JC3504 Robot Technology

Lecture 7: Computer Vision (1)

Dr Xiao Li xiao.li@abdn.ac.uk

Dr Junfeng Gao junfeng.gao@abdn.ac.uk

Outline

- Images as Signals
- Image Feature Detection
- Semantic Feature Detection
- Convolutional Neural Network (CNN)

Images as Signals

Images as Signals

Vision is one of the most information-rich sensor systems both humans and robots have available.

However, efficiently and accurately processing the wealth of information that is generated by vision sensors is still a key challenge in the field.

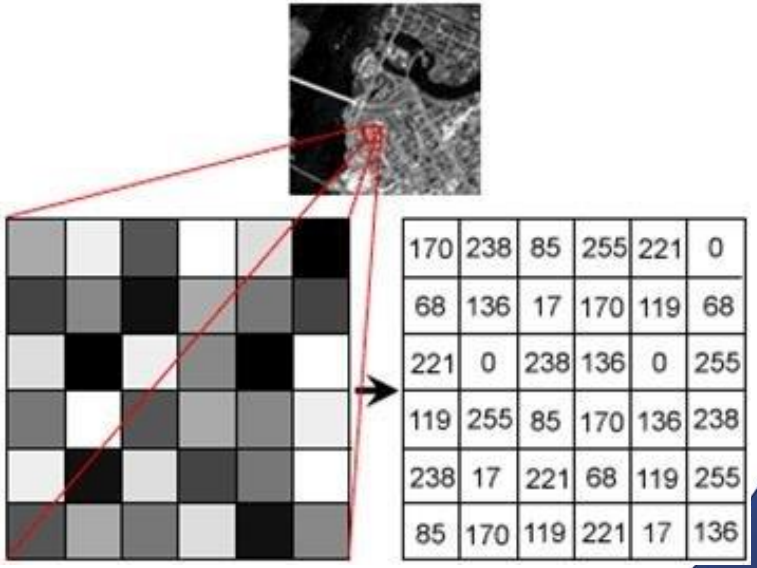
Images are captured by cameras containing matrices of charge-coupled devices (CCD) or similar semi-conductors (e.g. complementary metal–oxide semiconductor, CMOS) that can turn photons into electrical signals.

Grayscale Images

A grayscale image can be viewed as a 2D matrix where each element of the matrix corresponds to a single pixel in the image.

The value of each element (pixel) represents the intensity of the light at that point, with lower values indicating darker shades and higher values indicating lighter shades.

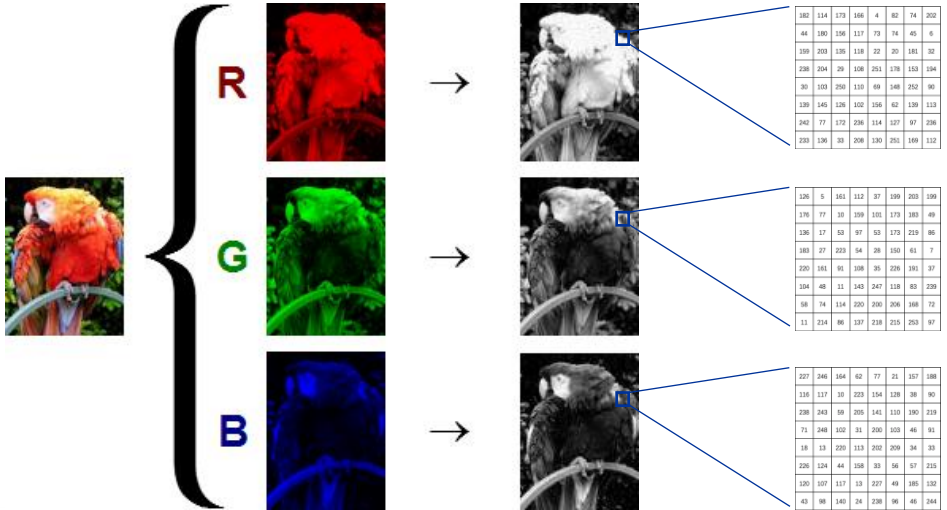
Grayscale values typically range from 0 (black) to 255 (white) for 8-bit images, which are common in digital imaging. NB: in deep learning approaches, the grayscale values are 0.0 to 1.0 usually.



Colour Images

Colour images have a more complex structure and can be represented as a 3D tensor. This tensor has three dimensions: the height and width of the image and a third dimension that represents colour channels.

In most colour images, there are three channels: Red (R), Green (G), and Blue (B). Each channel is itself a 2D matrix (the same size as the grayscale image), and the three matrices together form the 3D tensor.



Colour Images with Additional Channels

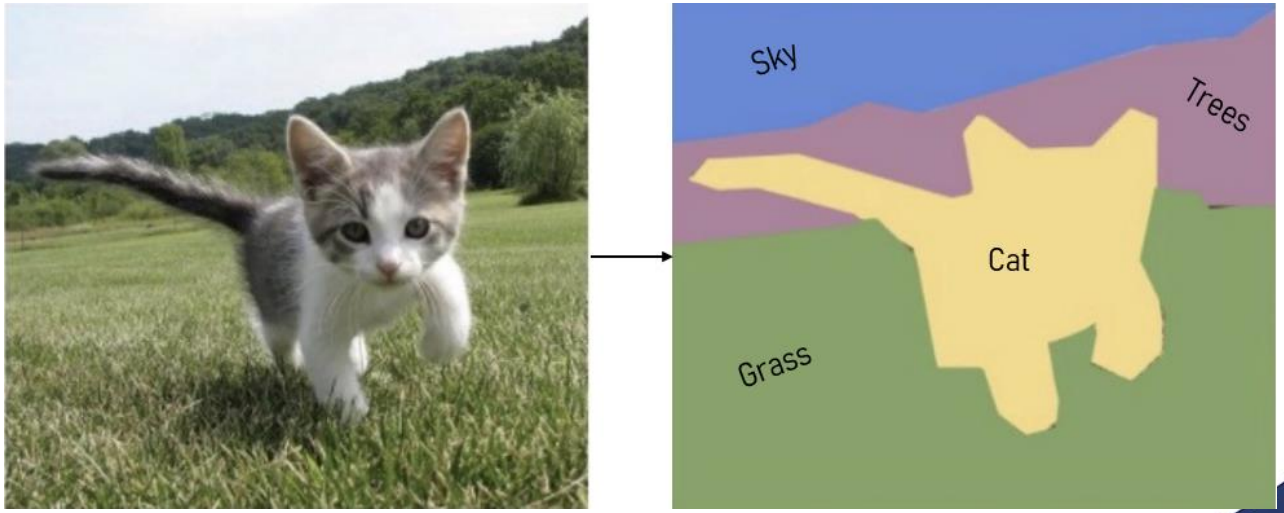
In some situations, images can indeed have additional channels found in colour images.

