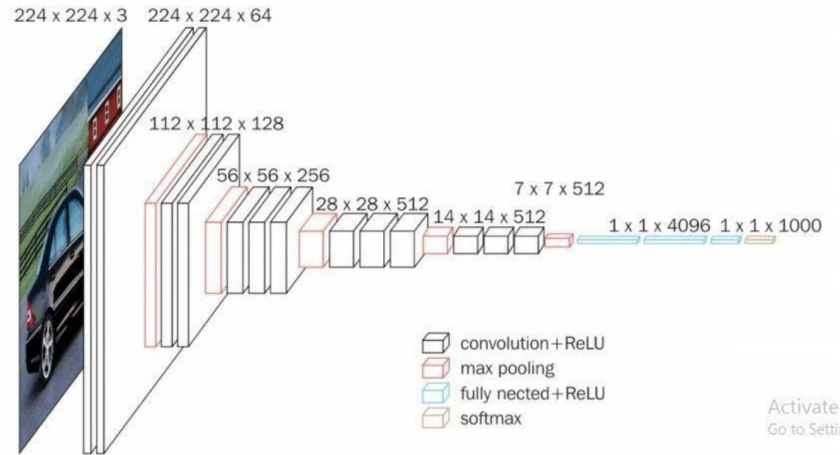- Flexibility to learn complex relationships in data.
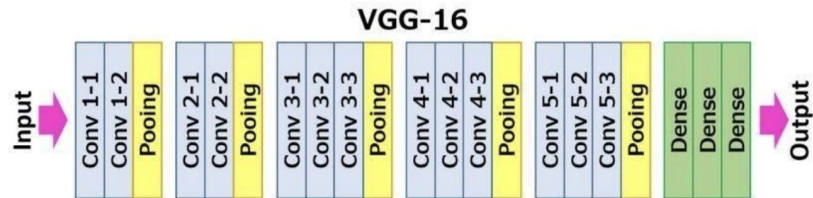
Disadvantage:

- Too many weights!

Convolutional Neural Network (CNN)

Input

15	16	1	20
4	2	6	5
12	3	7	
0	15	12	

Filter

1	0	1
0	1	4
2	0	3

Output

87		

Receptive Field

Layer1

Layer2

Layer3

5x5

3x3

1x1

Popular CNN architectures

224 x 224 x 3 224 x 224 x 64

112 x 112 x 128

56 x 56 x 256

28 x 28 x 512 14 x 14 x 512

7 x 7 x 512

1 x 1 x 4096 1 x 1 x 1000

convolution+ReLU
max pooling
fully nected+ReLU
softmax

Activate
Go to Setti

36 61
79
91
82 Scale 1
Stride: 32
94 Scale 2
Stride: 16
106 Scale 3
Stride: 8

Concatenation
Addition
Residual Block
Detection Layer
Upsampling Layer
Further Layers

YOLO v3 network Architecture

VGG-16

Input → Conv 1-1 | Conv 1-2 | Pooing | Conv 2-1 | Conv 2-2 | Pooing | Conv 3-1 | Conv 3-2 | Conv 3-3 | Pooing | Conv 4-1 | Conv 4-2 | Conv 4-3 | Pooing | Conv 5-1 | Conv 5-2 | Conv 5-3 | Pooing | Dense | Dense | Dense → Output

