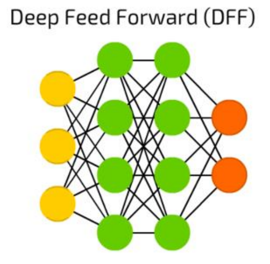
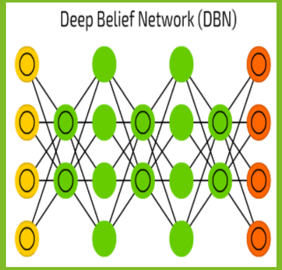
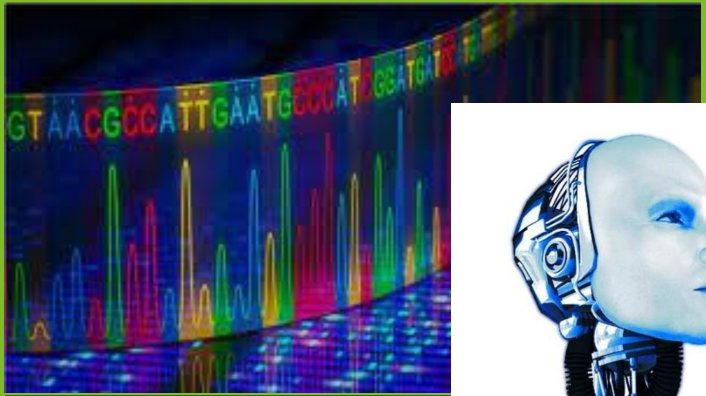
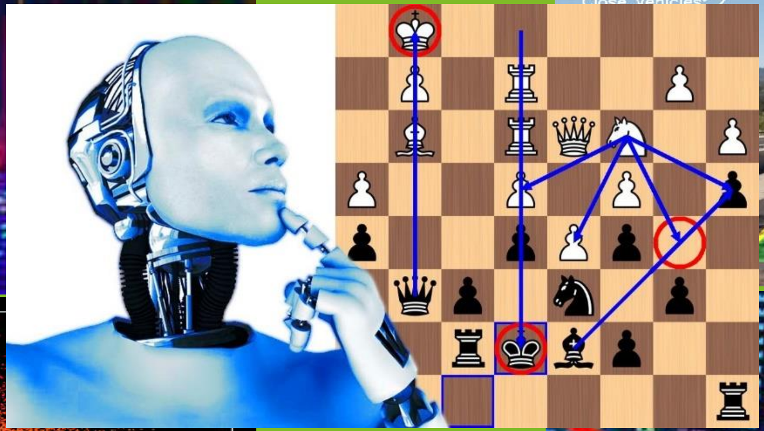
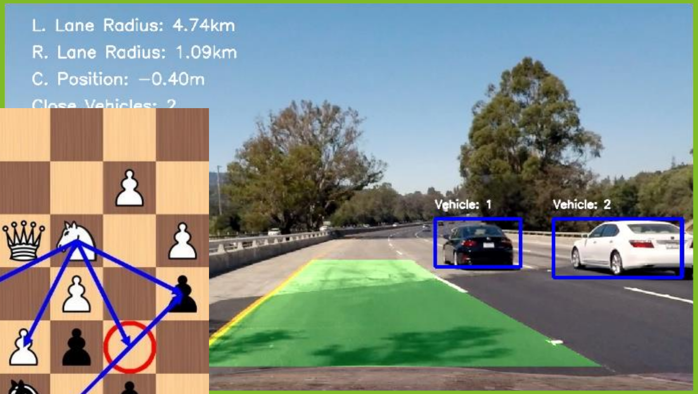
Deep Learning CSI_7_DEL

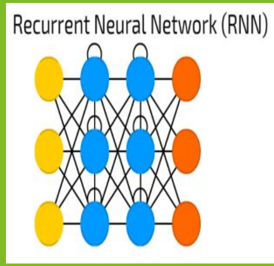


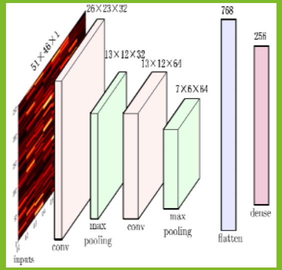


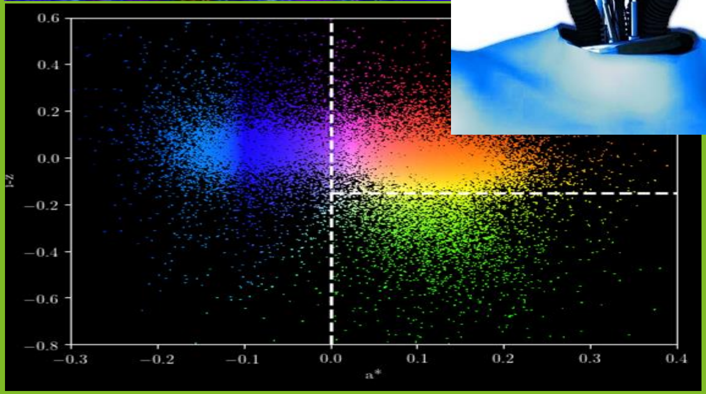


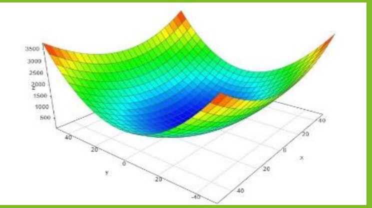












Week 6:Convolutional Neural Networks (CNN)

London South Bank University
EST 1892

Neural Network Architecture Hyperparameters

- Number of **hidden layers** and units **start small and then increase the** number of neurons/layers until the accuracy is improved.
- Network weight initialization (Mostly **uninform distribution** is used check out others).
- Activation functions (Choose them based on lecture 5)
- **Dropout** a regularisation technique that drops random neurons in each layer to **prevent overfitting.** (Recommended between 0.5 and 0.8 for the hidden layers)

London South Bank University
EST 1892

Hyper-parameters: Learning rate

- Learning rate specifies how quickly the network should update it parameters.
- This parameter is denoted by the Greek letter η (Eta)
- A low learning rate slows down the learning process but produces a smooth convergence.
- A High learning rate speeds up the learning process, but convergence is not guaranteed.
- Usually, a **default learning rate** in most software's is set to 0.01 as it’s considered **a good starting point**.
- Starting with **a large learning rate** and slowly reducing it is a commonly used technique called **learning rate decay**.

