Introduction to Computer Vision

Dr Ian Cornelius

Hello

Hello (1)

Learning Outcomes

1. Understand the concept of computer vision
2. Understand and apply the concept and theory behind a common computer vision framework
3. Demonstrate knowledge on how to use a computer vision framework in a body of work

Computer Vision

Computer Vision (1)

- A field to gain a level of understanding from digital media
  - represents an object and their characteristics
- Situated within **Artificial Intelligence** (AI)
- Initial computer vision experimentation began in 1959
  - cats where shown an array of images to correlate a response in its bran
  - they responded to hard edges or lines
  - computers were able to scan images to digitize them during this time
- 1963 saw the transformation of 2D images into 3D
- 1974 saw the introduction of optical character recognition (OCR)
- 1982 saw the introduction of algorithms to discern edges and shapes
- 2000 focused upon object recognition
  - applications released in 2001 focussing on facial recognition

Computer Vision (2)

Computer Vision vs. Image Processing

- Computer vision is distinct from image processing
- *Image Processing* creates a new image from pre-existing images
  - i.e. simplification or enhancing of content
  - often not concerned with the content of an image
- *Computer Vision* is concerned with automating tasks
  - i.e. object recognition and tracking of a said object

Computer Vision (3)

Why do we need Computer Vision?

- Thousands of images/videos are now publicly available
  - cameras exist on smartphones and laptops
  - sharing it is becoming easier
- Digital world enabled to interact with physical
- Indexing and searching text is easy, images are not
  - knowledge on what an image contains is required
- Machines need to *see* an image and understand its content
- It allows us to understand the content of digital images
  - via the use of algorithms to reproduce human vision
  - thus being able to discern objects and people

Computer Vision (4)

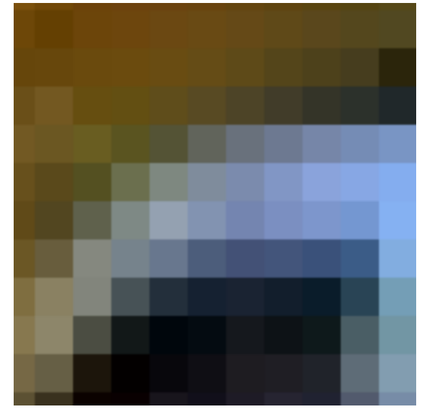
Human Vision

- Humans can easily perceive the world, machines on the other hand cannot
  - it is second nature for us to gather information from our surroundings
- Objects can be perceived in less than a second
  - describe the content of photos and videos after a single glance

Computer Vision (5)

Computer Vision

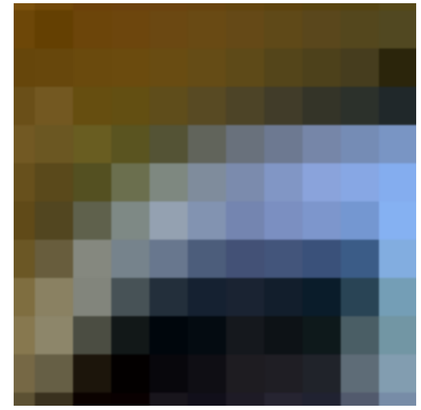
- It is more complex for computers to *see* an image
  - instead, it processes text and images as numbers
- These numbers are otherwise known as *pixels*



Structure of an Image

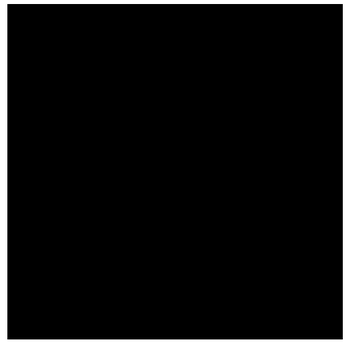
Structure of an Image (1)

- Images are made up of lots of small elements known as *pixels*
- Each pixel corresponds to a given value
  - a single-bit pixel is grayscale
  - a three-bit pixel is colour
    - i.e. red, green, blue (RGB) or blue, green, red (BGR)
- Each bit of a pixel is interpreted as an integer
  - for grayscale, a value between 0 and 255
    - i.e. `(255)`
  - for RGB, a value between 0 and 255, but in three components
    - i.e. `(255, 201, 154)`



Structure of an Image (2)

- Images can be defined as a two-dimensional matrix
- The matrix on the screen represents a ten-by-ten image
  - known as a *single channel* image
- Each value in this image is 0, therefore, the resulting image is a black square



```
blackSquare = [
    # Row 1                         # Row 2
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    # Row 3                         # Row 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, 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, 0, 0, 0]
]
```

Structure of an Image (3)

