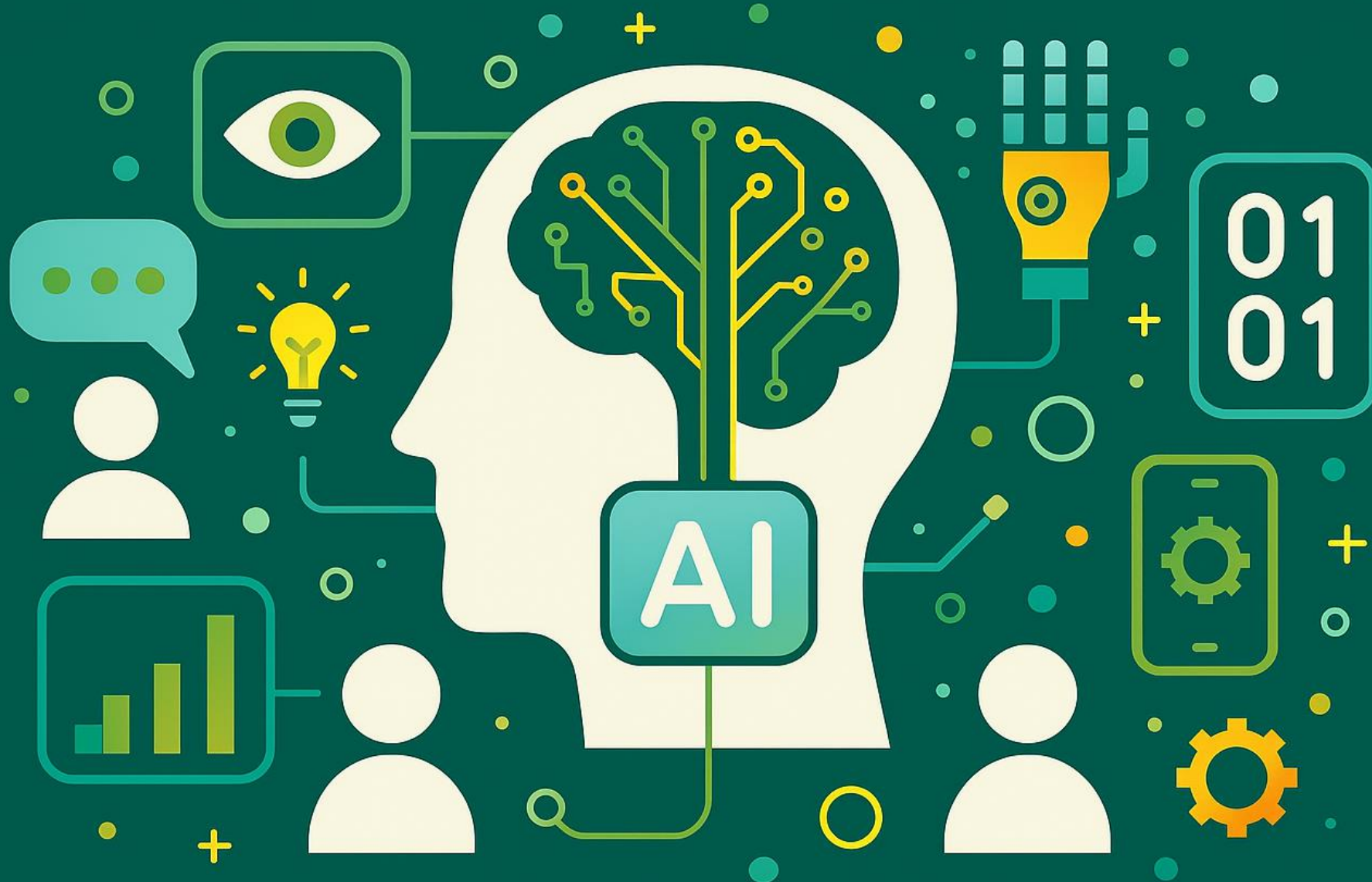


# Computer vision



## Licence Disclaimer

This document is distributed under the terms of the GNU General Public Licence, Version 3, dated 29 June 2007. It is intended to promote freedom to use, study, modify, and share the content herein, in accordance with the principles of free and open-source documentation. By accessing, reproducing, or modifying this document, you agree to comply with the conditions set forth in the GNU GPL v3. A full copy of the licence is available at <https://www.gnu.org/licenses/gpl-3.0.html>.

This licence applies to the document as a whole, including any derivative works, unless otherwise stated. No warranties are provided, and the document is offered “as-is” without liability for its use or interpretation.

## Attribution Requirement

When using, sharing, or adapting this document for any individual, group, or organisation, proper citation of the original author is required. Please cite as follows:

Alsaggaf, I. (2025) *Introduction to Artificial Intelligence*. Available at:  
<https://github.com/ibrahimsaggaf/Introduction-to-Artificial-Intelligence> (Accessed: [insert date]).

