Image Processing and Computer Vision with Python & OpenCV

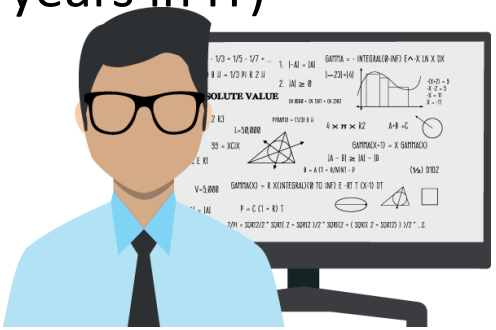
About Me

- Name: Shiv

- BSc and MSc in Mathematics from IIT Kharagpur, India

- PhD in Analytics

- Experience: 19 years (9 Years in Analytics and 10 years in IT)



- Current Role: Chief Data Scientist

Prerequisites

i. Knowledge of Python (Basic)
ii. Knowledge of basic Statistics
iii. Expert in NumPy package

Take any course similar to following

AI, Basic Statistics, Basic Python, Basic R, ML (Overview)
https://www.udemy.com/ai-basic-statistics-basic-python-basic-r-ml-overview/?couponCode=AISPYRML

iv. Awareness of ML (Nice to have in for Data Science professional)

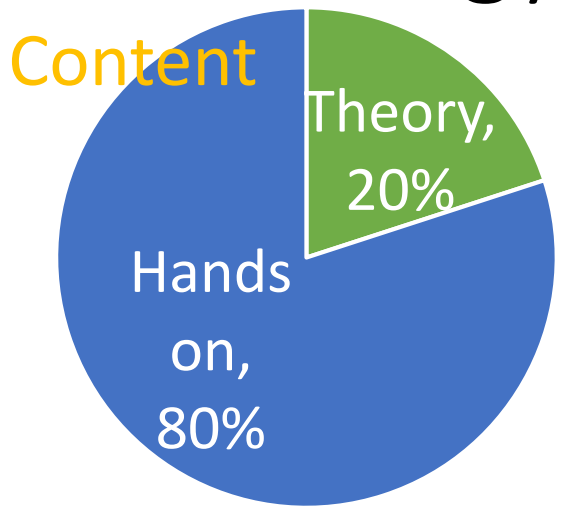
Data science for AI and Machine learning using Python
https://www.udemy.com/data-scientist-for-ai-and-machine-learning-using-python/?couponCode=AIMLPY

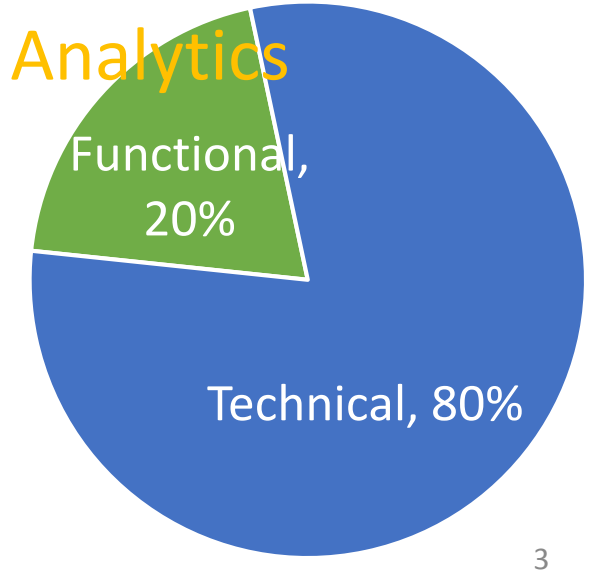
v. Awareness of DL (Nice to have in for Data Science professional)

Deep Learning by TensorFlow (tf.keras) & Keras using Python
https://www.udemy.com/deep-learning-by-tensorflow-tfkeras-keras-using-python/?couponCode=DLPY010

vi. Good Knowledge of any programming technology
vii. Ideal count of trainee 10 (classroom or webex)

Methodology





Training Content (Hands on marked in green color)

Part 1: Foundation

i. Technology - Skimage and OpenCV-Python
ii. Definition of image processing
iii. Definition of Computer vision
iv. Explanation of Image's attributes
v. Black vs Grey vs Color image
vi. Color Spaces
    i. RGB
    ii. HSV: Hue Saturation Value
    iii. CMYK
vii. skimage
viii. Exploration of image
ix. Boolean arithmetic on image
x. Various image conventions
xi. 2D/3D grayscale/multichannel
xii. Example: Counting of White dots
xiii. Color image manipulation
xiv. Image Histograms
xv. Image Filters

Part 2: OpenCv

i. Exploration of image
ii. Exploration of Video
iii. Manipulation of Video
iv. Draw different geometric shapes
v. Playing with color
vi. Various transformations
vii. Arithmetic on image
viii. Overlapping of two images
ix. GrabCut Foreground
x. Contours
xi. Hu-Moments
xii. Clustering using KMeans

Hands on marked in green color

Training Content (Hands on marked in green color)

Part 3/4: Advance OpenCV

i. Blurring
ii. Low/High pass filter
iii. Histogram equalization
iv. Morphological operations
v. Face Detection using Haar Cascades
vi. Image/Video Denoising
vii. Feature detection
viii. Shi-Tomasi Corner Detector
ix. Features from accelerated segment test(FAST)
x. ORB (Oriented FAST and Rotated BRIEF) corner detectors
xi. Image Segmentation (TBD)

x. TensorFlow Object detection
xi. Object detection on image
xii. Object detection on video/webcam
xiii. Digit recognition on image/video
xiv. Build model using LinearSVC
xv. Build model using Random forest

xvi. Harris Corner Detector

xvii. Edge Detection
xviii. First Derivative Filters
xix. Second Derivative Filters
xx. Transformations
xxi. Understanding Features
xxii. Extract features/key points
xxiii. Histogram equalization
xxiv. Object Detection
xxv. Tensorflow Object Detection API
xxvi. Best Practices

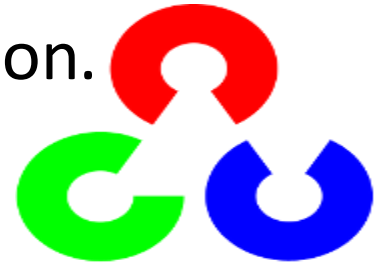
Hands on marked in green color

Installations & Technology

- Install.txt – Please follow the steps

- **Skimage:** Well known library in Python for image processing
  - https://scikit-image.org/

- **OpenCV-Python: A** Python wrapper around original C++ implementation.
  - https://docs.opencv.org/3.4.5/index.html

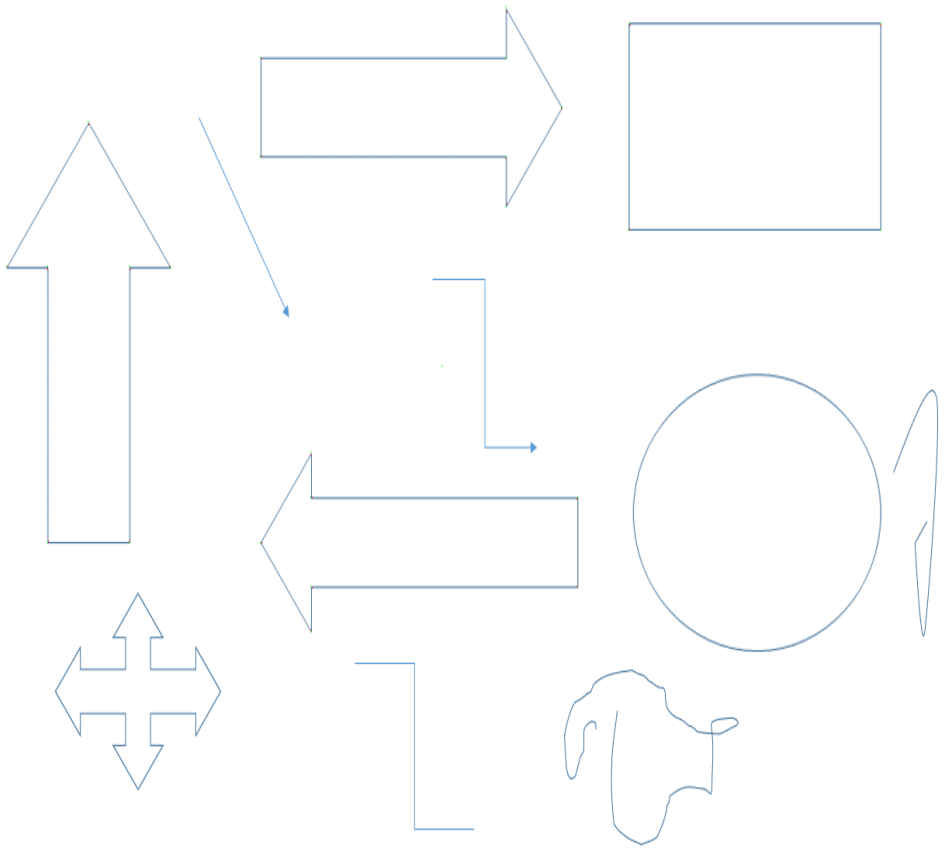


- Anaconda (https://www.anaconda.com/download/) or any Python IDE

- Presentations and Code are uploaded in section (1 or 2)

