Python: Artificial Intelligence & Deep Neural Networks

Installing OpenCV

Frame Differencing

Tracking Objects Using Colorspaces

Object Tracking Using Background Subtraction

Building an Object Tracker Using the CAMShift Algorithm

Optical Flow Based Tracking

Face Detection and Tracking

OpenCV

OpenCV was started at Intel in 1999 by Gary Bradsky and the first release came out in 2000. Vadim Pisarevsky joined Gary Bradsky to manage Intel’s Russian software OpenCV team. In 2005, OpenCV was used on Stanley, the vehicle who won 2005 DARPA Grand Challenge. Later its active development continued under the support of Willow Garage, with Gary Bradsky and Vadim Pisarevsky leading the project. Right now, OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning and it is expanding day-by-day.

Currently OpenCV supports a wide variety of programming languages like C++, Python, Java etc and is available on different platforms including Windows, Linux, OS X, Android, iOS etc. Also, interfaces based on CUDA and OpenCL are also under active development for high-speed GPU operations.

OpenCV-Python is the Python API of OpenCV. It combines the best qualities of OpenCV C++ API and Python language.

pip install opencv-python

import cv2

import matplotlib

import numpy

Loading image

# colored Image

Img = cv2.imread (“Penguins.jpg”,1)

# Black and White (gray scale)

Img\_1 = cv2.imread (“Penguins.jpg”,0)

# Black and White (gray scale)

Img = cv2.imread (“Penguins.jpg”,0)

Print(img.shape)

Resizing the image

# Black and White (gray scale)

img = cv2.imread (“Penguins.jpg”,0)

resized\_image = cv2.resize(img, (650,500))

cv2.imshow(“Penguins”, resized\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()

img = cv2.imread('watch.jpg',cv2.IMREAD\_GRAYSCALE)

cv2.imshow('image',img)

cv2.waitKey(0)

cv2.destroyAllWindows()

import cv2

import numpy as np

from matplotlib import pyplot as plt

img = cv2.imread('watch.jpg',cv2.IMREAD\_GRAYSCALE)

plt.imshow(img, cmap = 'gray', interpolation = 'bicubic')

plt.xticks([]), plt.yticks([]) # to hide tick values on X and Y axis

plt.plot([200,300,400],[100,200,300],'c', linewidth=5)

plt.show()

cv2.imwrite('watchgray.png',img)