Introduction to Image Processing

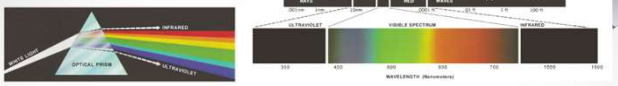
DLAI3 Workshop, 1 July 2020

Somnuk Phon-Amnuaisuk
School of Computing and Informatics
Centre for Innovative Engineering
Universiti Teknologi Brunei

1

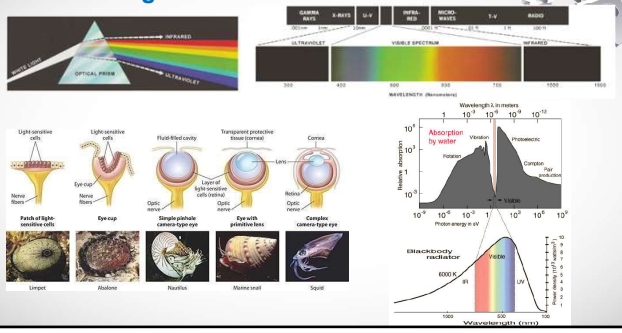
2

Electromagnetic Wave



3

Electromagnetic Wave



4

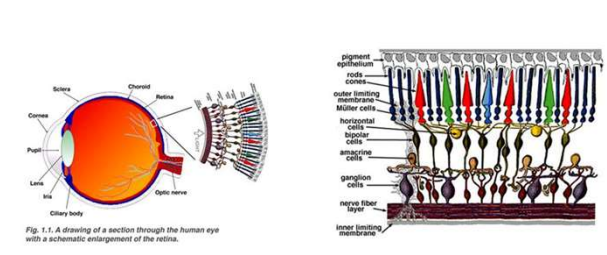


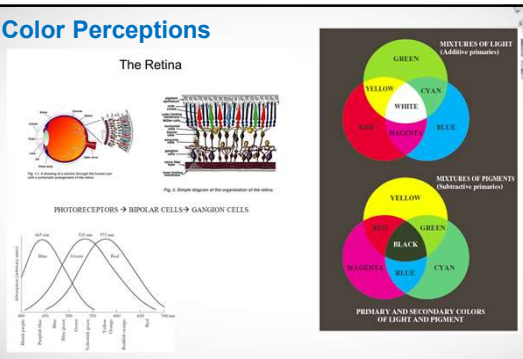
Fig. 1.1. A drawing of a section through the human eye with a schematic enlargement of the retina.

Fig. 2. Simple diagram of the organization of the retina.

PHOTORECEPTORS → BIPOLAR CELLS→ GANGION CELLS

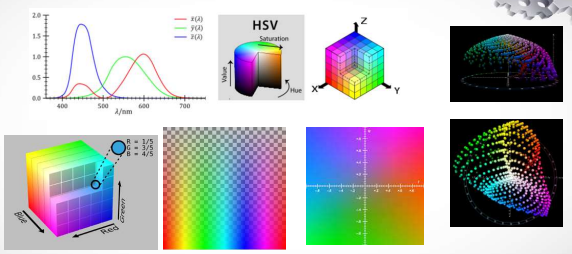
5

Color Perceptions



6

Color Representations



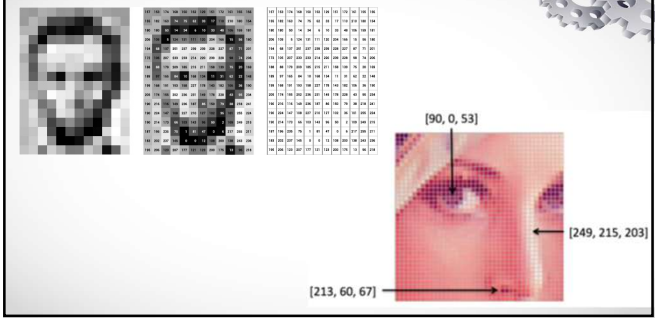
7

File Formats

V · T · E	Graphics file formats [hide]
Raster	ANI · ANIM · APNG · ART · BMP · BPG · BSAVE · CAL · CIN · CPC · CPT · DDS · DPX · ECW · EXR · FITS · FLIC · FLIF · FPX · GIF · HDRi · HEVC · ICER · ICNS · ICO / CUR · ICS · ILBM · JBIG · JBIG2 · JNG · JPEG · JPEG-LS · JPEG 2000 · JPEG XR · JPEG XT (JPEG-HDR) · KRA · MNG · MIFF · NRRD · PAM · PBM / PGM / PPM / PNM · PCX · PGF · PICtor · PNG · PSD / PSB · PSP · QTVR · RAS · RGBE (Logluv TIFF) · SGI · TGA · TIFF (TIFF/EP · TIFF/IT) · UFO/ UFP · WBMP · WebP · XBM · XCF · XPM · XWD
Raw	CIFF · DNG
Vector	AI · CDR · CGM · DXF · EVA · EMF · Gerber · HVIF · IGES · PGML · SVG · VML · WMF · Xar
Compound	CDF · DjVu · EPS · PDF · PICT · PS · SWF · XAML
Metadata	Exchangeable image file format (Exif) · Extensible Metadata Platform (XMP)
Category · Comparison	

