Imperial College London

Tutorial for Computer Vision Technical Project

03/02/2025

Chengliang Dai

Key Tasks in Medical Image Analysis

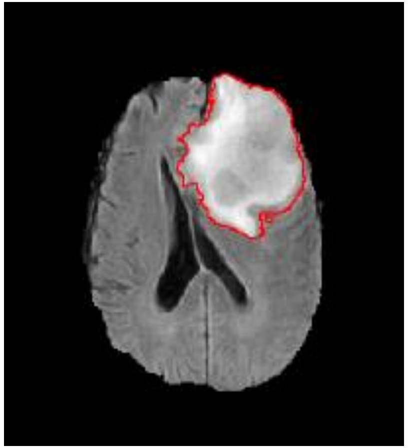
Localisation

Output: bounding boxes

Classification

Output: tumour/no tumour

Semantic Segmentation



Output: segmentation map

Imperial College London

Other Tasks in Computer Vision

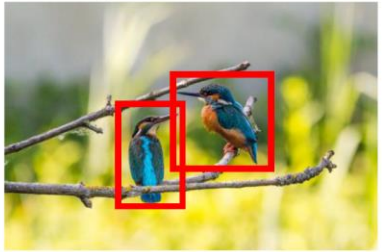
Image Classification

Output: Category (e.g., “bird”)

Object Detection

Output: Coordinates (e.g., centroid)

Object Localisation

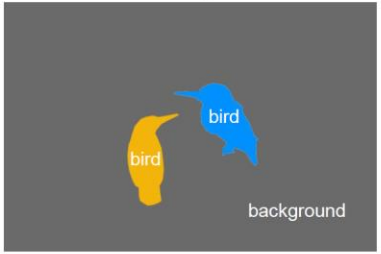


Output: Coordinates (e.g., bounding box)

Object Recognition

Output: Category (e.g., “kingfisher”)

Semantic Segmentation

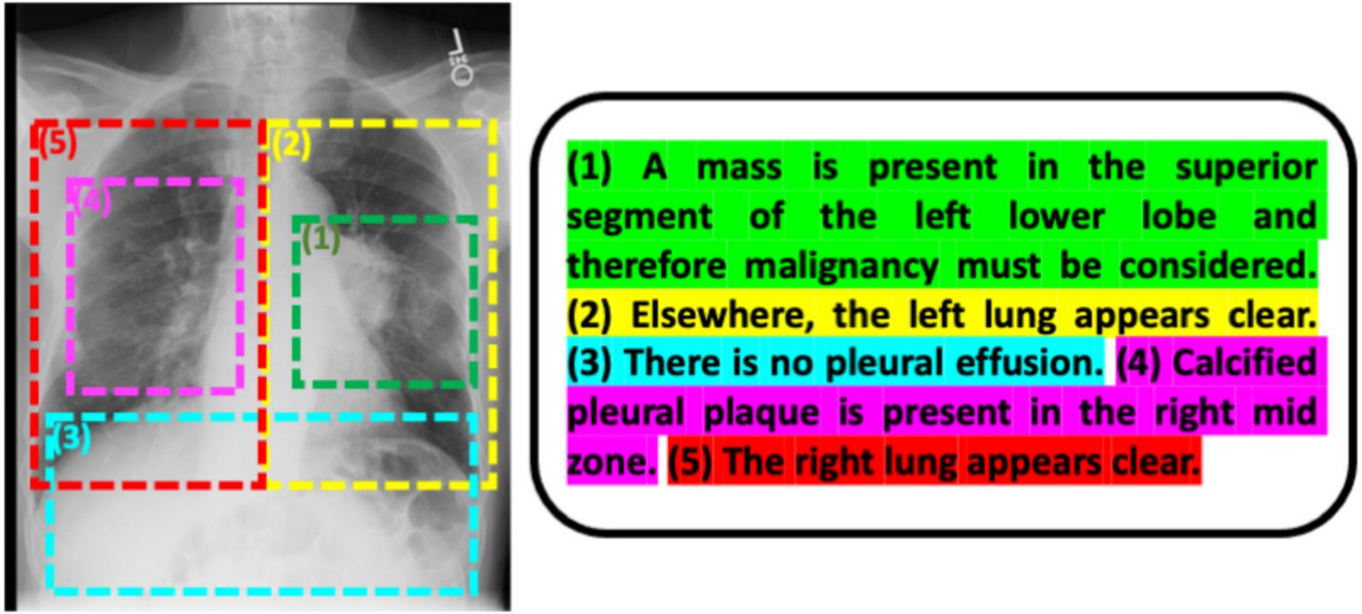


Output: Labelmap

Image Captioning

Output: Text
(e.g., “two birds sitting on a branch”)

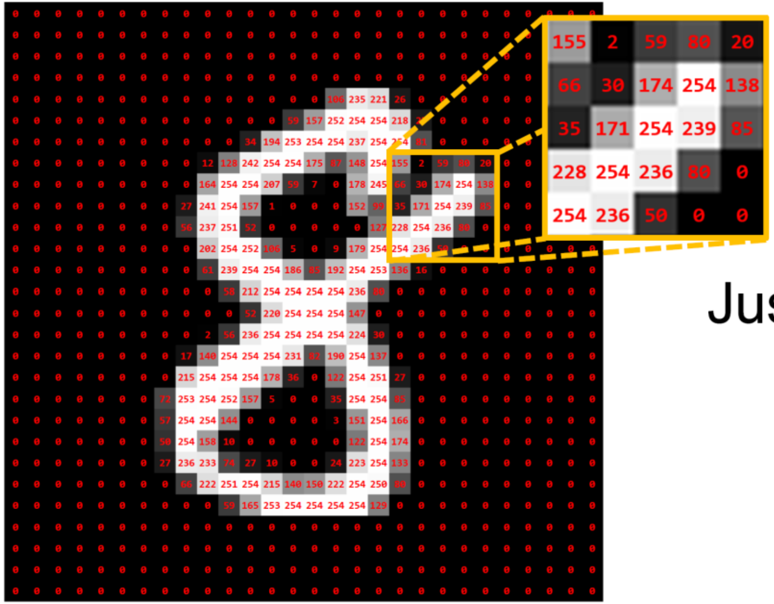
Imperial College
London



Liao, R., Moyer, D., Cha, M., Quigley, K., Berkowitz, S., Horng, S., Golland, P. and Wells, W.M., 2021. Multimodal representation learning via maximization of local mutual information. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part II 24* (pp. 273-283). Springer International Publishing.

Example: Image Classification Algorithms

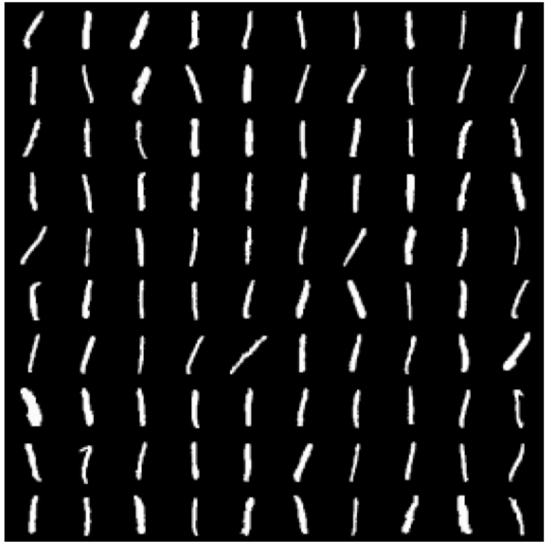
→ Image Classification Algorithms → With/without tumour