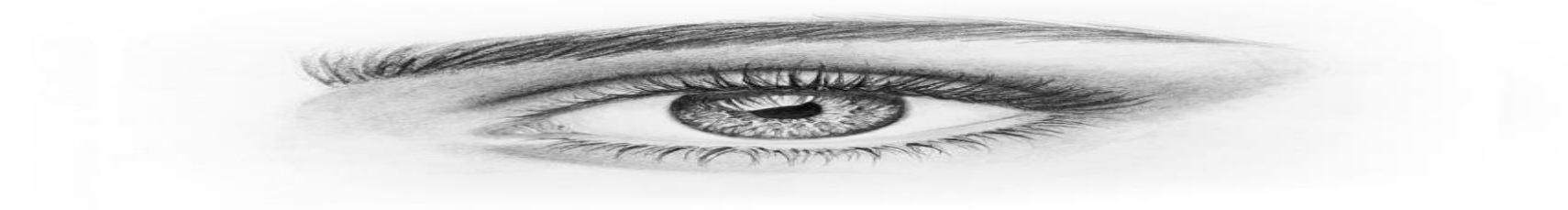
Definition of image processing

The analysis and manipulation of a digitized image, especially in order to improve its quality. *As per Dictionary.com*



123

Definition of Computer vision



Process

https://en.wikipedia.org/wiki/Computer_vision

Computer vision is an interdisciplinary scientific field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do.

Eye image source: https://ogloszenia-belgia.info/how-to-draw-eyes-sketch/7562

Explanation of Image's attributes

i. The term **pixel** is generally applied to 2D and is replaced by **voxel** in 3D images.

ii. **Bit Depth**: The pixel range of a given image format is determined by its bit depth. For example, an 8-bit image will have a range of [0; 255]. Most of the common photographic formats such as jpeg, png etc. use 8-bit for storage and only have positive values.

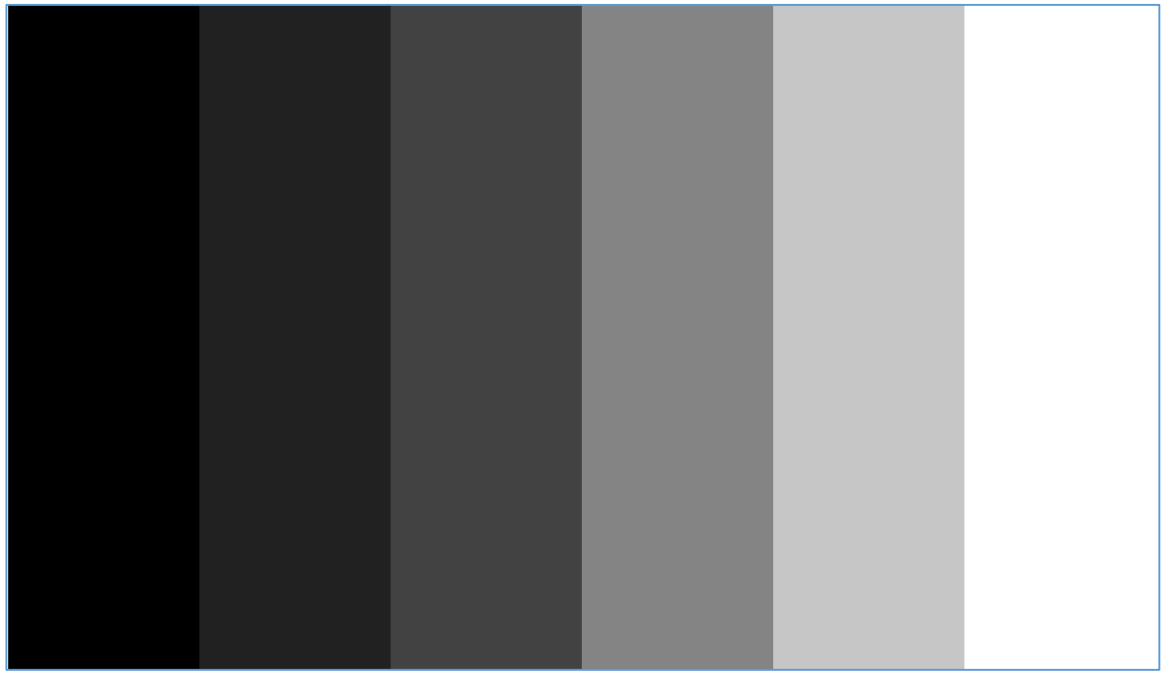
iii. **Medical** and microscope images use a higher bit depth, as scientific applications demand higher accuracy. A 16-bit medical image will have values in the range [0; 65535].

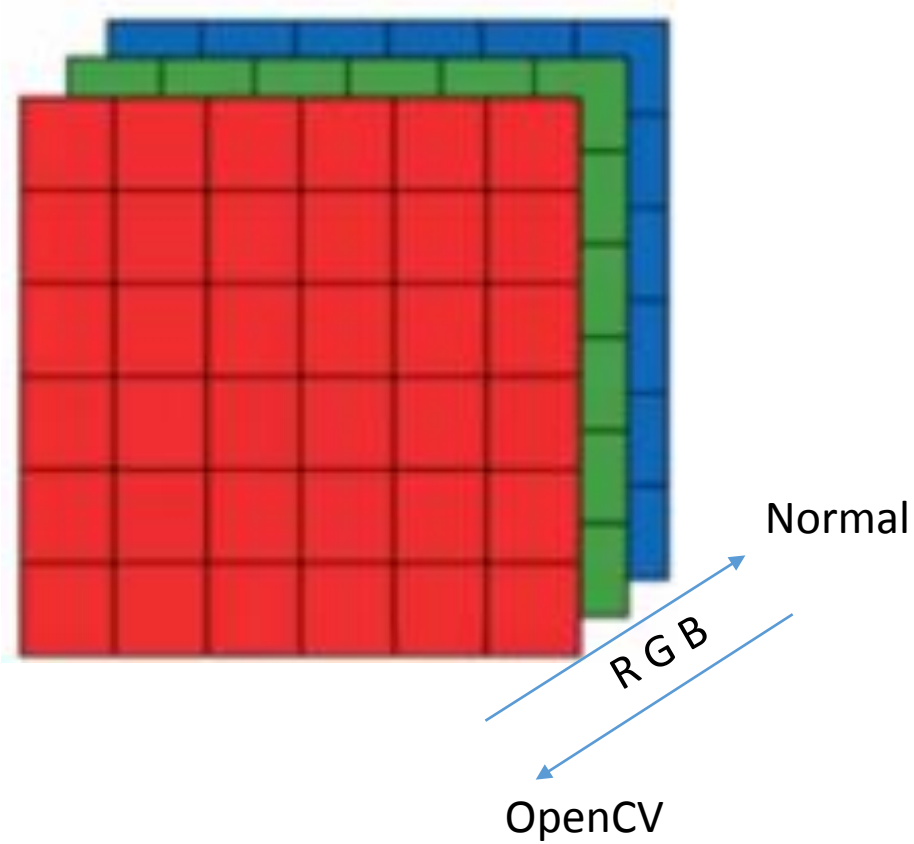
iv. **Scientific** image formats store the pixel values at high precision not only for accuracy but also to ensure that physical phenomenon that it records is not lost. In CT, for example a pixel value of > **1000 indicates bone**.

v. There are a few image formats that store images at even higher bit-depth such as 32 or 64. For example, a jpeg image containing **RGB** (3 **channels**) will have a bit-depth of 8 for each channel and hence has a total bit-depth of **24**.

vi. A microscope image with 5 **channels** (say) with each channel at **16-bit** depth will have a total bit-depth of **80**.