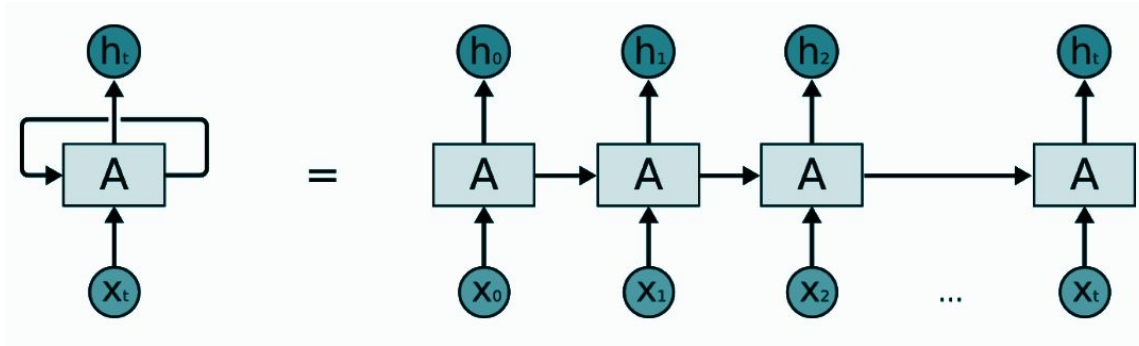Image Source: https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918