Hyper-parameters: Momentum

- **The momentum** is a decimal number used to **help the optimisation algorithm choose the next step** based on the previous step.
- This is used to **prevent oscillations** when descending or ascending the convex space.
- A typical value for the momentum is **between 0.5 and 0.9**

Hyper-parameters: Number of epochs

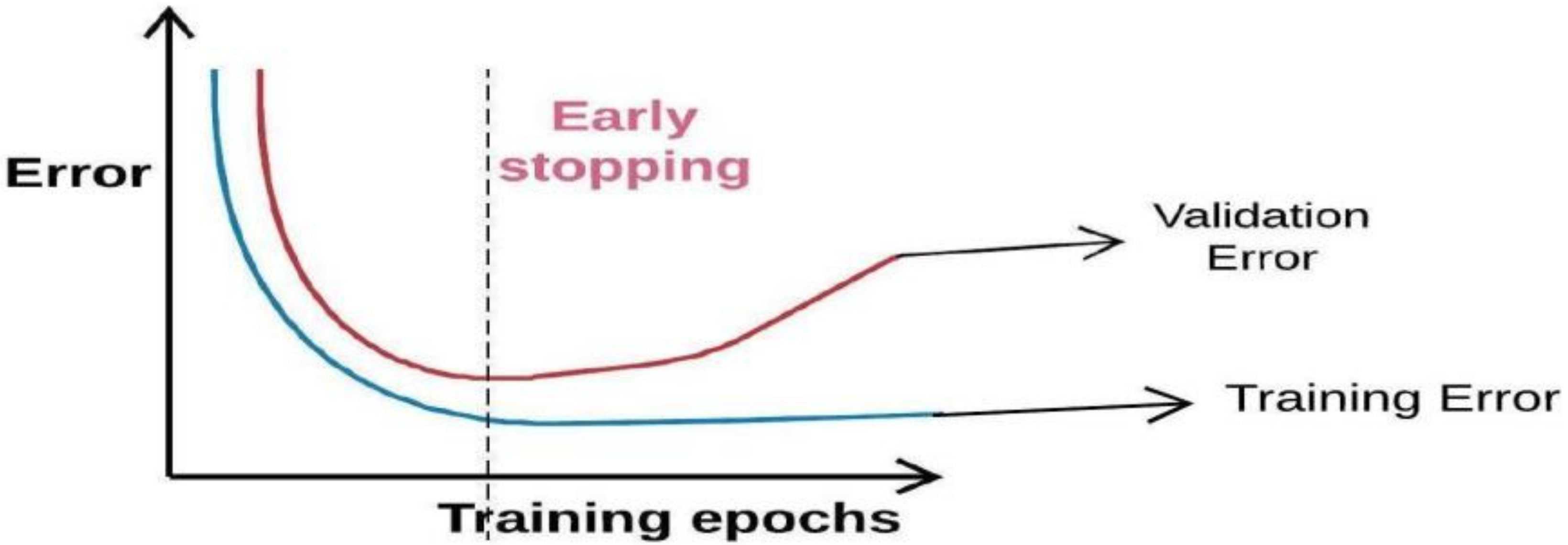
- The number of epochs specifies the number iterations the training dataset is shown to the network while training,
- Iteration, epochs and cycles are terms used interchangeable in textbooks.
- There is not set number of epochs to solve all problems at once.
- **Keep increasing the number of epochs** until the validation accuracy begins to decrease. (32, 64, 128, 256, 512,1024 etc(

London South Bank University
EST 1892

Hyper-parameters: Batch size

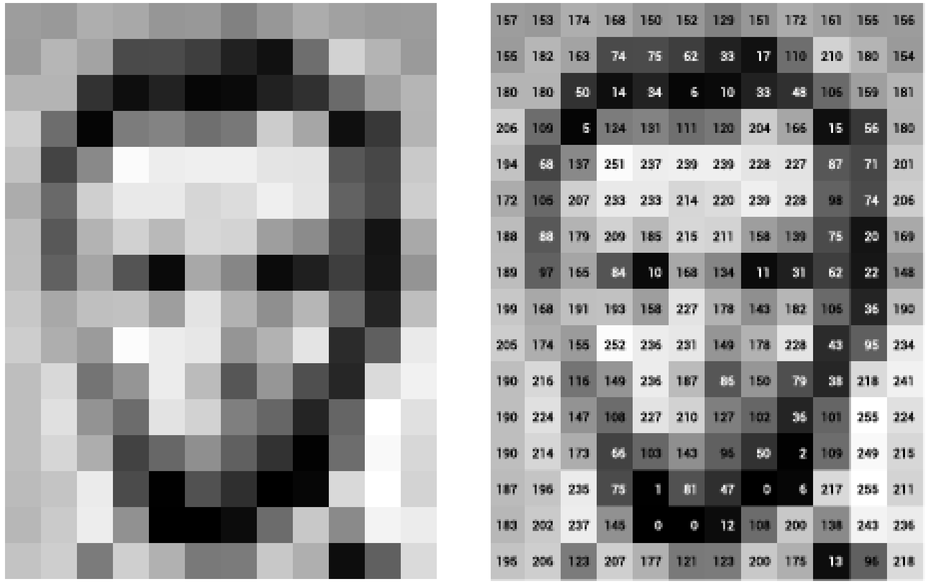
- A batch size is the number of sub samples of the dataset given the deep neural network
- after each batch size the model parameter update happens.
- Generally, 32 is considered a good starting batch size.
- If the results are not improving consider trying 64, 128, 256, and so on.

