Introduction to image analysis

Dr Maria Anastasiadi
(m.anastasiadi@cranfield.ac.uk)

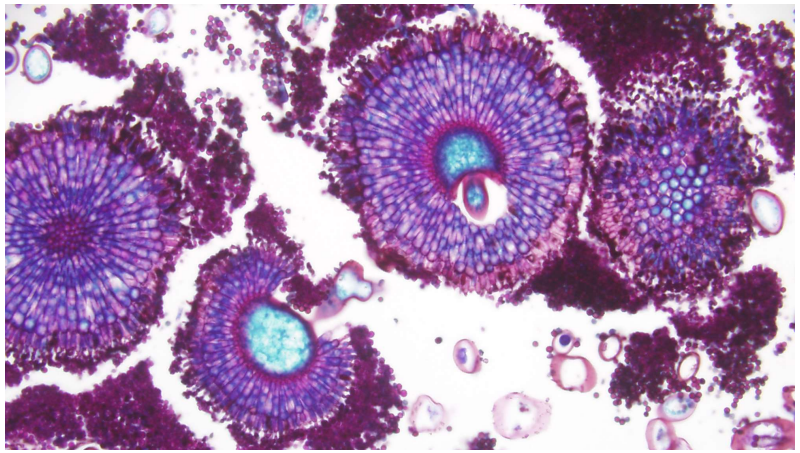
15th January 2025

www.cranfield.ac.uk

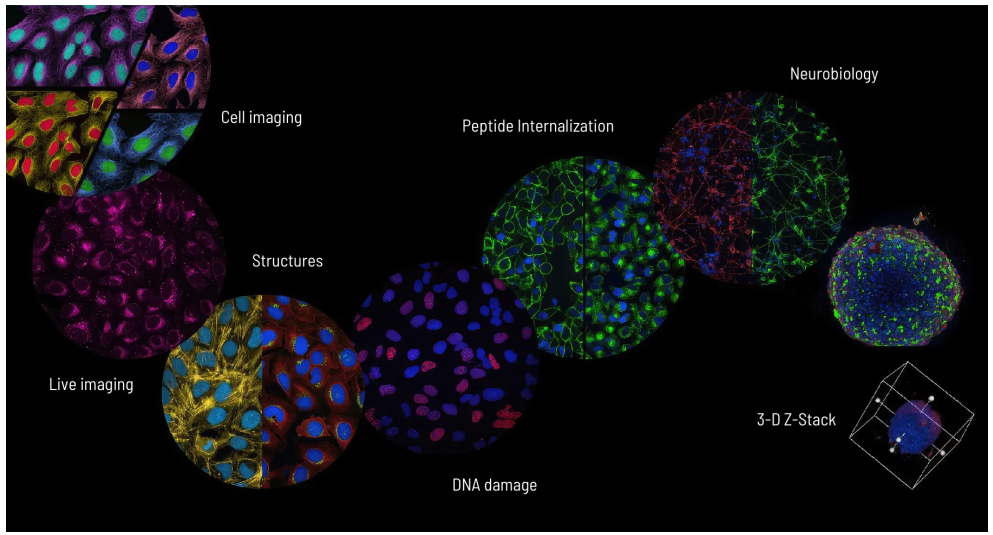
What are images used for in bioinformatics?

Some examples:

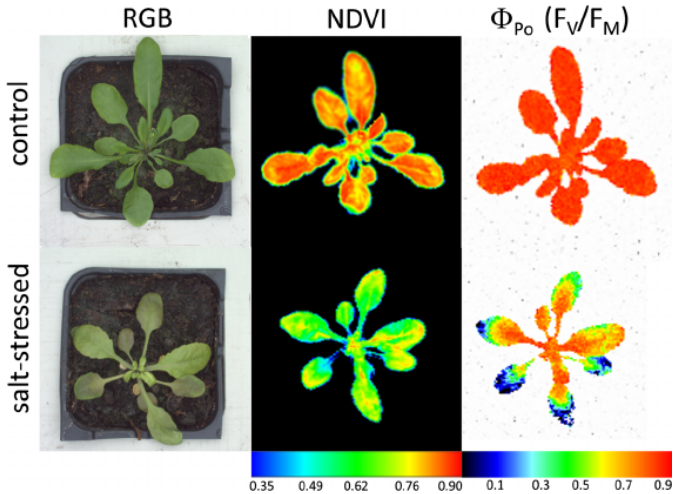
Medical Imaging



Histopathology



High Content Cell Imaging



Plant Phenotyping

DOI 10.1186/s13007-015-0072-8

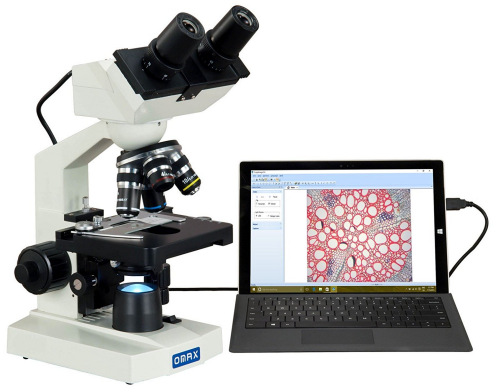
Sources of images in bioinformatics

Some examples:

MRI Scan

Multispectral Camera

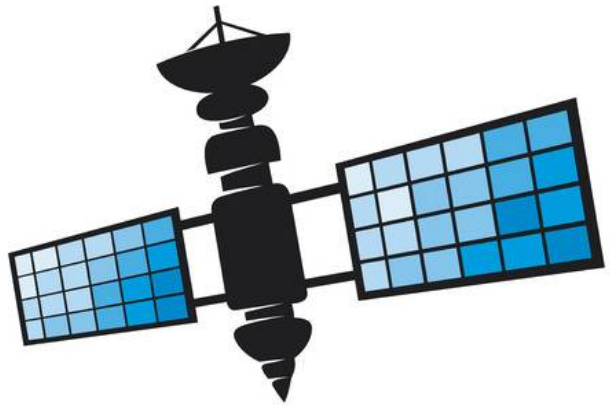
Digital Microscope



X-Ray Machine

Scanning Electron Machine

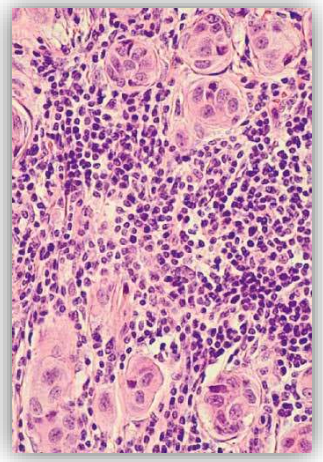
Satellite Imagery



Current challenges

Example: Computational Pathology

- Using mathematical models and different data sources to diagnose disease.
- One of the current challenges in Bioinformatics is building an accurate statistical model to identify different types of cancer cells using computer vision.
- Model doesn’t just have to be as accurate as a pathologist, it needs to be faster and cheaper than a pathologist too.
- Biggest issues:
  - o Computers aren’t fast enough;
  - o Models aren’t accurate enough;
  - o There isn’t enough public data to test models on.



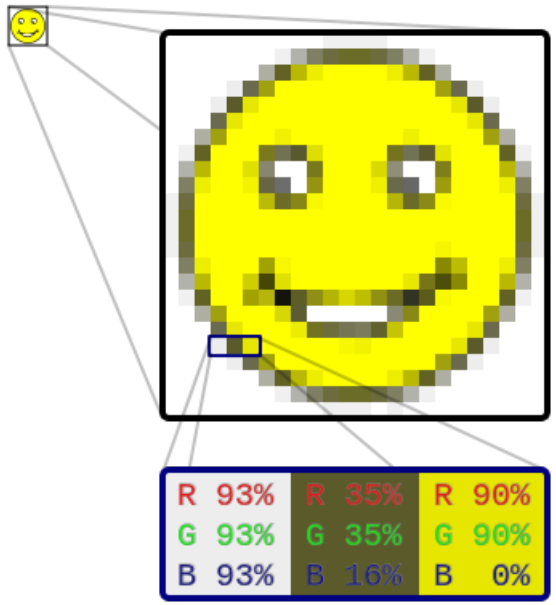
What is an image?

2 common formats: Raster or Vector Image

Raster Images

Dot matrix data structure usually depicting a rectangular grid of pixels.

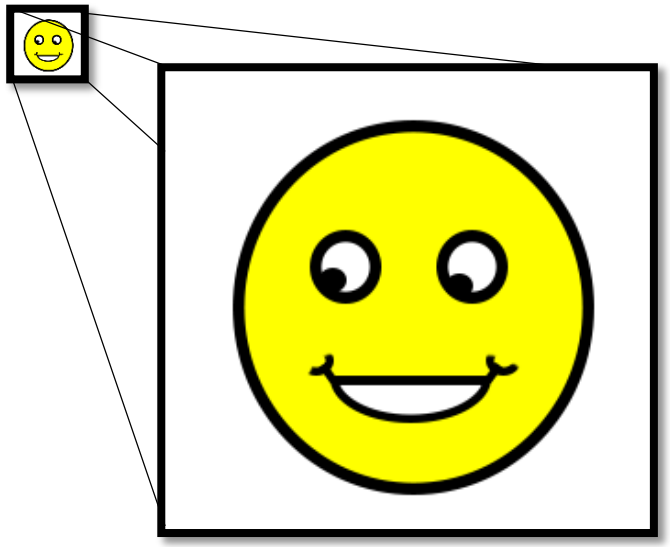
- PNG
- JPEG
- TIFF
- BMP



Vector Images

Geometrical description of the image, allowing it to be rendered smoothly for any display size.

- SVG
- PDF
- EPS
- AI

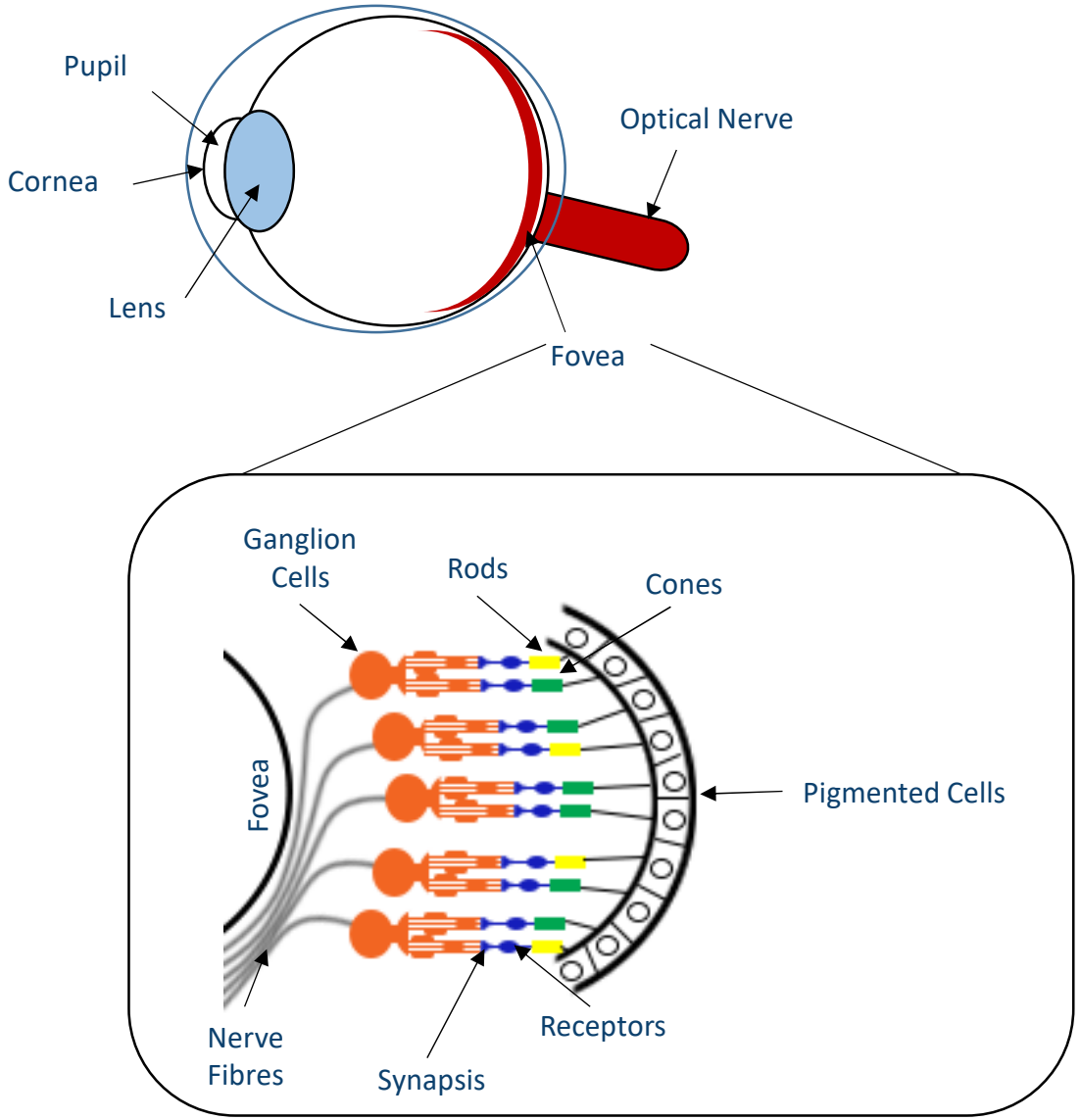


How do we see?

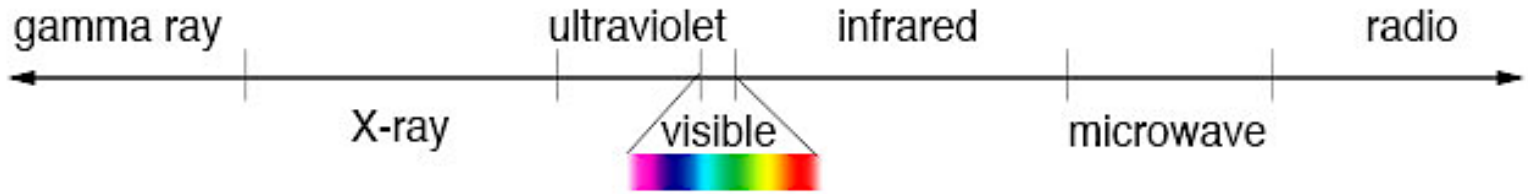
Two types of photoreceptors in the eye:

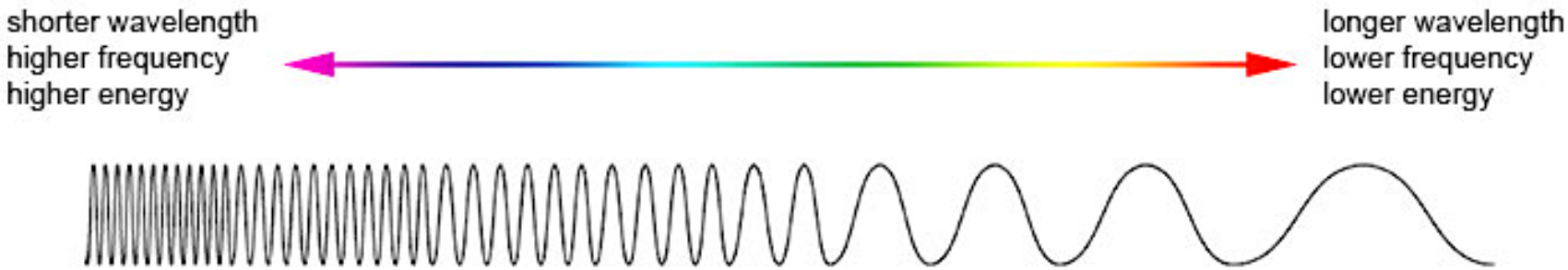
- **Rods**
  - Responsible for vision at low light levels;
  - Not capable of colour vision;
  - Low spatial acuity.
- **Cones**
  - Responsible for vision at high light levels.
  - Are capable of colour vision.
  - High spatial acuity.
  - 3 types of cones:
    - Short-wavelength sensitive;
    - Medium-wavelength sensitive;
    - Long-wavelength sensitive.