Just an array of numbers

Imperial College London

Classification Example : Digit Recognition

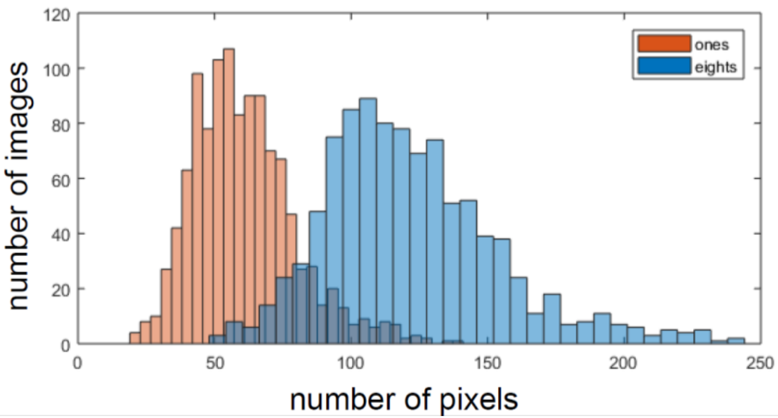


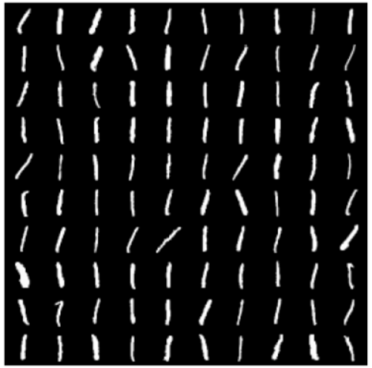
ones

eights

Imperial College London

Classification Example : Digit Recognition



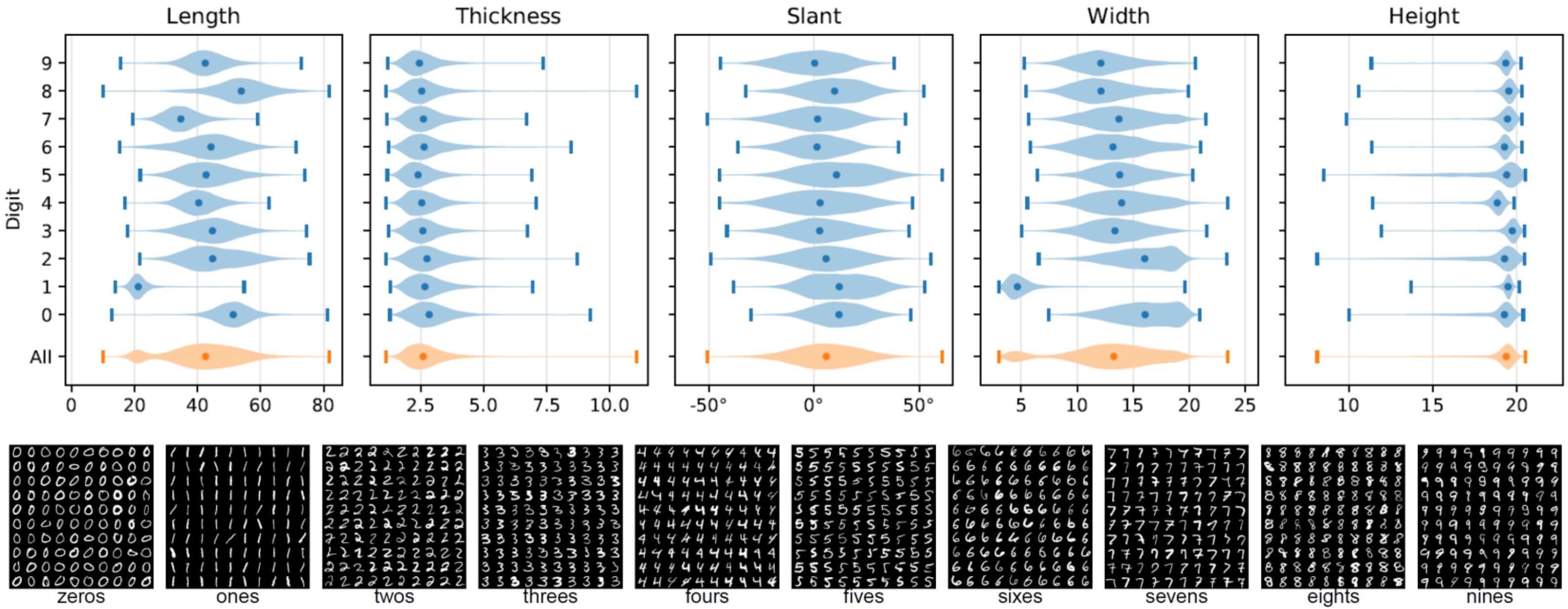


ones

eights

Imperial College London

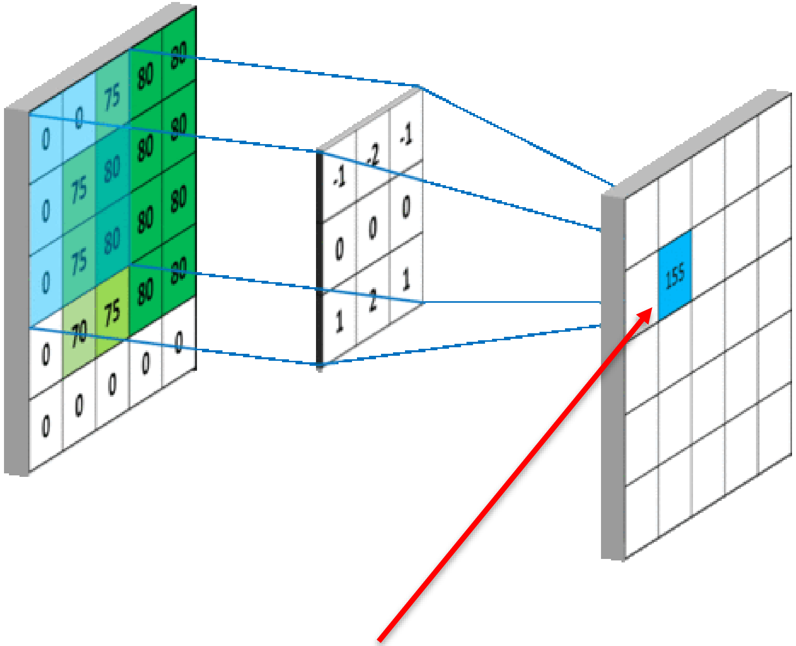
Classification Example : Digit Recognition



Source: Castro et al. 2018. https://arxiv.org/abs/1809.10780

Imperial College London

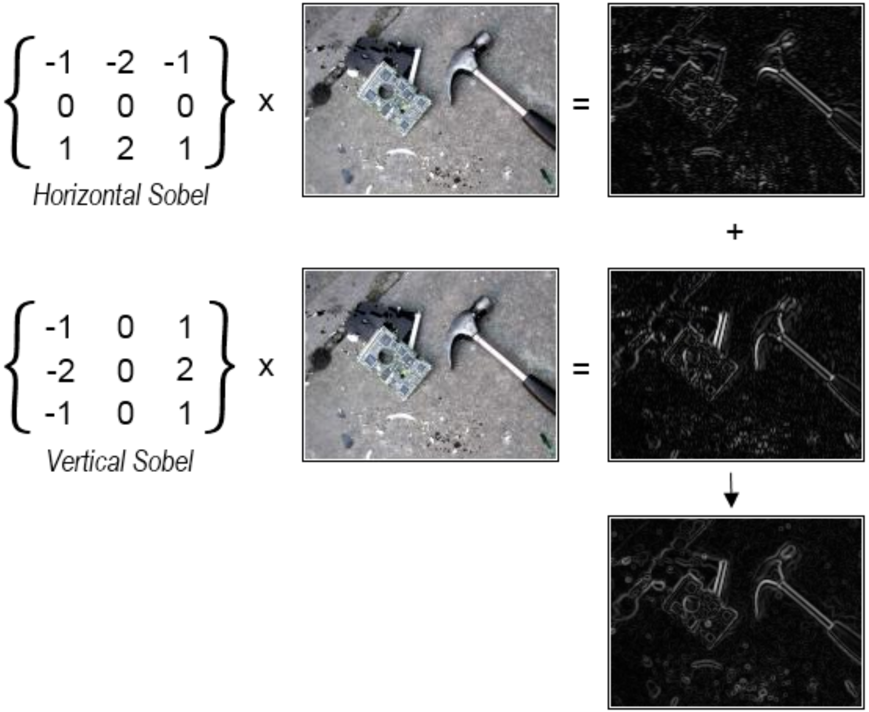
Convolutional Operations in Image Processing