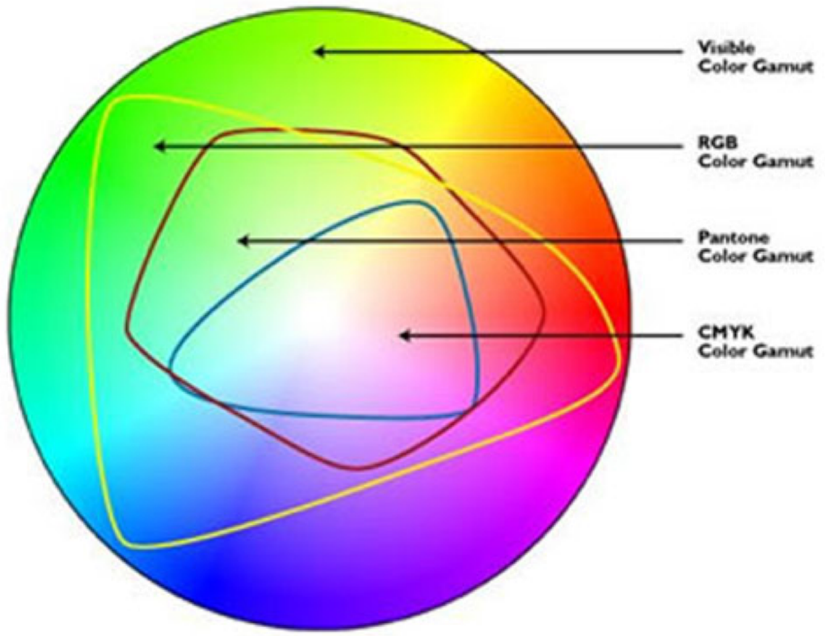
- The matrix on the screen represents a 10x10 image
  - known as a *three channel* image
- Each value in this image is a different value, which results in the image being a red square
  - a tone of red...

```
redSquare = [
    # Row 1
    [   # Column 1     # Column 2     # Column 3     # Column 4     # Column 5
        [31, 31, 187], [31, 31, 187], [31, 31, 187], [31, 31, 187], [31, 31, 187],
        # Column 6     # Column 7     # Column 8     # Column 9     # Column 10
        [31, 31, 187], [31, 31, 187], [31, 31, 187], [31, 31, 187], [31, 31, 187]
    ],
    "...",
    # Row 10
    [   # Column 1     # Column 2     # Column 3     # Column 4     # Column 5
        [31, 31, 187], [31, 31, 187], [31, 31, 187], [31, 31, 187], [31, 31, 187],
```

Color Models

Color Models (1)

- A protocol for representing colours, making them easily reproducible
- Popular models in computer vision are RGB, BGR and gray-scale
- Other color models such as HSV may be used
  - video compression and device independent storage
- Colour models discussed in this lecture:
  - RGB
  - CMYK
  - Grayscale
  - HSV/HSB



Color Models (2)

RGB

- Abbreviation for: Red, Green and Blue
- An additive color model
- Uses a collection of three intensities for each pixel
  - red, green and blue
  - intensities of each value are mixed in this color space

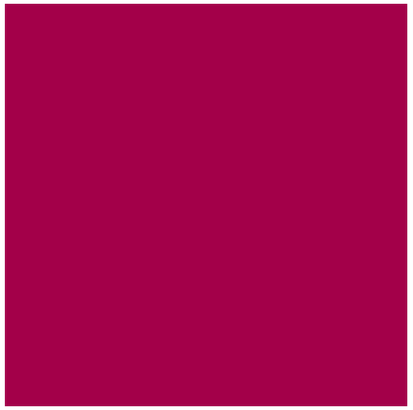
```
Red = 165
Green = 34
Blue = 90
```

```
rgbColor = (165, 34, 90)
```

Color Models (2)

CMYK

- Abbreviation for: Cyan, Magenta, Yellow, and Black
- A subtractive color model
- Calculate colors by a process of subtraction



```
C = 0
M = 79%
Y = 45%
K = 35%
```

```
cmykColor = (0, 0.79, 0.45, 0.35)
```

Color Models (3)

Grayscale

- Uses a single intensity for each pixel

```
Red = 115
Green = 115
Blue = 115
```

```
grayscaleColor = (115, 115, 115) # or (115)
```

Color Models (4)

HSV/HSB

- Abbreviation for: Hue Saturation Value
  - or Hue Saturation Brightness
- Colours and intensity are provided separately
  - provides a robustness to any lighting changes that may occur

```
Hue = 334°
Saturation = 79%
Value/Brightness = 65%
```

Computer Vision Library: OpenCV

Computer Vision Library: OpenCV (1)

- OpenCV is an abbreviation for **Open** Source **C**omputer **V**ision
  - originally a research project at Intel
- Library consisting of computer vision and machine learning tools
  - Consists of over 2,500 algorithms
- The library consists of interfaces for Python and C++
  - and Java! (👍 or 👎?)
- Not a requirement for computer vision
  - however, it is one of the easiest, capable and well-supported options

OpenCV Contribution - Extended Modules

- A separate collection of modules that consists of ‘non-free’ algorithms
  - SIFT, SURF etc.
- It can be unstable as it is not well-tested

Computer Vision Library: OpenCV (2)

Applications of OpenCV

- Used for a wide variety of applications, such as:
  - Image and Video Processing
    - i.e. color space/model conversion, image smoothing, and transformations
  - Facial Recognition
  - Object Recognition

Computer Vision Library: OpenCV (3)

Installing OpenCV

- Installation will depend upon your platform
  - major platforms will have pre-built libraries

Windows Installation

- Three methods of installation for Windows:
  1. third party installer if you are using C++
  2. via the Python package manager
     - i.e. `python3 -m pip install python-opencv`
  3. compiling/building from the OpenCV sources

Linux Installation

- Two methods of installation for Linux:
  1. via the package manager of your distribution
     - i.e. `apt install libopencv-dev python3-opencv`
  2. compiling/building from the OpenCV sources

Computer Vision Library: OpenCV (4)

Compiling and Building from Source

- OpenCV is an open-source computer vision framework
  - hence the **open** in its name
- Compilation of the latest version from source is an option for enthusiasts
- Extra benefits are provided:
  - i.e. CUDA and CuDNN support, as well as cross-compilation for different architectures
- Feeling brave?
  - follow this guide for instructions on how to compile from source

Goodbye

Goodbye (1)

Questions and Support

- Questions? Post them on the **Community Page** on Aula
- Additional Support? Visit the Module Support Page
- Contact Details:
  - Dr Ian Cornelius, ab6459@coventry.ac.uk