These extra channels are used to encode supplementary information for each pixel that can be critical for specific tasks, such as semantic segmentation, object detection, and scene understanding.

Segmentation Channels

This can be particularly useful in tasks like semantic segmentation, where the goal is to classify each pixel of an image as belonging to a particular category.

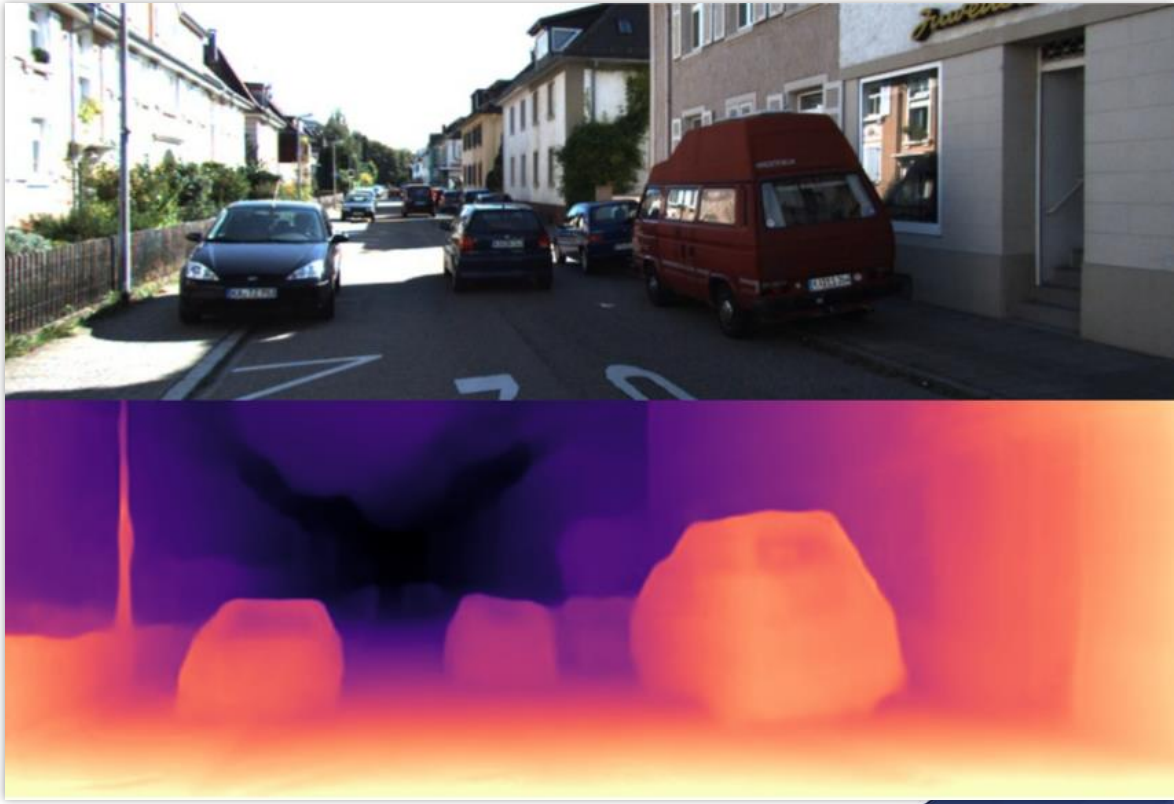
In this case, the additional channel consists the classification indexes for each pixel



Depth Channels

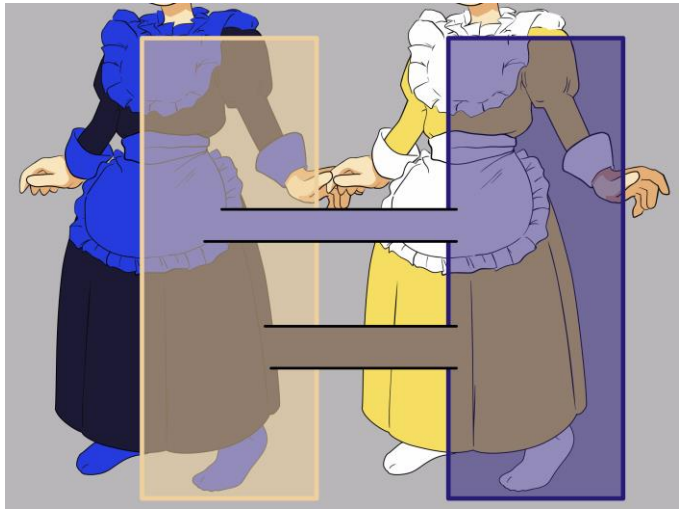
Depth channels provide information about the distance of objects in the scene from the viewpoint of the camera.

This information is crucial for understanding the 3D structure of a scene and for tasks that require spatial understanding, such as obstacle avoidance in robotics, 3D reconstruction, and augmented reality applications.