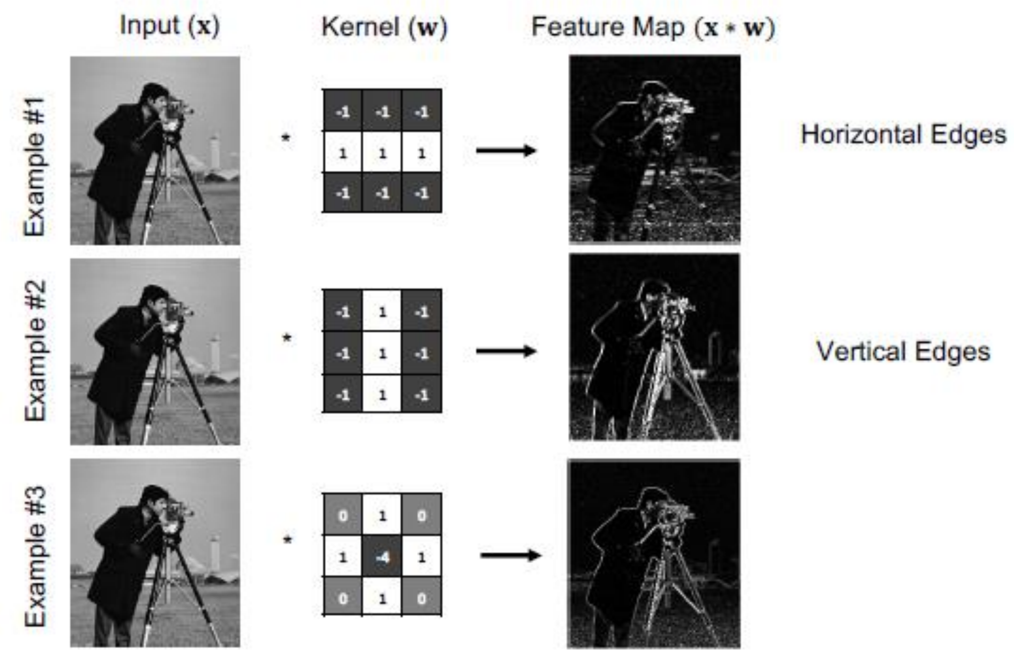
$$-1 * 0 + (-2) * 0 + (-1) * 75 + 0 * 0 + 0 * 75 + 0 * 80 + 1 * 0 + 2 * 75 + 1 * 80 = 155$$

Imperial College London

Convolutional Operations with a Sobel Filter



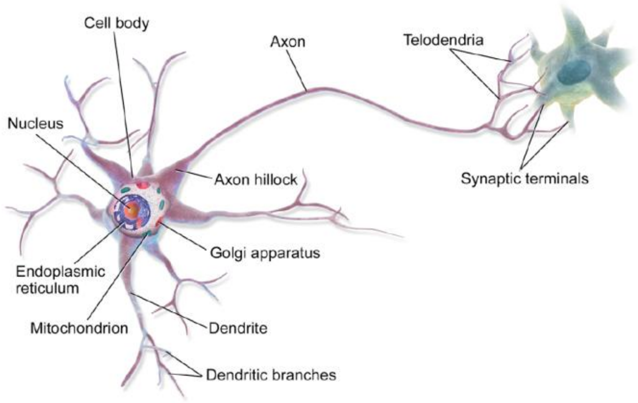
Applying Convolutional Operations



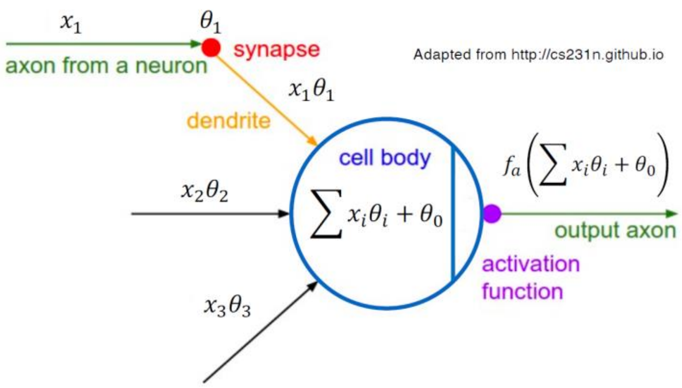
Imperial College London

Neural Network

- First machine learning methods were inspired by how the brain works:



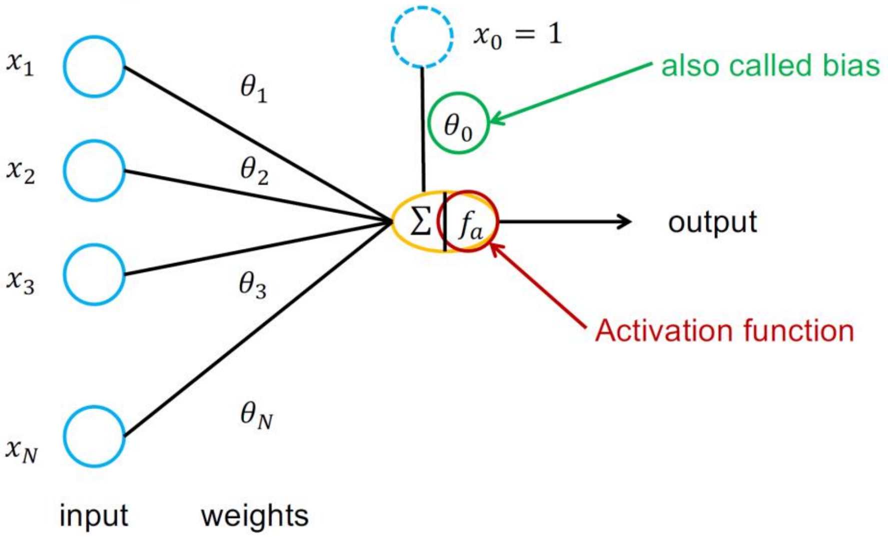
Biological neuron



Artificial neuron

Single-layer Perceptron

- A single-layer perceptron looks as follows:



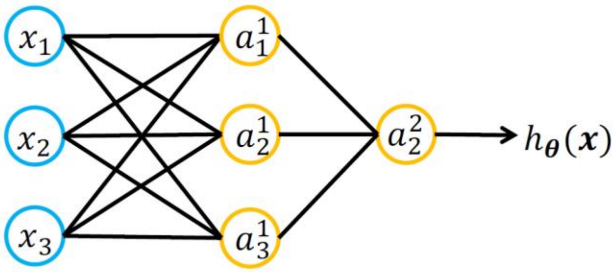
Imperial College London

Single-layer Perceptron

- Training a perceptron means to iteratively updating the weights associated with each of its inputs
- This allows to progressively approximate the underlying relationship in the given training dataset
- Once properly trained, it can be used to classify entirely new samples

Imperial College London

Neural Network



$\mathbf{a}_j^i$ activation of neuron j in layer i

$\boldsymbol{\Theta}^i$ matrix of weights controlling mapping from layer $i-1$ to i

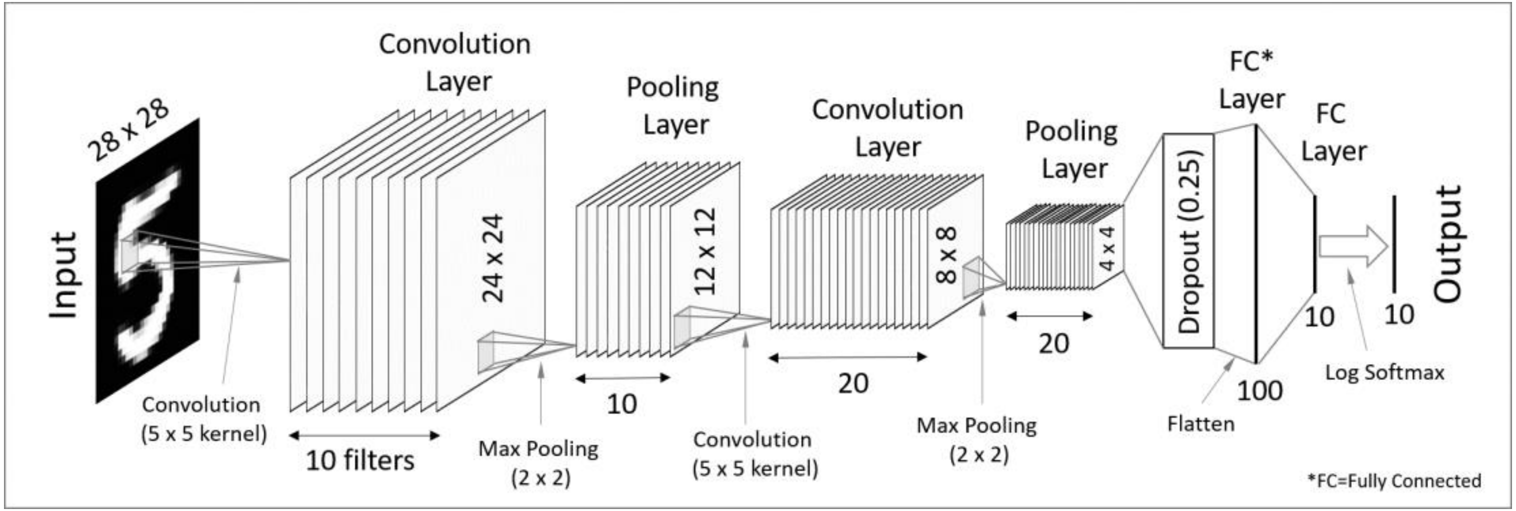
$$a_1^1 = f_a(\theta_{0,0}^1 + \theta_{1,0}^1 x_1 + \theta_{2,0}^1 x_2 + \theta_{3,0}^1 x_3)$$

$$a_2^2 = f_a(\theta_{0,1}^1 + \theta_{1,1}^1 x_1 + \theta_{2,1}^1 x_2 + \theta_{3,1}^1 x_3)$$