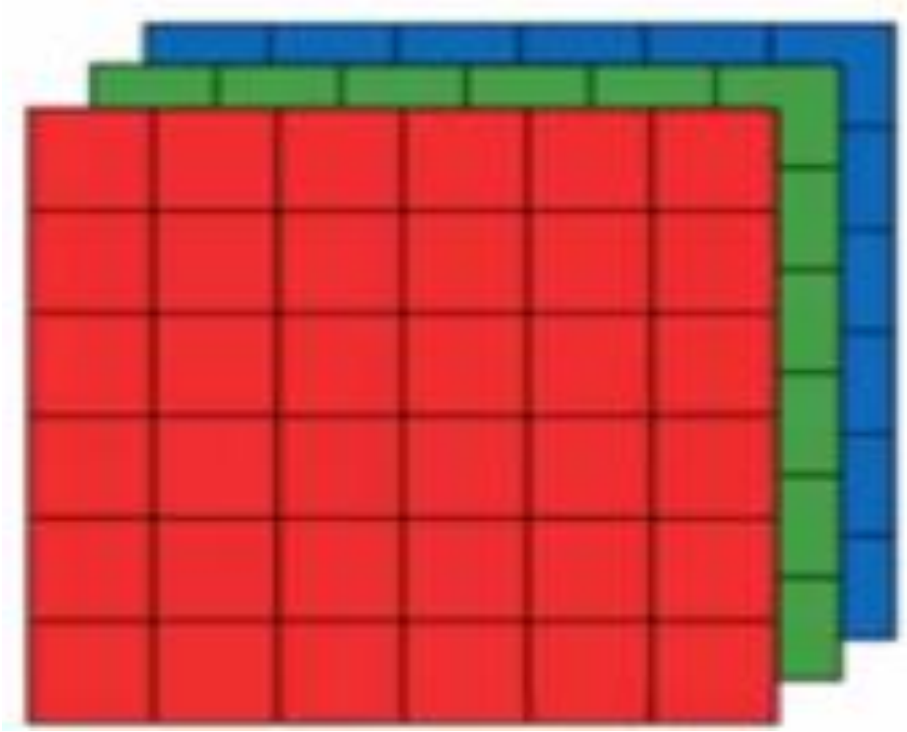
Black vs Gray vs Color image





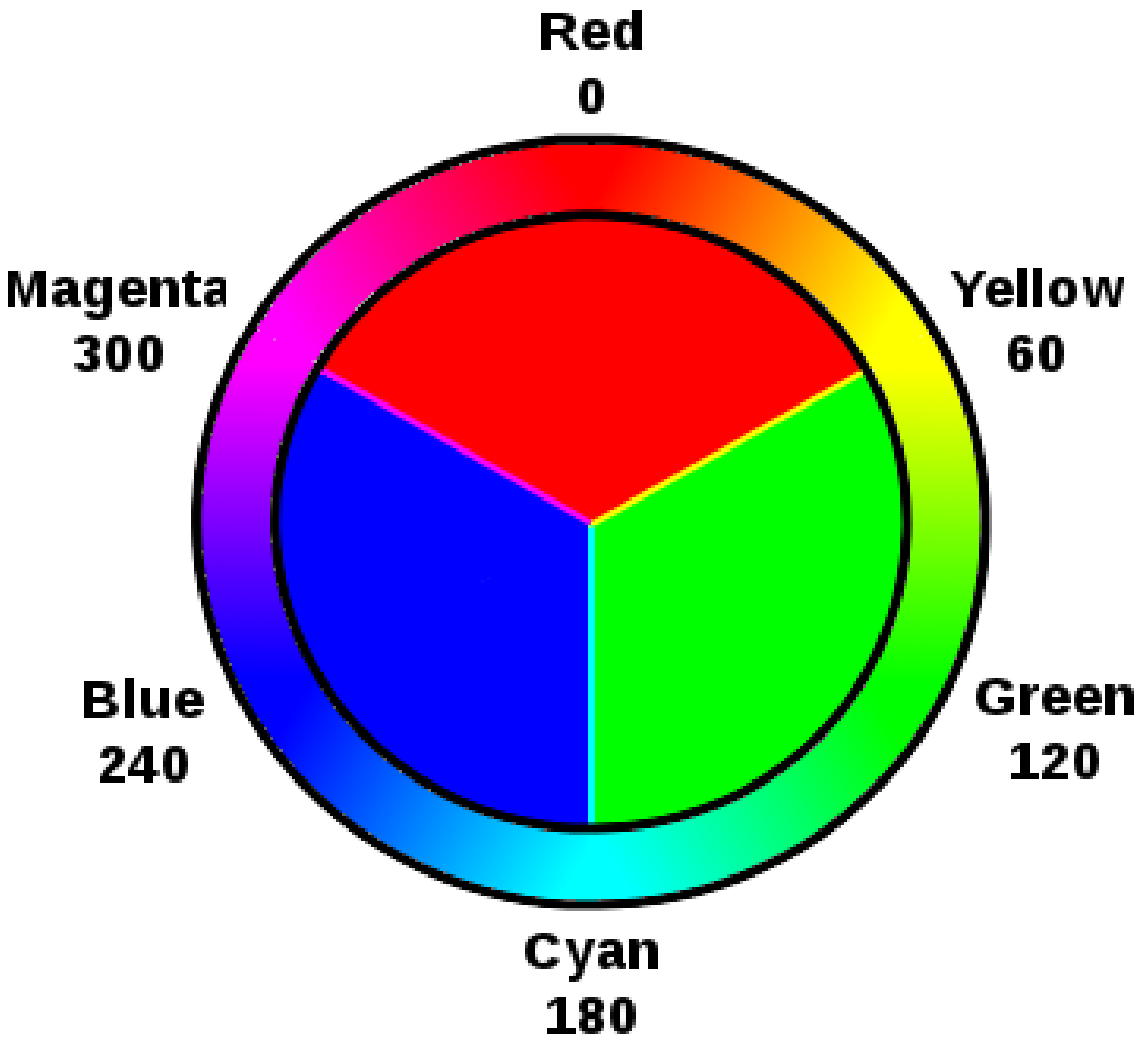
Color Spaces - RGB

- The most common color space is RGB (Red Green Blue). Each component can take a value between 0 and 255, where the tuple (0, 0, 0) represents black and (255, 255, 255) represents white, Red (255, 0, 0), Orange(255, 128, 0) and Pink (255, 153, 255)
- RGB is considered an "additive" color space, and colors can be imagined as being produced from shining quantities of red, blue, and green light onto a black background.



HSV: Hue Saturation Value (https://en.wikipedia.org/wiki/HSL_and_HSV)

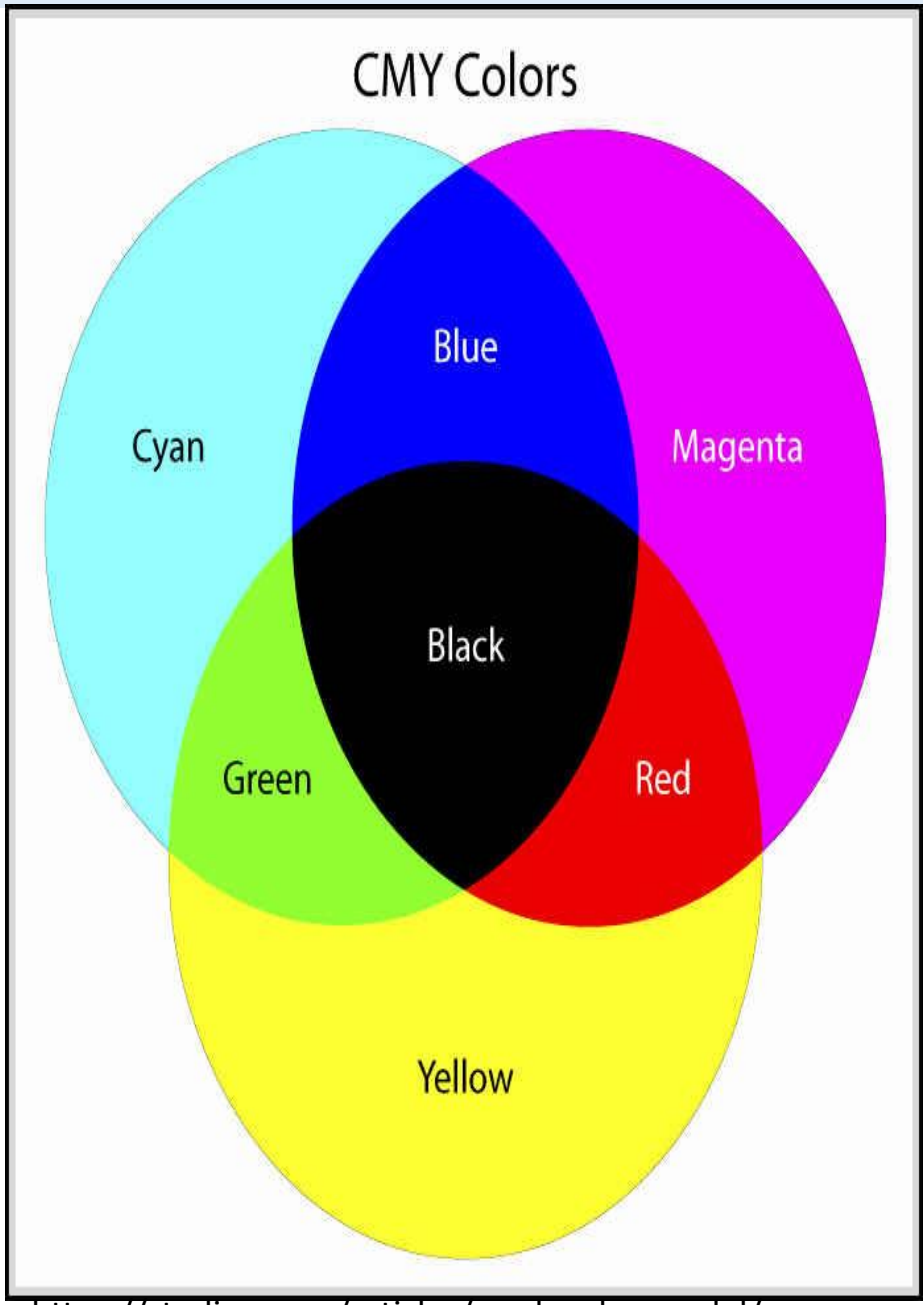
- It help to pinpoint a specific color, based on hue and saturation ranges, with a variance of value. Used in color selection tools in software and for web design
- **H**ue for color, **S**aturation for the strength of the color, and **V**alue for light
- In OpenCV, Hue range is [0,179], Saturation range is [0,255] and Value range is [0,255]. Different software use different scales