Early Stopping

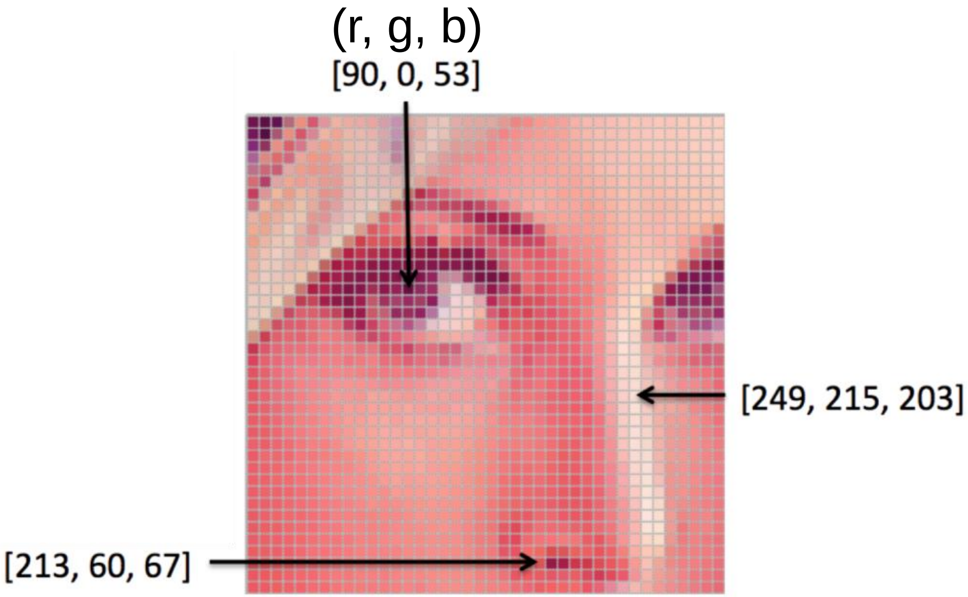


A digital image

- A binary representation of visual data
- Contains a series of pixels arranged in a grid like fashion
- Each pixel value denotes how bright and what color each picture should be.
- In neural network every pixel is a potential input value (ex. 28x28=784px input)



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218



Data augmentation

- A technique to increase the size of the training sets.
- Lack of enough training datasets to solve real-life complex problems(e.g., Medical dataset)
- More training data, more skillful model.
- Reduces costs related to data collection.
- There are several techniques for data augmentation:

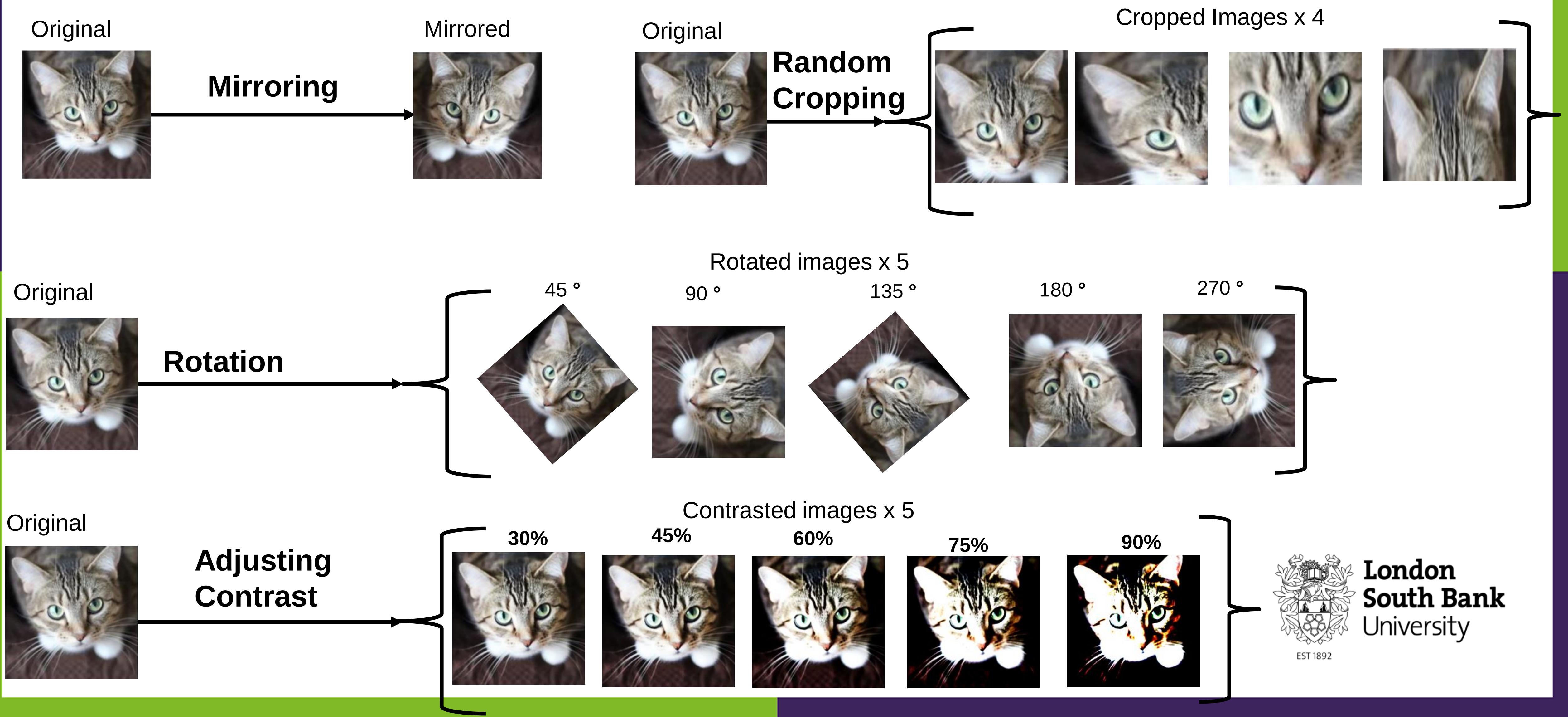
London South Bank University
EST 1892

Data Augmentation Common Techniques

- Cropping (Randomly cropped regions
- Rotation(45 °, 90 ° , 180 °, 270 °)
- Flipping.
- Contrast adjustment (25%, 50%, 75%, 90% etc.)
- Scaling etc.