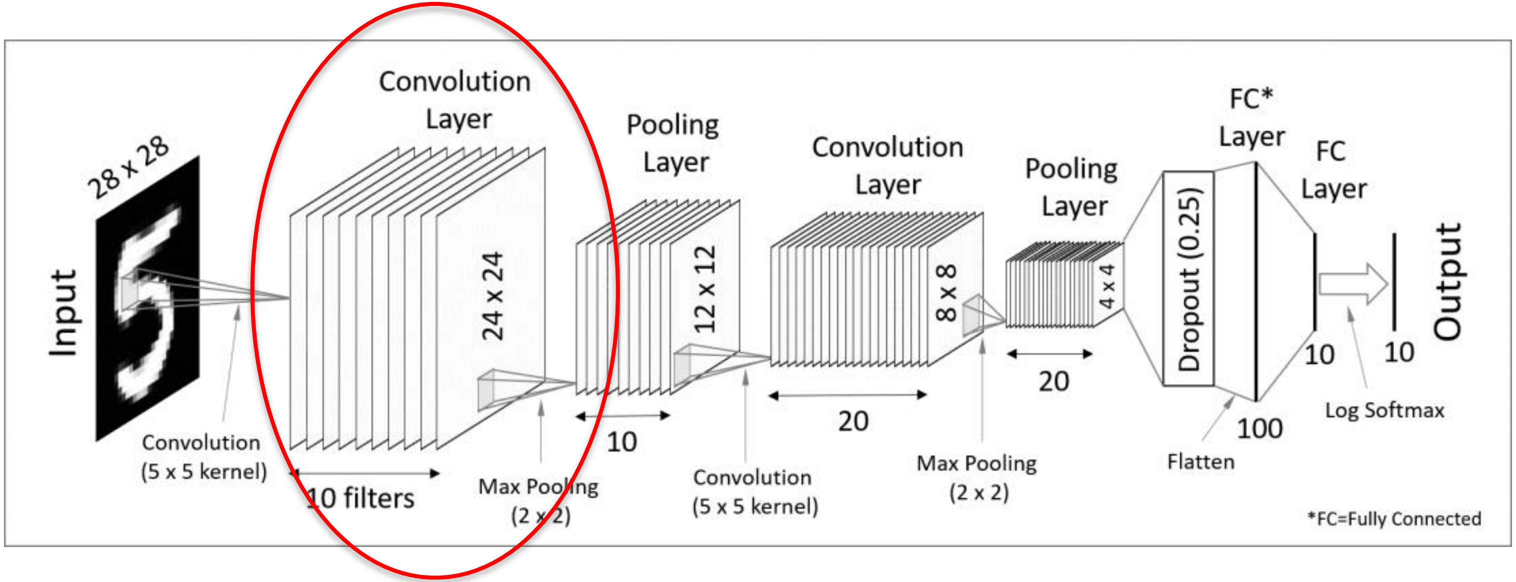
$$a_3^1 = f_a(\theta_{0,2}^1 + \theta_{1,2}^1 x_1 + \theta_{2,2}^1 x_2 + \theta_{3,2}^1 x_3)$$

$$h_{\boldsymbol{\theta}} = a_1^2 = f_a(\theta_{0,0}^2 + \theta_{1,0}^2 a_1 + \theta_{2,0}^2 a_2 + \theta_{3,0}^3 a_3)$$

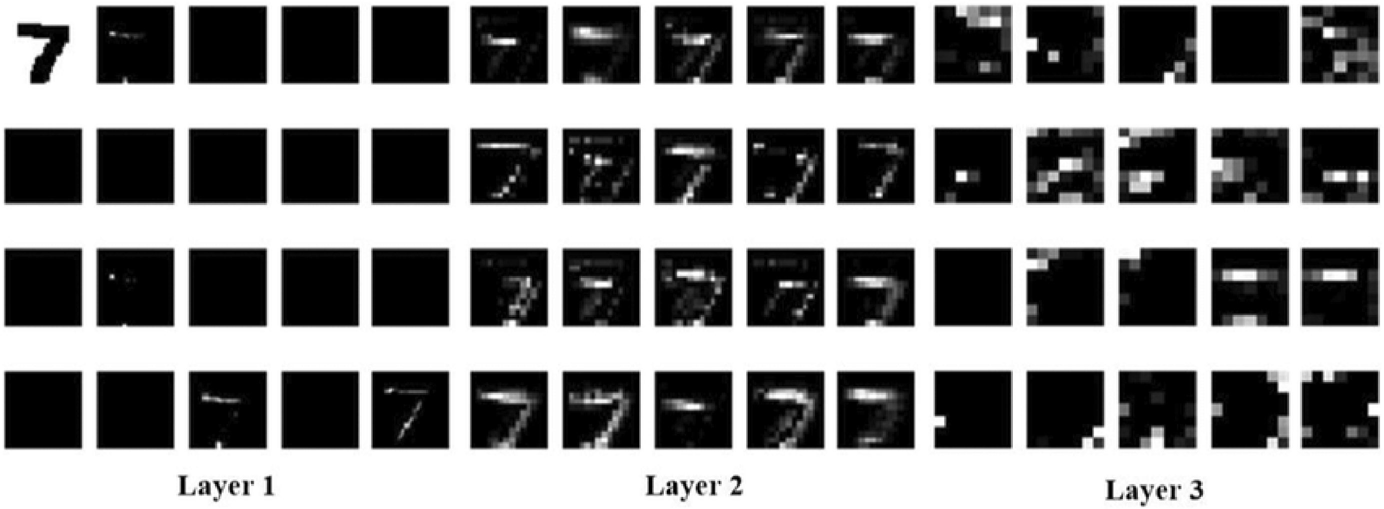
Convolutional Neural Network (CNN) Model