Structure of the eye

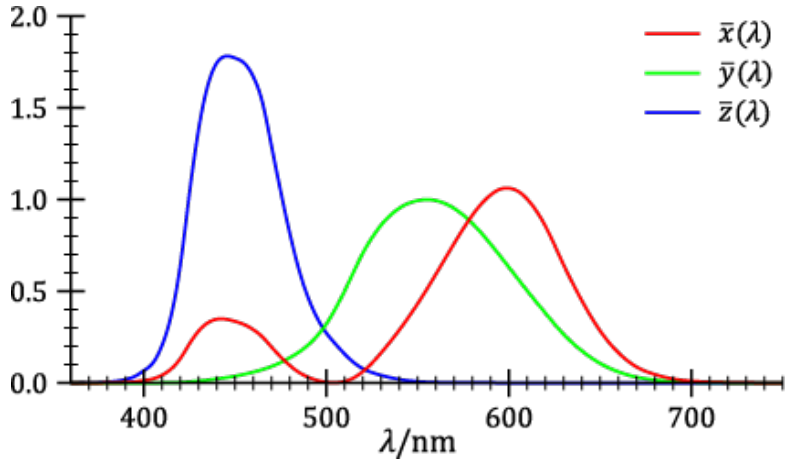


What is light?





Visible range is between 350nm (Blue), and 750nm (Red)



Colour space

All the colours a given system can reproduce.

One of the first defined colour-spaces was
CIE XYZ 1931 Colour Space which uses 3 channels:

- Luminescence
- Two Colour coordinates from the chromaticity diagram.

Currently, RGB colour spaces are much more common and is simpler to understand. 3 colour channels, one for each type of cone:

- Short-wavelength — Blue
- Medium-wavelength — Green
- Long-wavelength — Red

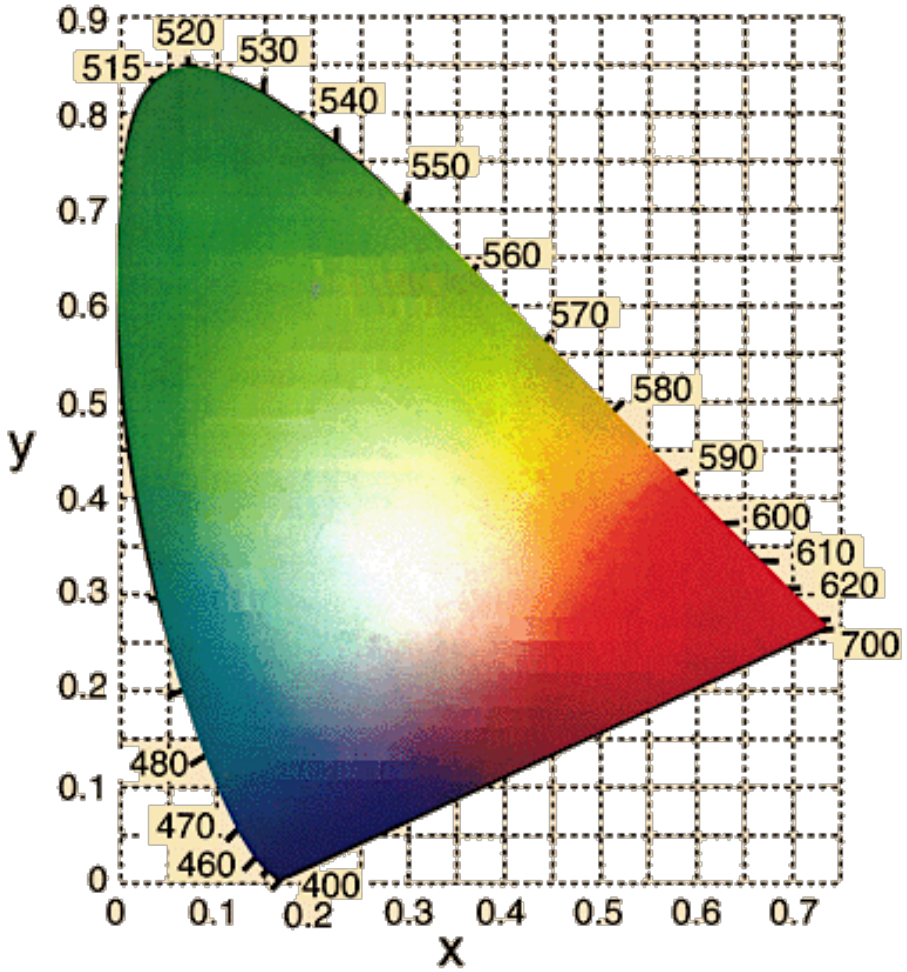
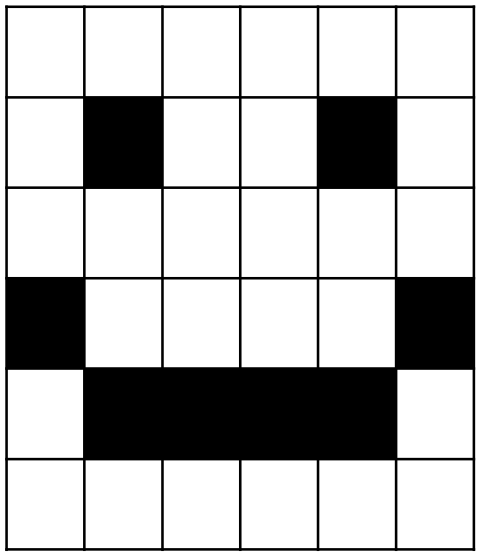
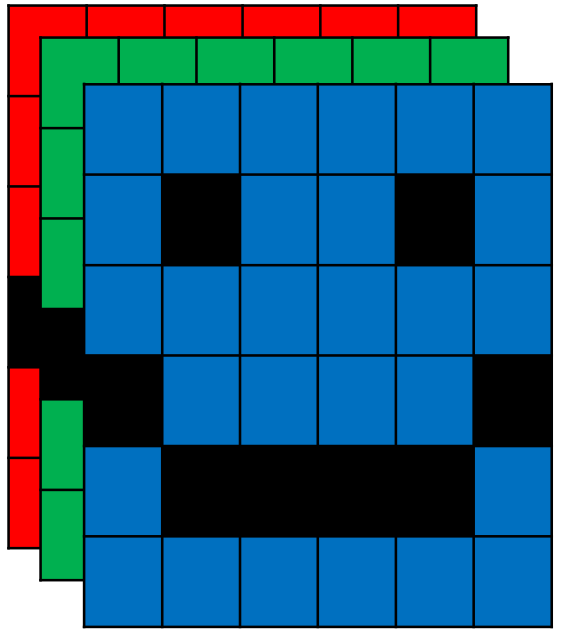


Image structure

3 Channels: Red, Green Blue



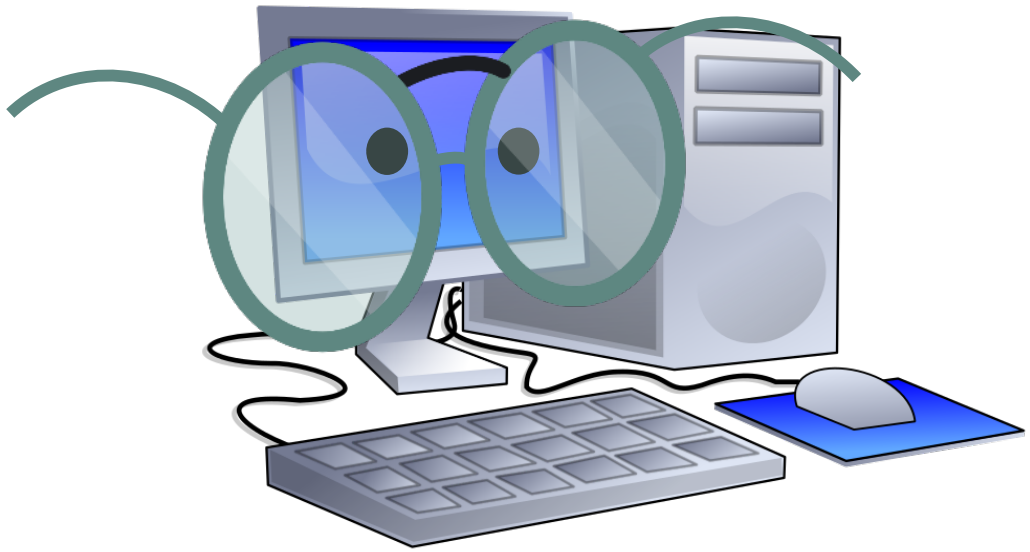


- Images are manipulated using a matrix structure.
- Greyscale images use values $0 \leq I \leq 1$ in a 2D matrix.
- RGB colour images use values $0 \leq I \leq 255$ in a 3D matrix.

What can we do with an image?

How can the computer see? Computer Vision

- False colour images;
- Noise Reduction;
- Contrast & Brightness Correction;
- Geometric Correction / Extraction (3D);
- Image segmentation;
- Edge Detection;
- Shape identification;
- Machine learning for object identification.



Background removal

Also called foreground detection

Most images will have redundant information in them.

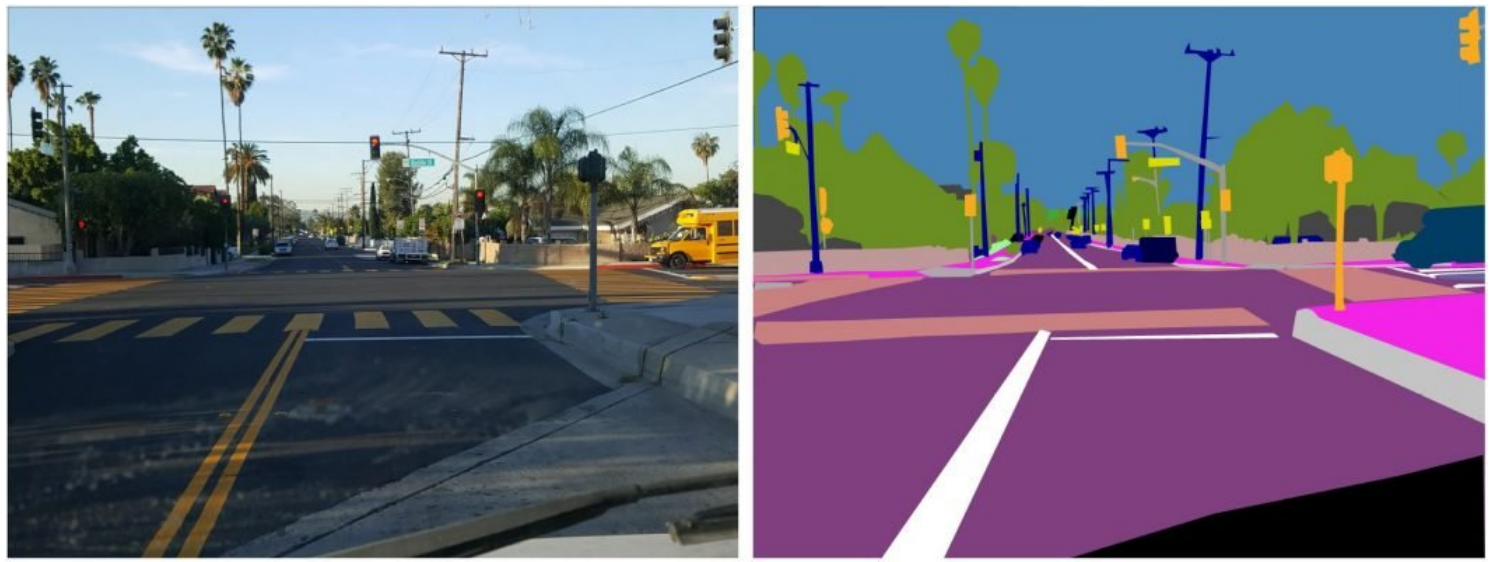
Before analysis can take place, this information needs removing.

How best to remove it?

- Frame Differencing
- Mean Filtering
- Running Gaussian Average

Image segmentation

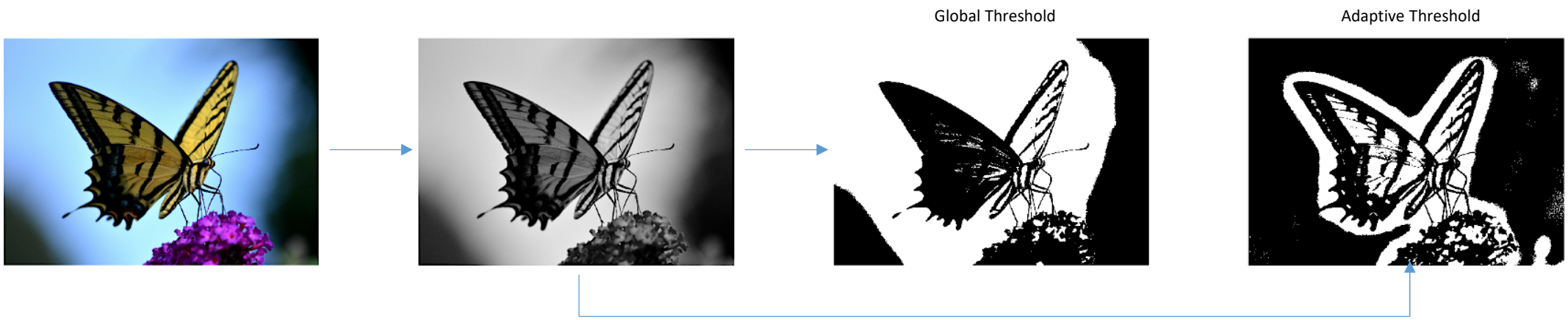
- Segmenting an image can help differentiate between objects in an image;
- Useful for computer vision and machine learning;
- Also used for background removal in simple images;
- Precursor step to object identification.



Thresholding (greyscale to binary)

Most basic form of segmentation, used on greyscale images; if the image is not initially greyscale, it is converted before thresholding takes place.

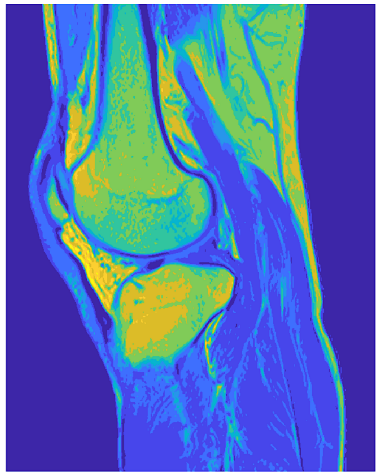
$$T = 0.5$$
$$I_{x,y} \geq T \rightarrow I_{x,y} = 1$$
$$I_{x,y} < T \rightarrow I_{x,y} = 0$$



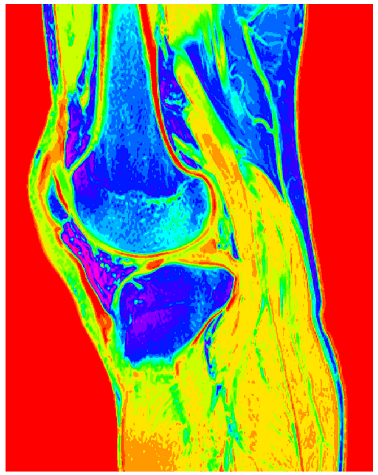
False colour images

- As there are some light wavelengths the human eye cannot detect that contain useful information, we have to convert them into a pseudo RGB image.
- In other cases, there are greyscale images that can be scaled to see the differences between pixels more clearly, for example in MRI images the more dense the tissue, the more white it is, but it can be made more clear by applying a colour map.

Original Scan



Parula(10)



Hsv(20)

Image enhancement

Multiple methods for enhancing images, including but not limited to:

- **Geometric Corrections:** When taking photos over time, it's possible to measure growth or shrinkage of objects, but only if the geometric space is the same in all images.
- **Noise Reduction:** Used to smooth images to make segmentation more effective.
- **Contrast Enhancement:** Used to exaggerate visible differences in order to enhance edge detection techniques.
- **Light/Luminosity Correction:** Used to enhance foreground detection techniques.

Contrast & Brightness Enhancement

- Brightness is the overall intensity of the image, changing this can help reduce noise.
- Contrast is the difference in light between pixels, changing this value can really enhance certain edge detection methods.