London South Bank University
EST 1892

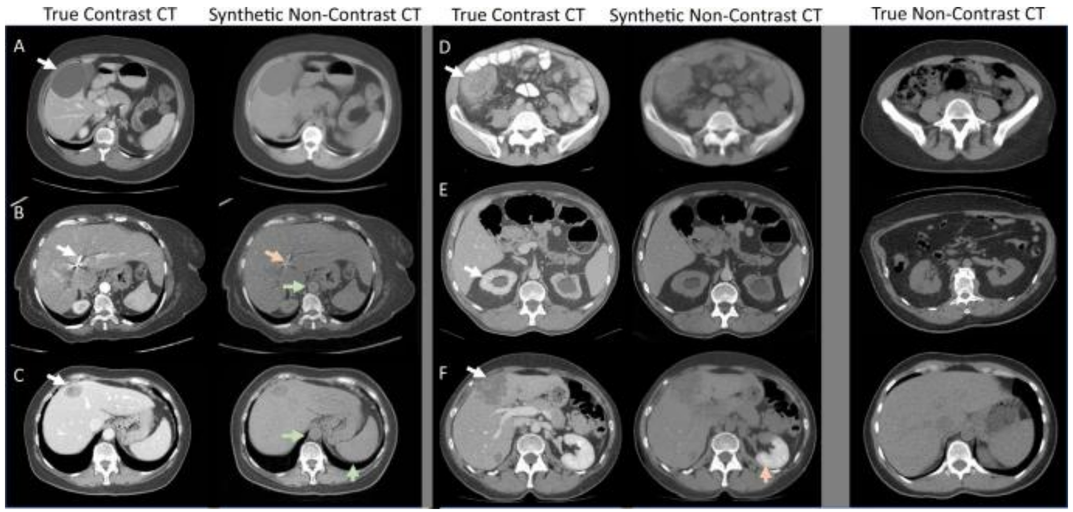
Common Data Augmentation Techniques



London South Bank University
EST 1892

Other Data augmentation methods

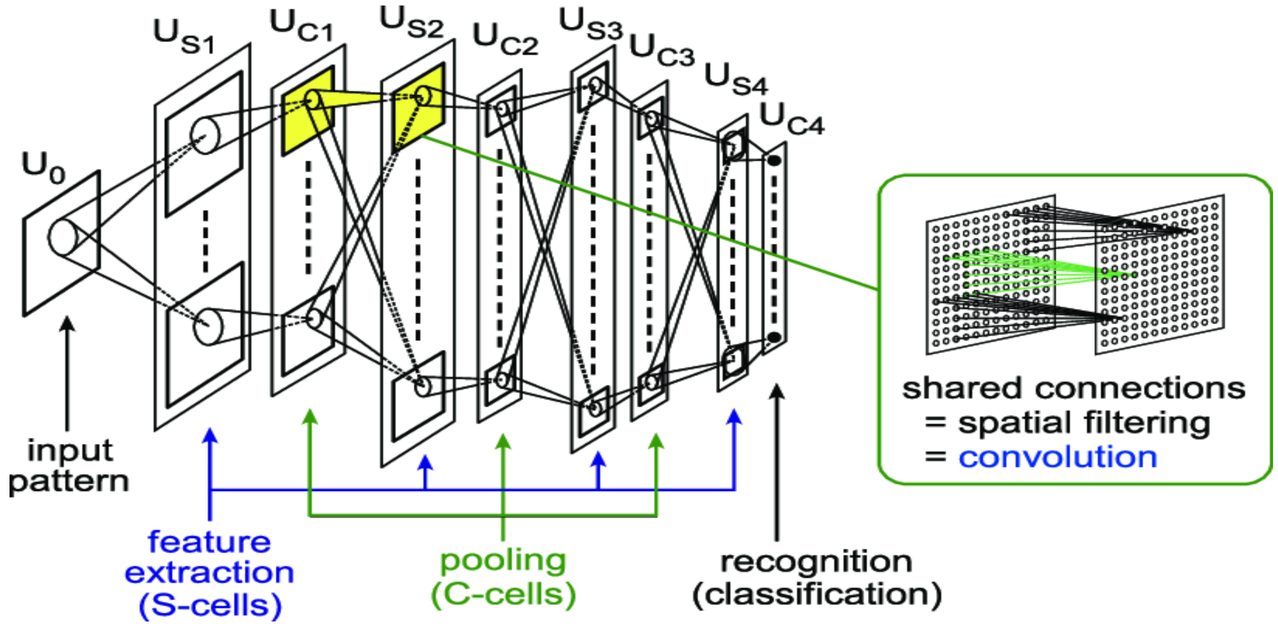
- Shearing
- Grey scaling
- Color shifting
- Implementing distortion and filters.
- Synthetic data. (GAN)



A computerized tomography scan high-resolution image generated by GAN

CNN Definition

- A class of neural network evolved from neocognitron (Fukushima 1979)
- A Convolution Neural Network, also known as CNN or ConvNet.
- Learns highly abstracted features of data objects especially spatial data.
- Useful for processing grid-like topology data:
  - Medical imaging
  - Video captioning
  - Audio processing
  - Synthetic data generation (GAN)
- Efficient feature identification

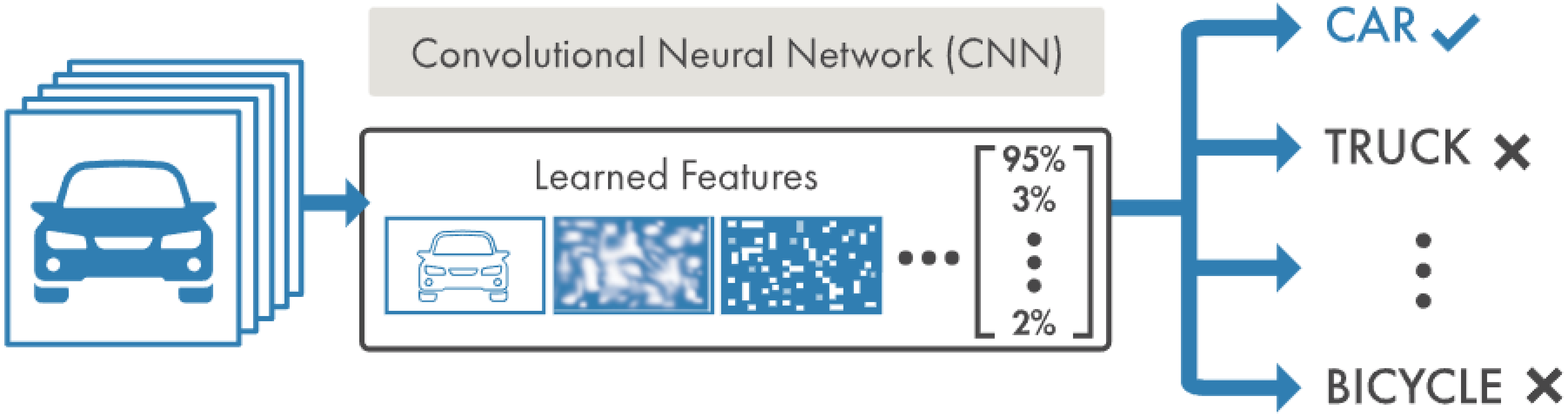


Advantages of CNN:

- Weight sharing (less overfitting, reduced the number of trainable parameters.)
- Classification layers and feature extraction layers learn together.
- Implementation of a large neural network on images is complicated compared to CNN
- Useful for tasks like: Image classification, object detection, face detection, speech recognition, vehicle recognition and facial expression recognition, text recognition etc.