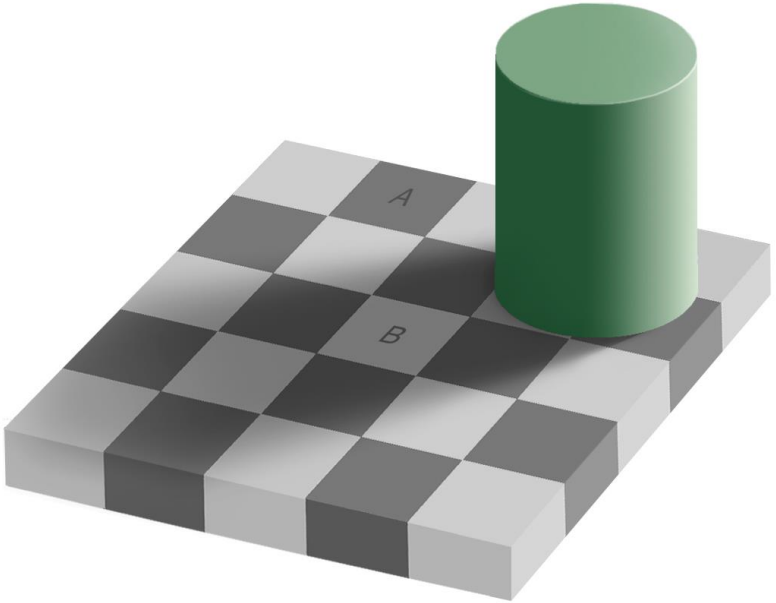


Human vs. Computer Colour Perception

Humans perceive colours in a context-sensitive manner, where our understanding of colour can be influenced by surrounding colours, lighting conditions, and other environmental factors.



Blue and black / White and gold



Human vs. Computer Colour Perception

Humans perceive colours in a context-sensitive manner, where our understanding of colour can be influenced by surrounding colours, lighting conditions, and other environmental factors.

Computers, on the other hand, interpret colours in a photograph based on fixed numerical values assigned to each pixel, without the context or subjective interpretation that affects human vision.

This objective approach means computers cannot be “fooled” by optical illusions in the same way humans can. This results in the **information that a computer extracts from an image being fundamentally different from that of a human being**.

Types of Image Feature

Image features, the critical elements used for interpreting and analysing visual data, encompass a spectrum from low-level to high-level representations.

- **Low-level features**, such as edges, corners, and textures, are derived directly from the raw pixel data and represent fundamental visual attributes like brightness, colour, and gradient. These features are foundational for image processing and are often used for initial tasks like segmentation, edge detection, and basic object recognition.
- **High-level features** are complex and **semantic** information about the image, emerging from the combination and abstraction of low-level features, encapsulating. Examples include the presence of specific objects, scenes, or activities.

Image Feature Detection

1495 UNIVERSITY OF ABERDEEN

Image Feature Detection

The detection of image features across different levels—ranging from simple, low-level features to complex, high-level features—necessitates diverse methodologies tailored to the intricacies of each feature type.

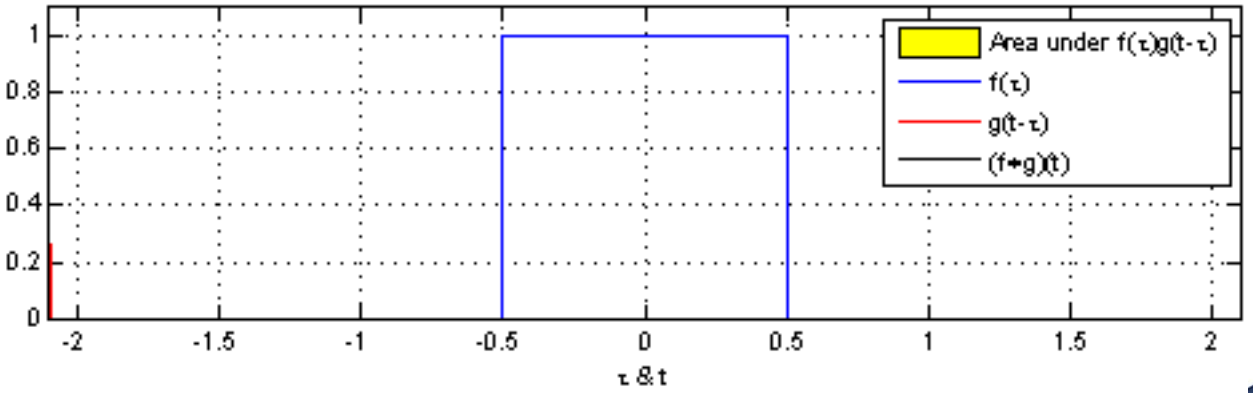
- Low-level features, such as edges, corners, and textures, are typically detected using algorithms that analyse the image's pixel intensity values and gradients. Methods like the Sobel edge detector, Harris corner detector, and Gabor filters for texture analysis are classic examples that operate directly on the raw pixel data to extract fundamental visual cues.
- The higher-level features often require the use of machine learning and deep learning models, such as convolutional neural networks (CNNs), which can learn from large datasets to identify and extract features that are meaningful within the context of specific tasks or applications.

Convolution

Convolution operation is the most used operation in image processing (and even signal processing).

Convolution is a mathematical operation on two functions f and g that produces a third function $f * g$, where g is called the filter or the kernel.

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\tau)g(x-\tau)d\tau$$

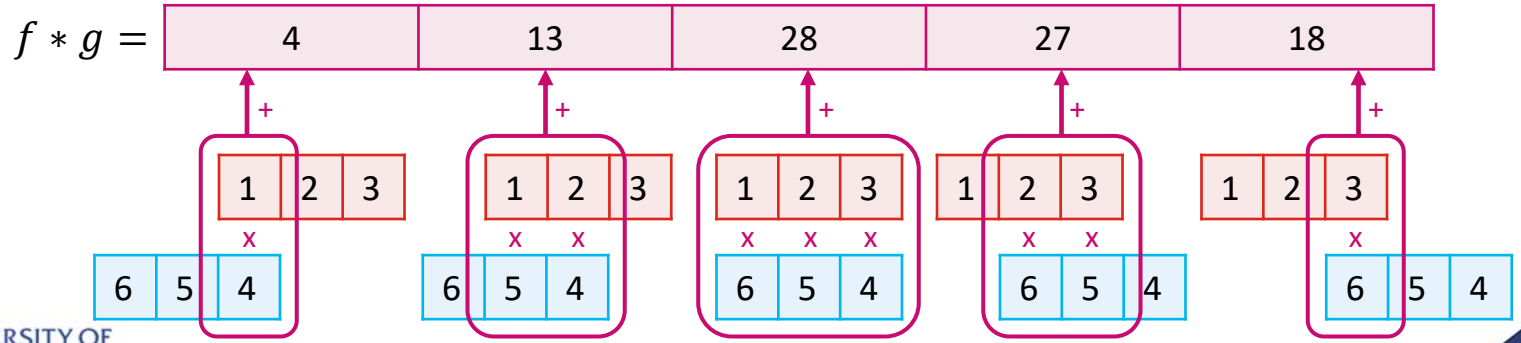


Discrete Convolution

As images are discrete signals, the convolution is consequently discrete:

$$(f * g)[x] = \sum_{i=-\infty}^{\infty} f[i]g[x-i] = \sum_{i=-\infty}^{\infty} f[x-i]g[i]$$