The formula for changing the contrast is:

$$NI_c = (\alpha \times CI_c) + \beta$$

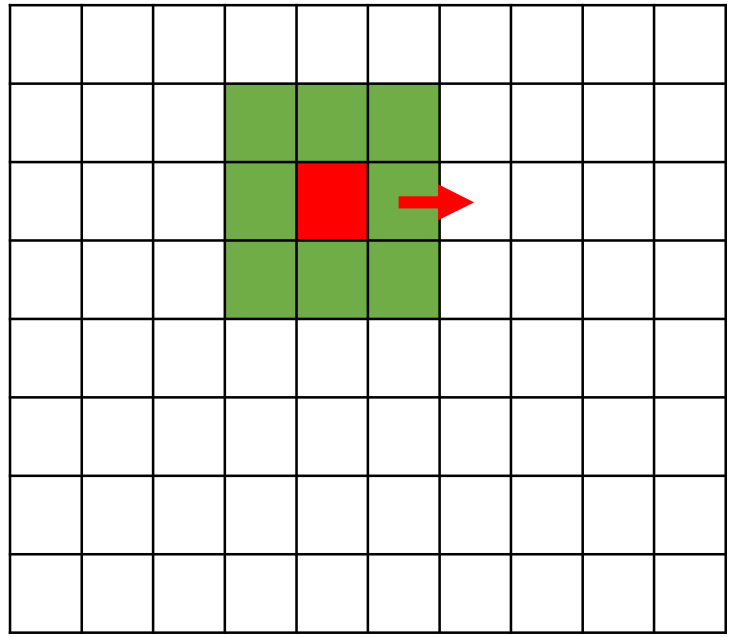
Where NI is the new image matrix, c is the colour channel (R,G,B), **α is the contrast scalar**, CI is the current image, and **β is the brightness scalar**.

$$0 < \alpha < 3$$

$$0 < \beta < 255$$

Noise Reduction

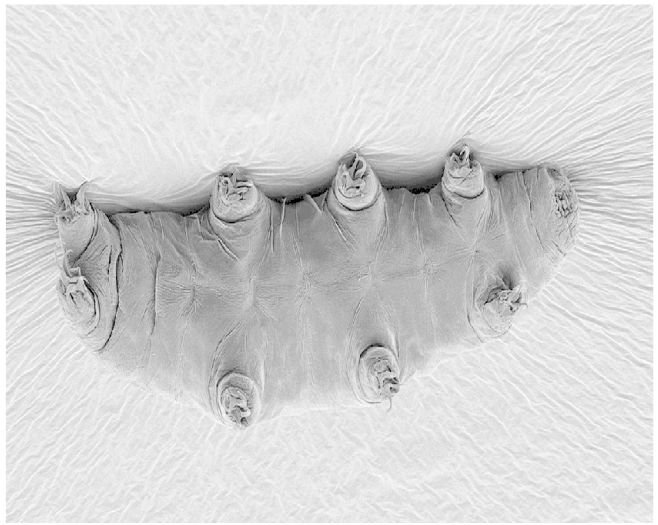
- Noise reduction is important as edge detection techniques can be sensitive to it, meaning that a lot of useless data is extracted too.
- One effective technique is Median Filtering, this involves moving a ‘sliding window’ over the image, extracting the median pixel of the window and storing it as the pixel at (x,y) in the new image.



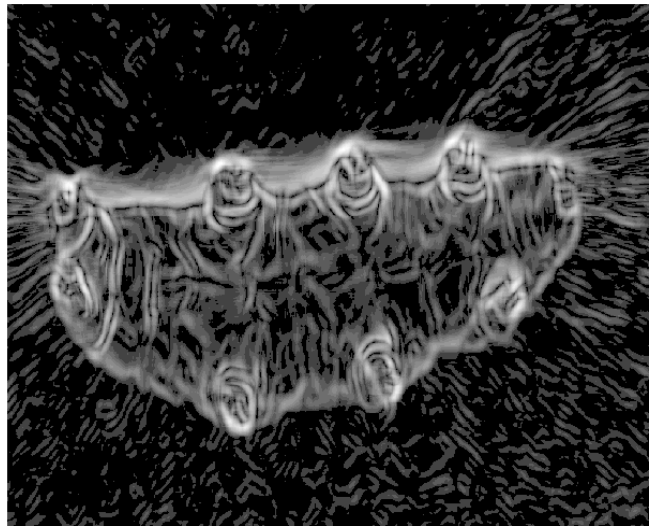
Edge Detection

- Finding both horizontal and vertical edges requires combining a pair of **convolutional kernels.**
- Sobel Edge Detection works by performing a 2D spatial gradient measurement on an image, emphasising regions where there is a high spatial frequency which indicate edges.

Greyscale image

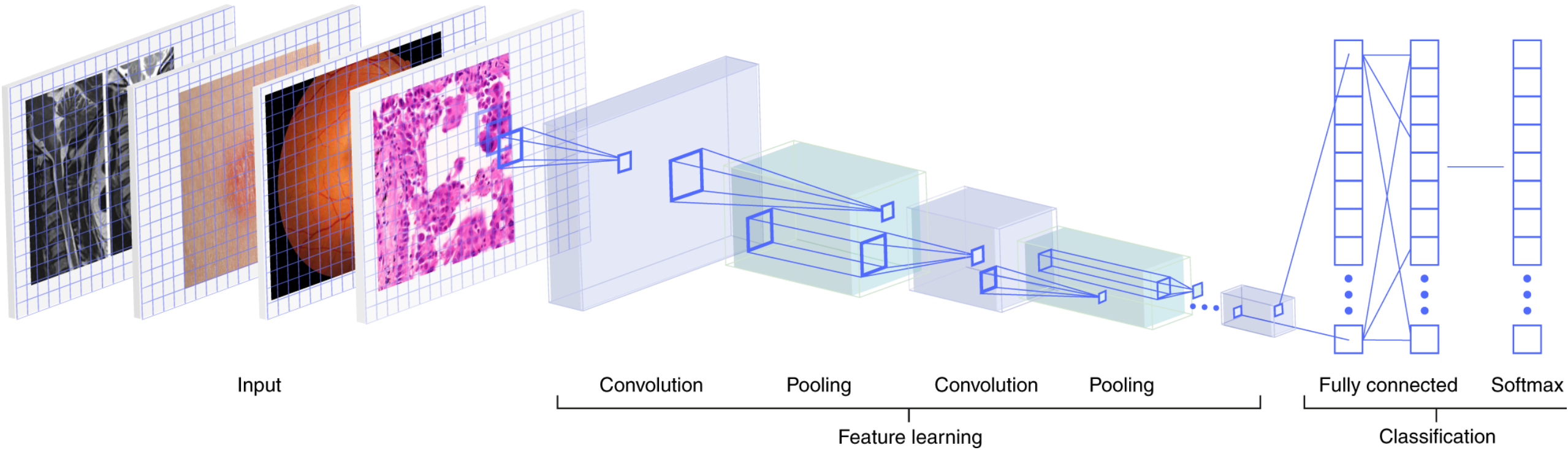


Sobel image



Convolutional Neural Networks (CNNs)

- The idea behind CNNs is that objects in the same category will have similar features as each other.
- Convolve the image down to its most basic features and the edges should be similar to each other.

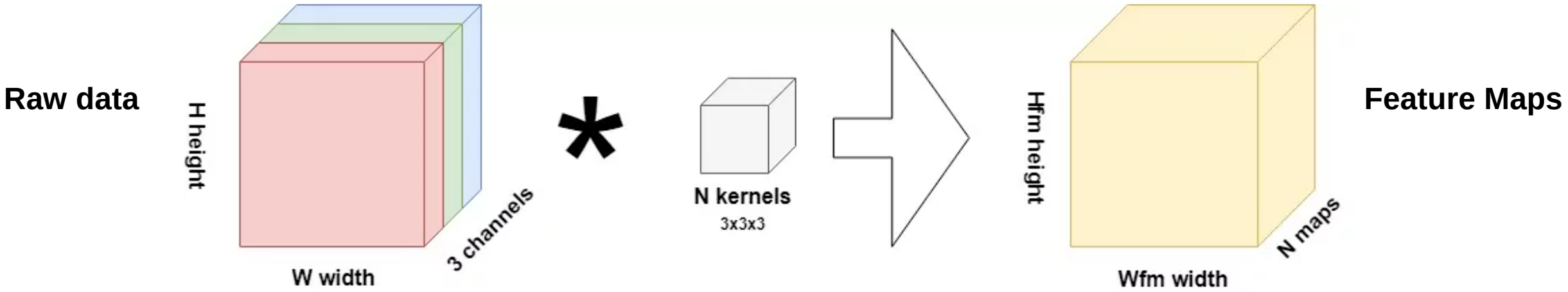


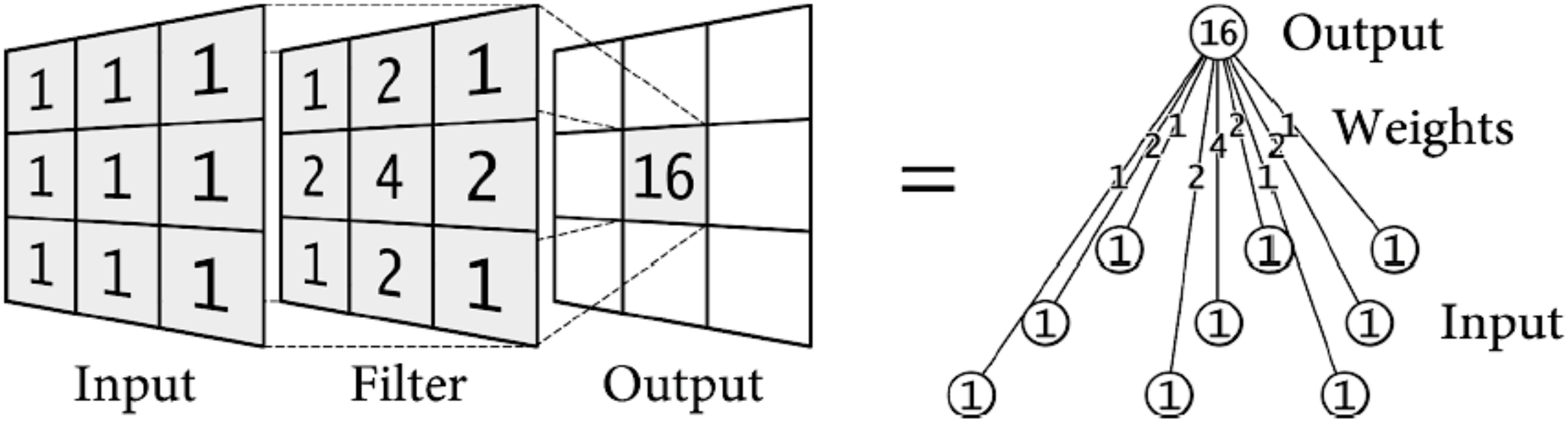
(Figure ref: Esteva A. et al., *Nat Med* (2019). https://doi.org/10.1038/s41591-018-0316-z)

Convolutional Neural Networks (CNNs)

Operations:
First convolutional layer - Filtering

1. Input: image height, width, and RGB colour channels.
2. Kernel: a matrix that moves over the input data to extract high-level features (such as vertical or horizontal lines and edges) from the images.
3. As the output, each feature map learns **one particular feature** in the image.
4. Together, all feature maps contribute to the prediction results.





- A 3×3 convolution filter is equivalent to a node with 9 weighted connections.
- The filter values correspond to the weights.
- A convolution layer applies this filter to every location in the input image, producing a new (filtered) image.

Convolution

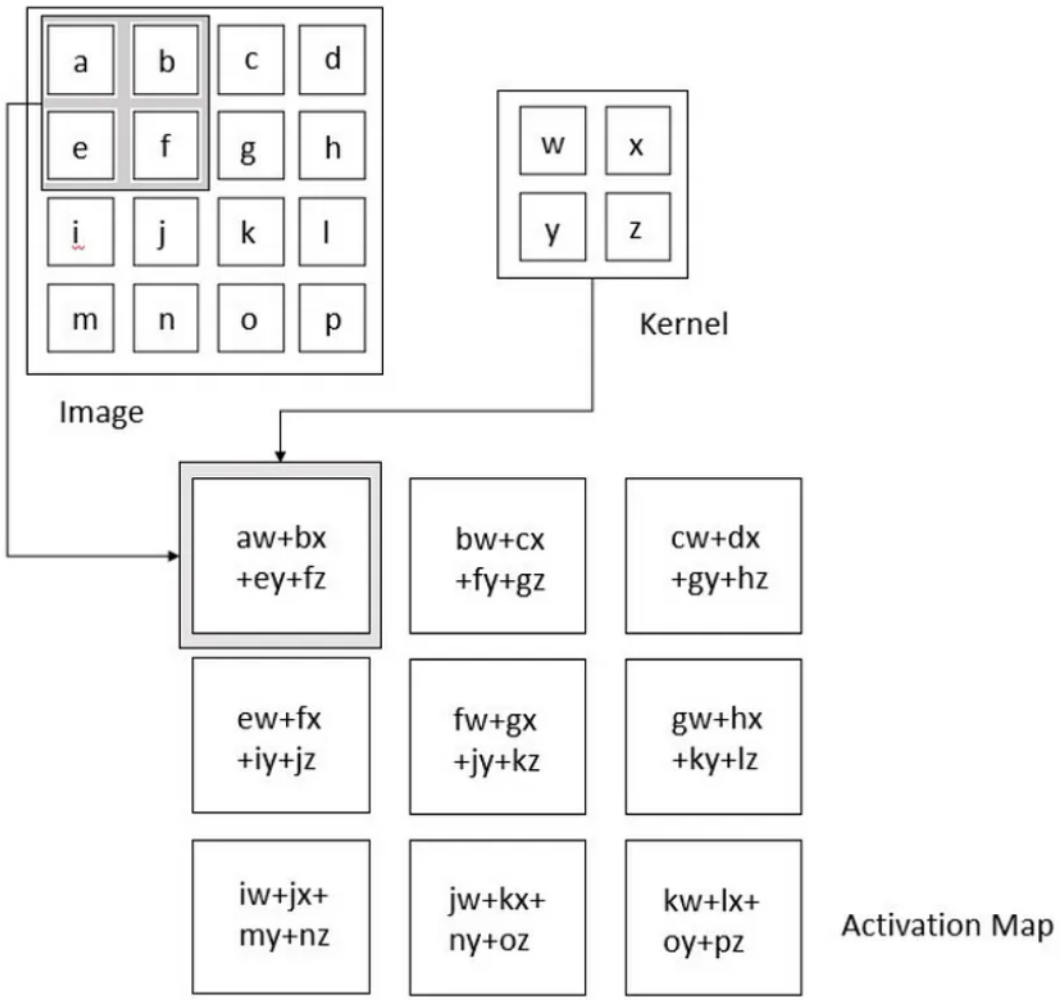


Figure 3: Convolution Operation (Source: Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