# Content

- Image data
- Convolutional Neural Networks (CNNs)
- Q&A

Lab session: Tiny ImageNet challenge

# Image data

1. How do computers represent and process images?
2. What is the difference between grayscale images and colour images?
3. How many channels does an RGB image contain?
4. What does the term image size refer to?



# Image data

Coloured image



Grayscale image





# Image data

RGB image



Grayscale image

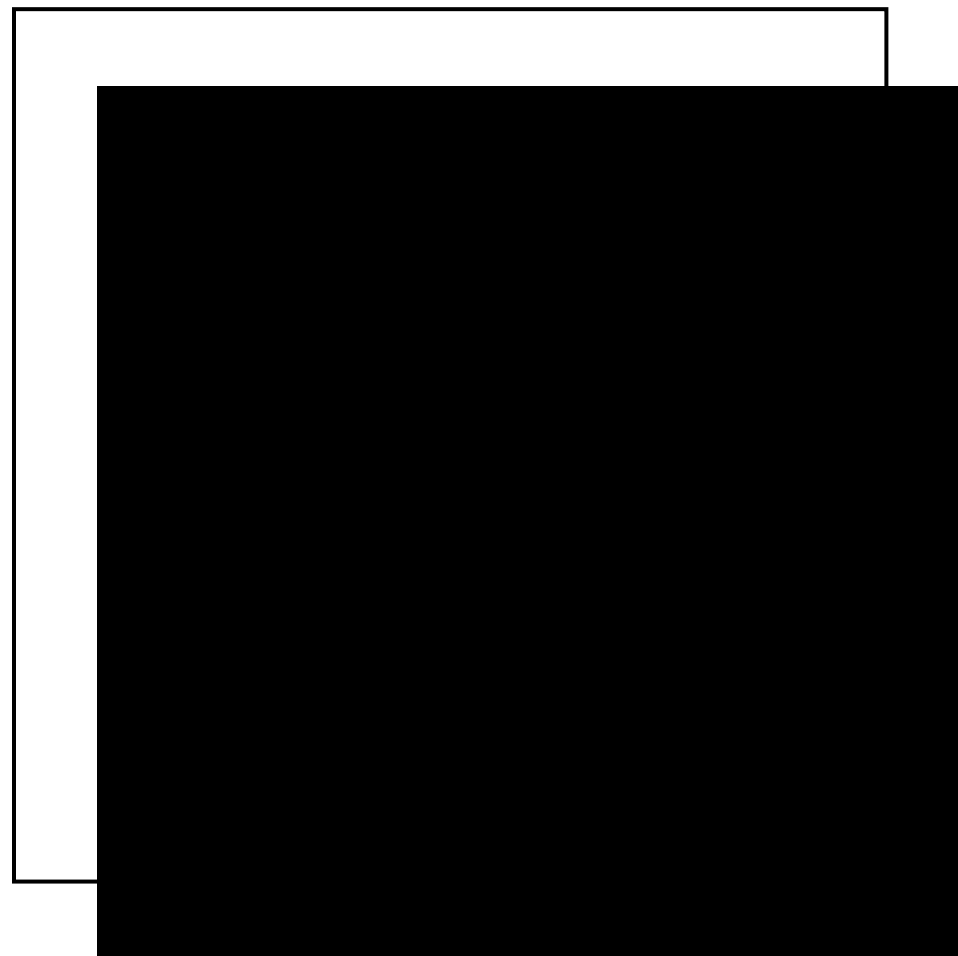


# Image data

RGB image



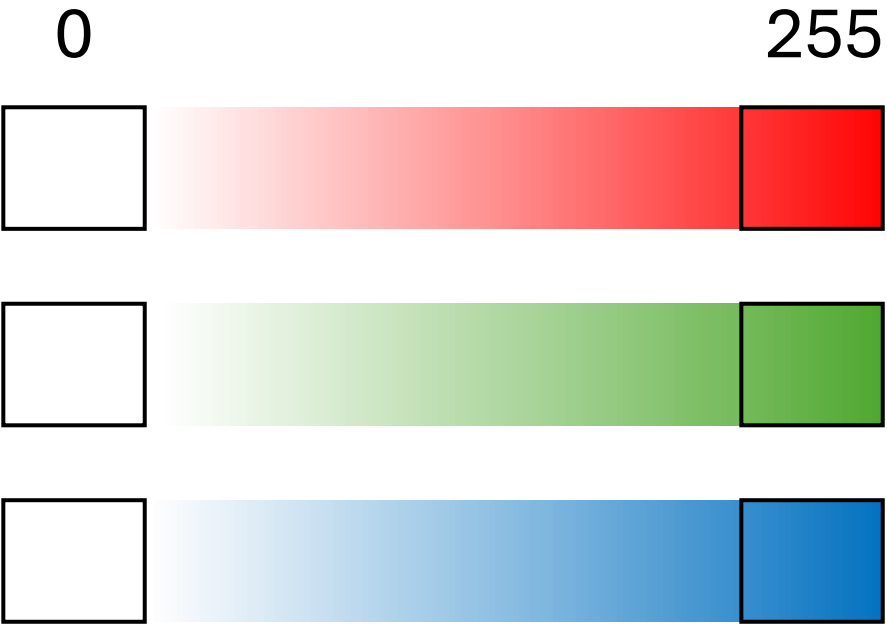
Grayscale image



# Image data

## Colour intensity

RGB image



Grayscale image





# Image data

## Colour intensity

RGB image

**R** channel

0	255
255	0

**G** channel

0	255
255	0

**B** channel

0	255
255	0

Grayscale image

**One** channel

0	255
255	0

# Image data

## Image size

2x2 image with 3 channels

R channel

0	255
255	0

G channel

0	255
255	0

B channel

0	255
255	0

2x2 image with 1 channel

Grayscale

0	255
255	0

# Image data

## Image size

(Channel, Height, Width)  
(3, 2, 2)

R channel

0	255
255	0

G channel

0	255
255	0

B channel

0	255
255	0

(Channel, Height, Width)  
(1, 2, 2) = (2, 2)

Grayscale

0	255
255	0



# Image data

## Image size

Size = (3, 64, 64)

RGB or grayscale?

What is the height?

What is the width?

# Image data

## Image size

Size = (3, 64, 64)

RGB or grayscale?

RGB



What is the height?

64



What is the width?