London South Bank University
EST 1892

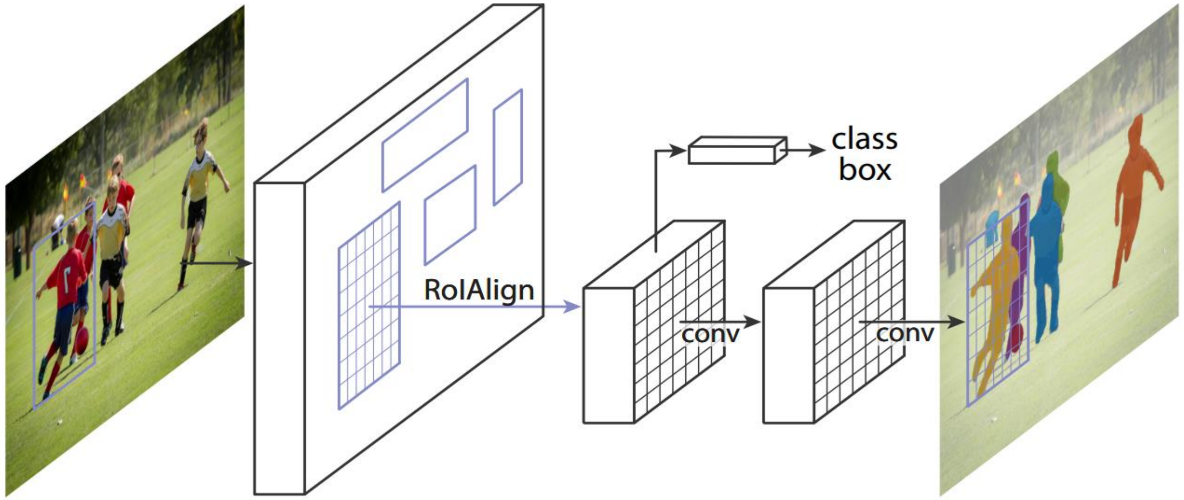
Domains of application: Object Detection

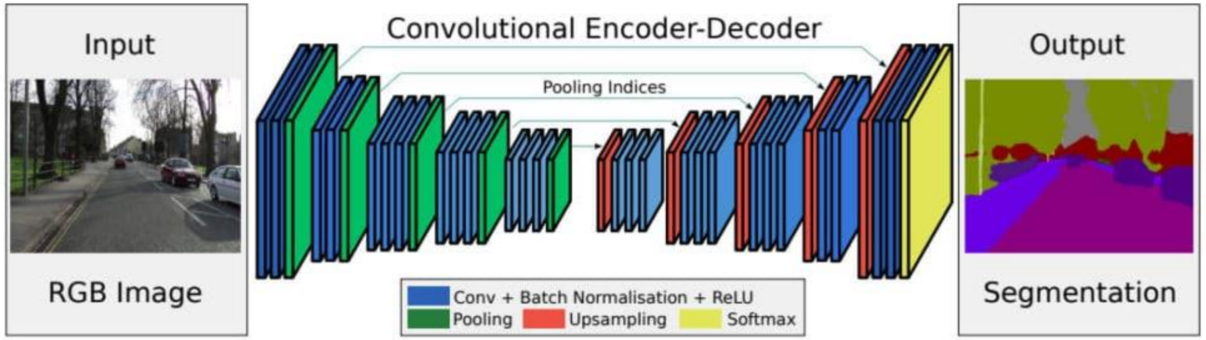


Source: https://uk.mathworks.com/discovery/deep-learning/_jcr_content/mainParsys/band_2123350969_copy_1983242569/mainParsys/columns_1635259577/1/image_792810770_copy.adapt.1200.medium.svg/1662737450098.svg

London South Bank University
EST 1892

Domain of application: Object detection (R-CNN)





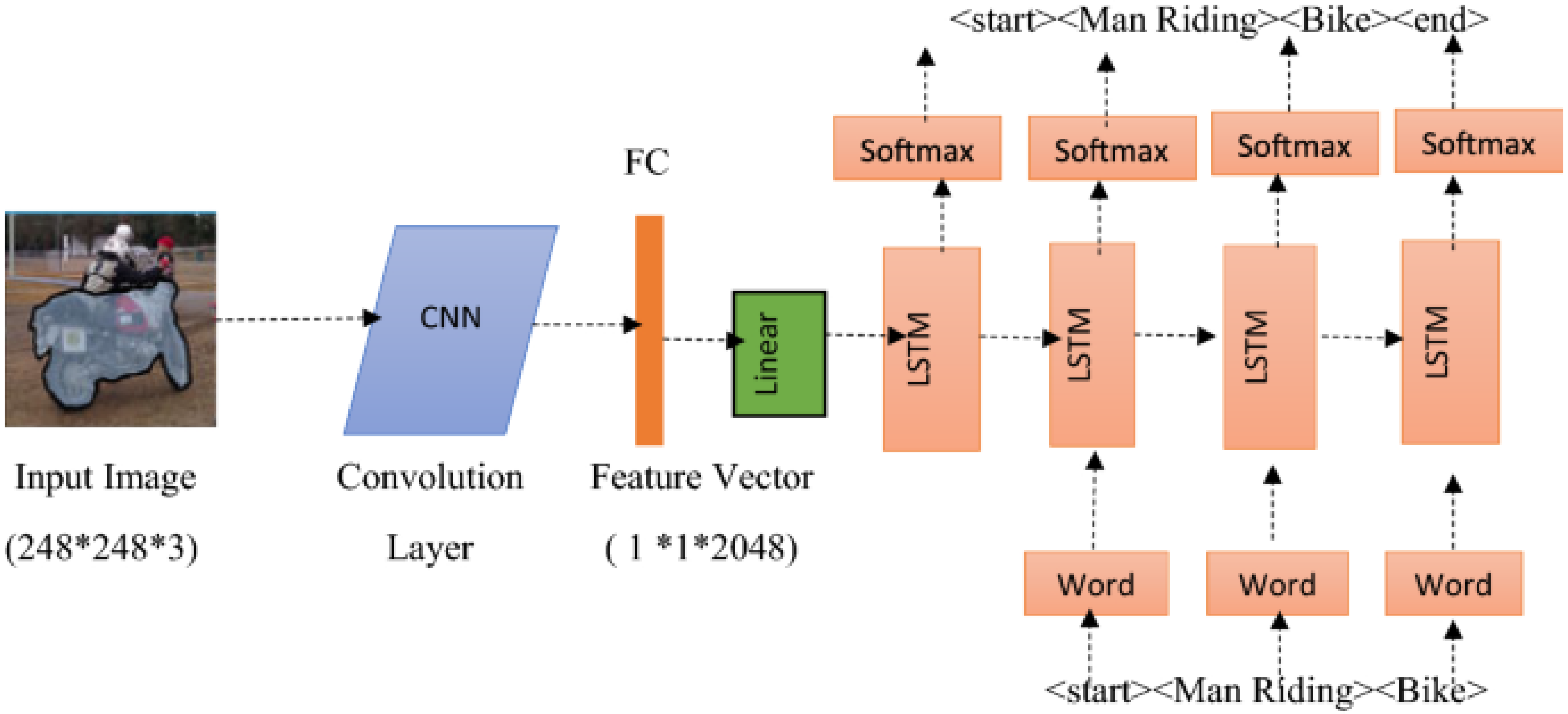
Source: https://arxiv.org/abs/1703.06870