Formula for Convolution Layer

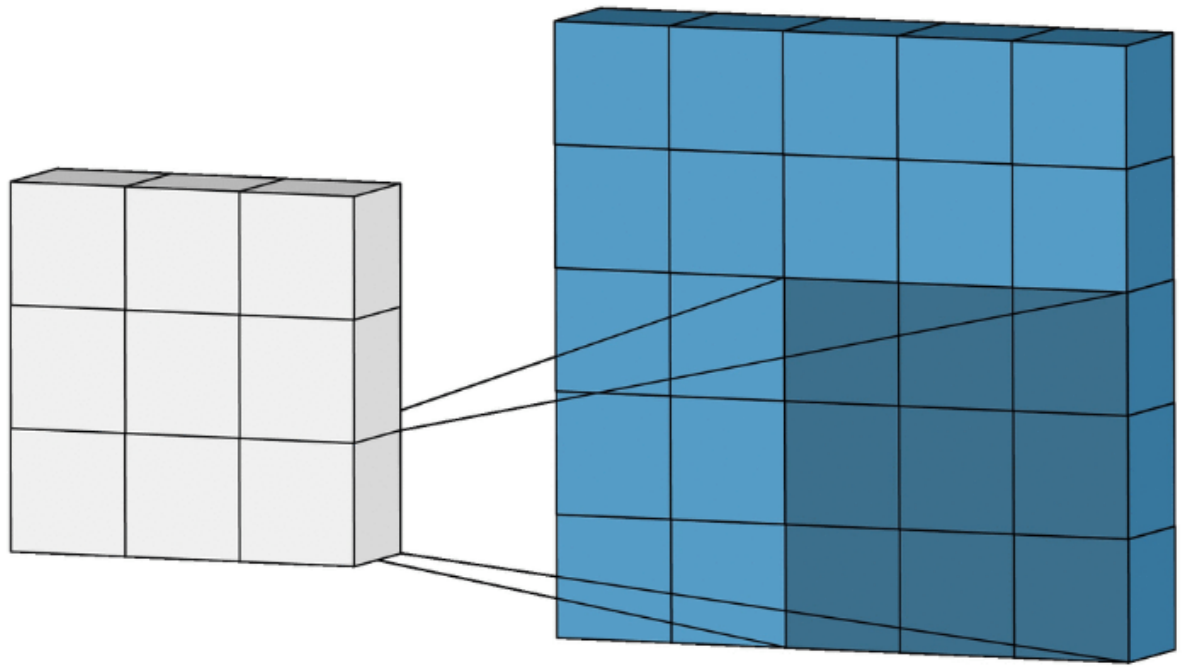


Image Source: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

Convolution

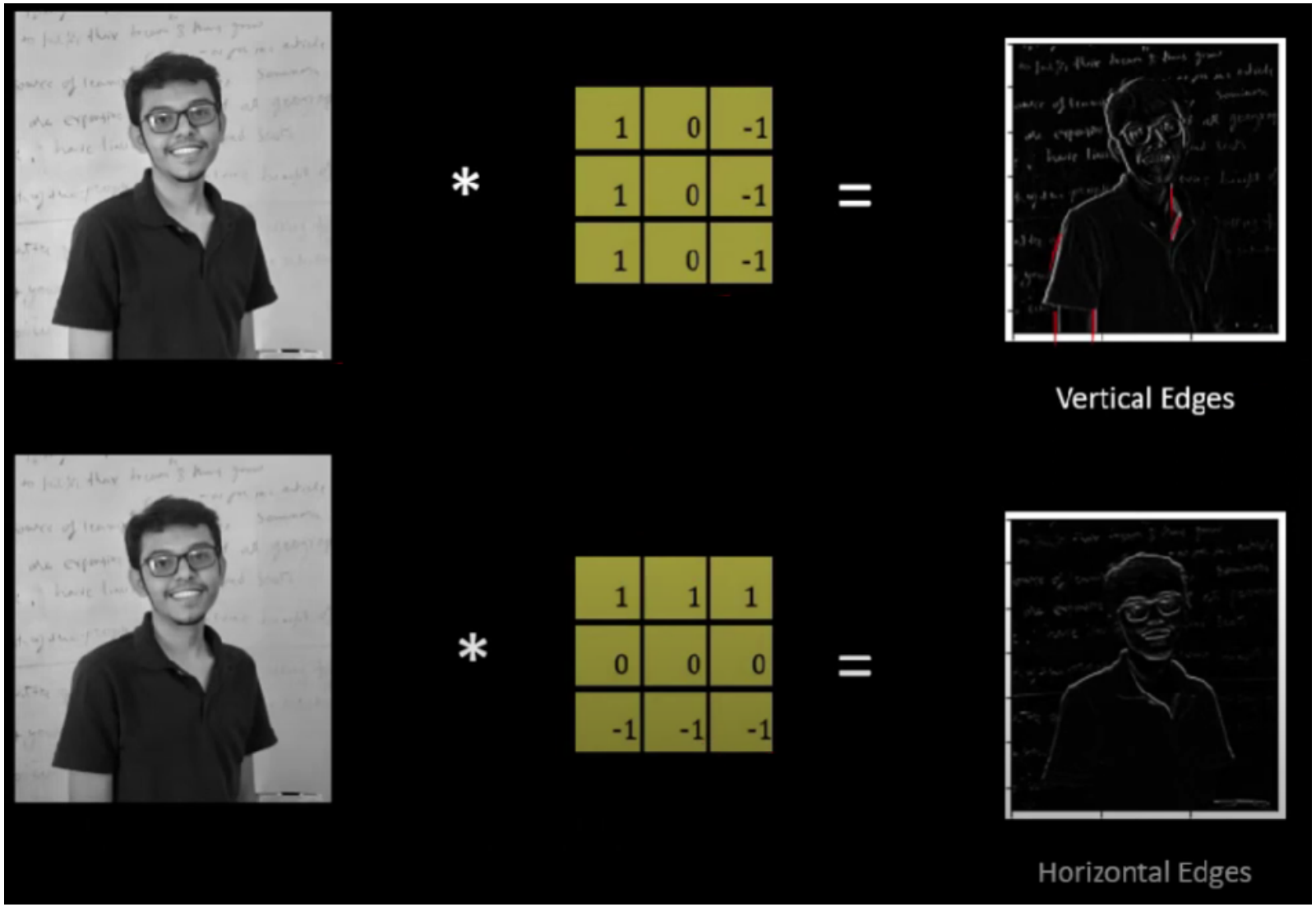


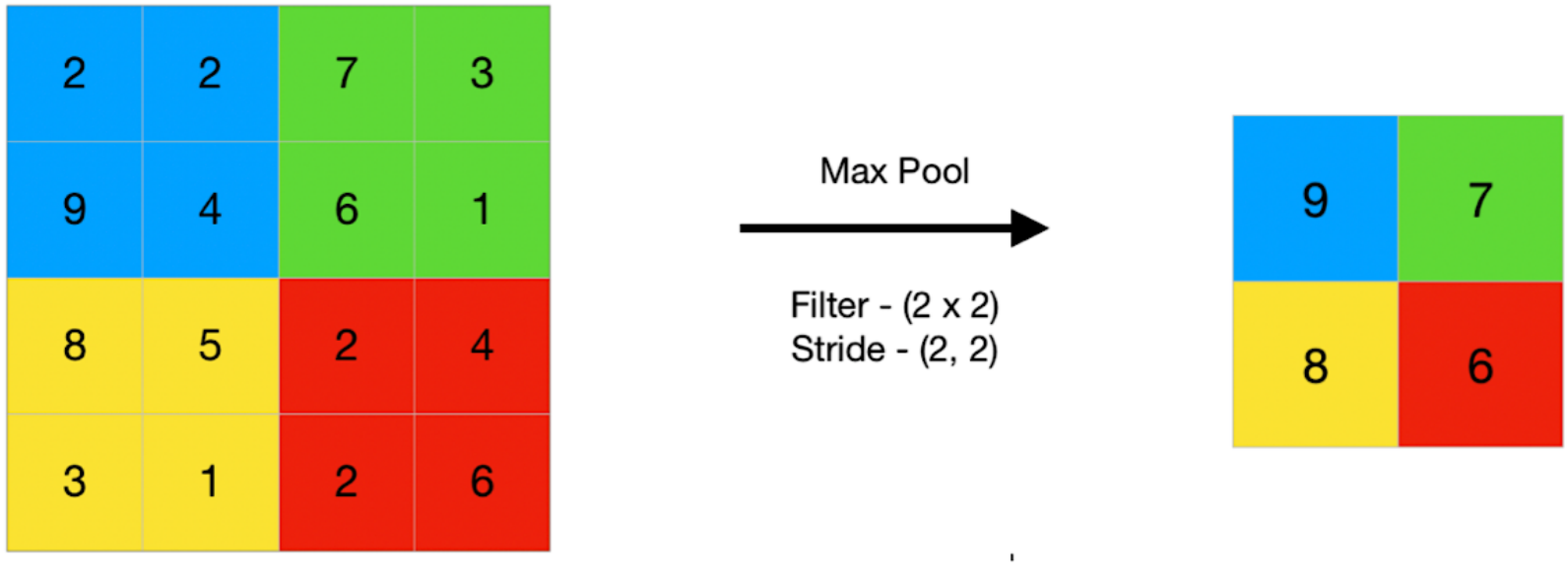
Image Source: https://www.youtube.com/watch?v=gLwX3zHkims&list=PLuhqtP7jdD8CD6rOWy20INGM44kULvrHu&index=5&ab_channel=CodingLane

Convolutional Neural Networks (CNNs)

Operations:
First convolutional layer - Pooling (Downsampling)

1. Sliding a two-dimensional filter over each channel of feature map.
2. Summarising the features lying within the filter region.
3. Types of pooling: Max, Min, Average.

$$W_{out} = \frac{W - F}{S} + 1$$



The output:
A feature map containing the most prominent features of the previous feature map

Fully Connected Layer in CNN

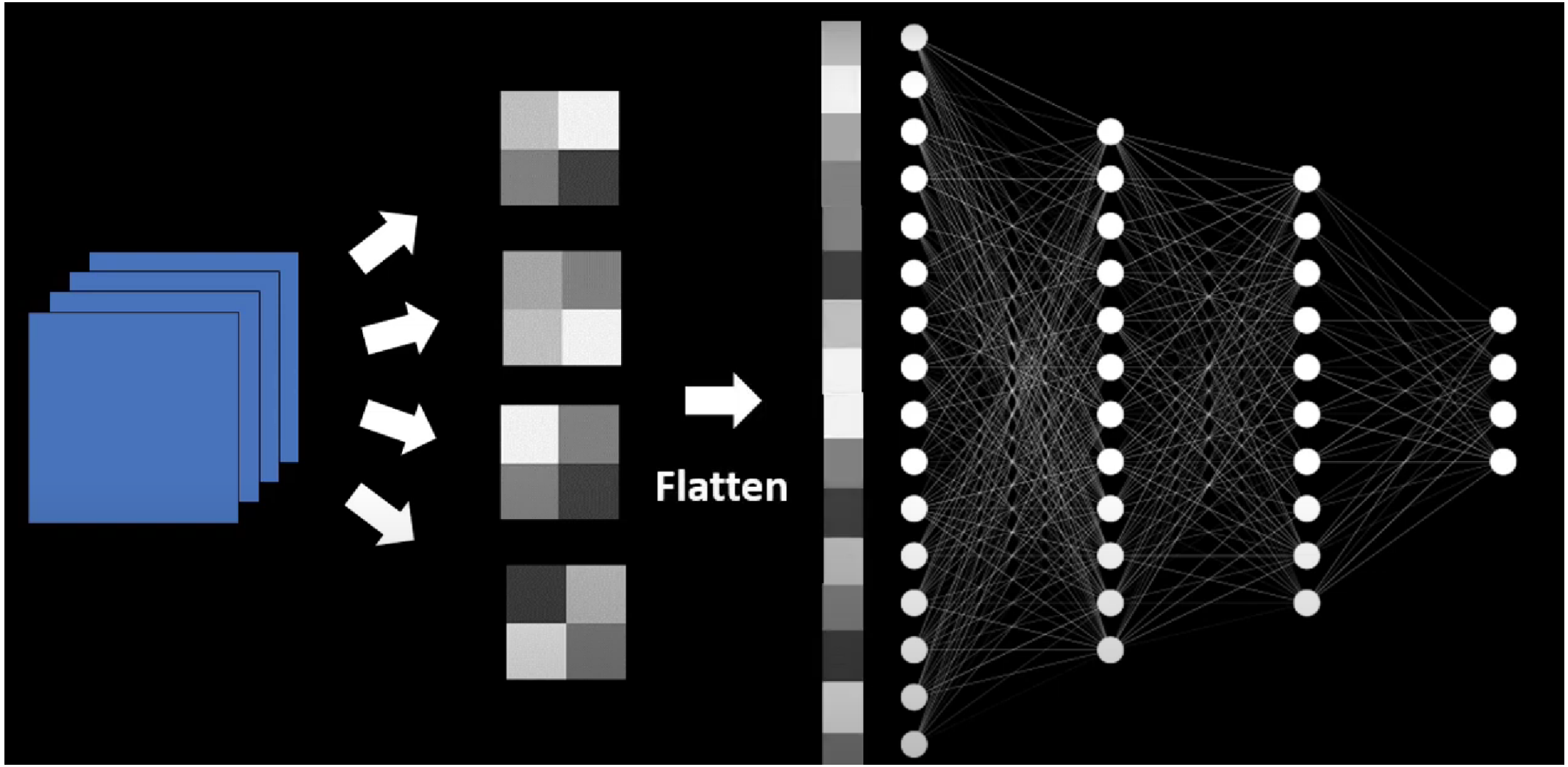
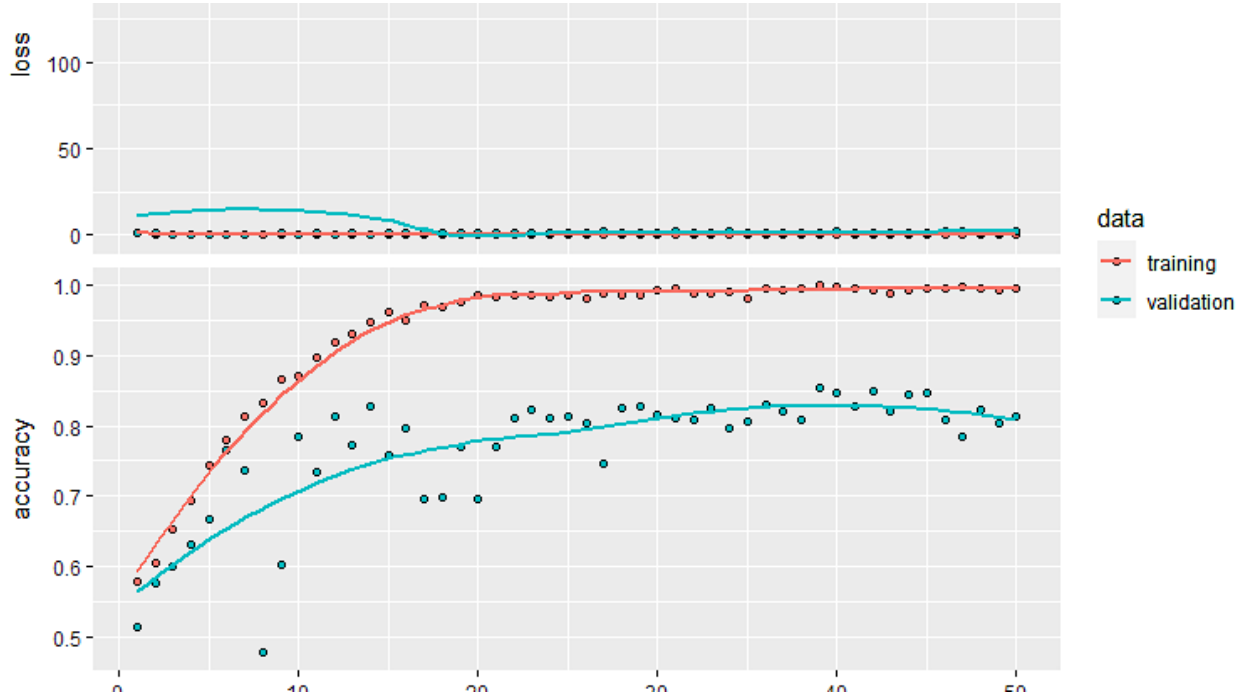


Image source: https://www.youtube.com/watch?v=rxSmwM7z0_4&list=PLvr2XuTk0zOJs9zYoECbGZMfKONVE_5nh&index=12&t=95s

Convolutional Neural Networks (CNNs)

Operations: Classification

- Flatten the pooled feature map into a long vector.
- Input the vector into our fully connected ANN.
- Measure Accuracy of prediction over a number of epochs.



Data Augmentation

NN require a large number of data for training an effective model.
Data Augmentation can mitigate this problem.