Color Spaces - cont

- In the printing world, CMYK is useful because it describes the color combinations required to produce a color from a white background. While the 0 tuple in RGB is black, in CMYK the 0 tuple is white. Our printers contain ink canisters of cvan. magenta, yellow, and black.

- In certain types of medical fields, glass slides mounted with stained tissue samples are scanned and saved as images. They can be analyzed in HED space, a representation of the saturations of the stain types—hematoxylin, eosin, and DAB—applied to the original tissue.

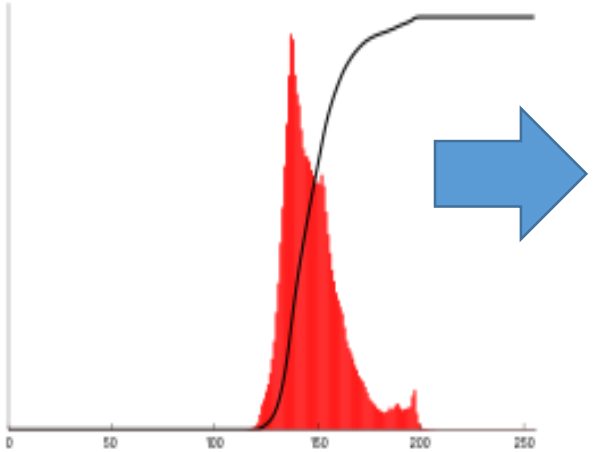


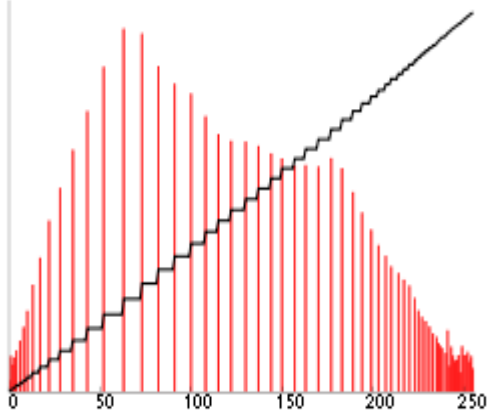
https://stodioo.com/articles/cmyk-color-model/

Hands on with file 1.1 image_sklearn_basics.py

Histogram equalization

- A method in image processing of contrast adjustment
- This method usually increases the global contrast and hence gain lower local contrast to gain a higher contrast.
- It spreads out the most frequent intensity values. (like Normalisation)
- The method is useful in images with backgrounds and foregrounds that are both bright or both dark.
- The calculation is not computationally intensive.
- It may increase the contrast of background noise, while decreasing the usable signal.





https://en.wikipedia.org/wiki/Histogram_equalization

Corresponding histogram (red) and cumulative histogram (black)

Image Filters

Filtering means - applying some visual effects to images as follows

i. Blurring, ii. Detecting edges, iii. Sharpening, iv. Enhance image contrast, v. Erode and Dilate an image *(All terms will be defined in detail during hands on)*

Convolution is a mathematical operation that is performed on two functions to produce a third function.

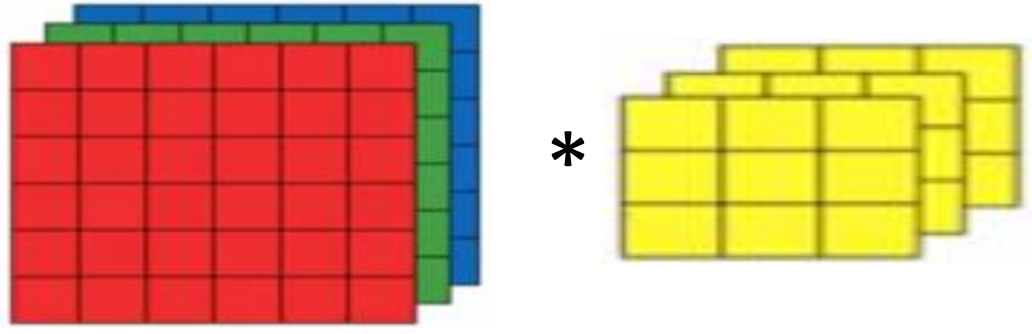
kernel is another matrix.

Following process is known as convolution.

i. Multiply the pixels under that kernel by the values in the kernel

ii. Sum all the those results

iii. Replace the top left pixel with the summed result



See the tab 'convolution" in "miscellaneous.xlsx file

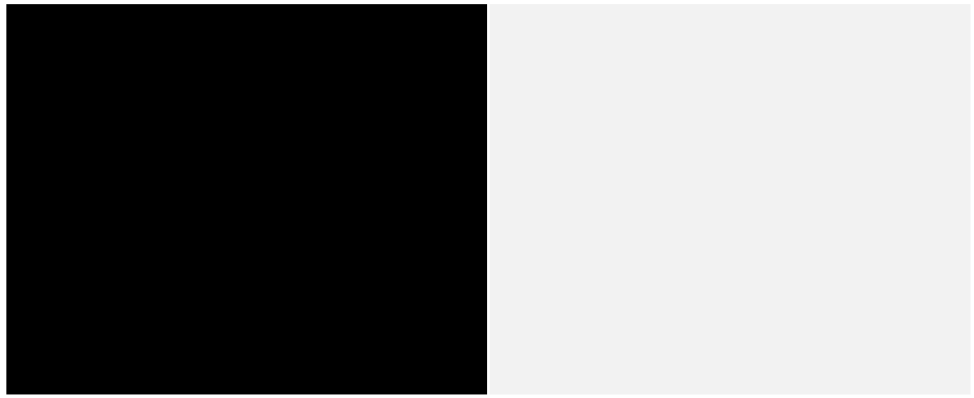
Types

1. Linear filters: Mean, Laplacian and Laplacian of Gaussian.
2. Nonlinear filters: Median, maximum, minimum, Sobel, Prewitt and Canny filters.

Hands on with file 1.2 image_sklearn_processing.py

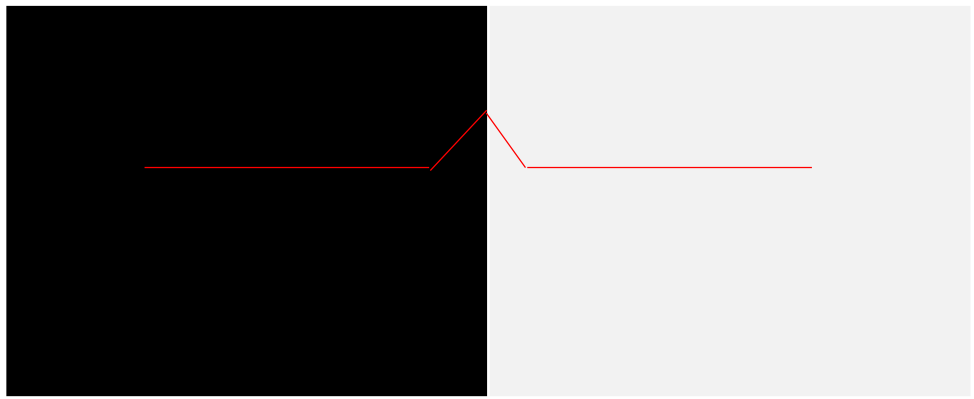
Edge Detection

Edges are a set of points in an image where there is a change of intensity. From calculus, we know that the changes in intensity can be measured by using the first or second derivative.