London South Bank University
EST 1892

Domain of Application: Image captioning

Label – [<start>, man, riding bike.]

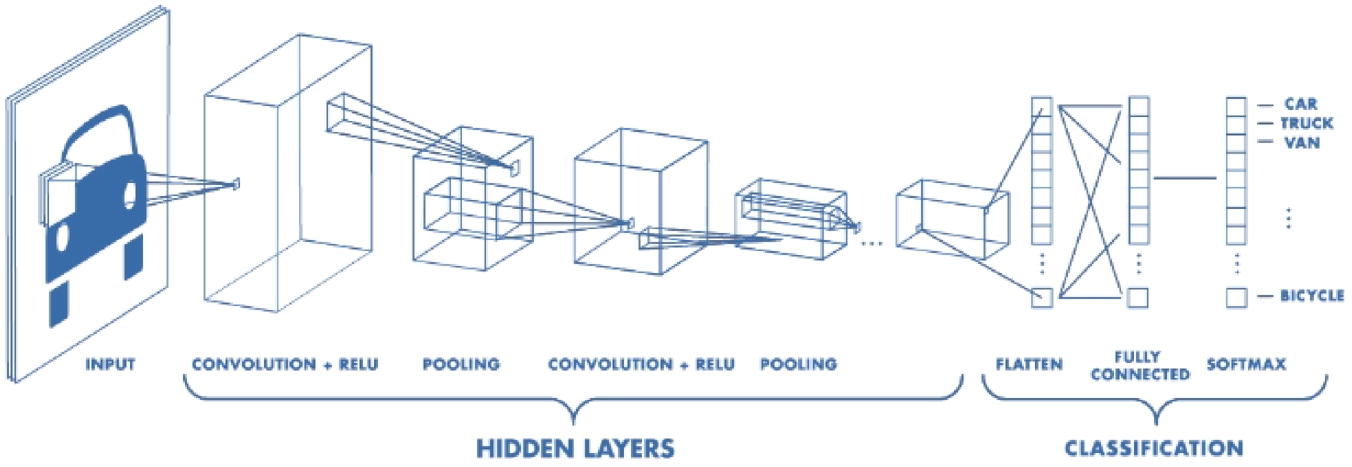
Target – [man, riding bike, <end>]



London South Bank University
EST 1892

Convolutional Neural Network Architecture

- A CNN typically has three layers:
  - A convolutional layer
  - Pooling layer
  - Fully connected layer (FC).



Source: https://uk.mathworks.com/discovery/convolutional-neural-network-matlab.html

London South Bank University
EST 1892

Convolutional Layers

- The **convolution layer** is a core building block of CNN.
- Made up of a set learnable parameters referred to as kernels (also called filters).
- Each **input image** (e.g. 28x28) gets **convolved with a kernel** to generate an output **feature map**. (input_image $\otimes$ kernel)
- There are **multiple filters** in a convolutional layer **to extract different features**.
- The goal of a convolution is to **detect features** like **edges**, **lines**, blobs of color or any other visual elements.

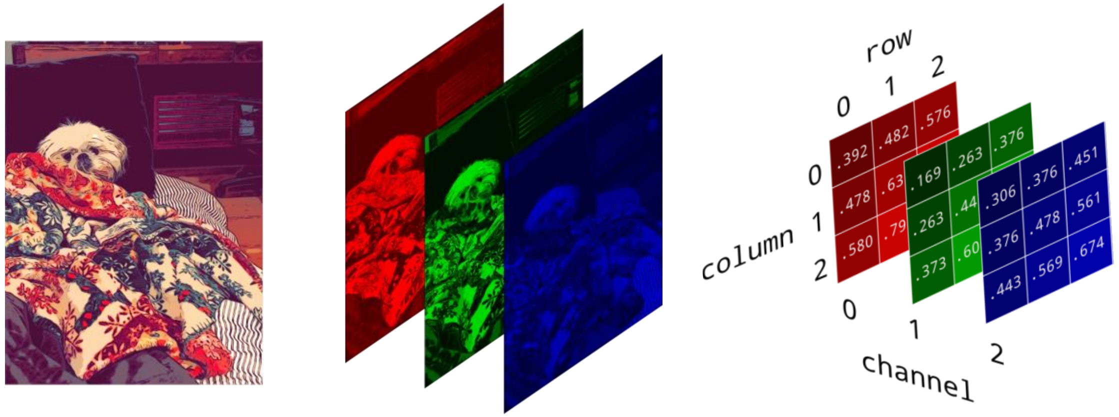
1	**0**
-1	**2**

Example of 2x2 kernel

London South Bank University
EST 1892

A Convolution Operation

- Unlike traditional neural networks, CNN input is multi-channel image(e.g. RGB 3 channels.
- For grey-scale images like MNIST it’s a single channel.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	13	18	18	18	134	150	255	169	37	0	0	0	0	0	0
7	0	0	0	0	0	5	92	36	36	140	154	154	154	224	253	253	253	253	253	253	253	170	0	0	0	0	0	0
8	0	0	0	0	0	36	253	253	253	253	253	253	253	253	253	253	253	253	253	253	253	145	0	0	0	0	0	0
9	0	0	0	0	0	9	144	202	253	223	182	182	182	182	182	138	65	161	253	253	235	38	0	0	0	0	0	0
10	0	0	0	0	0	0	0	13	47	27	0	0	0	0	0	0	0	130	253	253	147	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	235	253	253	70	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166	253	253	230	36	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	54	243	253	251	131	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	176	253	253	224	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	78	253	253	253	141	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	168	253	253	208	25	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	95	244	253	253	124	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	177	253	253	203	14	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	2	107	252	253	253	81	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	38	253	253	253	253	92	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	24	163	253	253	253	236	60	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	125	253	253	253	233	63	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	218	253	253	253	137	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	243	253	253	101	8	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	187	253	234	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Convolution operation