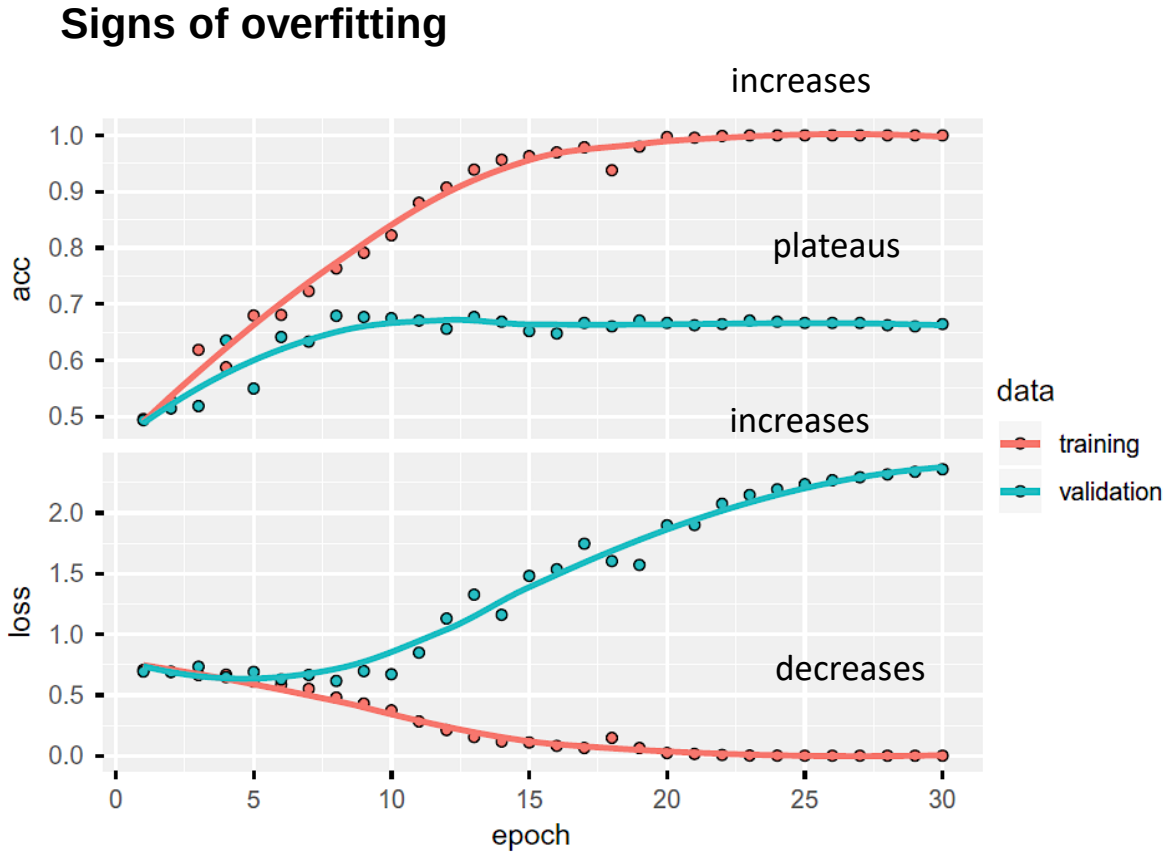
(a) General:

- o Mirroring, Shearing, Randomly cropping;
- o Rotating the image, etc.

b) Shifting the RGB color range of the image.

c) Applying PCA for color augmentation.

In keras, data augmentation is performed by the **`image_data_generator()`** function.



Traditional CNN Architecture

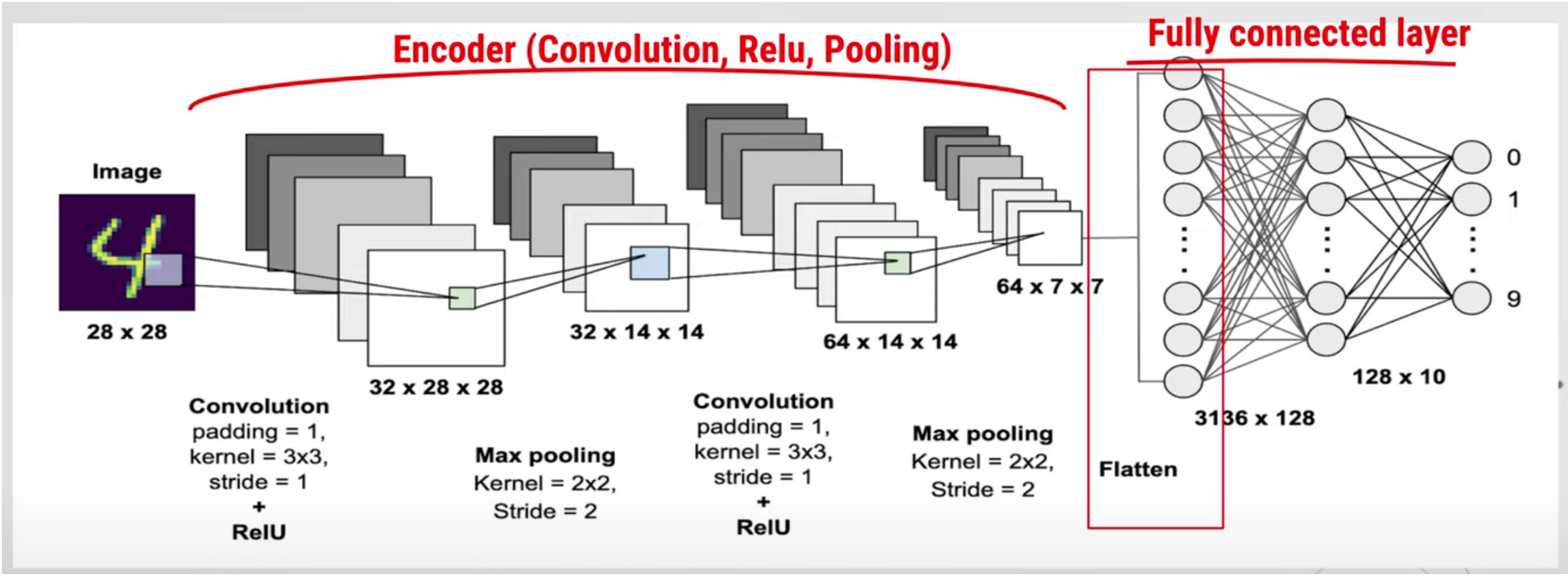
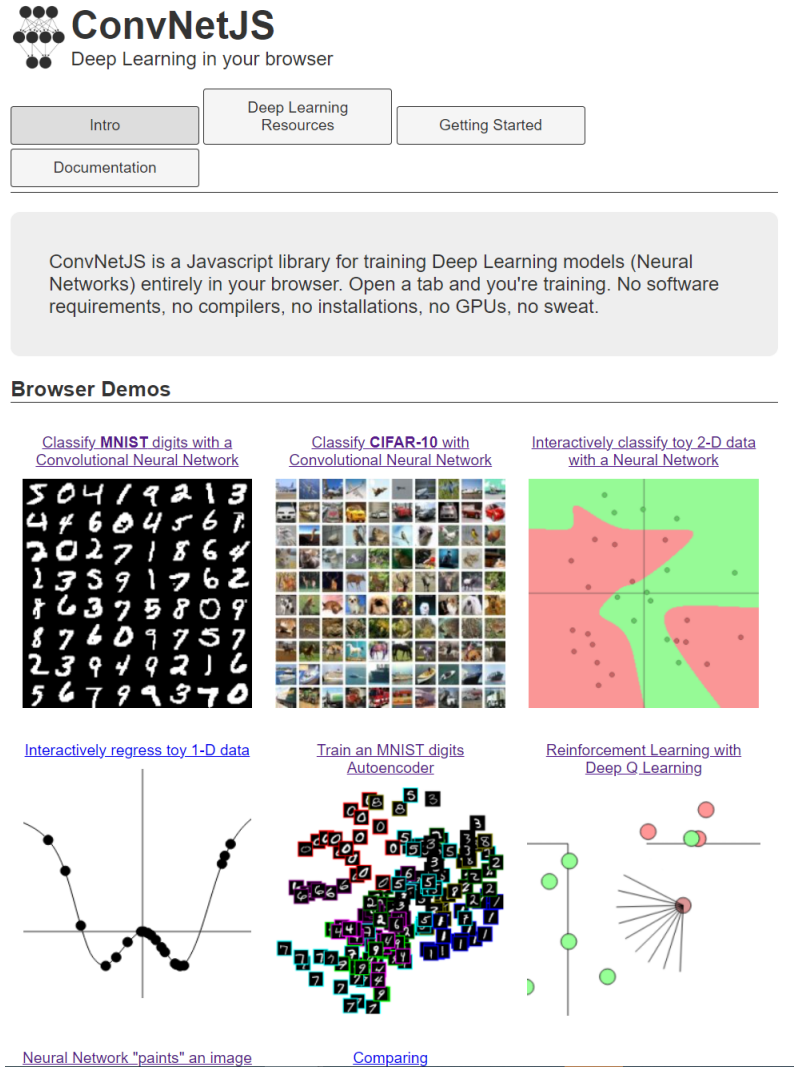
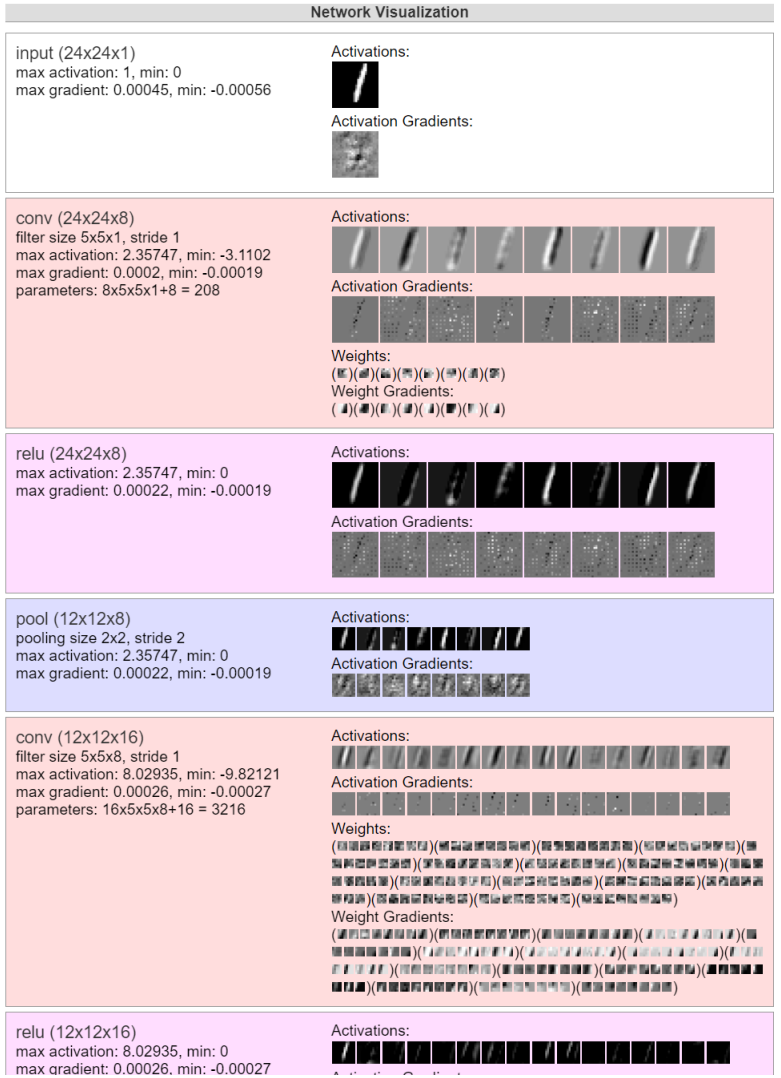


Image Source: https://www.youtube.com/watch?v=Ahge3GzQ3Kg&list=PLvr2XuTk0zOJs9zYoECbGZMfKONVE_5nh&index=16&t=1196s

Convolutional Neural Networks (CNNs)



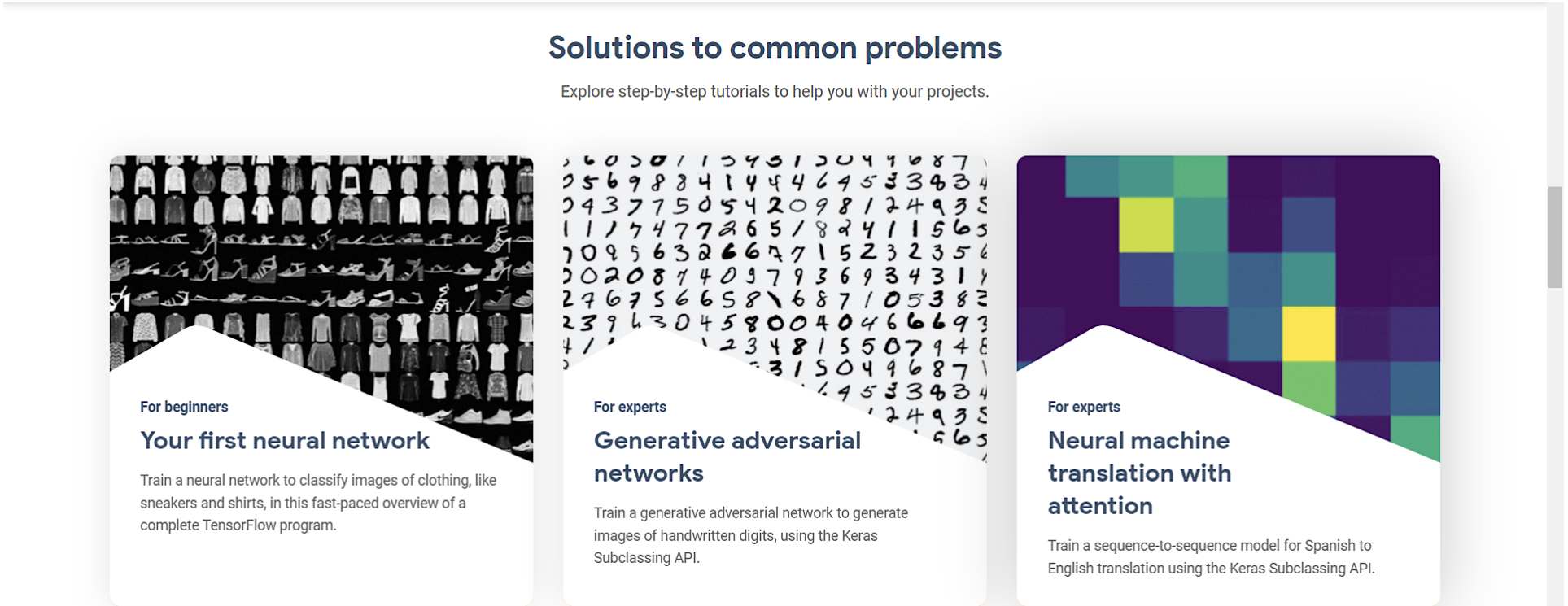


https://cs.stanford.edu/people/karpathy/convnetjs/index.html

R interface to Keras & Tensorflow

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow.

https://www.tensorflow.org/



Examples of Pretrained CNN: ResNet50

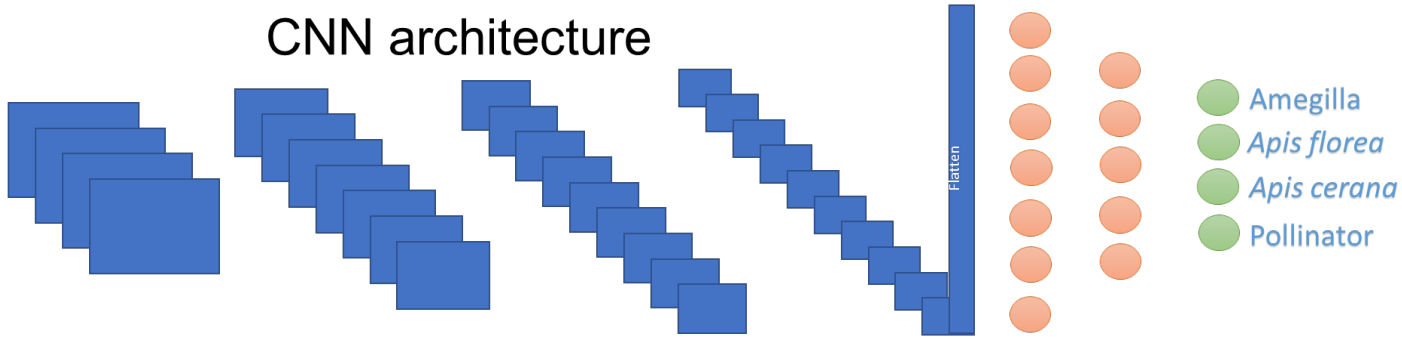
- Keras already contains some pretrained CNNs.
- ResNet-50 is a CNN that is 50 layers deep.
- You can load a pretrained version of the network trained on more than a million images from the ImageNet database.
- The pretrained network can classify images into 1000 object categories.