Convolution Layer

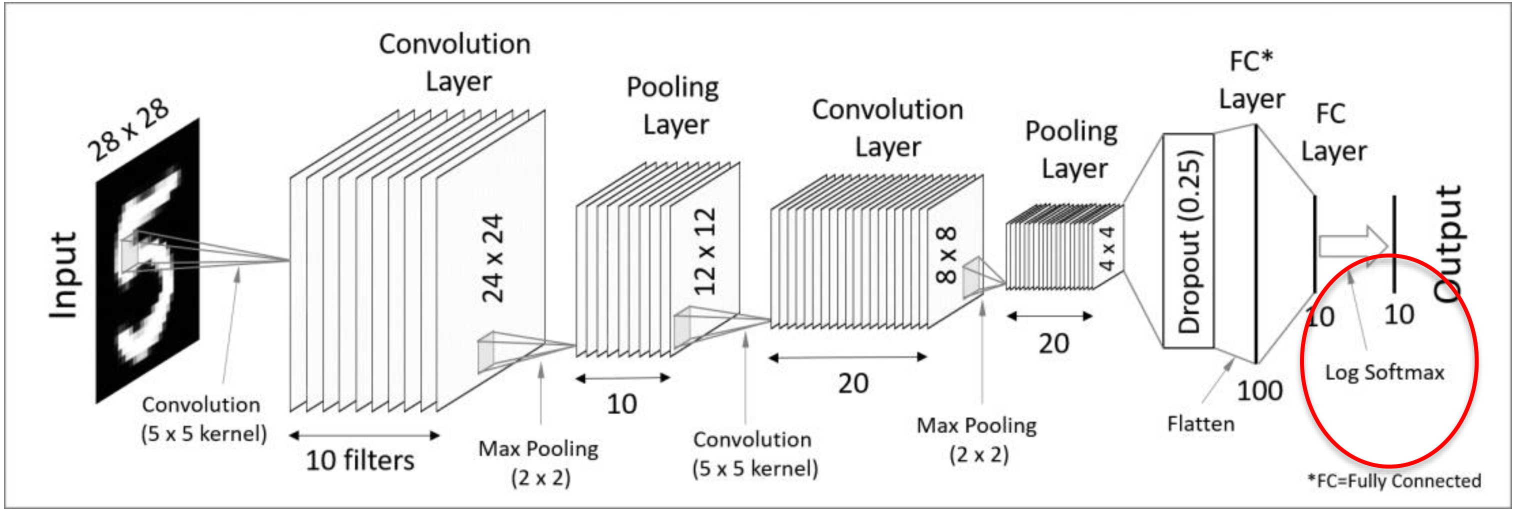


Feature maps



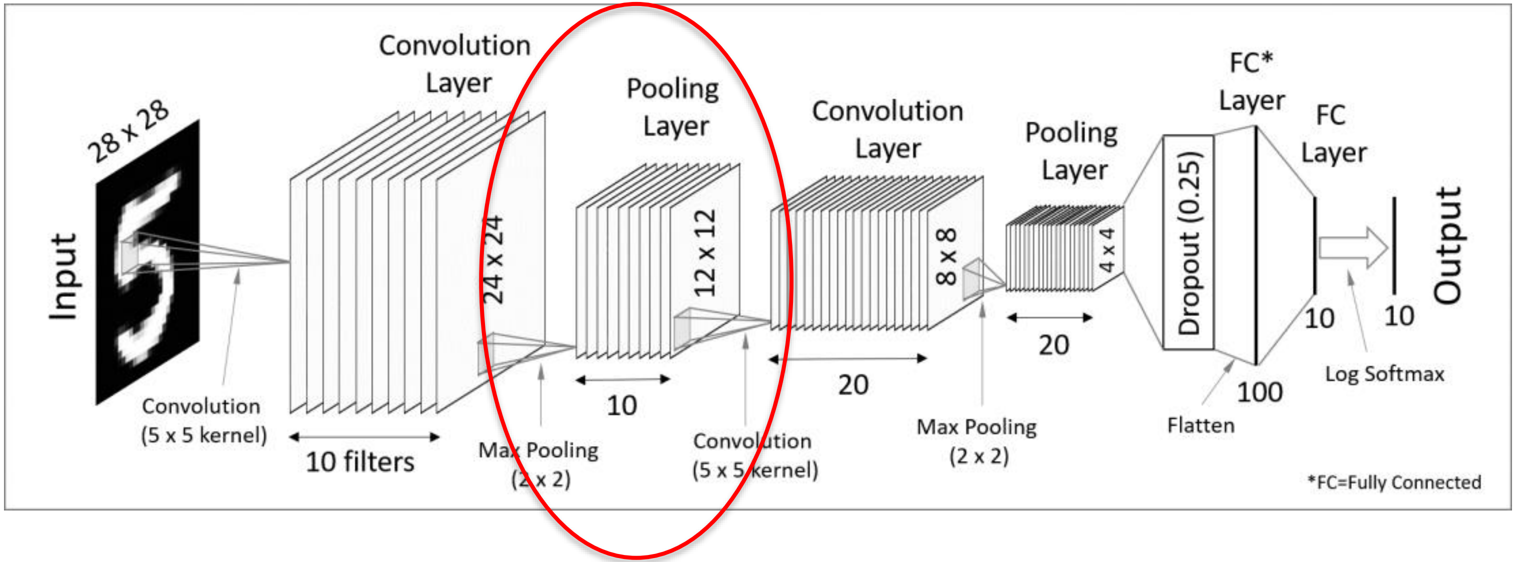
Liu, J., Shao, H., Jiang, Y. and Deng, X., 2022. CNN-based hidden-layer topological structure design and optimization methods for image classification. Neural Processing Letters, 54(4), pp.2831-2842.

