[Video Processing Course](http://hwshow-vp.blogspot.tw/2010/01/course-of-video-processing.html)

The goal of this course is to write programs of OpenCV based on Python to develop projects for videos.

Python is Now (July 2014) the Most Popular Introductory Teaching Language at Top U.S. Universities.   
Eight of the top 10 CS departments (80%), and 27 of the top 39 (69%), teach Python in introductory CS0 or CS1 courses.

OpenCV is an open source (see http://opensource.org) computer vision library available from http://SourceForge.net/projects/opencvlibrary. The library is written in C and C++ and runs under Linux, Windows and Mac OS X. OpenCV was designed for computational efficiency and with a strong focus on realtime applications. One of OpenCV’s goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. The OpenCV library contains over 500 functions that span many areas in vision, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision, and robotics.  
  
The evaluation of the score:  
1. Homeworks - 10%

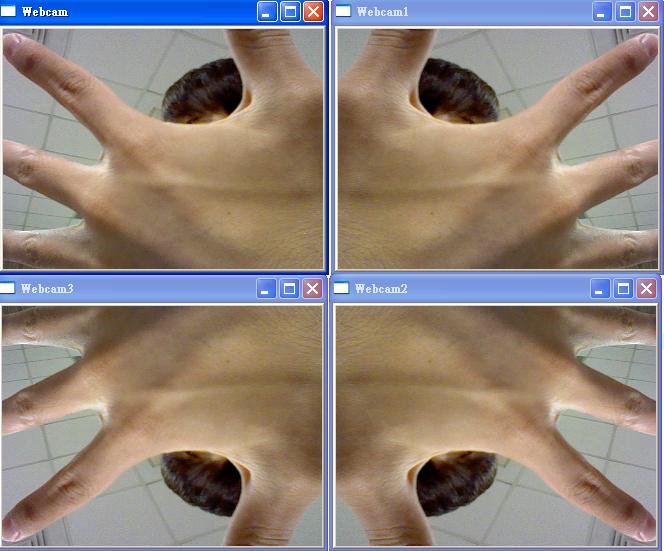
2. Participations – 30%  
3. Two exams - 20%  
4. Two projects - 40%

Rules of the course of Video Processing  
1. ‘Learning by coding’ is the distinguishing feature of the course.  
- Each class is divided into two sections: Instruction and experiment.  
2. Discussions are very much encouraged but ONLY during the section of experiment.  
3. No discussion is allowed while broadcasting demonstrations and instructions. For any violation, 20 points will be deducted from your final score each time.  
4. Students are asked to be on time in the class. For any violation, 5 points will be deducted from your final score each time.  
9:30 會點名 一次不到者 扣總成績 5 分

- Team work is encouraged in this course and two students are allowed in each team. Find your partner right now and have a seat with your partner in this semester.

### [HW01: Capture video from WebCam](http://hwshow-vp.blogspot.tw/2010/01/hw01-capture-video-from-webcam.html)

1. Download 11\_Capture\_Video\_from\_Camera.py.  
   Excute and trace the program.  
   Use PrintScreen to capture one image from the video of WebCam.  
   From Internet, study functions of cv2.VideoCapture(0), cap.read(), cv2.cvtColor(), cv2.imshow(), cv2.waitKey(), cap.release() and cv2.destroyAllWindows().  
   Use cv2.flip() and cv2.resize() functions to extend the code to generate three additional windows. Arrange these windows such that the upper left window is the original one and the other three windows have the effects like the following.



1. Download 13\_Saving\_a\_Video.py.  
   Excute and trace the program.
2. Download 14\_Playing\_Video\_from\_file.py.  
   Excute and trace the program.
3. Download 15\_Multi\_Camera.py.  
   Excute and trace the program.