Other pretrained CNNs:

- Inception V3
- MobileNet
- EfficientNet

Example: CNN for tropical pollinator identification

Image Pre-Processing → Data Split and Validation → Model Architecture → Hyper-parameter tuning → Model testing and Predictions



Low Level Processing:
- Image Cropping
- Quality Filtration
- **Total**: 637 images

Model Data Preparation
Training Set: 60%
Validation Set: 20%
Test Set: 20%

Model Implementation:
- Sequential Convolutional Neural Network
- InceptionV3 pre-trained network

Hyperparameter: Optimiser: RMSProp; Kernel: 3; Dropout: 0.2; Normalization: 0.9; Filters: 16, 32, 64, 128, 256, 512; Loss Function: categorical cross-entropy

Object Detection

- A process in computer vision that involves the detection of various objects in digital images or videos
- It seeks to answer two basic questions:
  1. Where is the object? → Object localisation
  2. What is the object? → Classification/Recognition

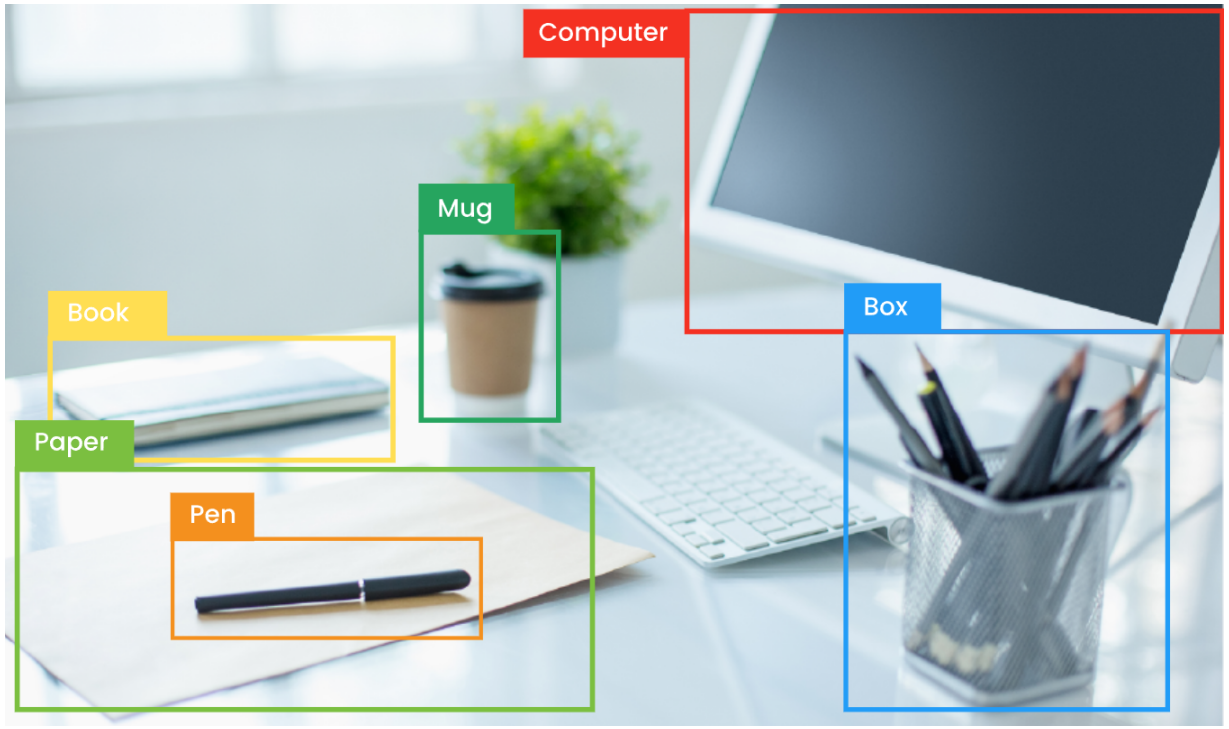


Image Source: https://axonator.com/object-detection-technology

Algorithms for Object Detection

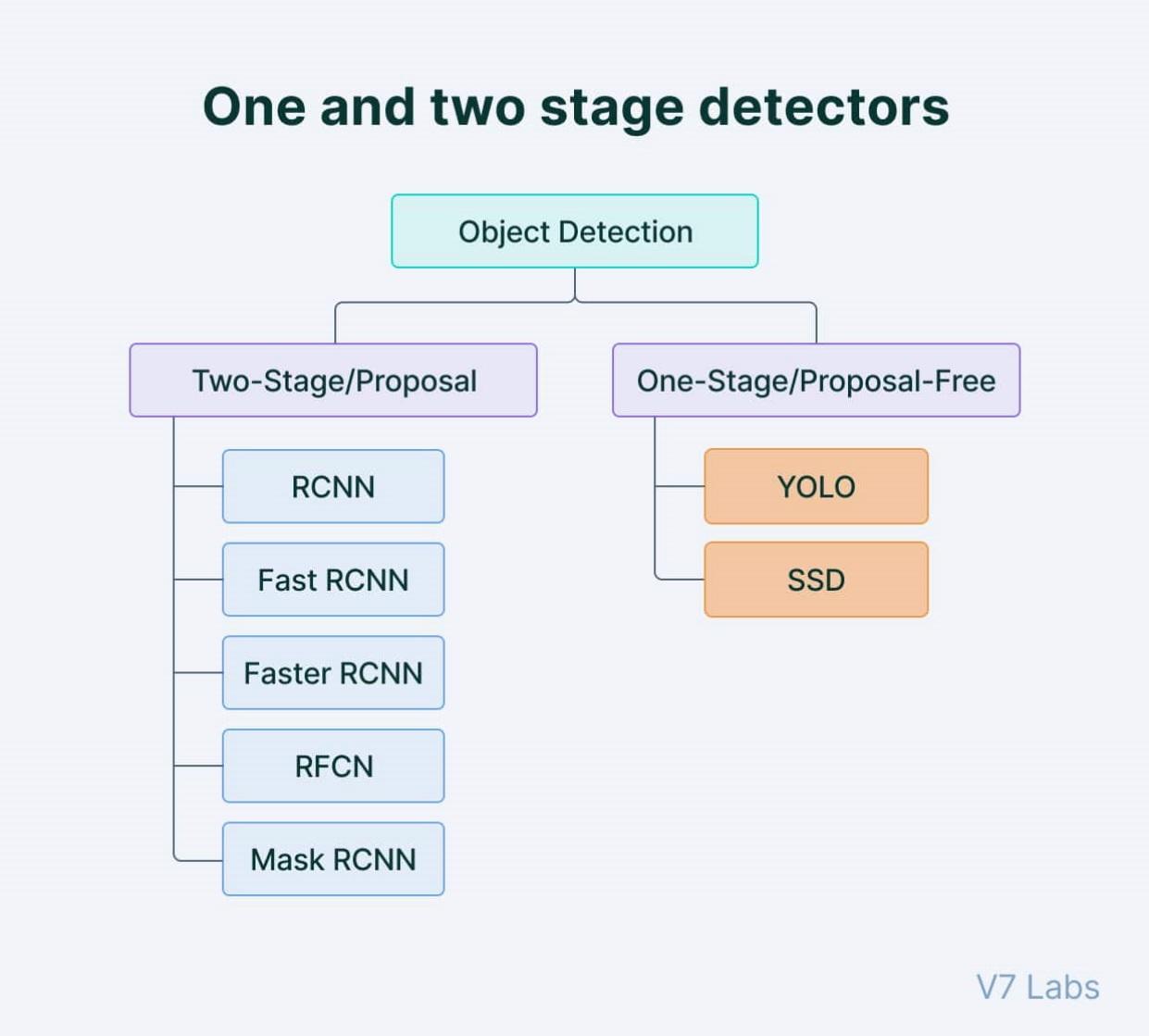


Image Source: https://www.v7labs.com/blog/yolo-object-detection

Object Detection using YOLO

"You Only Look Once"

- YOLO algorithm employs CNN to detect objects in real-time.
- First launch in 2015 by Joseph Redmond
- The CNN is used to predict various class probabilities and bounding boxes simultaneously.
- It requires only a single forward propagation through a neural network to detect objects.
- This means that prediction in the entire image is done in a single algorithm run

- Some YOLO variants:
- YOLOv3
- YOLOv5
- YOLOv7
- YOLOv8….YOLOv11

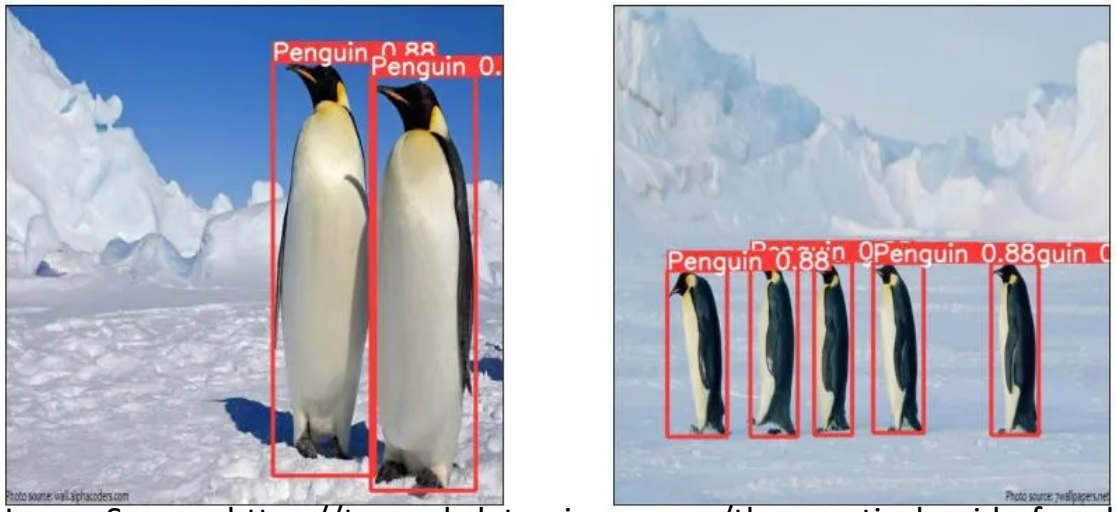


Image Source: https://towardsdatascience.com/the-practical-guide-for-object-detection-with-yolov5-algorithm-74c04aac4843

YOLO Model

- YOLO models detection as a regression problem.
- It divides the image into an S × S grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities.
- These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

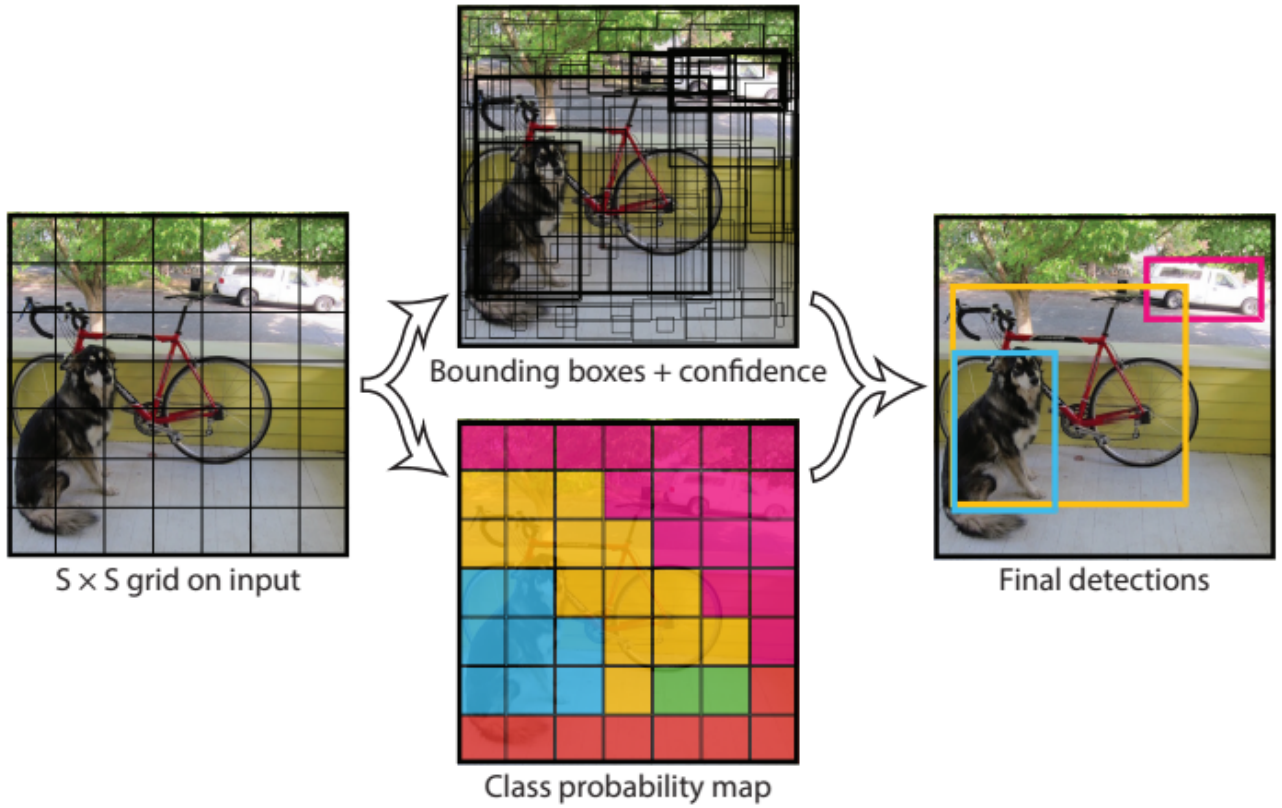
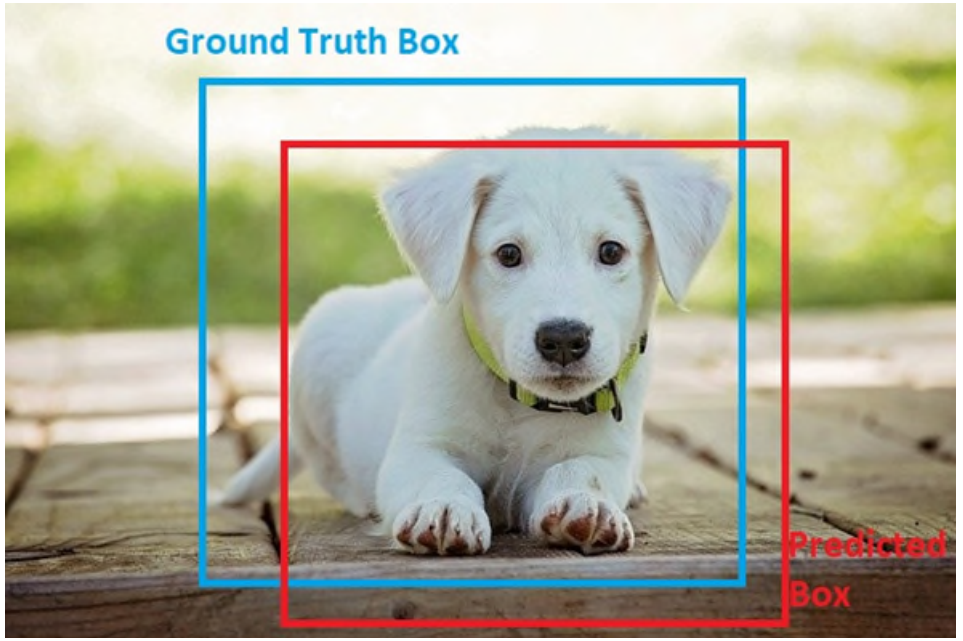


Image Source: https://arxiv.org/pdf/1506.02640.pdf

Intersection over Union (IoU)

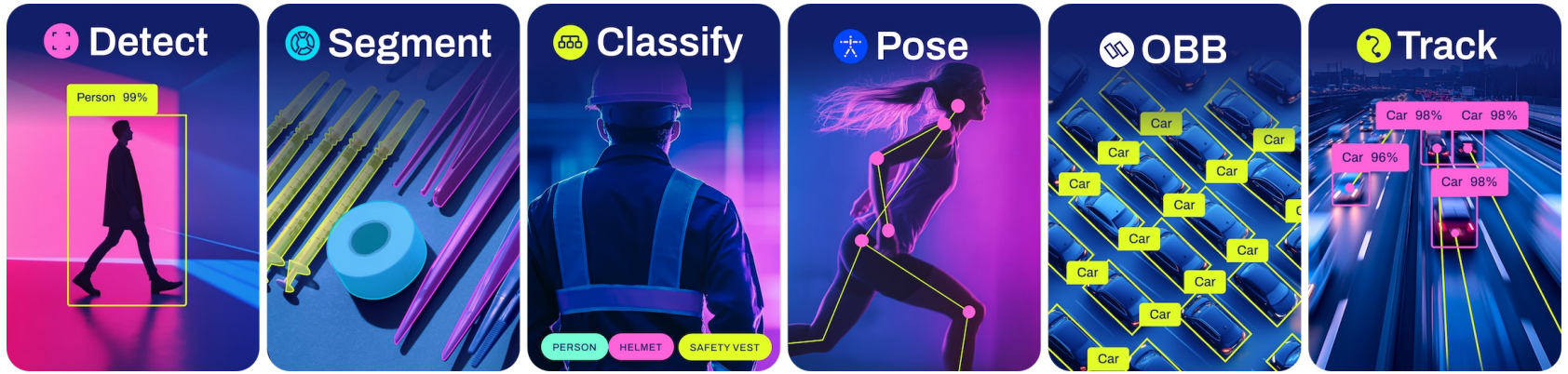
$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} =$$



https://www.v7labs.com/blog/yolo-object-detection

https://debuggercafe.com/evaluation-metrics-for-object-detection/

YOLO Applications



https://docs.ultralytics.com/tasks/

- Open-source tool
- Earlier versions were coded in C (versions 1-4)
- Now coded in Python using PyTorch framework
- Trained on COCO dataset (Common Objects in Context)
- Real-time Object Detection ++
- Used in various applications such as autonomous vehicles, security and surveillance, and medical imaging, wildlife, etc.
- Can be exported to different format like ONNX, CoreML, TF.js, TF Lite, etc.