64

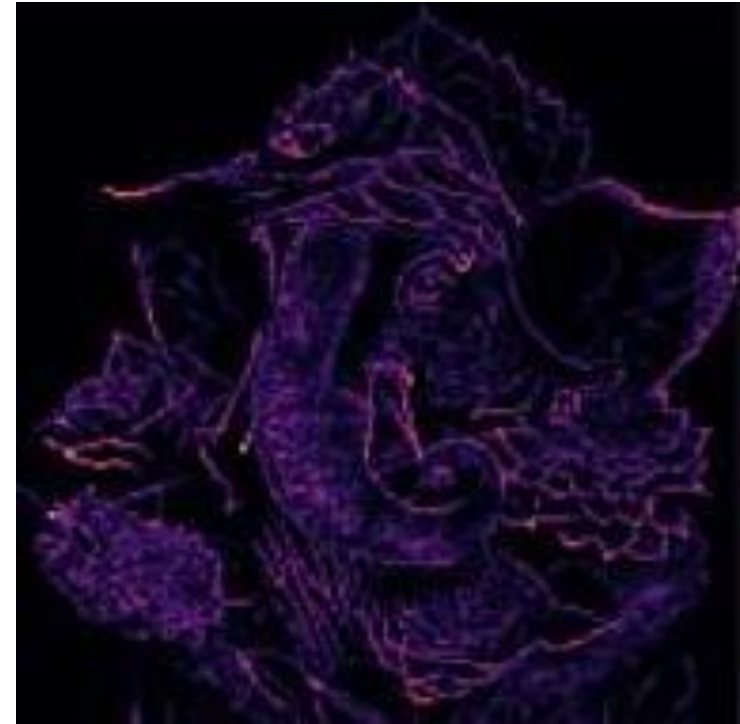


# The Convolution operation

Original image



Learnt feature maps





# The Convolution operation

<https://ezyang.github.io/convolution-visualizer/index.html>

# The average pooling operation

2	2	7	3
9	4	6	1
8	5	2	4
3	1	2	6

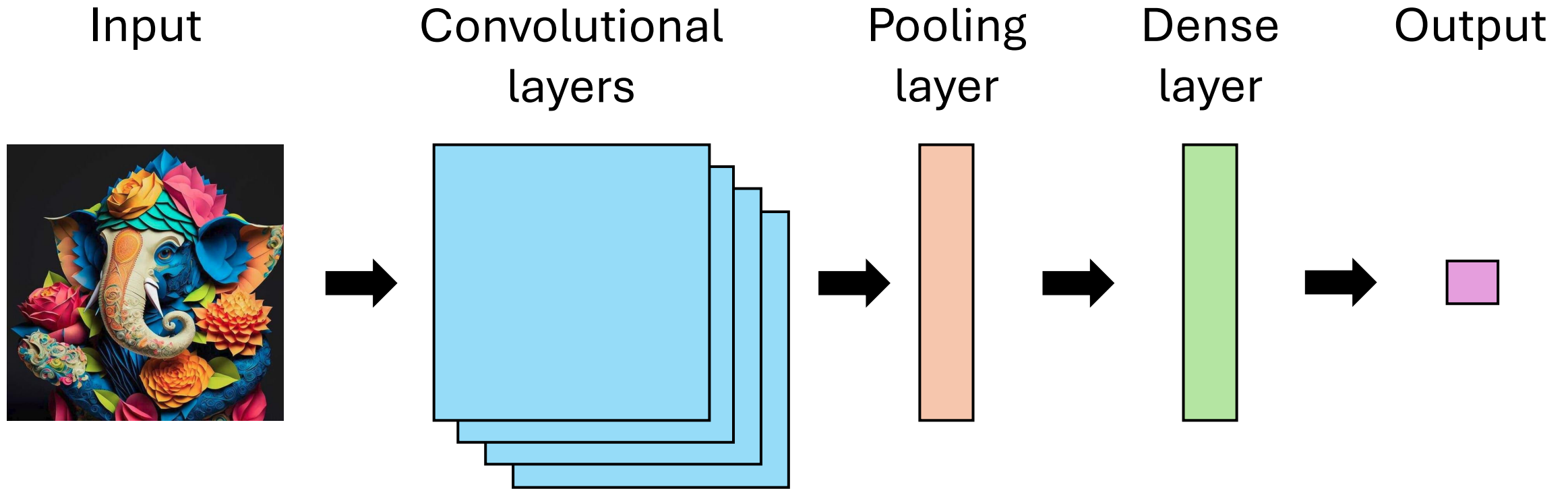
Average Pool



Filter - (2 x 2)  
Stride - (2, 2)

4.25	4.25
4.25	3.5

# CNN architecture





**Q&A**



# Lab Time

Lab 6: Tiny ImageNet challenge

# Lab 6: Dataset

## Tiny ImageNet

Tiny ImageNet contains 200 classes for training. Each class has 500 images. The training set contains 100,000 images, whilst the test set contains 10,000 images. All images are 64x64 coloured ones.



Goldfish



Goldfish



Goldfish



Goldfish

# Step 1

- Download the Lab6 directory from the GitHub repository <https://github.com/ibrahimsaggaf/Introduction-to-Artificial-Intelligence>
- Open the Lab6 directory in Visual Studio Code.
- The Lab6 directory contains 5 files:
  - ☐ main.py
  - ☐ model.py
  - ☐ network.py
  - ☐ utils.py
  - ☐ requirements.txt

Take your time examining these files.

## Step 2

- Create and activate a virtual environment under the name “lab6\_env” (see Lab 1)
- Install the below libraries inside the virtual environment using a requirements file (see Lab 4):
  - ☐ Datasets (load data from HuggingFace Hub)
  - ☐ Pillow (Image manipulation)
  - ☐ Numpy
  - ☐ Deep learning library Pytorch
  - ☐ Visualisation library Matplotlib

## Step 3

- Run the command:  
*python main.py*

This command runs a large-scale image classification task over 100,000 images with 200 different classes:

1. Download and preprocess the Tiny ImageNet dataset.