Activation Functions



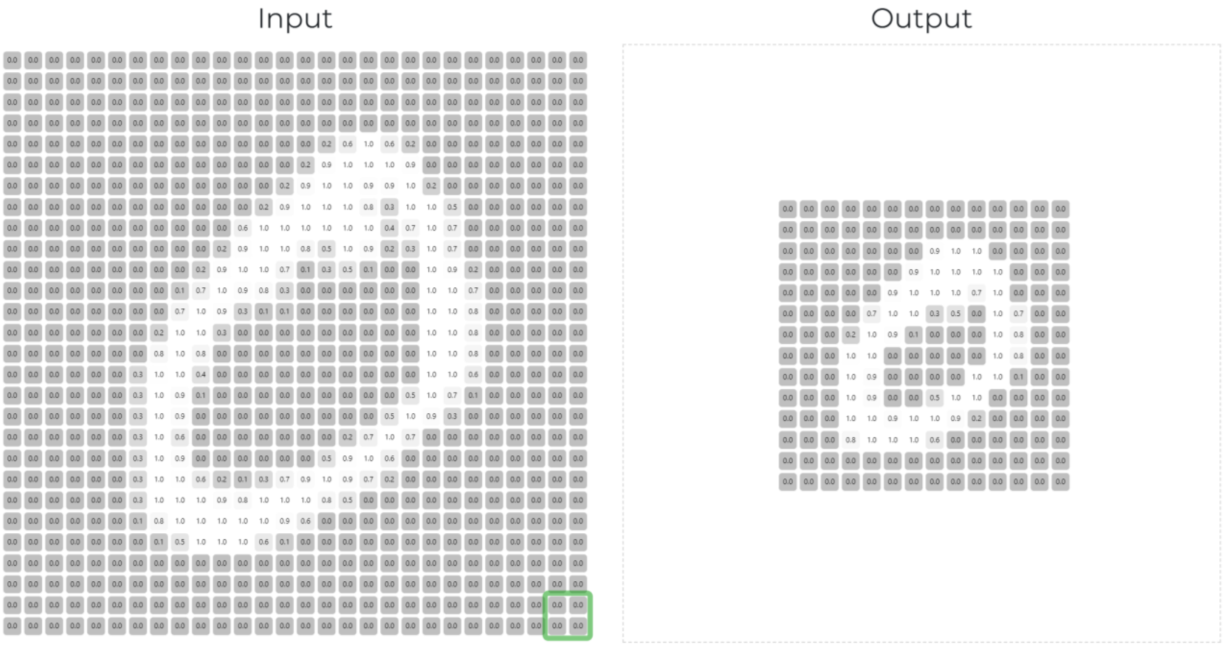
Imperial College London

Pooling Layer



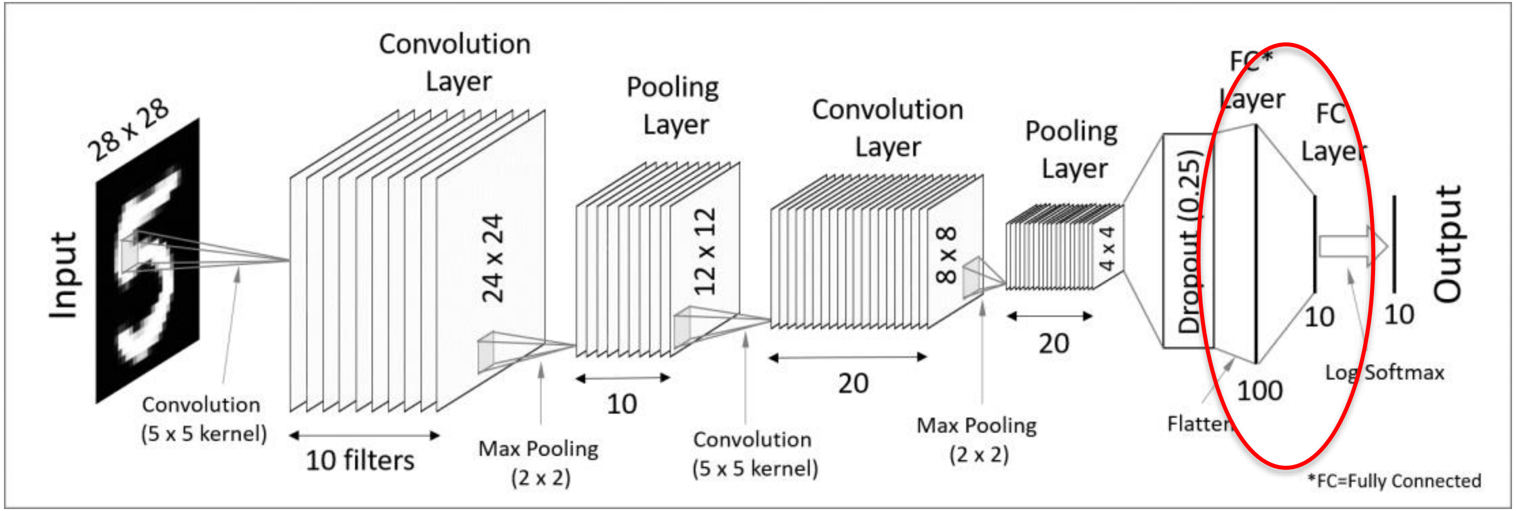
Imperial College London

Max Pooling



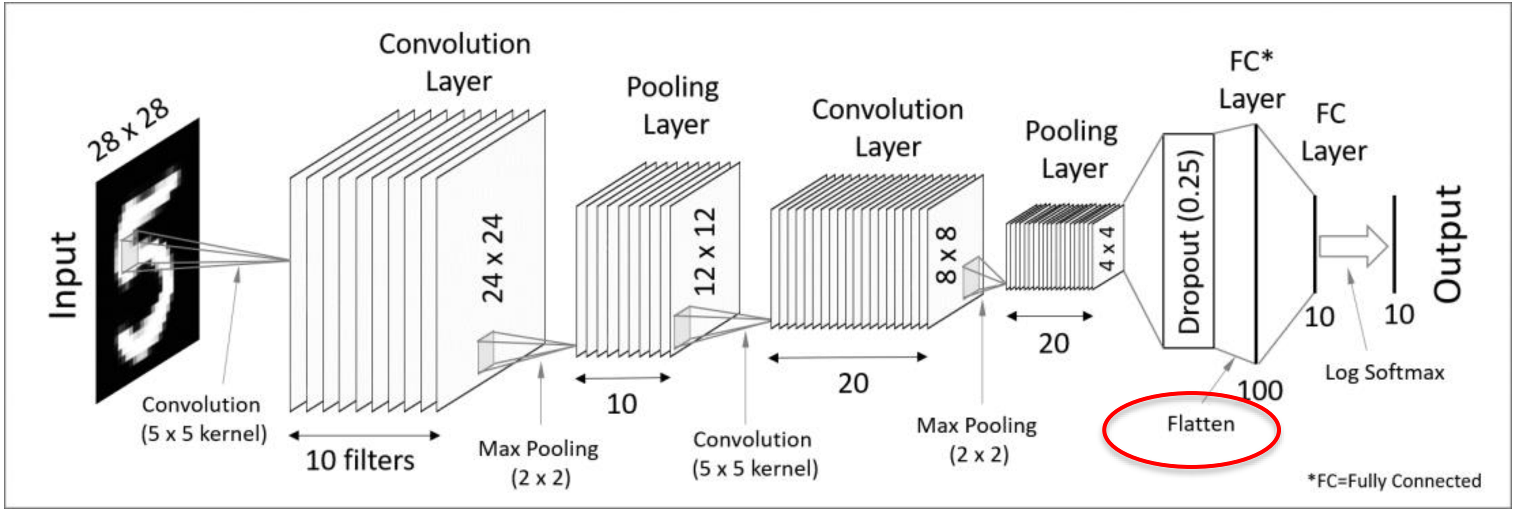
https://deeplizard.com/resource/pavq7noze3