YOLOv8 in AgriFood applications

Application of YOLOv8 for identification of asparagus “tip rot”

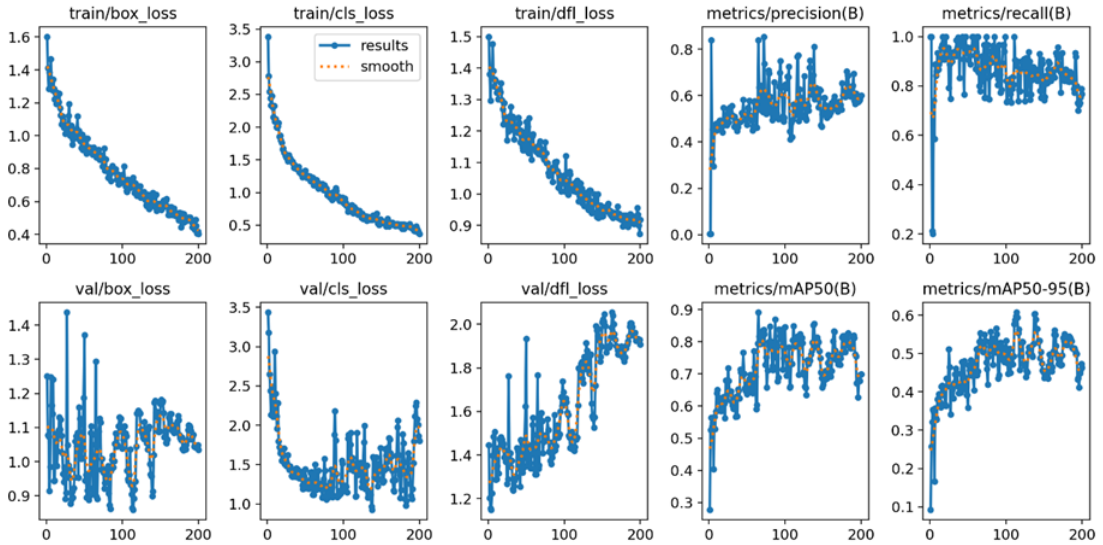
Data Preparation

- Object Detection

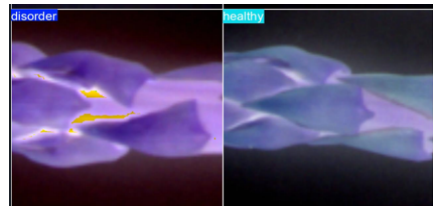
- Instance segmentation

- Image classification

Model(s) Training



Results



CASE STUDY: Automated Solutions for Monitoring Pollinators

Dr Maria Anastasiadi

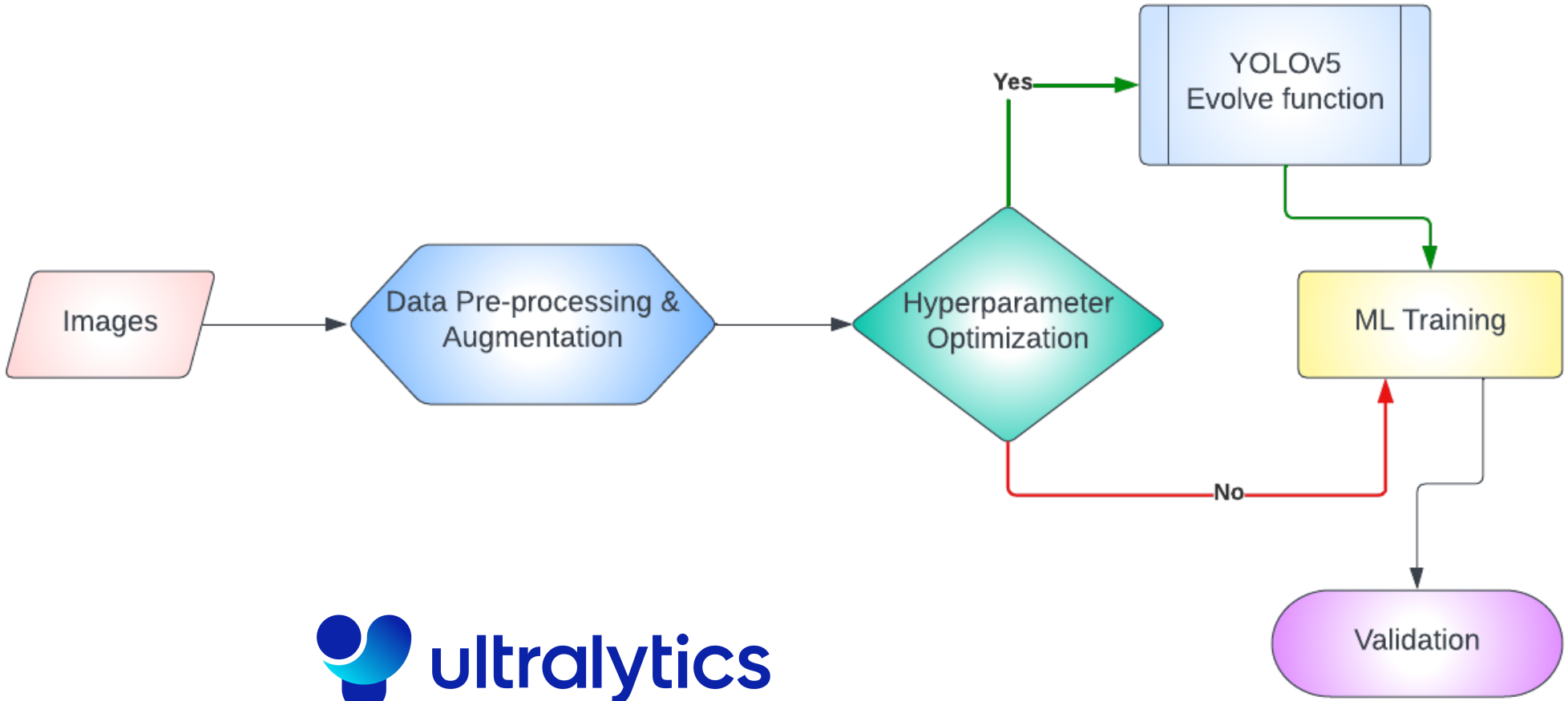
m.anastasiadi@cranfield.ac.uk

www.cranfield.ac.uk

Background

- Insect pollinators, provide critical pollination services to crops by improving yield and quality
- However, pollinator populations are declining worldwide
- Moreover, tropical regions are understudied
- Collecting data on pollinators is tedious and time-consuming
- New and effective ways of assessing pollinator activity are needed

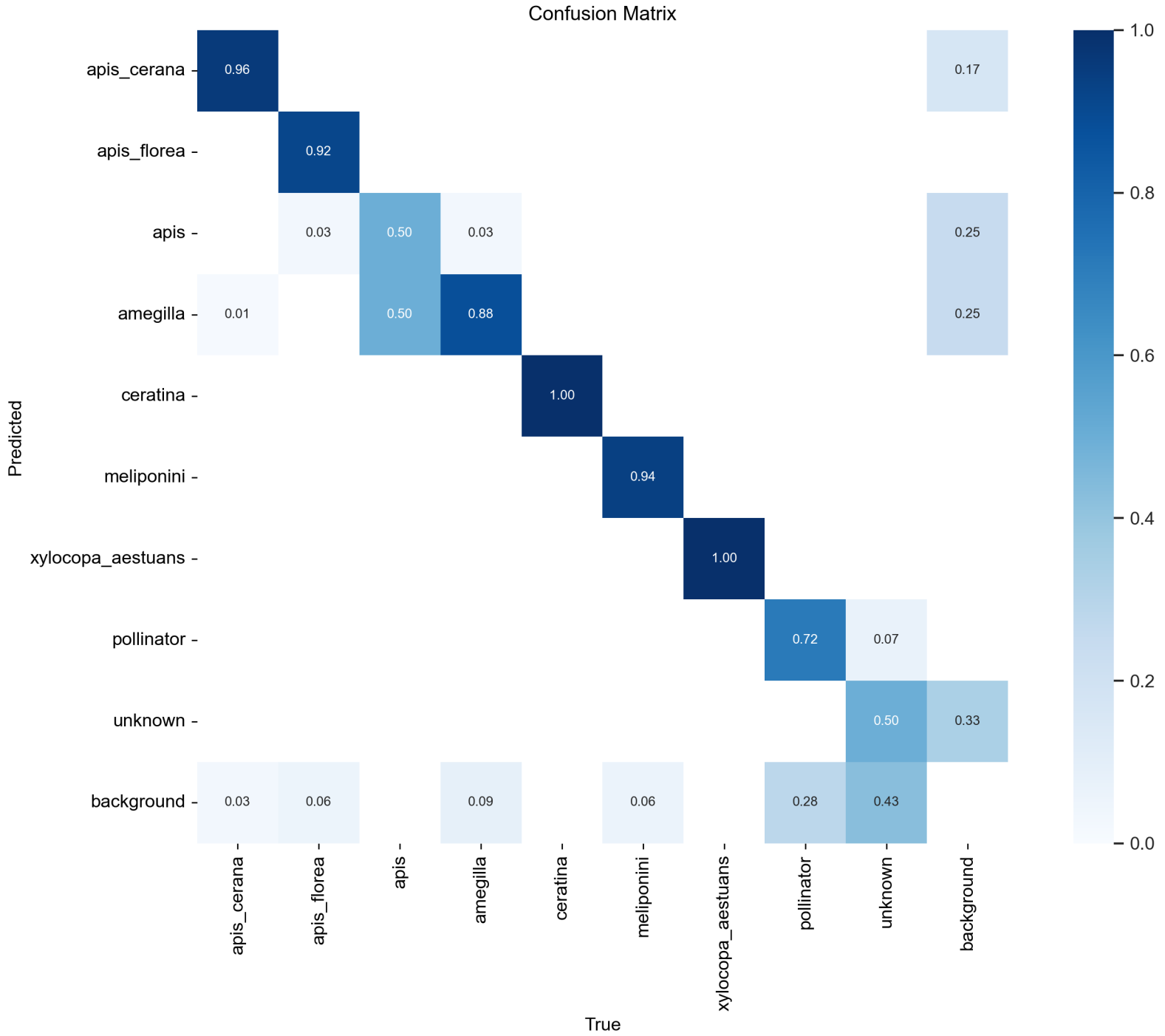
Build an object detection model for pollinator activity estimation