Suppose $f =$ 1 2 3 $g =$ 4 5 6



2D-Discrete Convolution

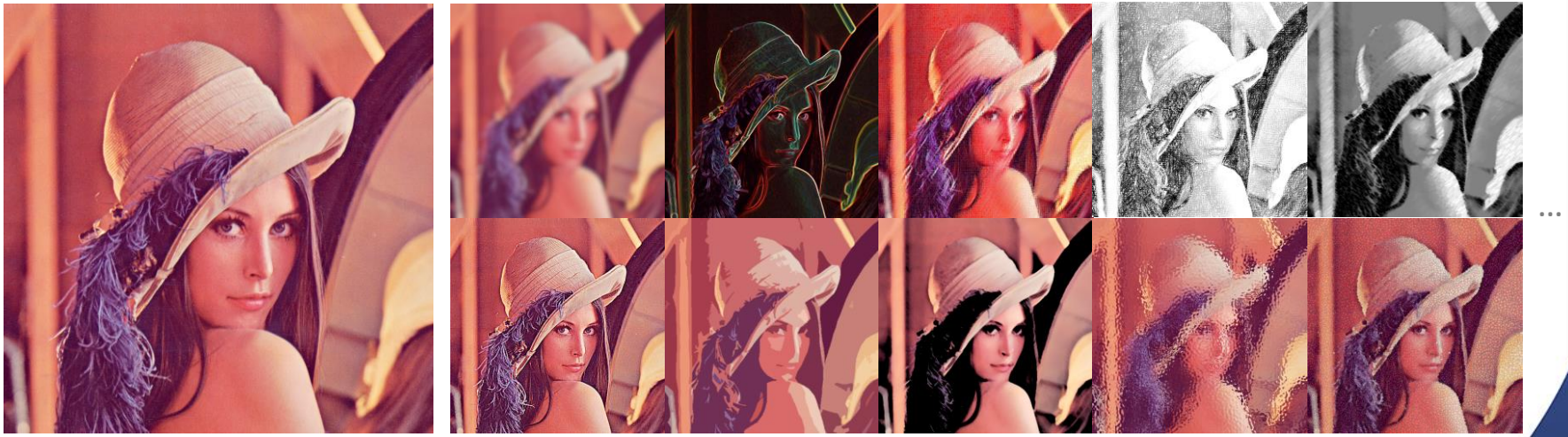
Additionally, given that images are two-dimensional signals, the convolution is two-dimensional as well:

$$(f * g)[x, y] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f[i, j] g[x - i, y - j] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f[x - i, y - j] g[i, j]$$

Although we have defined the convolution from $-\infty$ to ∞, both images and kernel are usually finite (filters are usually much smaller than the image).

Most of the image operations are based on the 2D discrete convolution, by apply the different kernels.

Basic Image Filtering Operations



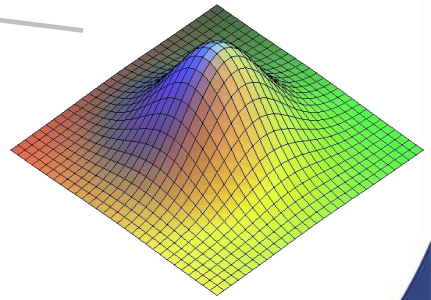
Lena Forsén

Photographed in 1960

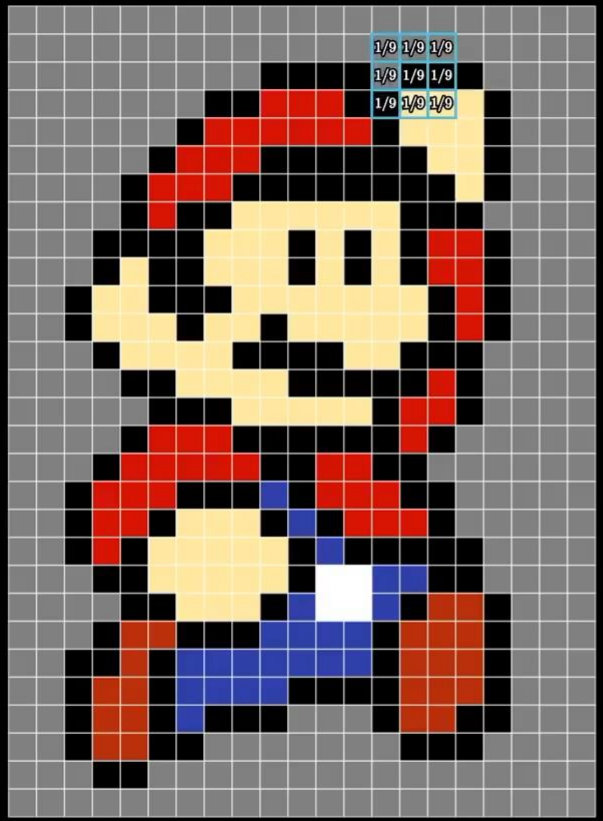
Gaussian Smoothing

When the 2D-kernel is shaped like the Gaussian bell function, the kernel is called Gaussian kernel. E.g.:

$$g(x,y) = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



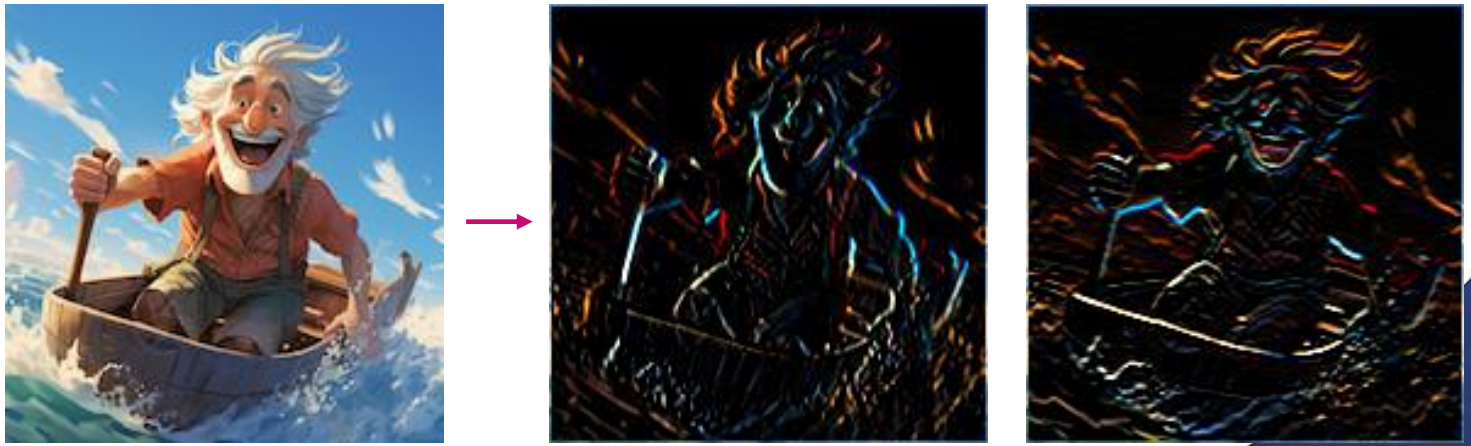
When g is applied to an image as f, a Gaussian blur is applied to the image.


1/9 1/9 1/9
1/9 1/9 1/9
1/9 1/9 1/9

Edge Detection

Edge detection can be achieved using another convolution-based filter, the Sobel kernel:

$$s_x(x,y) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad s_y(x,y) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$



Sharpen

A sharp kernel works by emphasizing the contrast between adjacent pixels in an image, effectively highlighting edges and fine details by subtracting a fraction of the adjacent pixel values from the central pixel value.

$$s_x(x, y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Semantic Feature Detection

Extracting semantic information (e.g. high-level features) from images involves analysing and interpreting visual content to understand the contextual and conceptual elements represented, such as identifying objects, scenes, and activities.

This process typically utilizes computer vision and machine learning techniques.

These machine-learning models can recognize patterns, shapes, and textures, and categorize and annotate images with high-level concepts, enabling applications such as image recognition, scene understanding, and content-based image retrieval, thereby bridging the gap between raw pixels and meaningful semantic content.

Machine-learning Models

The machine-learning (ML) models always require a training process instead of explicitly programmed for the task.

ML models can be regarded as functions, but the function logic is learned when training.

Given a function f with unknown parameters and training data (i.e. pairs of its inputs and targeting outputs), the training process will find the parameters ϕ for f based on the data.

$$f_\phi = train(f, training\ data)$$

The training is an iterative process. To obtain f_ϕ for high performance, the training process may require advanced structure for f and rich training data.

Machine-learning Models for CV

The machine-learning (ML) models for CV also use convolutional kernels. The core difference is that ML models requires a training process: using original images with the processed images (the target image) to find the aimed kernel(s).

E.g. given a ML model (consisting with an empty kernel), an image, and a target blurred image, the training process could find a Gaussian kernel.

$$train\left(\begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}, \right.$$

,

$$\left.\right) \rightarrow \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Convolutional Neural Network (CNN)

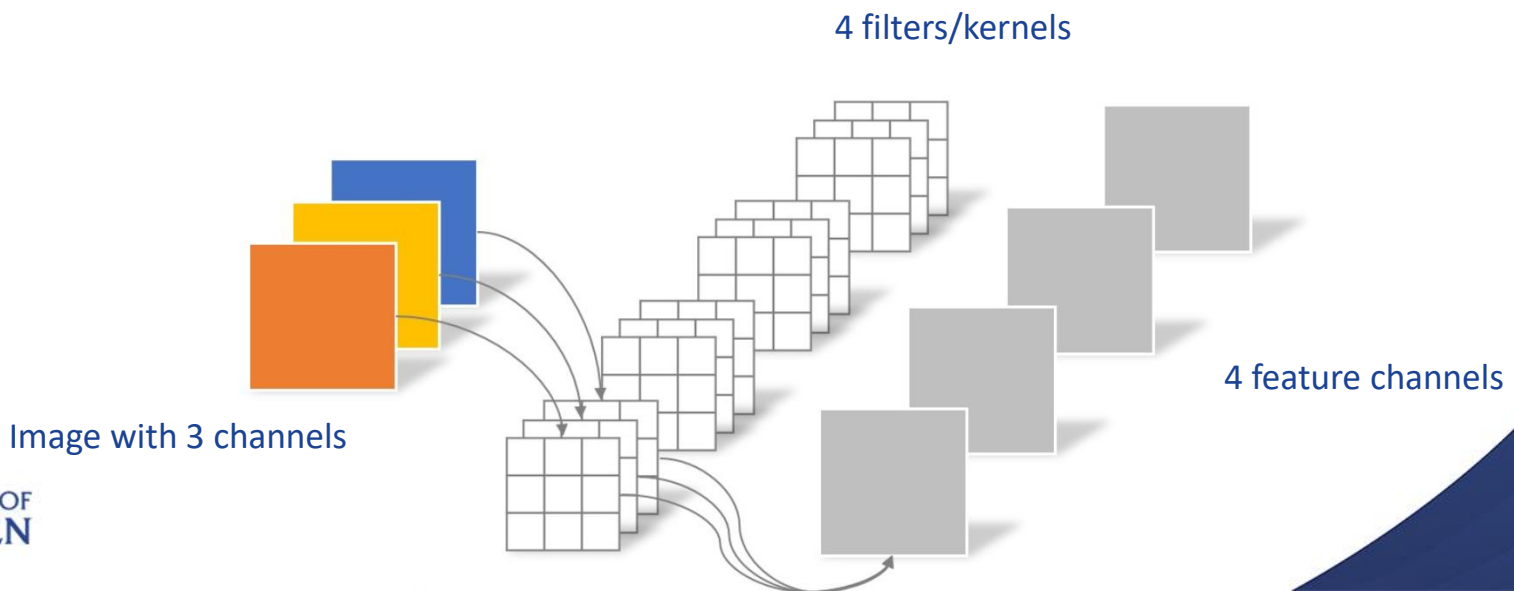
Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are a class of deep learning models that are particularly powerful for tasks involving image recognition, processing, and classification. They mimic the way human vision operates by applying a series of kernels to the input images to extract and learn hierarchical features.