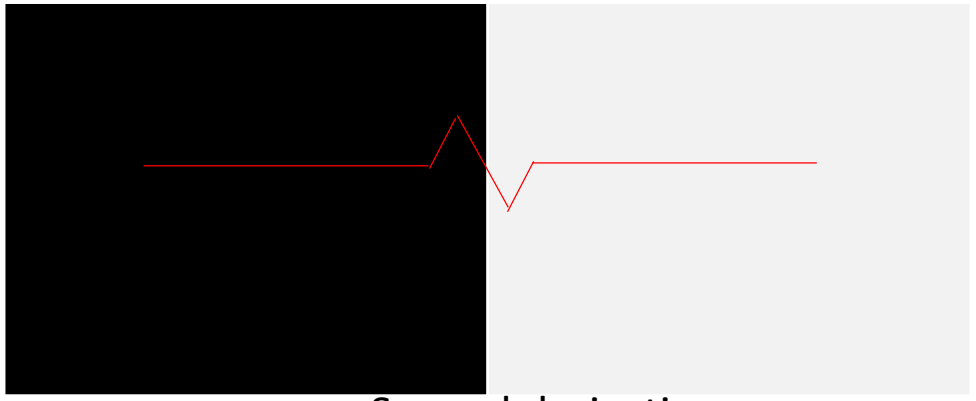


Image

Intensity



First derivative



Second derivative

First Derivative Filters

Sobel filter

- One of the most popular first derivative filters is the Sobel filter.
- The Sobel filter or mask is used to find horizontal and vertical edges as given here
- The sum of the coefficients in the mask image is 0. This means that the pixels with constant grayscale are not a affected by the derivative filter.
- The side effect of derivative filters is creation of additional noise. Hence, coefficients of +2 and -2 are used in the mask image to produce smoothing

-1	**-2**	**-1**
0	0	0
1	2	1

H

-1	**0**	**1**
-2	0	2
-1	0	1

V

Prewitt filter

The sum of the coefficients in the mask image is 0. However, the filter does not create noise as can be seen in the values of the coefficients.

-1	**-1**	**-1**
0	0	0
1	1	1

H

-1	**0**	**1**
-1	0	1
-1	0	1

V

Canny

This filter uses three parameters to detect edges. The first parameter is the standard deviation , ‘sigma’ for Gaussian filter. The second and third parameters are the threshold values, t1 and t2.

Source: Wikipedia

Second Derivative Filters: Laplacian filter

This filter uses three parameters to detect edges. The first parameter is the standard deviation , ‘sigma’ for Gaussian filter. The second and third parameters are the threshold values, t1 and t2.

Positive Laplacian Operator for outward edges

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

Negative Laplacian Operator for inward edges

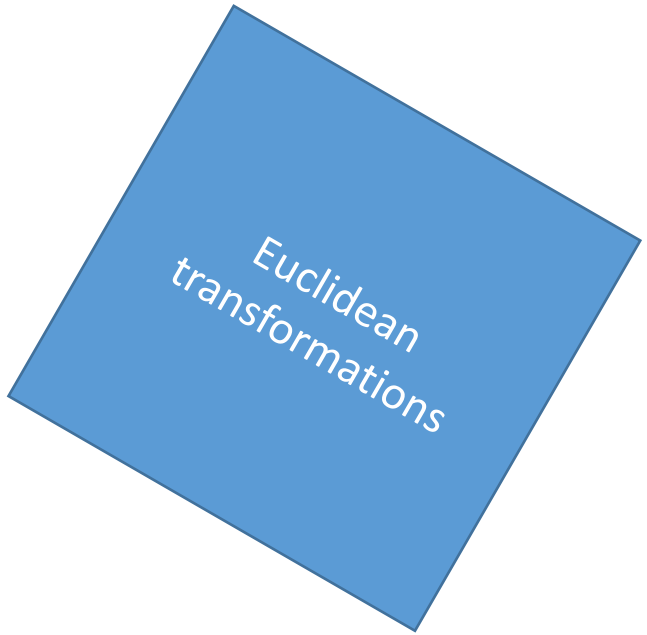
0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Hands on with file 2.1 image_opencv_basics.py