2. Create a ResNet model.
3. Train the model on the training set, then measure the accuracy on both training and testing sets.
4. Plot the learning curves and save the figure.



## Step 5

- Inspect the printed output.
- What is the size of the model?
- Is it taking forever just to finish one epoch?
- Try to reduce the data size by sampling:
  - ☐ In the main.py file, comment line 43 and uncomment line 46.
  - ☐ This will conduct stratified sampling by only considering 30% of the images in each class, reducing the number of training images from 100,000 to 30,000 and the number of testing images from 10,000 to 3,000.

# Lab 6: Tiny ImageNet challenge

Congrats! 🎉

- [✓] Understand the core concepts of CNN
- [✓] Conduct a large-scale experiment
- [✓] Train a ResNet model for image classification

# Quiz 5

Q1: Which of the following images would require the greatest computational cost to process?

- A) A 42x42 grayscale image
- B) A 32x32 RGB image
- C) A 32x32 grayscale image
- D) All of the above require the same computational cost

Q2: In this lab, what conclusions can be drawn from running such a large-scale experiment?

- A) The difficulty of the task increases with the increase in the number of classes
- B) The experiment runtime increases with the increase in the number of classes
- C) Training on grayscale images requires more convolutional operations compared to training on RGB images of the same size
- D) All of the above

# Reading list

Residual Networks (ResNet) - Deep Learning

<https://www.geeksforgeeks.org/deep-learning/residual-networks-resnet-deep-learning/>

But what is a convolution?

<https://www.youtube.com/watch?v=KuXjwB4LzSA>