CNNs consist of multiple layers, including **convolutional layers** that perform the filtering, pooling layers that reduce dimensionality, and fully connected layers that make decisions or predictions based on the extracted features.

Convolutional Layer

A convolutional layer uses **a set of** learnable filters to perform convolution operations on the input data, extracting important features such as edges and textures by preserving the spatial relationships between pixels.



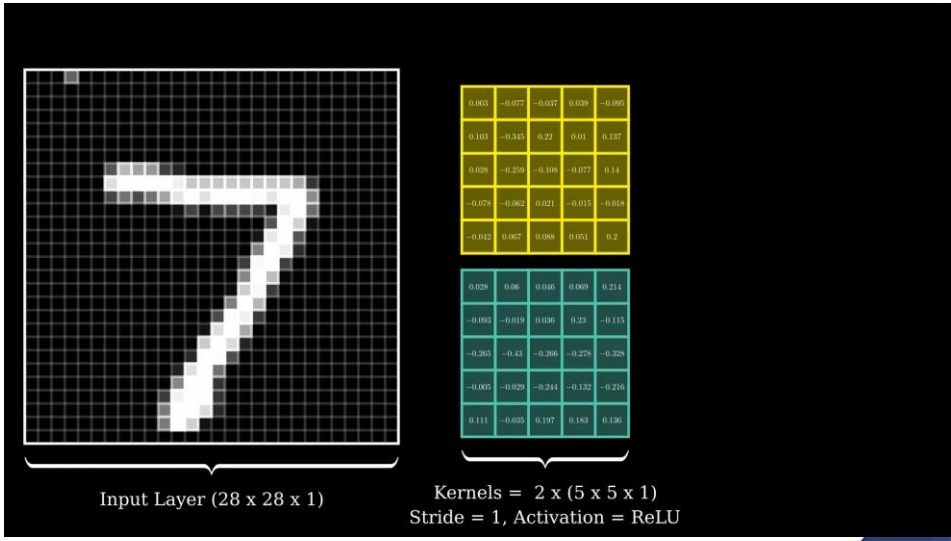
Convolutional Layer

Suppose we use 2 kernels (k_1, k_2) with each is 5x5 matrix, and the image is X, we will get 2 feature channels c_1, c_2.

$$c_1 = X * k_1$$
$$c_2 = X * k_2$$

The two channels are cucullated independently.

The output size of a convolutional layer is same/similar with the input, unless additional parameter is specified e.g. stride=1 means skip an entry after output an entry.



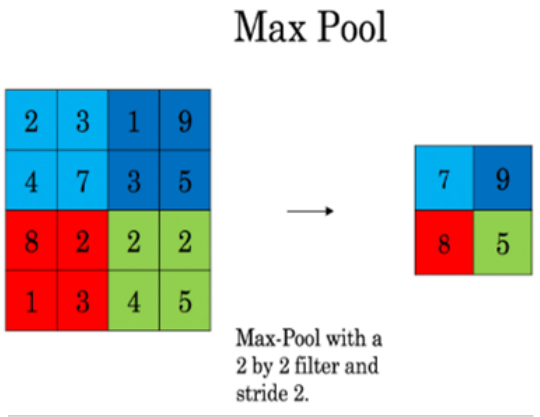
Pooling Layer

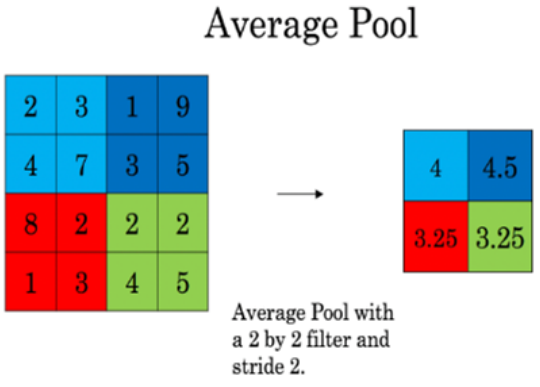
Pooling layers are used in CNNs to reduce information redundancy by summarizing local features within neighboring regions of an image, thus capturing the essence of those features while discarding irrelevant details.

It operates by aggregating the outputs of a neighbouring group of neurons at one layer into a single neuron in the next layer.

Common pooling operations include max pooling, which returns the maximum value from the portion of the image covered by the kernel, and average pooling, which calculates the average value.

NB: the pooling logic is definite, so pooling layers do NOT need the training. Training prosses will NOT modify pooling layers.





Flatten Layer and Dense Layer

When the output is not an image, a typical CNN-based model also includes Flatten Layers and Dense Layers to form the results.

- Flatten layer reshapes a tensor into a single vector by listing its entries literally.
- ReLu layer is also known as the activation function. It provides non-linear features to the model, such as filtering out negative probabilities.

$$\mathrm{ReLu}(\mathrm{X}) = \max(0, X)$$

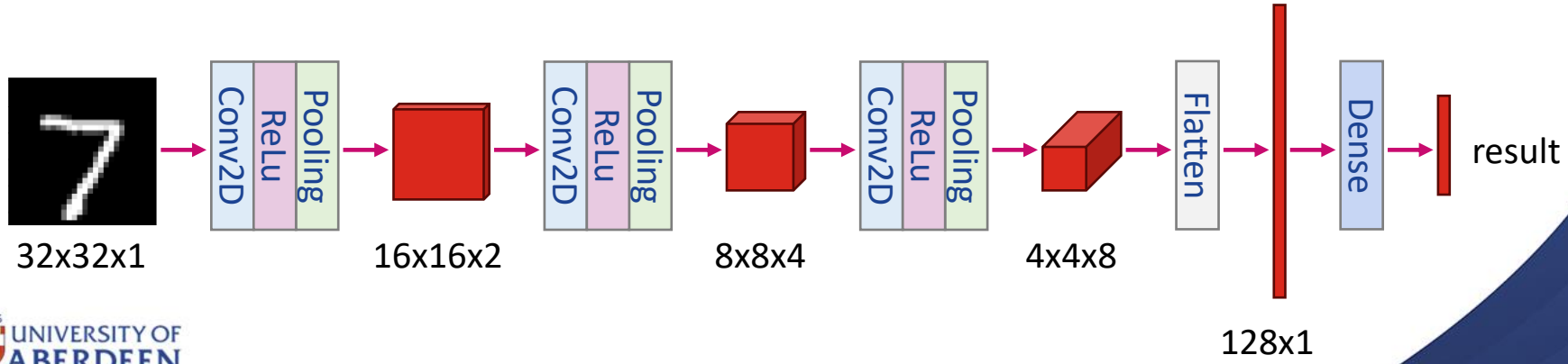
- Dense Layer (fully connected layer) is a fundamental neural network layer. It applies a matrix multiplication and a vector addition on the input vector. The matrix (W) and the vector (b) are learnable parameters.