Transformations

Euclidean transformations



Affine transformations

Similarity transformations

Similarity transformations

Data Augmentation

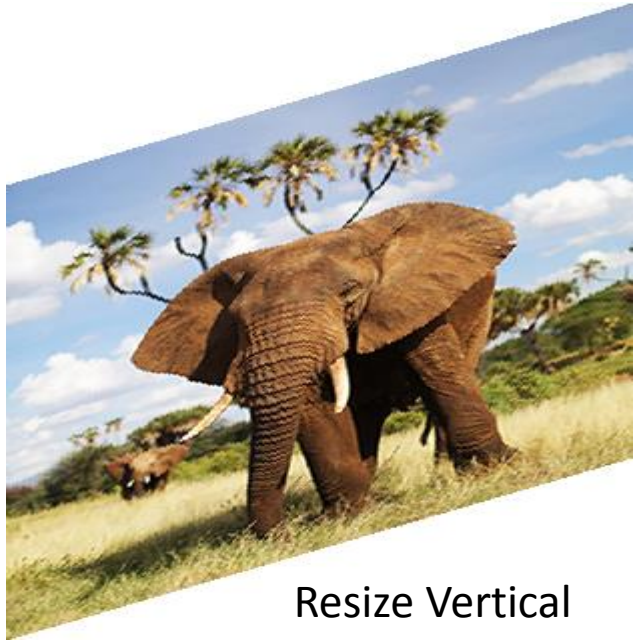
Original

Flip Horizontal

Resize Horizontal

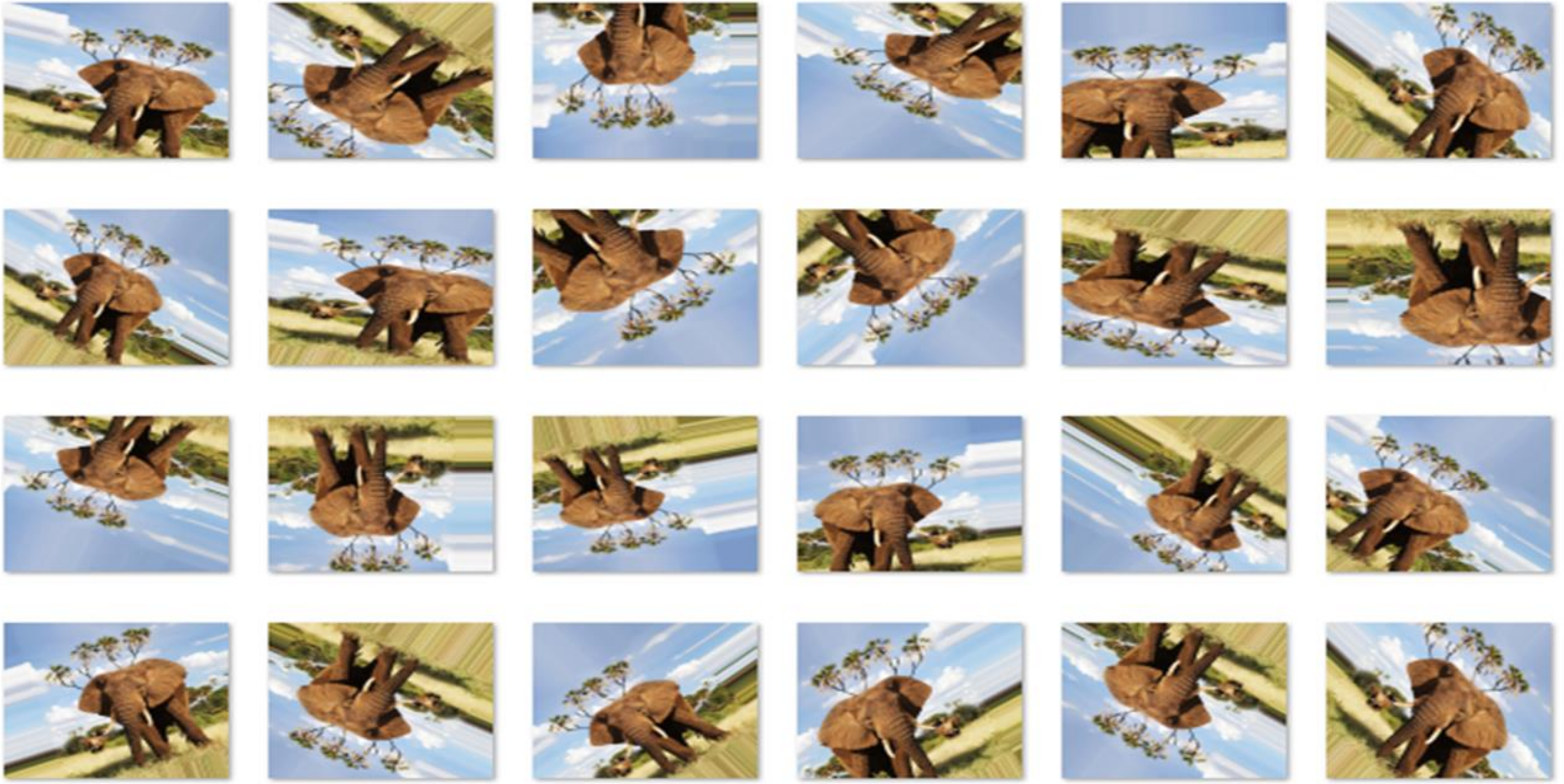
Random transformation

Cropping

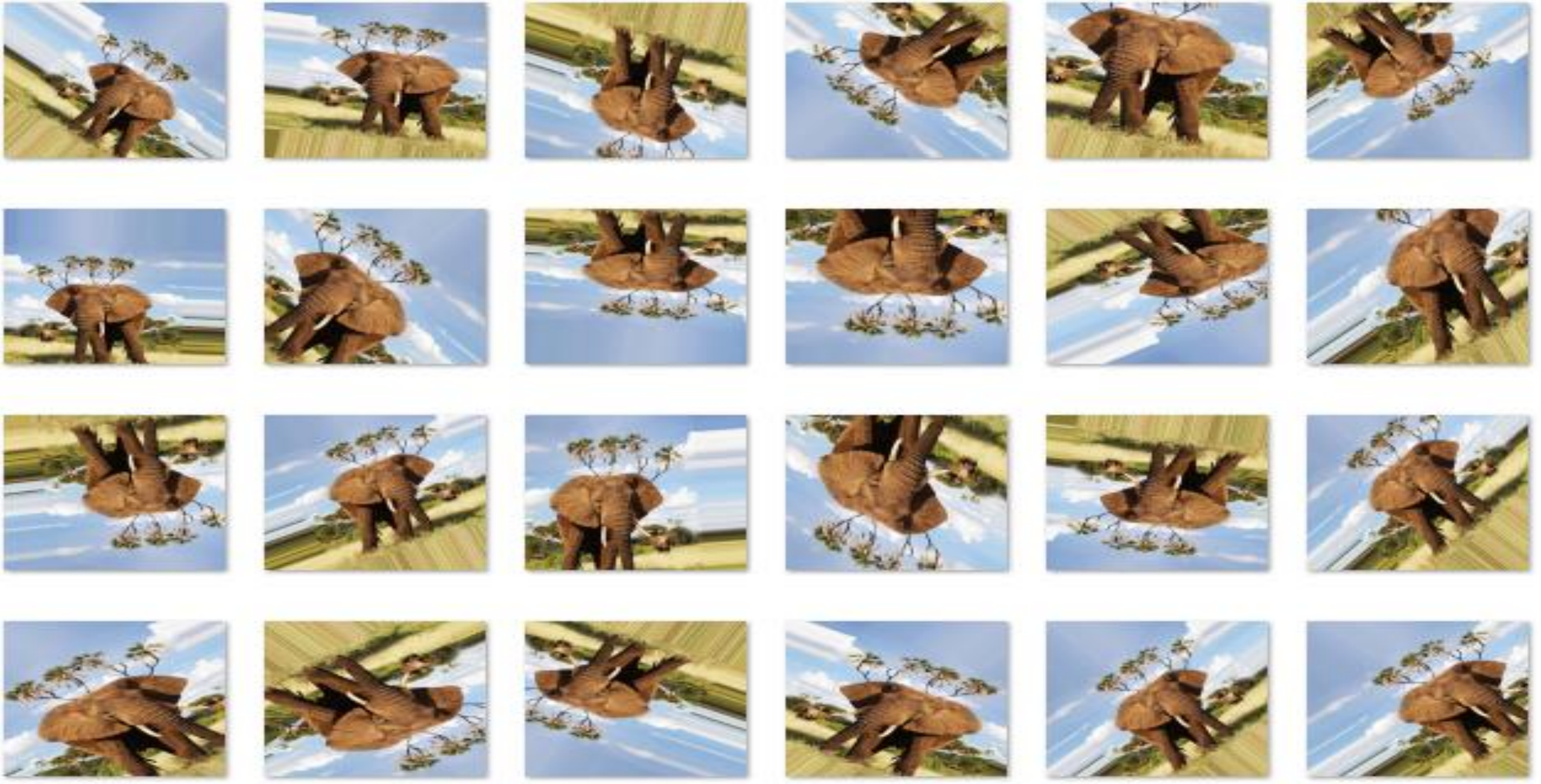


Resize Vertical

Data Augmentation cont



Data Augmentation cont



Hands on with file 2.2 image_opencv_filtering.py

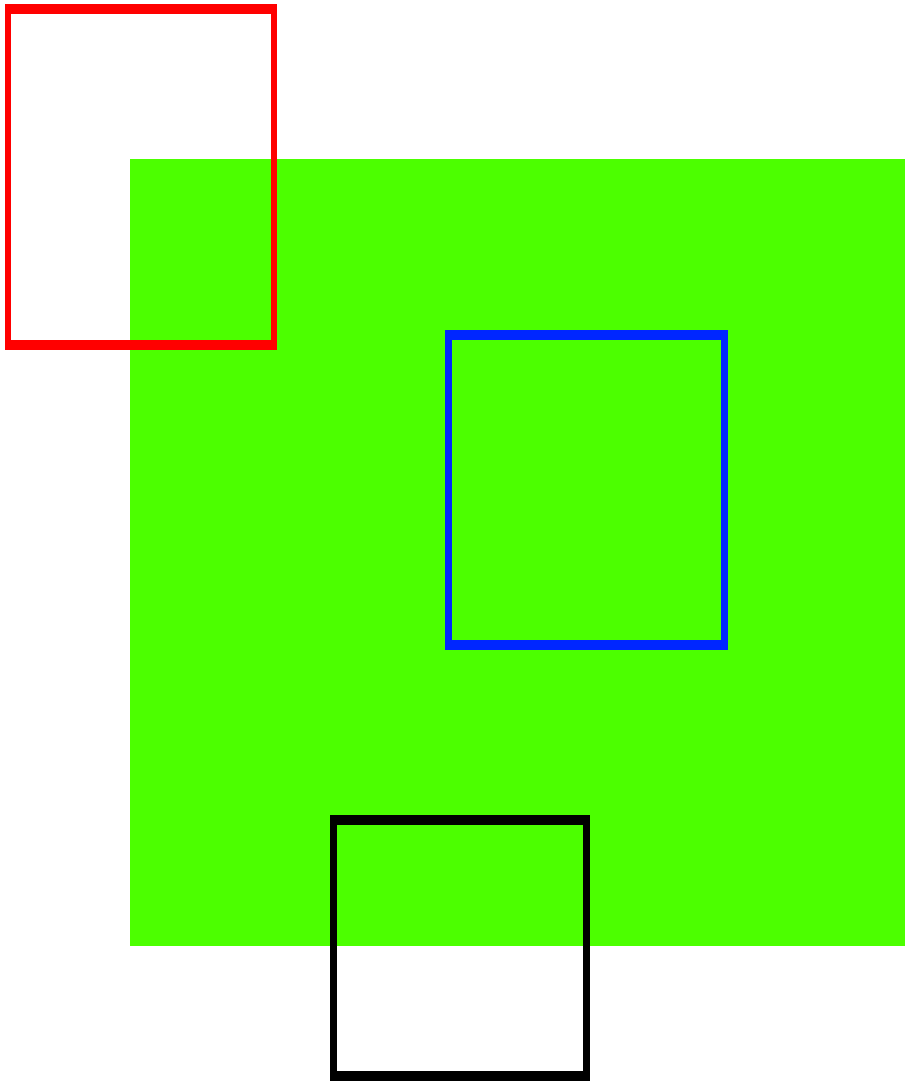
Hands on with file 2.3 image_opencv_arithmetic.py