Let's consider an input image for 4x4 (grey-scale) and a kernel size of 2x2

1	0	-2	1
-1	0	1	2
0	2	1	0
1	0	0	1

Grey-scale image 4x4

0	1
-1	2

A Kernel size 2x2

Convolution operation

Step one

$$\begin{bmatrix}1&0&-2&1\\-1&0&1&2\\0&2&1&0\\1&0&0&1\end{bmatrix} \otimes \begin{bmatrix}0&1\\-1&2\end{bmatrix} \rightarrow \begin{bmatrix}1&&\\&&\\&&\end{bmatrix}$$

Step two

$$\begin{bmatrix}1&0&-2&1\\-1&0&1&2\\0&2&1&0\\1&0&0&1\end{bmatrix} \otimes \begin{bmatrix}0&1\\-1&2\end{bmatrix} \rightarrow \begin{bmatrix}1&0&\\&&\\&&\end{bmatrix}$$

Step three

$$\begin{bmatrix}1&0&-2&1\\-1&0&1&2\\0&2&1&0\\1&0&0&1\end{bmatrix} \otimes \begin{bmatrix}0&1\\-1&2\end{bmatrix} \rightarrow \begin{bmatrix}1&0&4\\&&\\&&\end{bmatrix}$$

Step four

$$\begin{bmatrix}1&0&-2&1\\-1&0&1&2\\0&2&1&0\\1&0&0&1\end{bmatrix} \otimes \begin{bmatrix}0&1\\-1&2\end{bmatrix} \rightarrow \begin{bmatrix}1&0&4\\4&&\\&&\end{bmatrix}$$

Step five

$$\begin{bmatrix}1&0&-2&1\\-1&0&1&2\\0&2&1&0\\1&0&0&1\end{bmatrix} \otimes \begin{bmatrix}0&1\\-1&2\end{bmatrix} \rightarrow \begin{bmatrix}1&0&4\\4&1&\\&&\end{bmatrix}$$

1	0	4
4	1	1
1	1	2

Final Feature Map

London South Bank University
EST 1892

Convolution Operation

- In this example, the convolution operation was conducted on image with a **stride** but **no padding**.
- **A stride** is the number of steps taken along the horizontal and vertical positions in the input image by the kernel.
- **Padding** gives the border size information of the input image otherwise it gets washed away.
- **Padding** is used to increase the input image size (The output features also get increased)

London South Bank University
EST 1892

Feature map

- The feature map is the outcome of the convolution operation
- The formula to find out the feature map size after this operation is:

$$\mathrm{FeatureMap}Height = \left\lfloor \frac{InputImageHeight - FilterSize + Padding}{StrideSize} + 1 \right\rfloor$$

$$\mathrm{FeatureMap}Width = \left\lfloor \frac{InputImageWidth - FilterSize + Padding}{StrideSize} + 1 \right\rfloor$$

Convolution with Zero-Padding & 3 Stride

Step one

0	0	0	0	0	0
0	1	0	-2	1	0
0	-1	0	1	2	0
0	0	2	1	0	0
0	1	0	0	1	0
0	0	0	0	0	0

⊗

1	0	1
0	-1	2
-2	1	0

→

-2	

Step two

0	0	0	0	0	0
0	1	0	-2	1	0
0	-1	0	1	2	0
0	0	2	1	0	0
0	1	0	0	1	0
0	0	0	0	0	0

⊗

1	0	1
0	-1	2
-2	1	0

→

-2	-1

Step three

0	0	0	0	0	0
0	1	0	-2	1	0
0	-1	0	1	2	0
0	0	2	1	0	0
0	1	0	0	1	0
0	0	0	0	0	0

⊗

1	0	1
0	-1	2
-2	1	0

→

-2	-1
1	

Step four

0	0	0	0	0	0
0	1	0	-2	1	0
0	-1	0	1	2	0
0	0	2	1	0	0
0	1	0	0	1	0
0	0	0	0	0	0

⊗

1	0	1
0	-1	2
-2	1	0

→

-2	-1
1	0

Convolution layers advantages

- Convolution has **small number of weights** between two layers.
- **The amount of memory** used to store them is **small**
- **Weight sharing** between neurons of adjacent layers. (reduces the training time.)