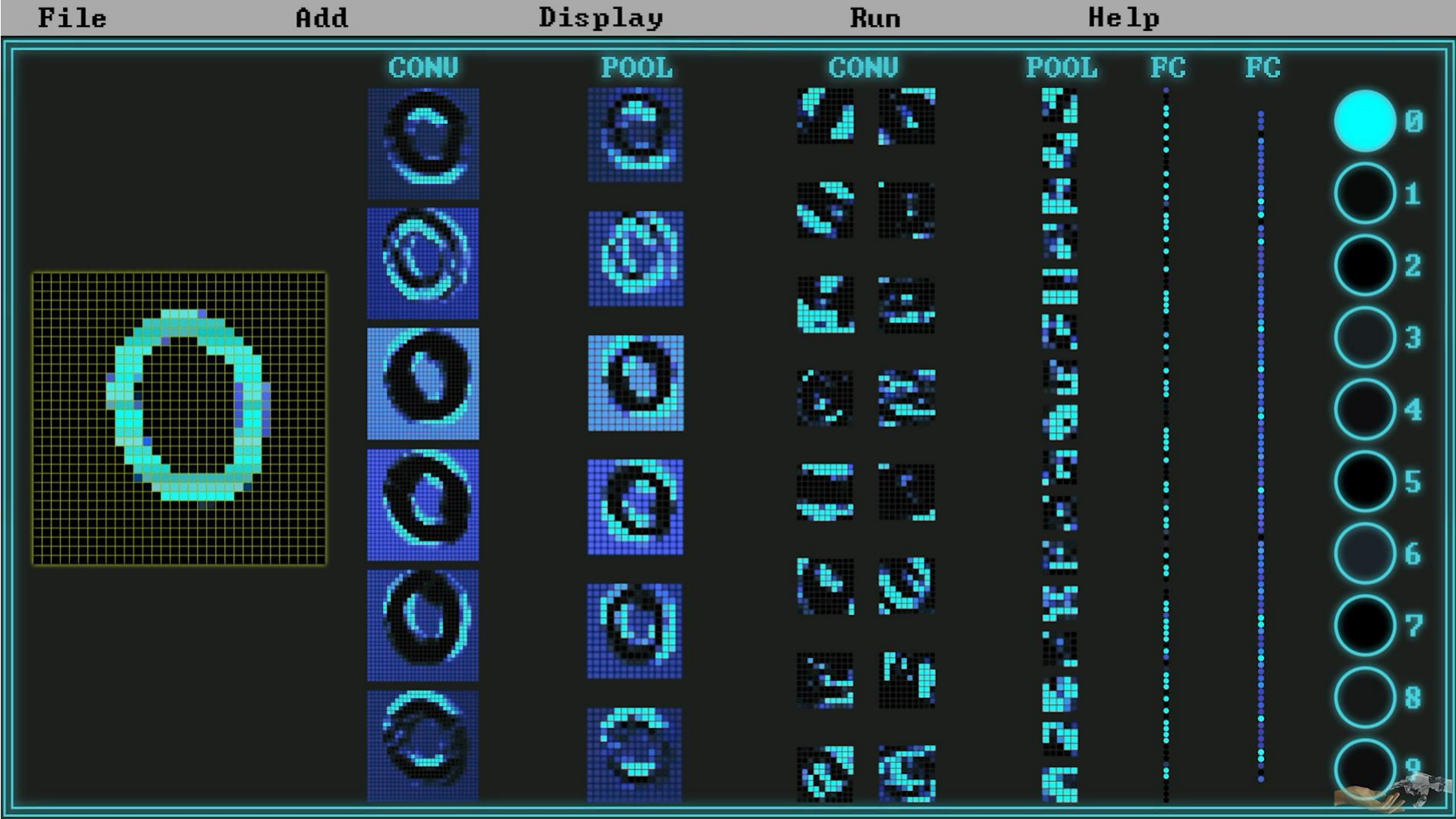
$$\mathrm{Dense}(\mathrm{X}) = \mathrm{WX} + b$$

Convolutional Neural Network (CNN)

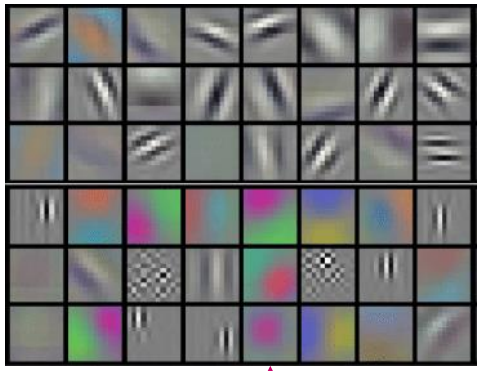
To obtain the complex features (semantic features), we need accumulate multiple layers. Each layer is applied on the output of last layer.

In this pipeline, the former Conv2D will extract the local patterns since the kernel window is small. Then, the follow Conv2D extract higher patterns from the previous patterns. Finally, the dense layer summarises the high-level patterns and makes the prediction.