8

Image Representation (R,G,B)



9

Features from Pixels

- Primitives
  - Pixel's values
  - Gradient
  - Edge, Point, Shape, Blob, Flow

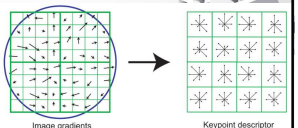
10

Extract Features using Filter

11

Features from Pixels

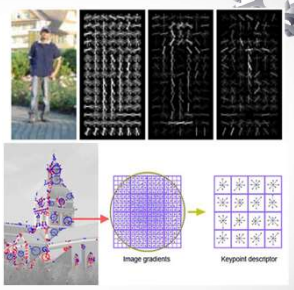
- Primitives
  - Pixel's values
  - Gradient
  - Edge, Point, Shape, Blob, Flow
- Structures from primitives or their combinations
  - Local and global min/max, Integrated image
  - Histogram, Correlation, Covariant
  - Spatial and temporal quantity



12

Extract Discriminative Features

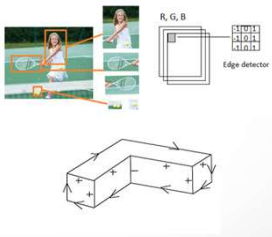
- Feature detection
  - Pixels information
  - Depth information
  - Edge, Corner, Blob, etc.
- Feature description
  - HOG (McConnell, 1986)
  - SIFT (Lowe, 1999)
  - SURF (Bay et al., 2006)
  - etc.



13

Extract Discriminative Features

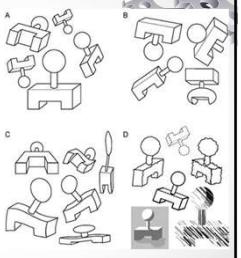
- Feature detection
  - Pixels information
  - Depth information
  - Edge, Corner, Blob, etc.
- Feature description
  - HOG (McConnell, 1986)
  - SIFT (Lowe, 1999)
  - SURF (Bay et al., 2006)
  - Understand 3D shape
  - etc.



14

Issues from Appearance-based Features

- Handling variations in appearances
  - Viewpoints, scale
  - Orientation, occlusion, deformation
- Handling temporal information
- Nature of features
  - Local vs Global
  - Static vs Dynamic



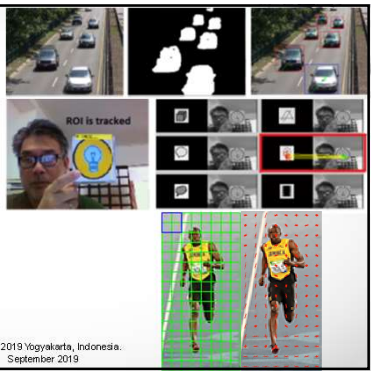
15

Prior Deep Learning

- Handcrafted features
- A long sequence of pre-processing, processing and post-processing.
- Feature representations are not invariant.

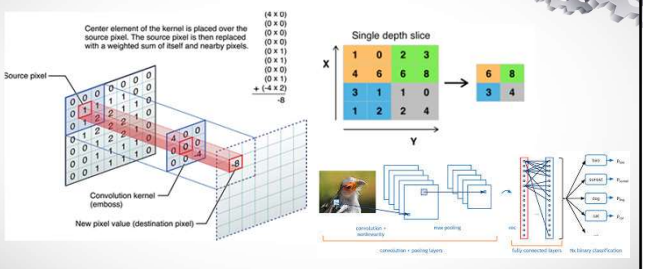
Hence

- Operate in a limited domain.
- Poor accuracy.
- Poor scalability.



16

Convolutional Neural Network



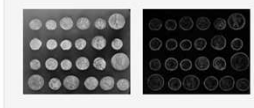
17

Image Processing with scikit-image 1

- scikit-image is a collection of algorithms for image processing. It is available free of charge and free of restriction. We pride ourselves on high-quality, peer-reviewed code, written by an active community of volunteers.

```
from skimage import data, io, filters

image = data.coins()
# ... or any other NumPy array!
edges = filters.sobel(image)
io.imshow(edges)
io.show()
```



18

Image Processing with scikit-image 2

- scikit-image represents images using NumPy arrays.
- Two-dimensional (2D) grayscale images are indexed by rows and columns with the lowest element (0, 0) at the top-left corner.
- Two-dimensional (2D) color images are indexed by rows, columns and the channel (corresponding to RGB, RGBA, HSV, etc.)