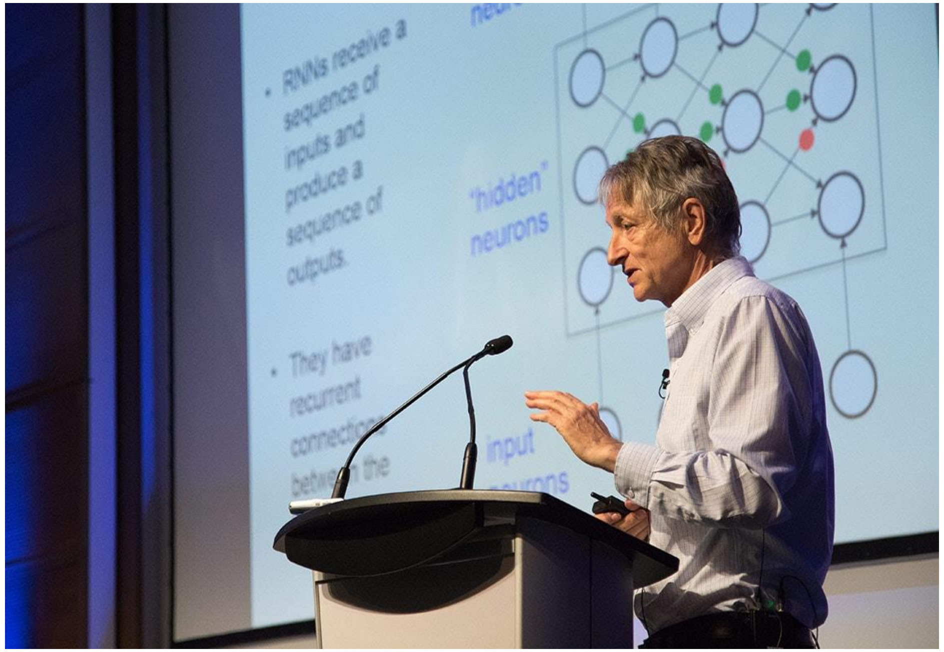
Pooling Layer

- **Sub-samples feature maps** (matrices produced after the convolution operation)
- **Pooling takes in large size** feature maps and **shrinks them to a lower size.**
- The pooling operation is done by specifying the **pooled region size** and **the stride of the operation.**
- Sometimes it can decrease the performance of the CNN.
- There are several **type** of pooling:
  - Max pooling
  - Min pooling
  - Average pooling
  - Gated pooling
  - Tree pooling

London South Bank University
EST 1892

Pooling and Hinton

" The pooling operation used in convolutional neural networks is a big ***mistake*** *and the fact that it works so well is a* ***disaster****."* – Geoffrey Hinton



Source: https://www.utoronto.ca/sites/default/files/2017-09-13Geoff-Hinton-%28web-lead%29.jpg

London South Bank University
EST 1892

Max pooling example

In this example, the size of the pooling region is 2x2 and the stride is 1.

1	**0**	-2	1
-1	**0**	1	2
0	2	1	0
1	0	0	1

Max(1,0,-1,0) →

1		

1	**0**	**-2**	1
-1	**0**	**1**	2
0	2	1	0
1	0	0	1

Max(0,-2,0,1) →

1	1	

1	0	**-2**	**1**
-1	0	**1**	**2**
0	2	1	0
1	0	0	1

Max(-2,1,1,2) →

1	1	2

1	0	-2	1
-1	**0**	1	2
0	**2**	1	0
1	0	0	1

Max(-1,0,0,2) →

1	1	2
2		

1	0	-2	1
-1	0	1	2
0	2	1	0
1	0	0	1

End Results →

1	1	2
2	2	2
2	2	1

London South Bank University
EST 1892

Feature map after pooling

- Calculating the feature map after any pooling operation is performed using the following formulas

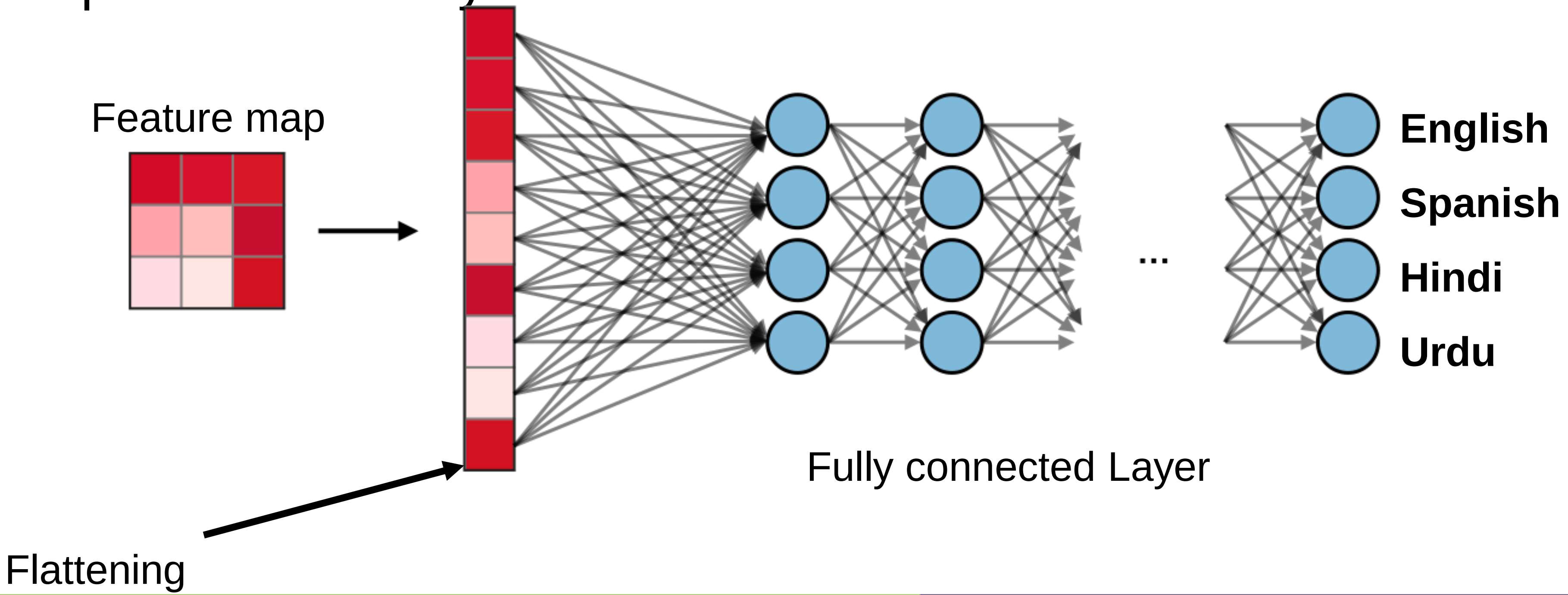
$$\mathrm{FeatureMap}Height = \left\lfloor \frac{InputFeatureMapHeight - PoolingRegionSize}{StrideSize} + 1 \right\rfloor$$

$$\mathrm{FeatureMap}Width = \left\lfloor \frac{InputFeatureMapWidth - PoolingRegionSize}{StrideSize} + 1 \right\rfloor$$

London South Bank University
EST 1892

Fully Connected (FC) Layer

- The last component or layer in every CNN architecture consists of a set of fully-connected layers.
- The last layer of the fully connected layers is used as an output layer (classifier) of the CNN Network.
- **The convolution output must be flattened** before it can be fed as an input to the FC layers.



CNN Summary

- CNN useful for spatial data
- Convolution layer extracts features using set of kernels
- Pooling shrinks the size of feature maps
- Fully connected layer is used for classification/ regression or labelling.

Questions & Answers

London South Bank University
EST 1892