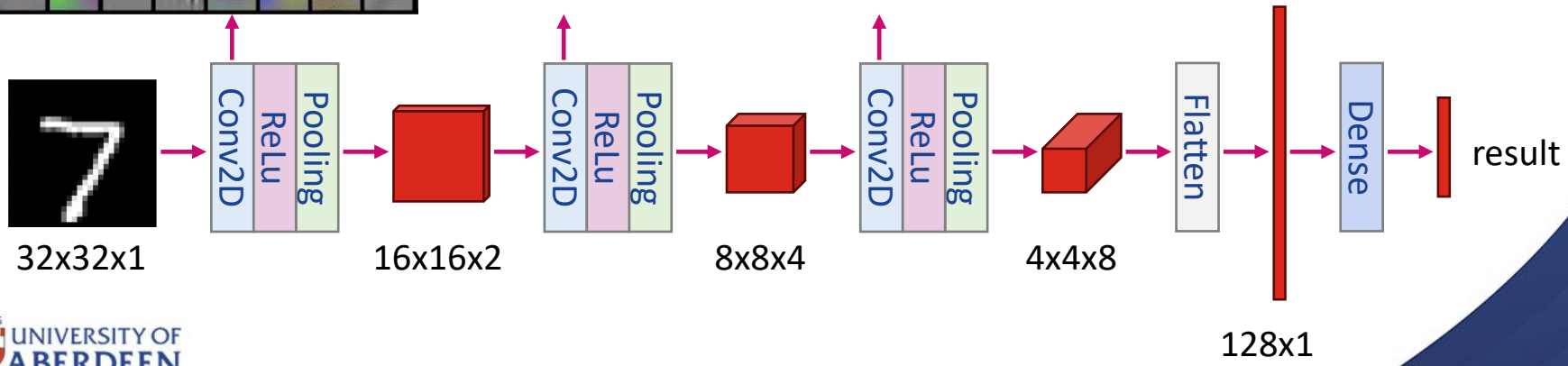

File
Add
Display
Run
Help
CONV
POOL
CONV
POOL
FC
FC
0
1
2
3
4
5
6
7
8
9

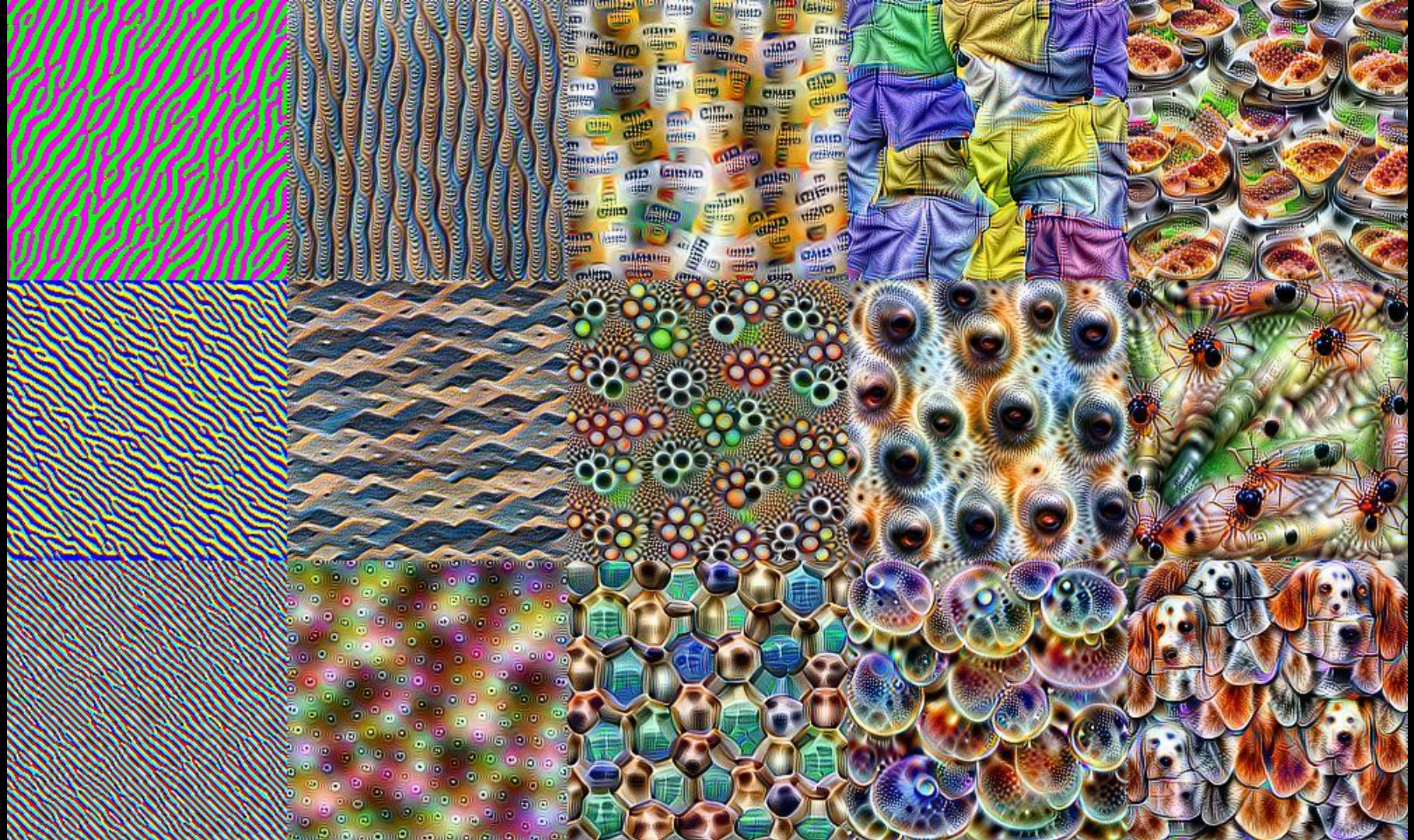
What the CNN Kernels Learnt?



Conv2Ds early on in the network (closer to the input) pick up the most basic features in the image e.g. lines, edges, angles, colours. The latter Conv2D captures texture, shapes, and patterns. Layers close to the output ,then, capture more complex patterns such as whole objects, animals.

To visualise the following layers requires complex technology (next page).





Conclusion

In conclusion, we've explored the concept of images as signals, laying the foundation for understanding how images are processed and analysed.

- We delved into image feature detection, highlighting methods to identify and utilize key points within images for various applications.
- Semantic feature detection was discussed, emphasizing the extraction of meaningful information from images beyond mere shapes and textures.
- Lastly, the role of Convolutional Neural Networks (CNNs) was examined, showcasing their power in image recognition and classification, driven by their ability to learn from and adapt to visual data.