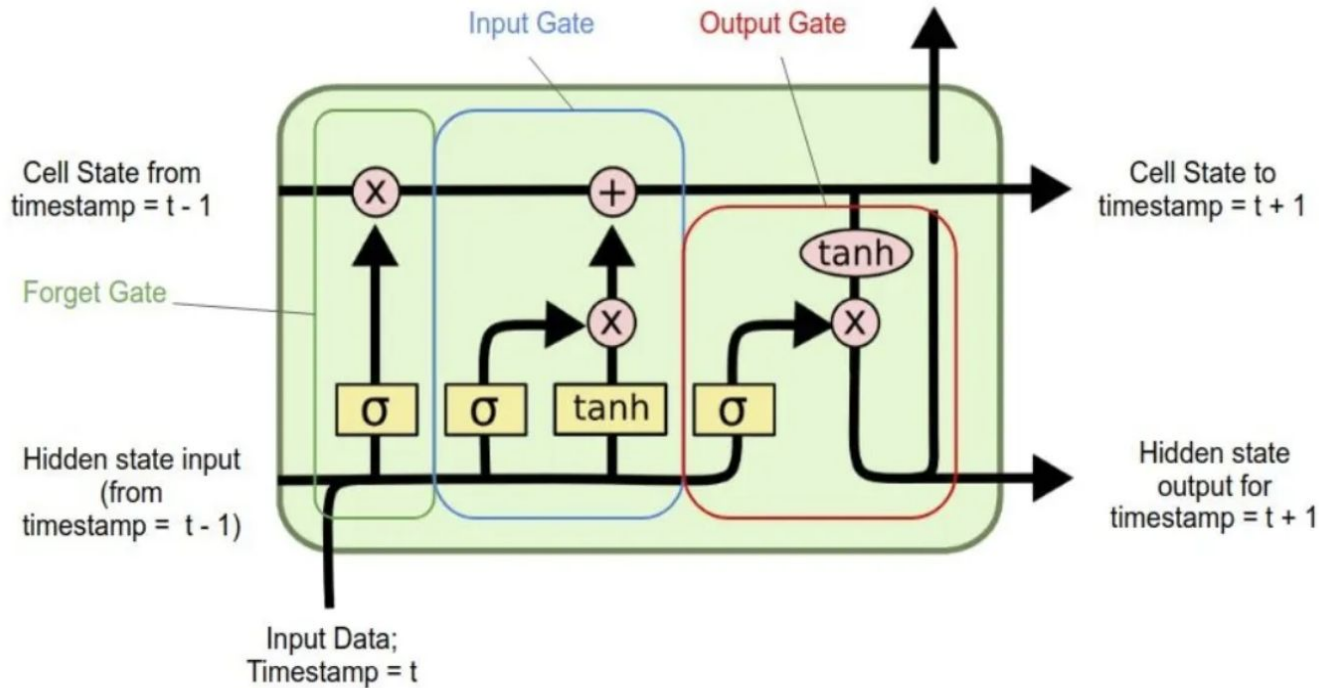
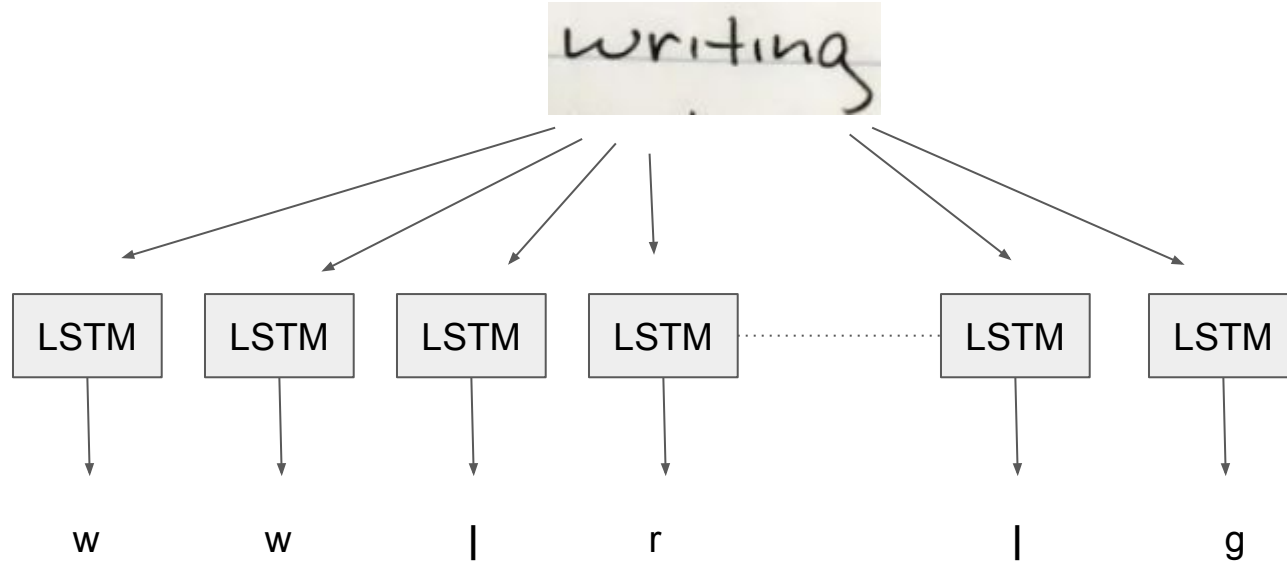# Recurrent Neural Networks (RNN)