19

Practical Session with scikit-image

- Load, Save
- Image cropping ad scaling
- Modify pixel information
- Filter and Convolution

20

Practical Session with scikit-image

- Load & display

```
import numpy as np
import matplotlib.pyplot as plt
from google.colab import files
from skimage import data, io, filters
```

```
img= io.imread('image1.jpg')
print(type(img), img.shape, img.size, img.min() , img.max())
#
plt.imshow(img)
```

```
<class 'numpy.ndarray'> (1040, 1920, 3) 5990400 0 255
<matplotlib.image.AxesImage at 0x7f9026c9fb70>
```

21

Practical Session with scikit-image

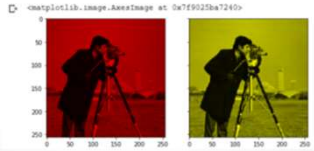
- Modify pixels

```
from skimage import color
from skimage import img_as_float
grayscale_image = img_as_float(data.camera()[::2, ::2])
image = color.gray2rgb(grayscale_image)

red_multiplier = [1, 0, 0]
yellow_multiplier = [1, 1, 0]

fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(8, 4), sharex=True, sharey=True)
ax1.imshow(red_multiplier * image)
ax2.imshow(yellow_multiplier * image)
```

```
<matplotlib.image.AxesImage at 0x7f9025ba7240>
```



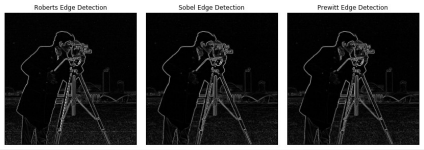
22

Practical Session with scikit-image

- Filters

```
from skimage import filters
from skimage import data
from skimage.exposure import rescale_intensity
import matplotlib.pyplot as plt


image = data.camera()
edge_roberts = filters.roberts(image)
edge_sobel = filters.sobel(image)
edge_prewitt = filters.prewitt(image)
```



23

Tensorflow Keras Utilities for Image

Module: tf.keras.preprocessing.image

Functions

- array_to_img(...): Converts a 3D Numpy array to a PIL Image instance.
- img_to_array(...): Converts a PIL Image instance to a Numpy array.
- load_img(...): Loads an image into PIL format.
- save_img(...): Saves an image stored as a Numpy array to a path or file object.

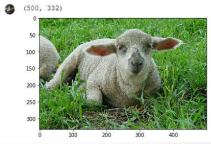
```
from google.colab import files
uploaded = files.upload()
```

Choose Files | No files selected. Upload widget is only available when the cell to enable.

Saving 2007_000175.jpg to 2007_000175.jpg

```
# load the image
img = tf.keras.preprocessing.image.load_img('2007_000175.jpg')
plt.imshow(img)
print(img.size)
```

(500, 332)

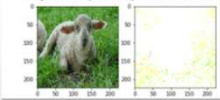


24

Tensorflow Keras Utilities for Image

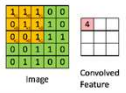
```
# load the image with the required shape
img = tf.keras.preprocessing.image.load_img('2007_000175.jpg', target_size=(224, 224))
plt.subplot(1,2,1),plt.imshow(img)
print(img.size, type(img))
img1 = img.copy()
# convert to array
img1 = tf.keras.preprocessing.image.img_to_array(img1)
print(img1.shape,img1.size, type(img1))
# expand dimensions so that it represents a single 'sample'
img1 = np.expand_dims(img1, axis=0)
print(img1.shape,img1.size, type(img1))
plt.subplot(1,2,2),plt.imshow(img1[0])
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
(224, 224) <class 'PIL.Image.Image'>
(224, 224, 3) 150528 <class 'numpy.ndarray'>
(1, 224, 224, 3) 150528 <class 'numpy.ndarray'>
(<matplotlib.axes._subplots.AxesSubplot at 0x7fb0ed37e1d0>,
 <matplotlib.image.AxesImage at 0x7fb0ed3d1f28>)
```



25

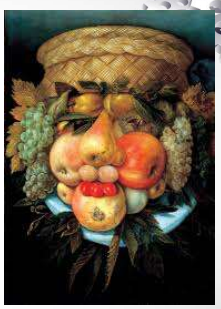
Convolution Process



https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

26

Q & A



The Fruit Basket is a c.1590 oil on panel still life by Giuseppe Arcimboldo

27