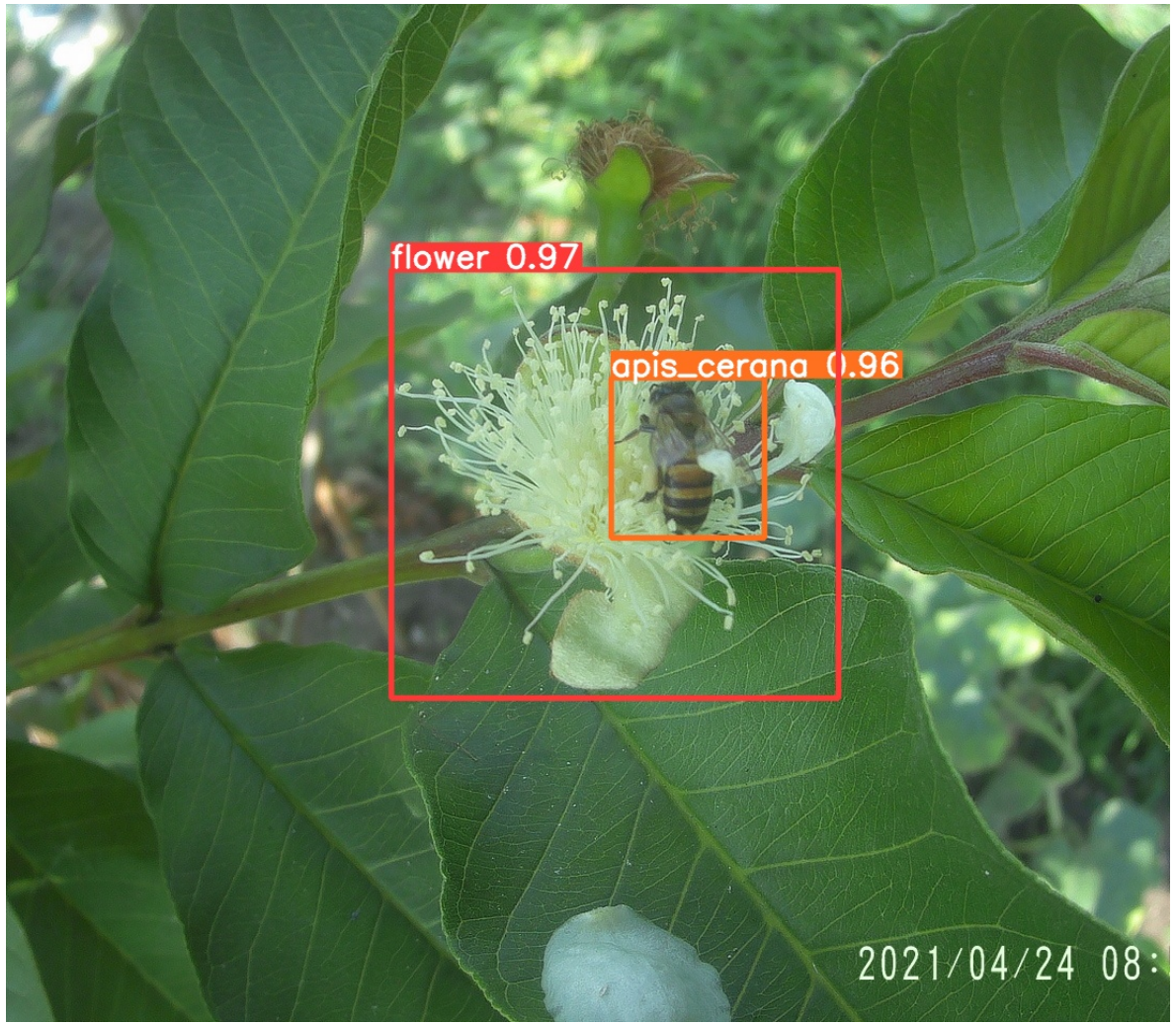
ultralytics
YOLOv5

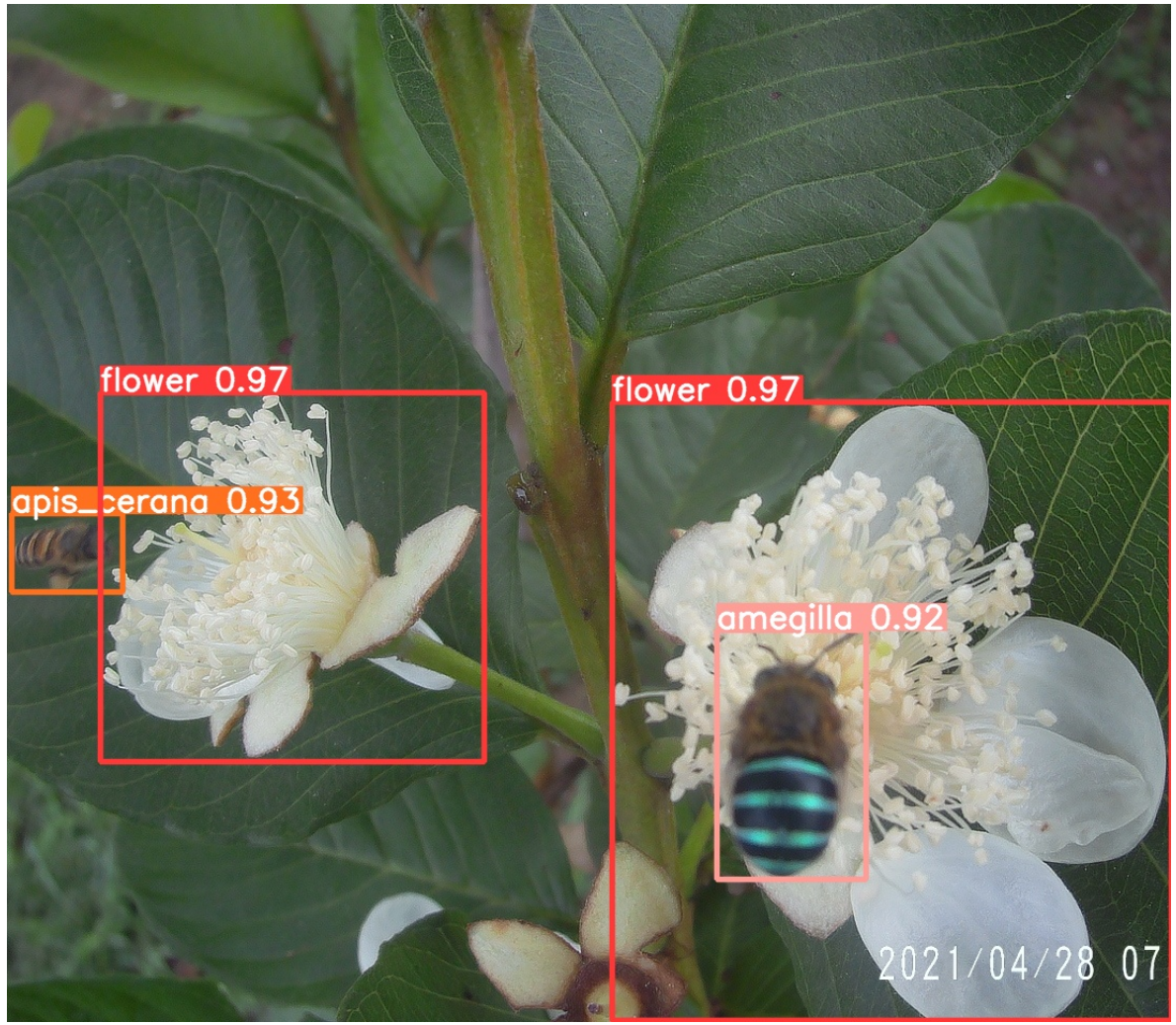
Results

- The Confusion Matrix shows the Accuracy of YOLOv5 model per species/taxa.
- Results are based on an independent test set and for most classes the accuracy is >90%



YOLO for pollinator monitoring





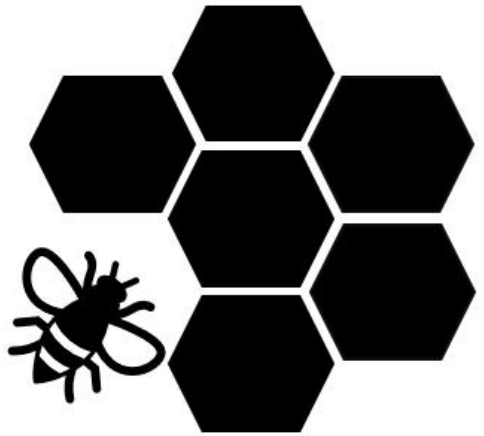
Tracking Algorithms - Implementation

Assigning unique IDs to pollinators helps estimate pollinator abundance

BeeTect: A Python Graphical User Interface (GUI) for automatic image processing and statistical analysis

BeeTect Features

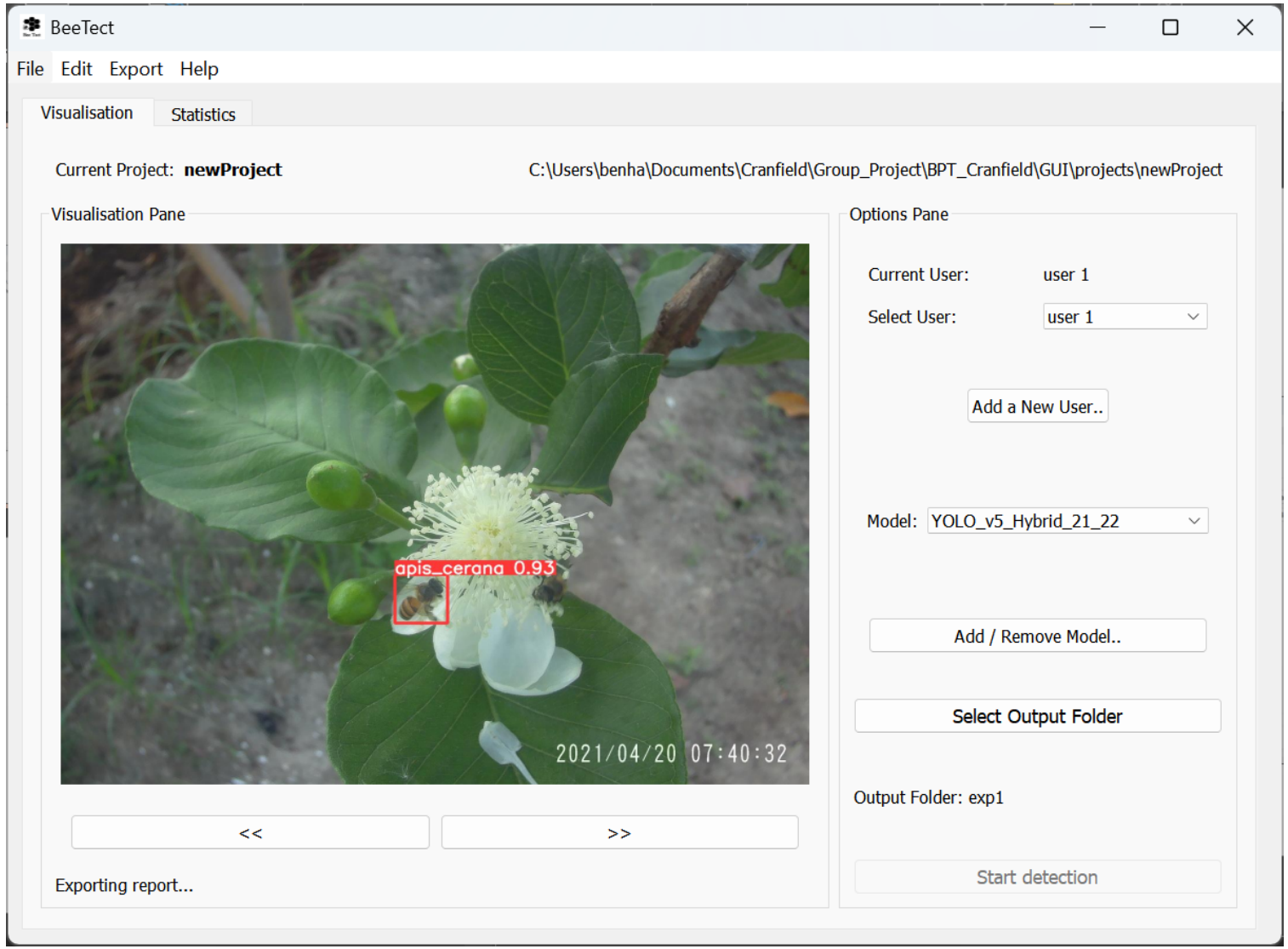
- Incorporates customised YOLO models to accommodate user requirements.
- Automatically processes multi-image and video data.
- Performs Exploratory Data Analysis and Formal Statistical Analysis.
- Exports detailed reports per project.
- Can be easily expanded to include further capabilities.



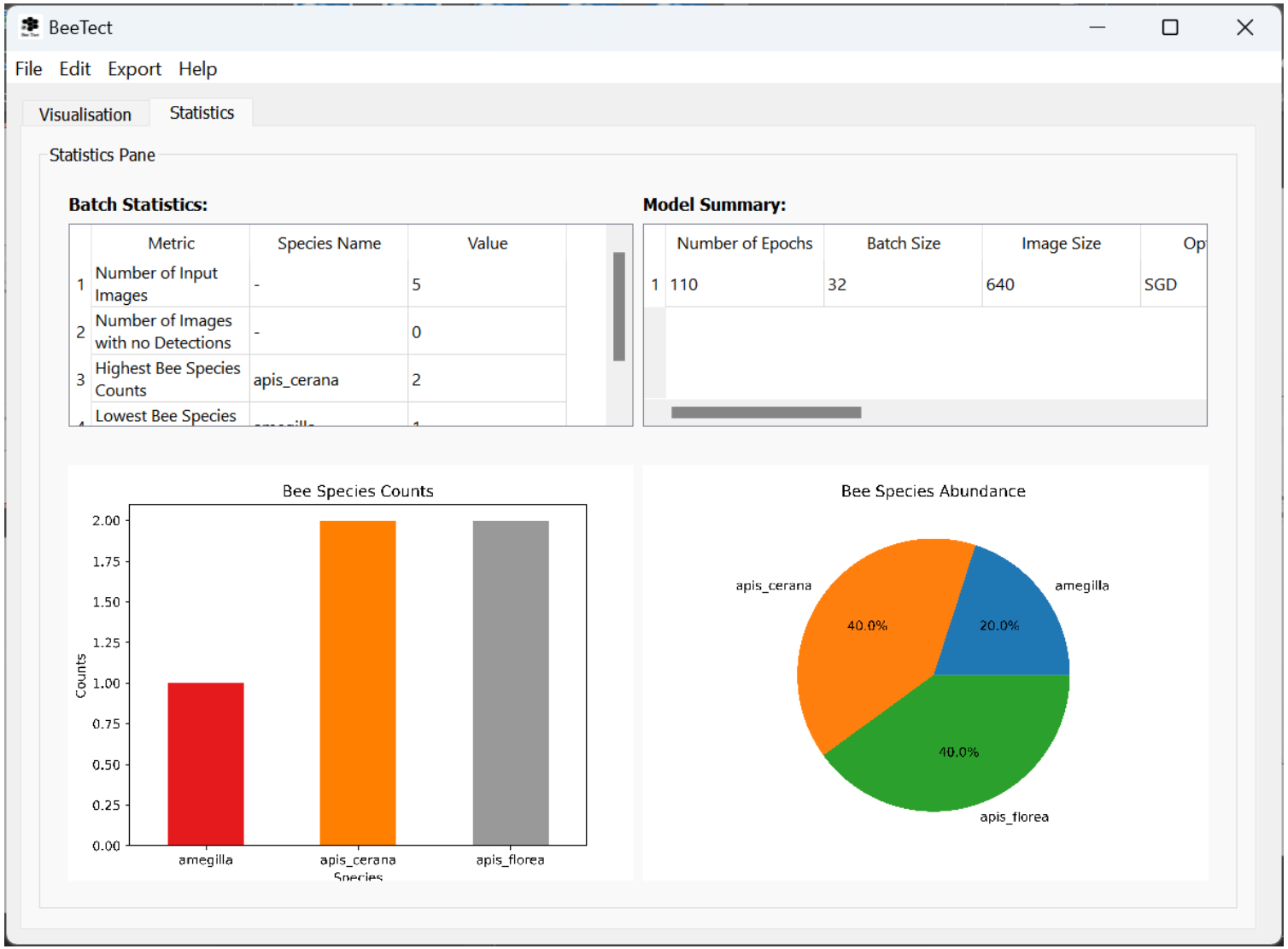
Bee Tect

PyTorch

BeeTect Visualization Pane



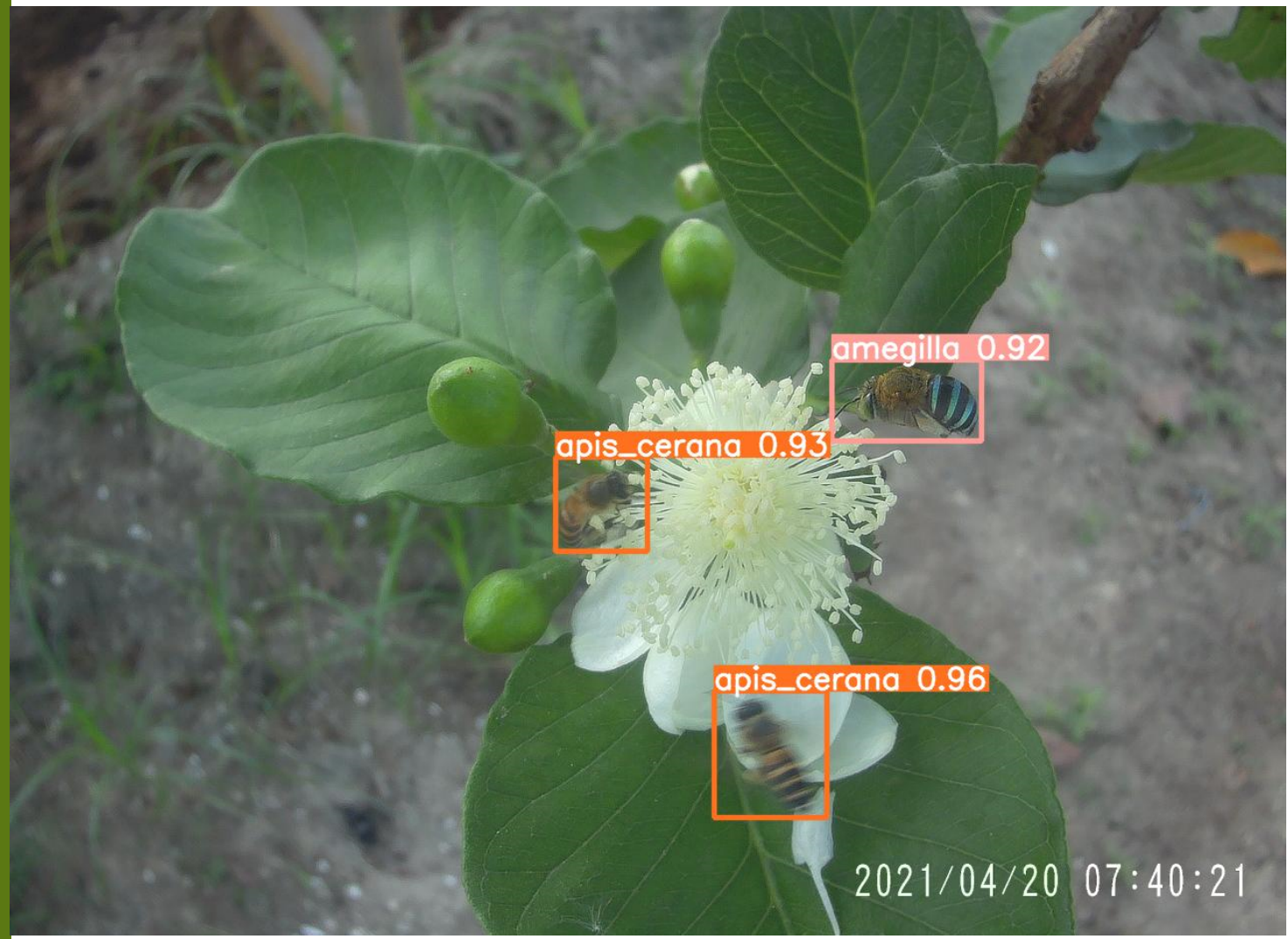
BeeTect Statistics Pane



GUI Statistics Pane.

Future Directions - BeeTect

- Implement video tracking to be able to quantify bee visitations.
- Incorporate models for different pollinators and geographical areas.
- Expand statistical and modelling capabilities.



Summary

- Applications of image analysis in Bioinformatics;
- Basics of Computer Vision;
- Image Manipulation Techniques:
    - Background subtraction;
    - Image Segmentation;
    - Thresholding;
    - Image Enhancement;
    - Noise Reduction;
    - Edge Detection.
- Basic Principles of CNNs;
- Examples of pretrained CNNs.
- Case Study: Pollinator Monitoring

www.cranfield.ac.uk

T: +44 (0)1234 750111

@cranfielduni

@cranfielduni

/cranfielduni