Network with loops

- Can learn short term dependencies in a sequence of data.
- Struggles to learn long term dependencies.
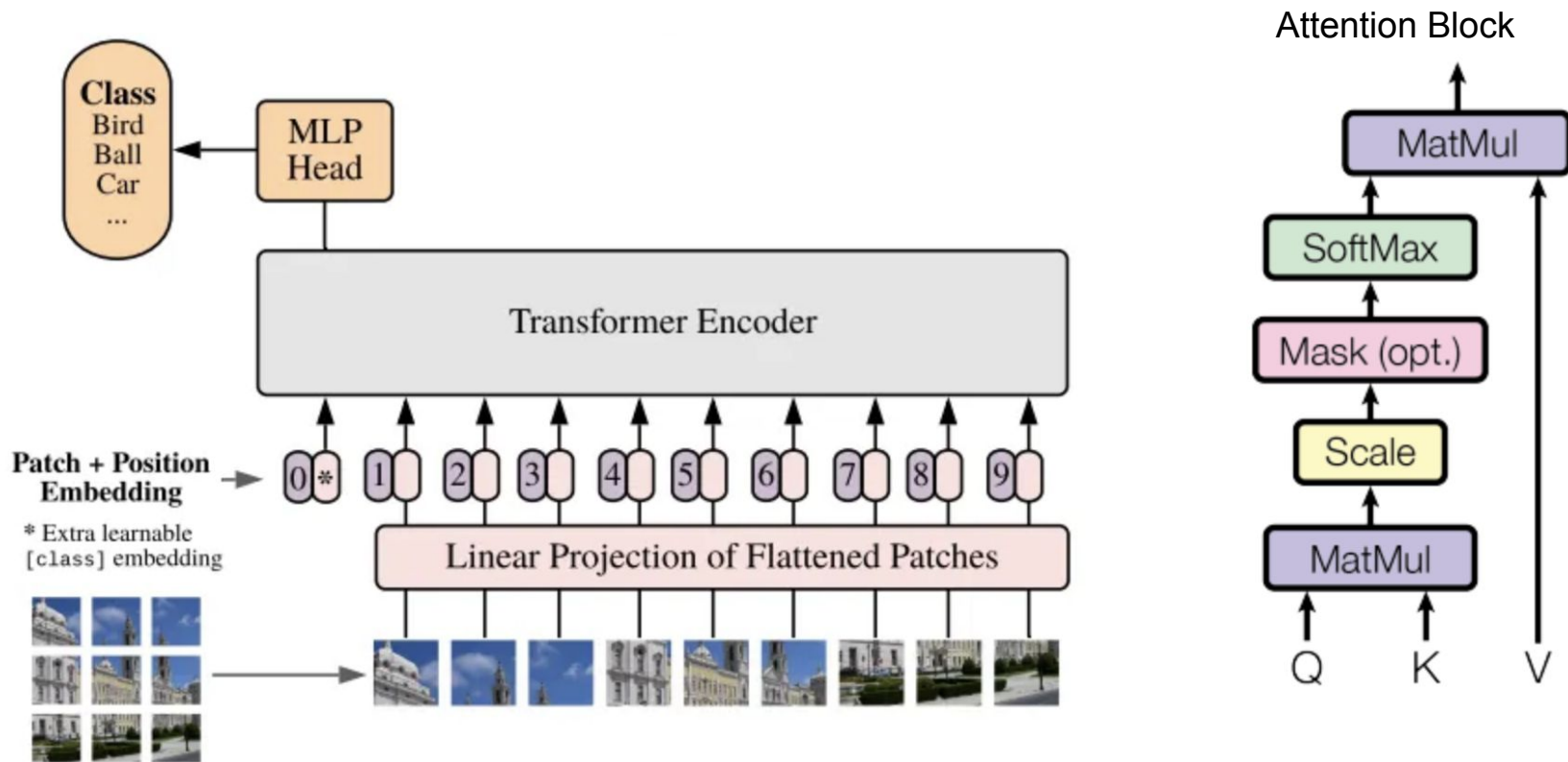- Difficult to train for long sequences as gradients need to be back-propagated in time.
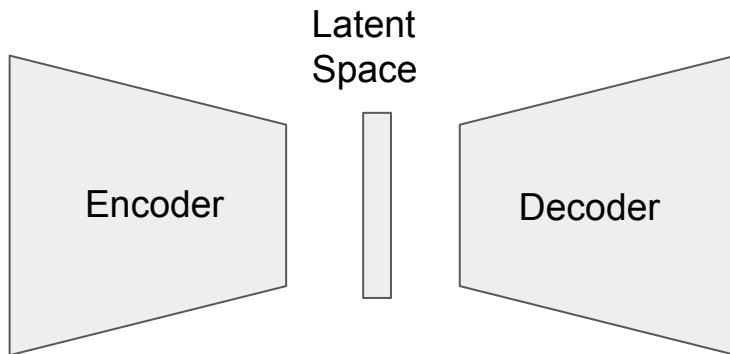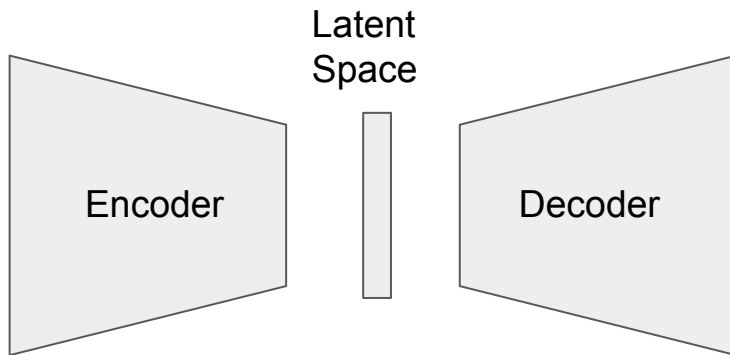
Long Short Term Memory Networks (LSTM)

Image Source: https://d2mk45aasx86xg.cloudfront.net/image17_11zon_2727417658.webp

Long Short Term Memory Networks (LSTM)

Vision Transformer

Attention Block

Representation Learning

Use Cases

Vision Transformer:

- Image classification
- Image Captioning
- Anomaly detection

Representation Learning:

- Compression
- Image restoration
- Synthetic data generation

Implementation

AutoEncoder:
https://colab.research.google.com/github/rickwierenga/notebooks/blob/master/autoencoders.ipynb

LSTM based Handwriting Recognition:

https://github.com/githubharald/SimpleHTR