Hands on with file 2.4 image_opencv_contours.py

Hands on with file 3. image_opencv_face_detection.py

Understanding Features

- See all 3 patches - blue patch (on flat area), black patch (on edge) and red patch (on corner) -> C**orners are considered to be good features in an image**
- How do we find the corners ->Find the regions which have maximum variation when moved (by a small amount) in all regions around it -> Finding these image features is called **Feature Detection**.
- Above will provide pattern -> Model is trained to find similar pattern in other images -> Called **Feature Description -> Now model** can find same features

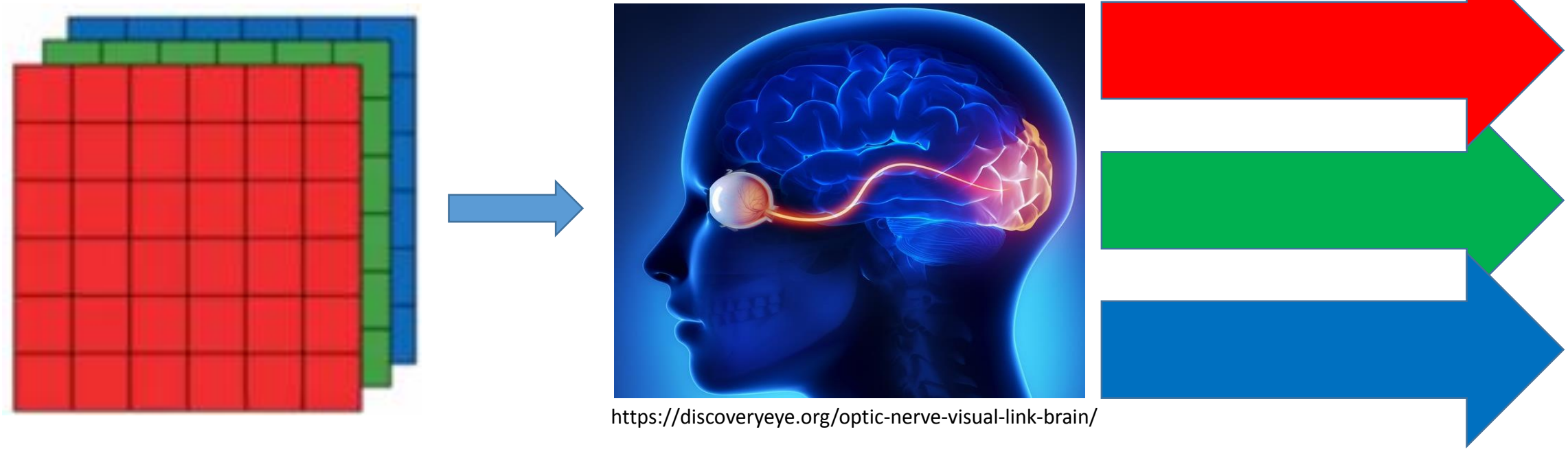


Few well known techniques

- **Harris Corner Detector** (Original Paper: http://www.bmva.org/bmvc/1988/avc-88-023.pdf)
  - 2x2 matrix based on partial derivatives of the grayscale image
  - Analyze the eigenvalues obtained
  - The high value of eigenvalues indicates - corner point
- **Harris Corner with Subpixel Accuracy**: To find the corners with maximum accuracy. It uses centroids (of bunch of pixels at a corner) to refine them. Harris corners are marked in red pixels and refined corners are marked in green pixels in code
- **ORB** is basically a fusion of FAST key point detector and BRIEF descriptor with many modifications to enhance the performance. First it use FAST to find key points, then apply Harris corner measure to find top N points among them.
- **ORB (Oriented FAST and Rotated BRIEF)**: it is a good alternative to SIFT and SURF in computation cost, matching performance and mainly the patents. Yes, SIFT and SURF are patented and you are supposed to pay them for its use. But ORB is not ☺

Extract features/key points

How humans do it. Our brain is an extremely powerful machine that can do complicated things very quickly. When we look at something, our brain automatically creates a footprint based on the interesting aspects of that image.



https://discoveryeye.org/optic-nerve-visual-link-brain/

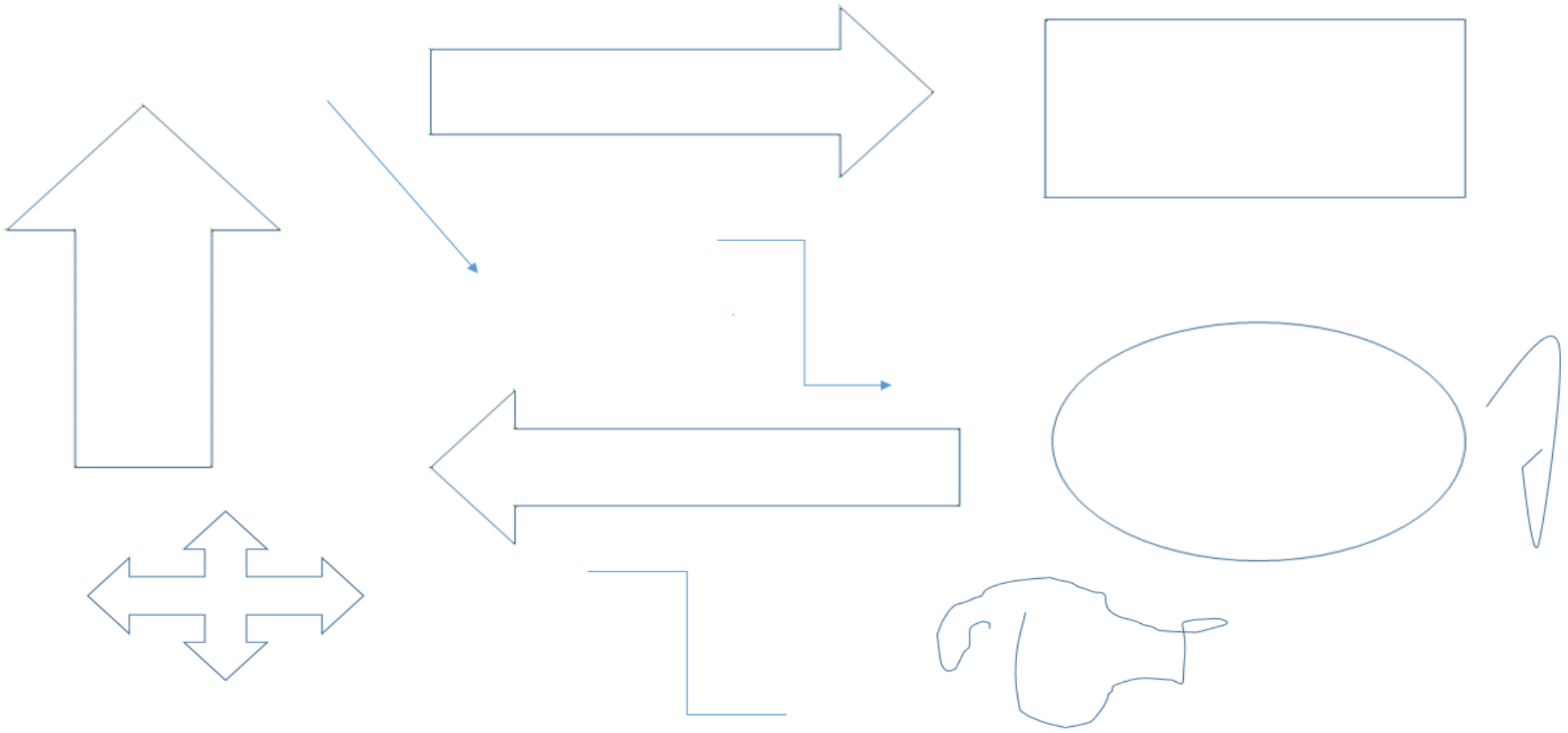
Extract features/key points: Blank image

http://www.outline-world-map.com/blank-world-maps-solid

Extract features/key points: Color image

Extract features/key points: What do you see?