Fully Connected Layer



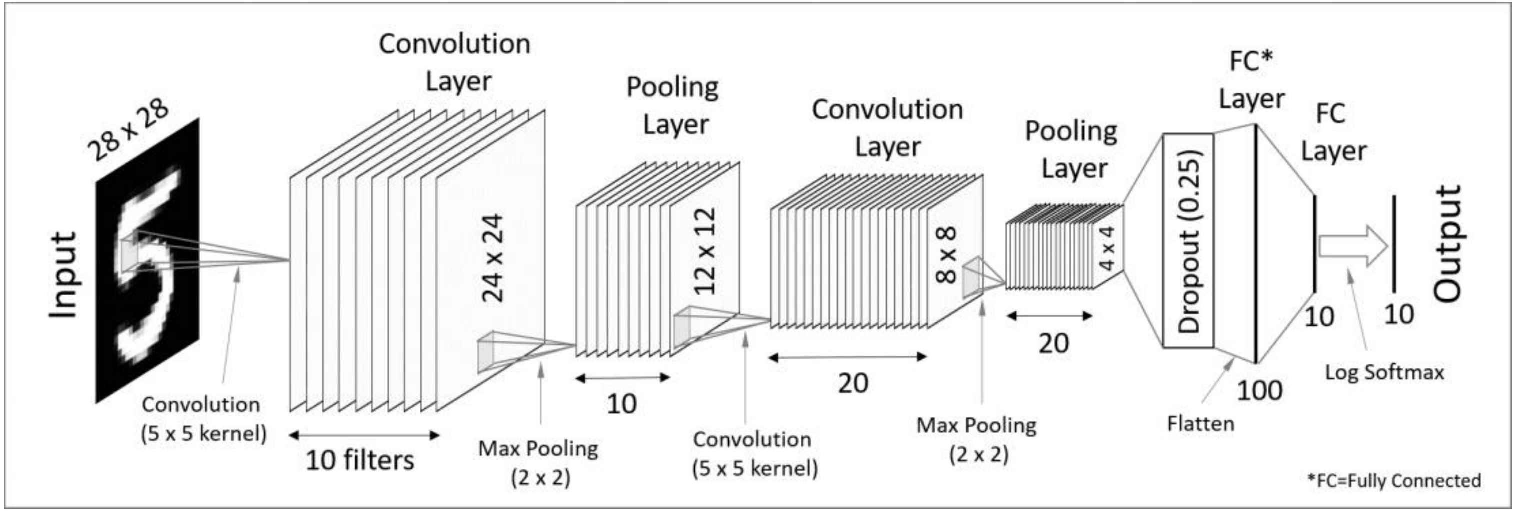
Imperial College London

Flattening



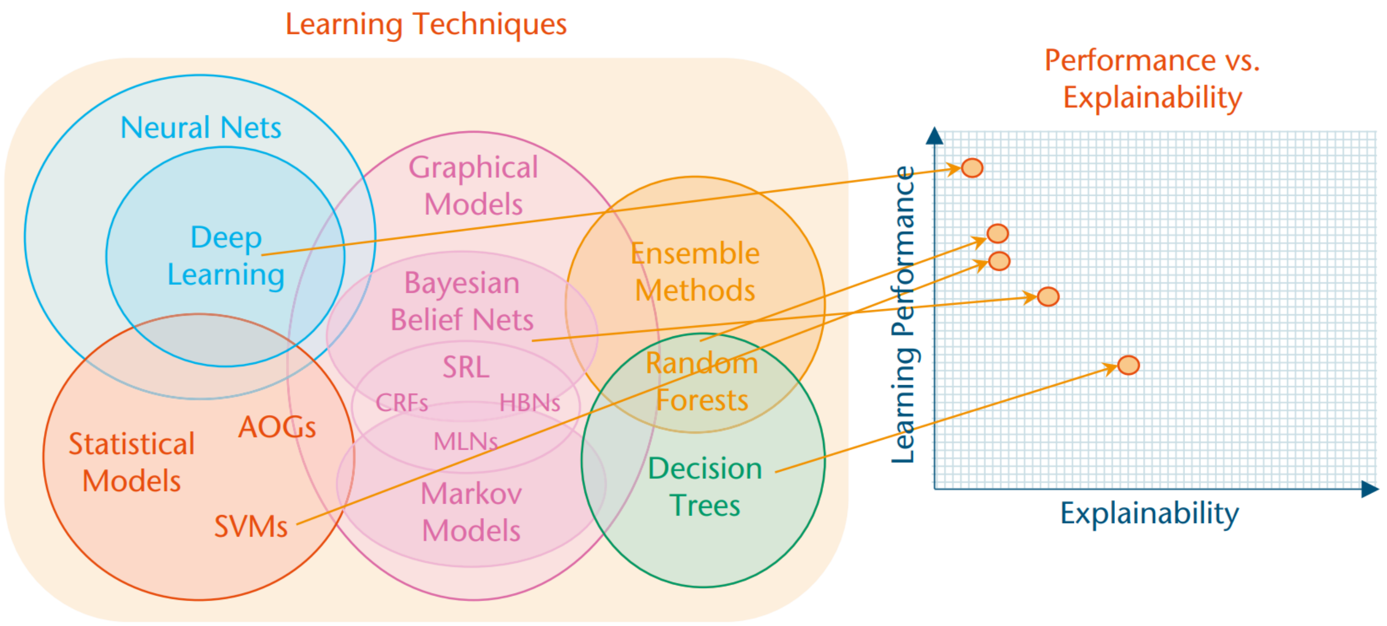
Imperial College London

Dropout Layer



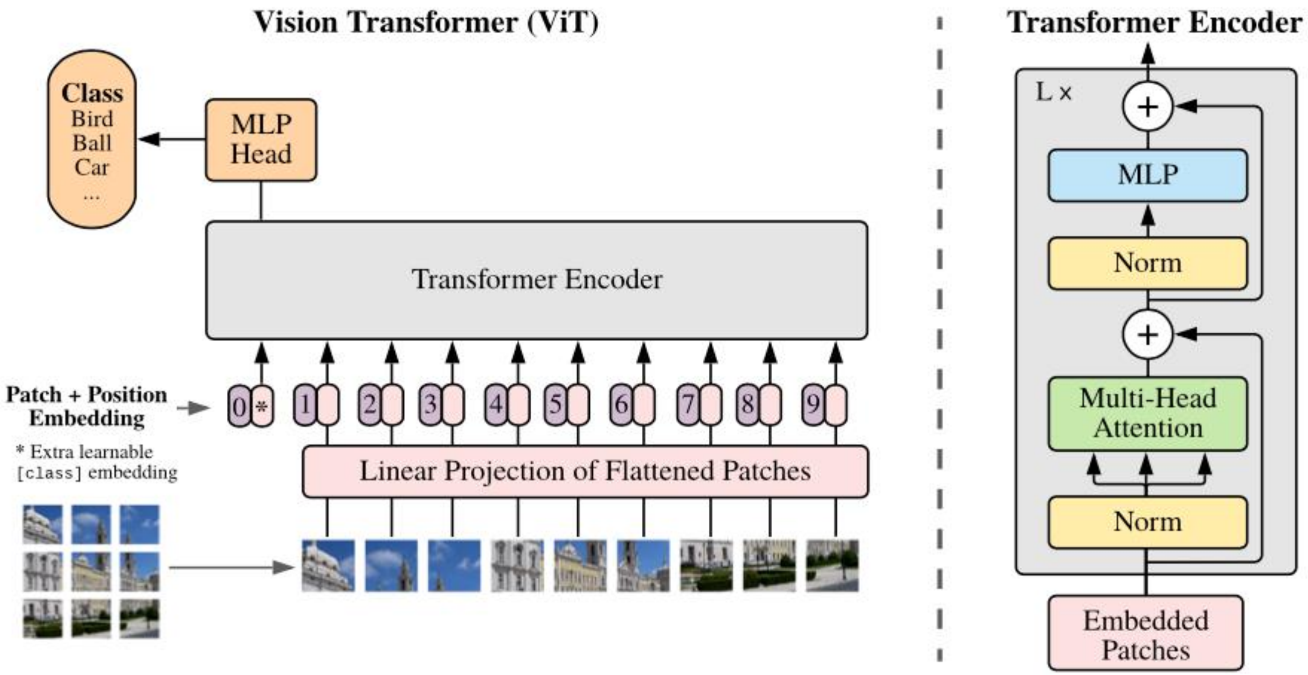
Imperial College London

Performance vs Explainability



Imperial College London

Vision Transformer (ViT)



Imperial College London

Project Workflow

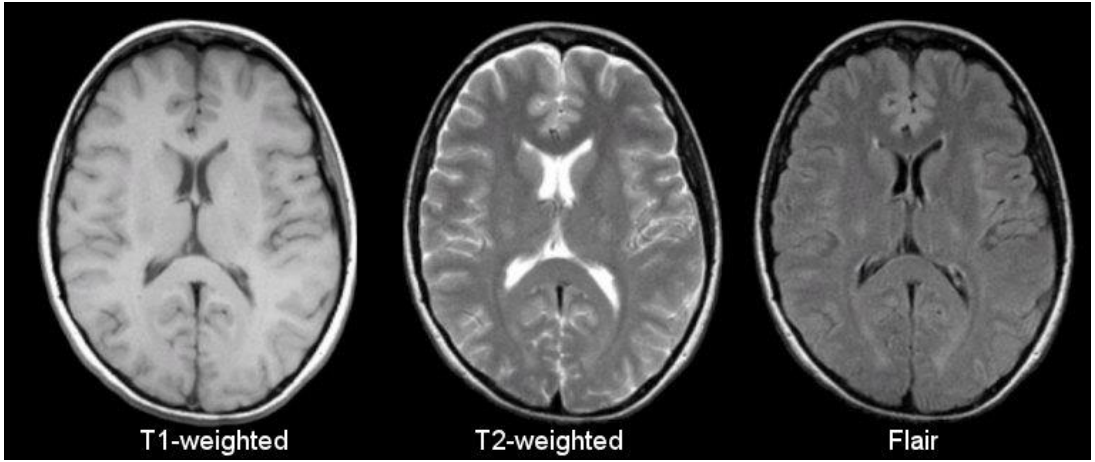
- Understanding and Preprocessing Your Dataset
- Building and Training the Model
- Optimising Hyperparameters
- Evaluating Performance
- Testing the Model

Imperial College London

Brain MRI

- Non-invasive imaging technology
- Providing three dimensional detailed anatomical images
- Used for disease detection, diagnosis, and treatment monitoring

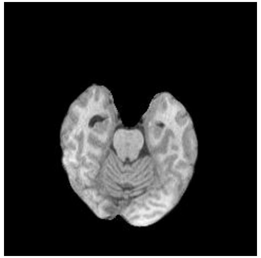
Different Sequences of Brain MR Images



Different MRI sequences

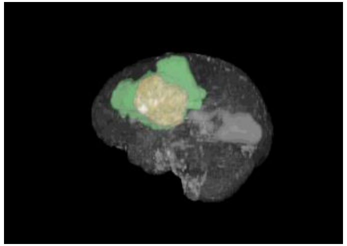
Dataset

- 210 MRI volumes will be given to you for training and evaluating the segmentation model.



Brain image with tumour

Brain image w/o tumour



Brain tumour in 3D

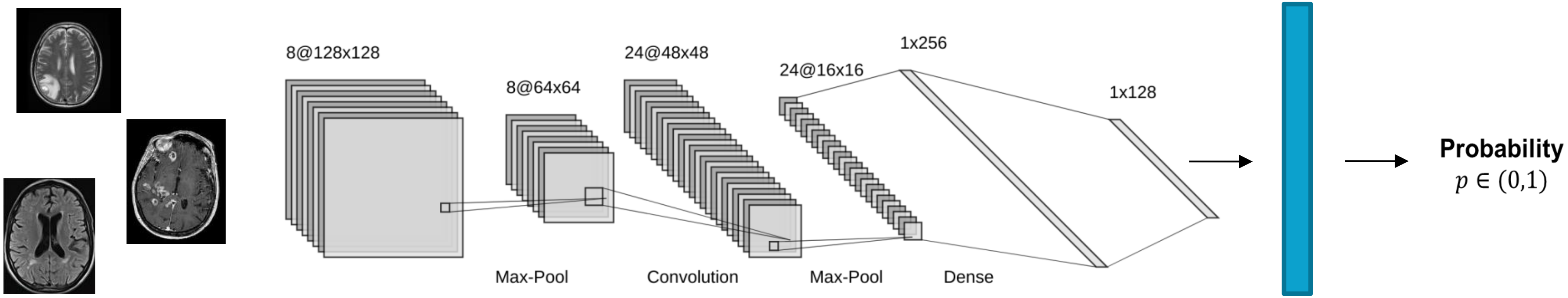
Imperial College London

Data preprocessing

- Denoising
- Image Standardisation (Resizing , Normalisation, etc.)
- Data Augmentation (Rotation, Shearing, Scaling, etc.)

Imperial College London

Building the Classification Model



Preprocessed dataset

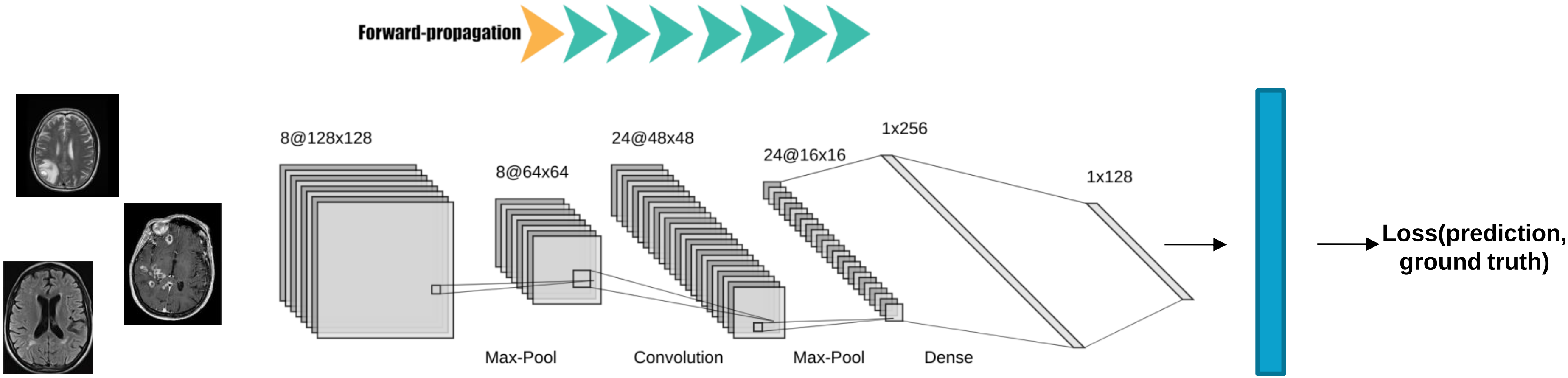
CNN-based backbone model for feature extraction (VGG16, AlexNet, ResNet, etc)