distinct

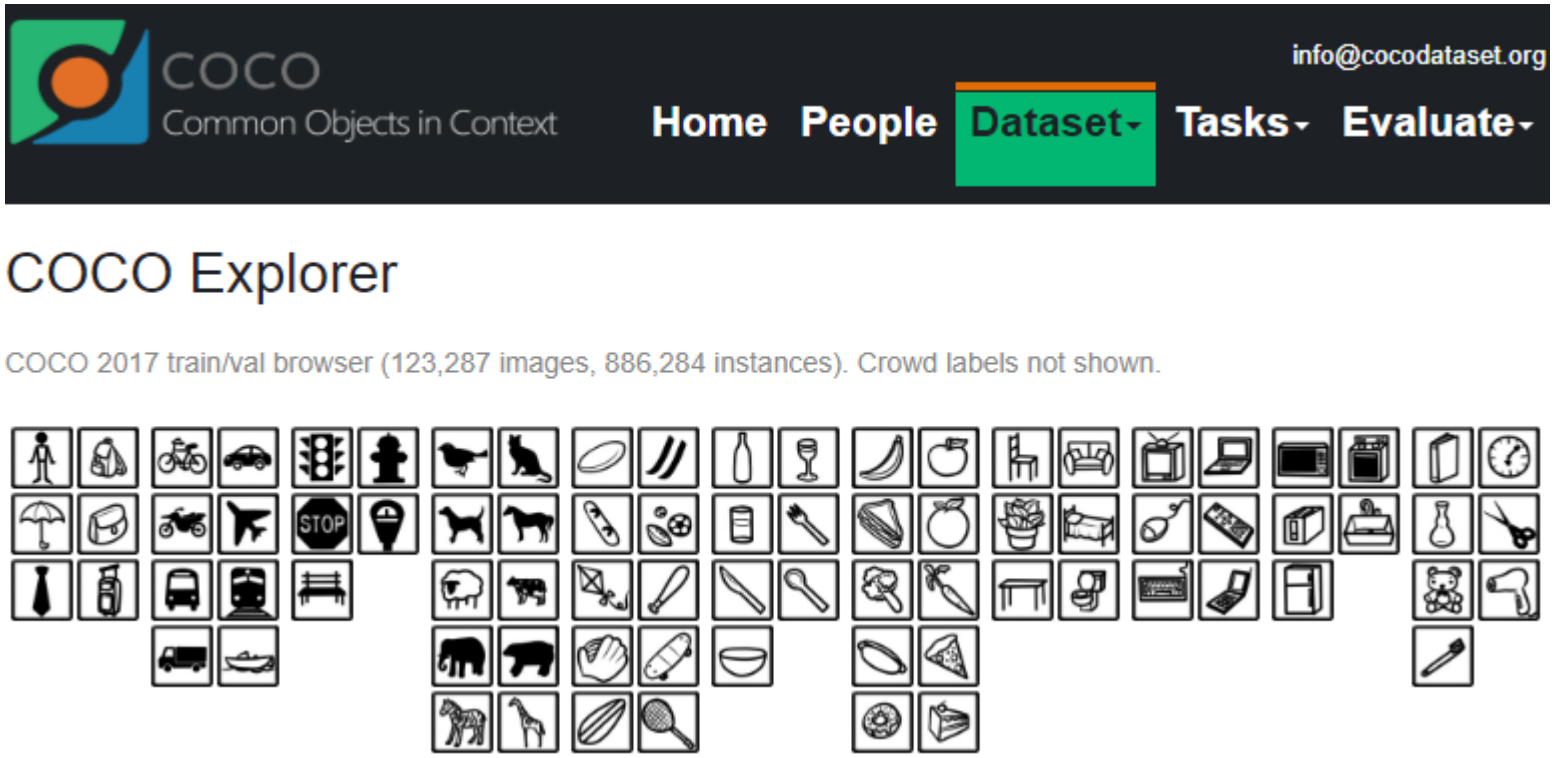


Hands on with file 4. image_opencv_feature_detection.py

Hands on with file 5. image_opencv_image_matching.py

Tensorflow Object Detection API

he API has been trained on the COCO dataset (Common Objects in Context) (http://cocodataset.org/). This is a dataset of 300k images of 90 most commonly found objects. Examples of objects includes:



Install steps are in ‘install.txt’. Please see section ‘TensorFlow Object detection’

Hands on with file 6. image_tensorflow_object_detection.py

Hands on with file 7. image_digit_recognition.py

8. image_opencv_segmentation.py

Miscellaneous

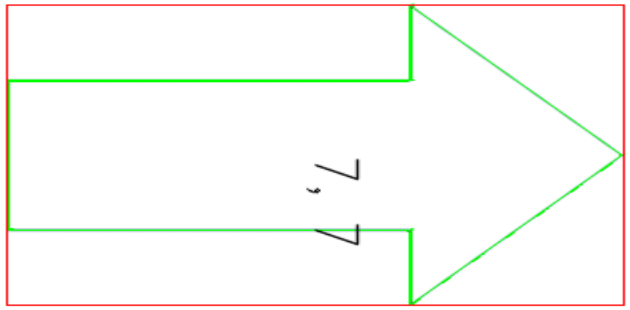
Definitions

- Morphological operations are the set of operations that process images according to their shapes.
- Erosion: Erosion is quite a similar process as dilation. But the pixel value computed here is minimum rather than maximum in dilation. The image is replaced under the anchor point with that minimum pixel value. With this procedure, the areas of dark regions grow in size and bright regions reduce. For example, the size of an object in dark shade or black shade increases, while it decreases in white shade or bright shade.
- Dilation: This procedure follows convolution with some kernel of a specific shape such as a square or a circle. This kernel has an anchor point, which denotes its center. This kernel is overlapped over the picture to compute maximum pixel value. After calculating, the picture is replaced with anchor at the center. With this procedure, the areas of bright regions grow in size and hence the image size increases.
- HOG is a feature extraction technique. That means HOG is a compressed & encoded version of your image.

https://www.quora.com/What-are-HOG-features-in-computer-vision-in-laymans-terms

Definitions cont

- Contours: it is set of boundaries of all the shapes that are found in the image. Contour analysis is a very useful tool in the field of computer vision. We deal with a lot of shapes in the real world, and contour analysis helps in analyzing those shapes using various algorithms.
- Convex (curves which are always bulged out or at-least flat) shapes (also known as Closed shapes) : For any two points within shapes and draw a line between them, and that line will always lie within that shape.
- Convex hull:



- Solidity factor: It is the ratio of the area of shape to the area of the convex hull. This metric will have a lower value for the irregular shapes because of the empty area that will be left out.
- Convexity defect: Any deviation of the object from convex hull

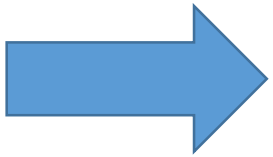
Best Practices

- Start with image and then move to Video
- To speed up video – process 1 frame per 29 frames (say 1sec)
- Infrastructure – as high as possible

Do you know Everything?

Seam carving

- https://en.wikipedia.org/wiki/Seam_carving
- Paper: http://dl.acm.org/citation.cfm?id=1276390



References

- https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_setup/py_intro/py_intro.html#introhttps://realpython.com/tutorials/computer-vision/
- https://www.pyimagesearch.com/
- https://pythonspot.com/vision/
- https://courses.learnopencv.com/p/opencv-for-beginners
- https://www.tensorflow.org/api_guides/python/image
- https://docs.opencv.org/3.4.5/d3/d05/tutorial_py_table_of_contents_contours.html
- https://en.wikipedia.org/wiki/Image_moment
- https://en.wikipedia.org/wiki/Histogram_equalization
- OpenCV 3.x with Python By Example by Gabriel Garrido and Prateek Joshi
- https://docs.opencv.org/3.4.5/d0/d86/tutorial_py_image_arithmetics.html
- https://pythonprogramming.net/image-arithmetics-logic-python-opencv-tutorial/
- HSV color range: http://answers.opencv.org/question/100647/how-to-remove-green-color-ie-to-set-it-to-0-in-an-image/
- https://stackoverflow.com/questions/10948589/choosing-the-correct-upper-and-lower-hsv-boundaries-for-color-detection-withcv
- https://pythonprogramming.net/grabcut-foreground-extraction-python-opencv-tutorial/
- http://www.ritsumei.ac.jp/~akitaoka/index-e.html
- https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/index.html
- https://www.pyimagesearch.com/2014/09/15/python-compare-two-images/
- http://scikit-image.org/docs/dev/auto_examples/features_detection/plot_blob.html#sphx-glr-auto-examples-features-detection-plot-blob-py
- http://scikit-image.org/docs/dev/auto_examples/segmentation/plot_chan_vese.html#sphx-glr-auto-examples-segmentation-plot-chan-vese-py
- http://scikit-image.org/docs/dev/auto_examples/
- https://github.com/scikit-image/skimage-tutorials/tree/master/
- https://www.pyimagesearch.com/pyimagesearch-gurus/

References

- Using TensorFlow: https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/index.html
- #Using retinanet -> https://www.analyticsvidhya.com/blog/2018/06/understanding-building-object-detection-model-python/?utm_source=feedburner&utm_medium=email&utm_campaign=Feed%3A+AnalyticsVidhya+%28Analytics+Vidhya%29
- # Using retinanet https://github.com/OlafenwaMoses/ImageAI/releases/download/1.0/resnet50_coco_best_v2.0.1.h5
- #and ImageAI pip install https://github.com/OlafenwaMoses/ImageAI/releases/download/2.0.1/imageai-2.0.1-py3-none-any.whl