Classifier

Output

Imperial College London

Training the Classification Model



Preprocessed dataset

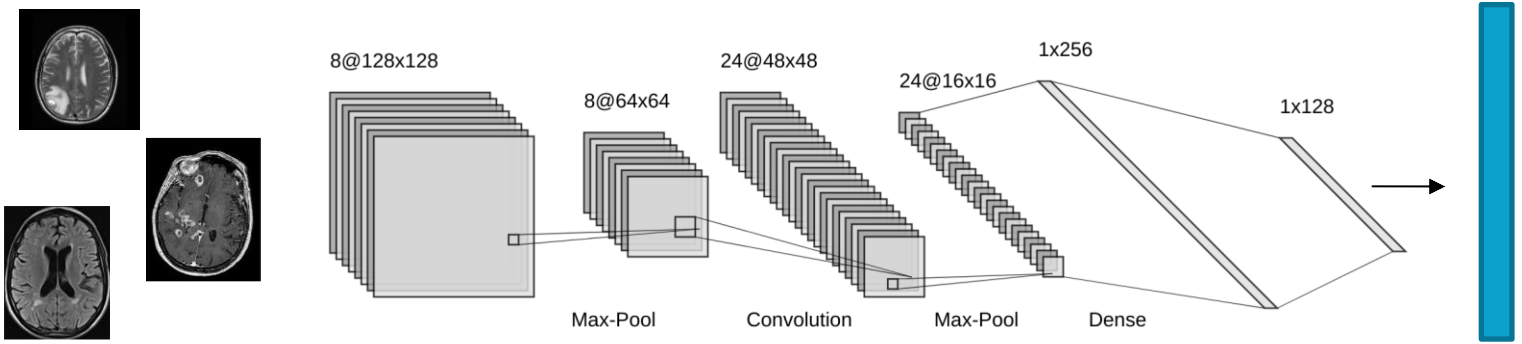
CNN-based backbone model for feature extraction (VGG16, AlexNet, ResNet, etc)

Classifier

Improving Prediction Accuracy

- Choose the right data augmentation methods;
- Find the optimal backbone model;
- Choose to finetune the model or train it from the scratch;
- Set the number of epochs to train the model and early stopping criteria;
- Tune other hyperparameters to achieve the best performance.

Test/Evaluate the Classification Model



Presence (1) or Absence (0) of Tumour

Preprocessed dataset

CNN-based backbone model for feature extraction (VGG16, AlexNet, ResNet, etc)

Classifier

Result

Imperial College London

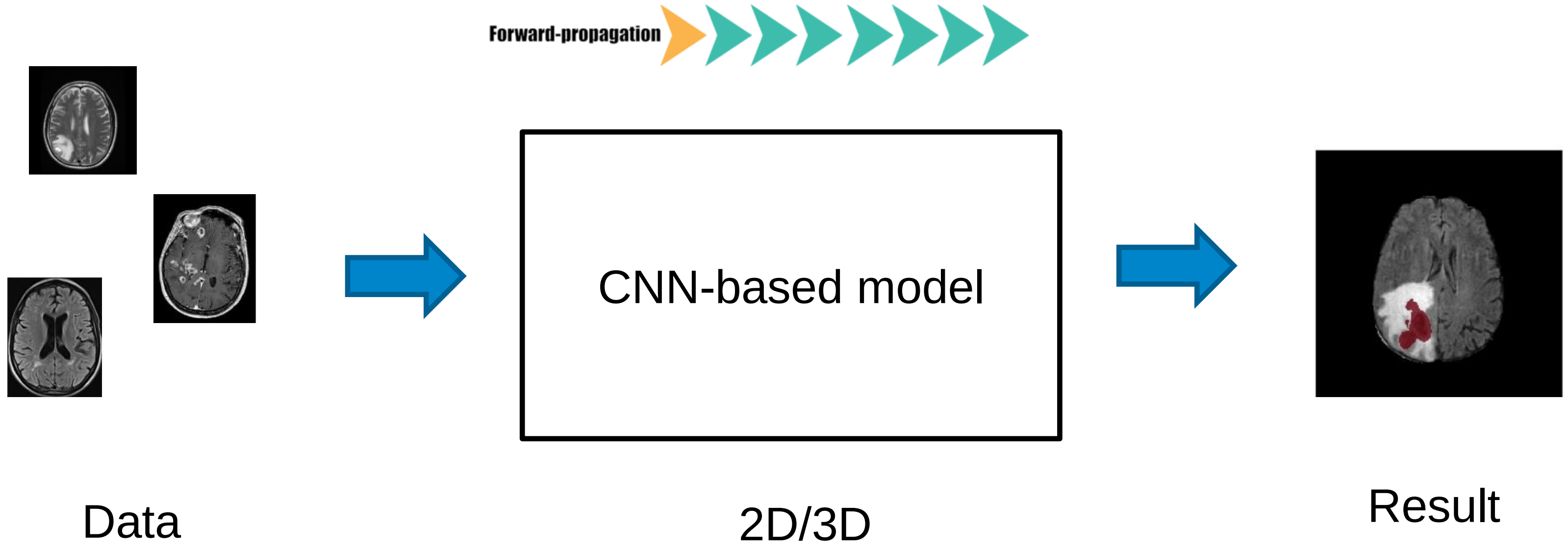
Result Evaluation

Accuracy is used to evaluate the performance of your classification model, which is defined as

$$Accuracy = \frac{Number\ of\ correctly\ classified\ images}{Number\ of\ total\ images} \times 100\%$$

Imperial College London

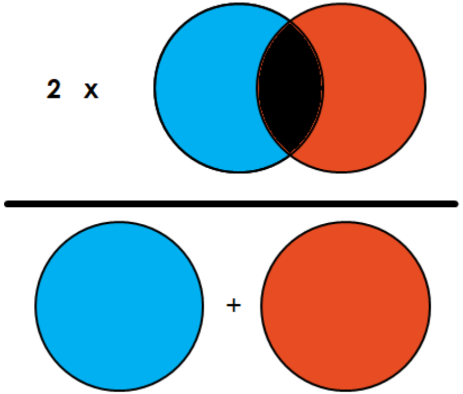
Building/Training Segmentation Model



Result evaluation

Dice score and 95% Hausdorff Distance are used to evaluate the performance of your segmentation model. Dice score is defined as

*Dice score = (2 * Area of Overlap)/(total pixels combined)*



Blue: Ground Truth
Red: Prediction

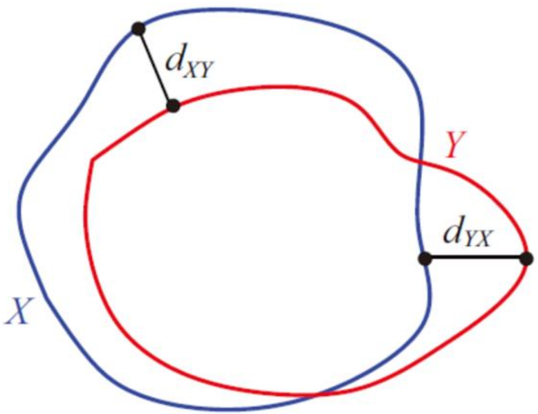
Source: https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2

Imperial College London

Result evaluation

The maximum Hausdorff Distance is defined as:

$$d_H(X,Y)=\max\{d_{XY},d_{YX}\}=\max\left\{\max_{x\in X}\min_{y\in Y}d(x,y),\max_{y\in Y}\min_{x\in X}d(x,y)\right\}$$



95% Hausdorff Distance it is based on the calculation of the 95th percentile of the distances between boundary points in X and Y.

Choice Considerations

- Model architecture: depth, width, scales, residuals,…
- Loss function: (weighted) cross-entropy, IoU, Dice,...
- Sampling strategy: equally per class, fore/background, uniform,...
- Optimization: optimizer, learning rate, momentum, regularization,…
- Data normalization/standardisation: z-score, bias field correction, histogram matching,...
- Post-processing: conditional random fields (CRF), smoothing,…

Imperial College London

Setting Up Python Development Environment

Anaconda and Tensorflow Installation (2020.02)
https://www.h2kinfosys.com/blog/how-to-download-and-setup-tensorflow-with-anaconda/
https://repo.anaconda.com/archive/

For using Nvidia GPU
https://saturncloud.io/blog/how-to-install-tensorflow-with-anaconda-on-windows/

Key Packages:
numpy>=1.19.2
tensorflow>=2.4.0
matplotlib>=3.3.2
scikit-learn>=0.24.0
tqdm>=4.50.2