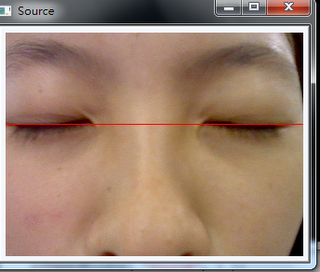
[HW02: Image-Level Processing in OpenCV](http://hwshow-vp.blogspot.tw/2010/01/hw02-image-level-processing-in-opencv.html)

1. Download 21\_Read\_Image.py, 22\_Read\_Image\_from\_System.py, 23\_Thresholding\_in\_Image, 24\_Trackbar\_Thresholding\_in\_Image.
2. Execute and trace the above programs.
3. Study cv2.Canny() to detect edge points in an image from Internet.
4. Extend your program to have the effect like 25\_Cany\_in\_Image.py.
5. Extend 11\_webcam.py to generate three videos. The first one is gray-level video. The second one is the black and white video. The third one is a video with edges only.   
   [](http://4.bp.blogspot.com/_jwZ3P1V1xNI/S52owjwyqwI/AAAAAAAAF_I/Ysqnc3oQZLE/s1600-h/000.jpg)

[Line Detection](http://hwshow-vp.blogspot.tw/2010/02/hw12-line-and-circle-detection.html):

Hough Transform: <http://en.wikipedia.org/wiki/Hough_transform>

How many lines and circles on your body?  
1. Download 27\_HoughLines\_in\_Image.py and line.jpg.  
2. Trace the above program.  
3. Modify it such that the program can find lines on your body.

[](http://4.bp.blogspot.com/-BCh6qpvObVM/TcQEe7sa7dI/AAAAAAAAGEI/pbJoiz-C8bU/s1600/2.png)

[HW03: Pixel-Level Processing in OpenCV](http://hwshow-vp.blogspot.tw/2010/01/hw03-pixel-level-processing-in-opencv.html)

* Download 31\_Splitting\_and\_Merging\_Image\_Channels.py.
* Execute and trace the above program.
* Extend your program to have the effect like 32\_Splitting\_Viedo\_Channels.py.



* Download 33\_Slpitting\_Image\_Channels\_by\_Pixel\_Processing.py.
* Execute and trace the above program.
* Design a program with pixel processing technique to have the effect like 32\_Splitting\_Viedo\_Channels.py.
* Download 34\_Insert\_Logo\_in\_Image.py.
* Execute and trace the above program.
* Extend your program to have the effect like 35\_Insert\_logo\_in\_Video.py.

[](http://4.bp.blogspot.com/-142CzlO3DSQ/UUKOL1BIC3I/AAAAAAAAGIk/IBk-yfayG74/s1600/111111.jpg)

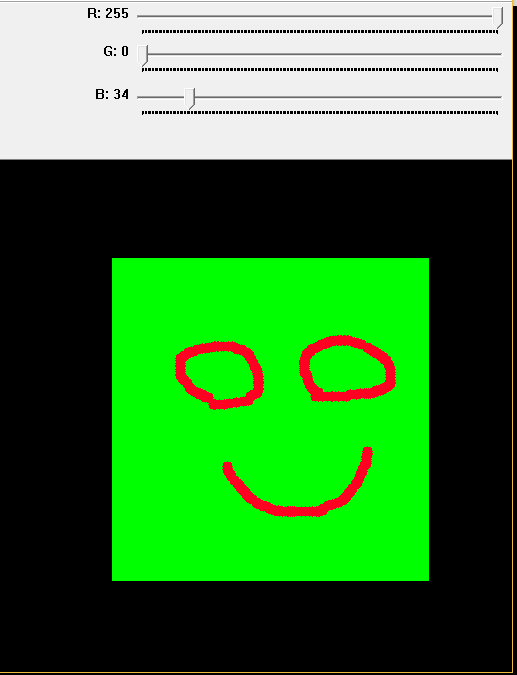
* Download 36\_Insert\_Logo\_in\_Video\_by\_Pixel\_Processing.py.
* Execute and trace the above program.
* Extend your program to have the effect like the following figure.



* Study functions of datetime.now() and cv2.putText() from Internet.
* Extend your program to have the effect like 37\_Put\_Text\_in\_video.py.
* Embed your English name in the video.



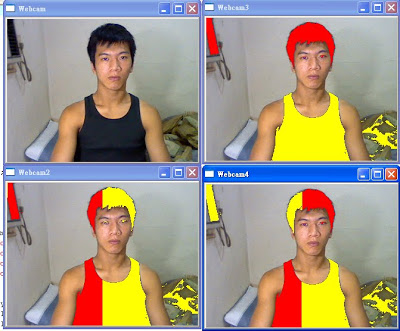
* Download 38\_Mouse\_callback\_function.py.
* Execute and trace the above program.
* Extend your program to have the effect like 39\_Drawing\_Board.py.



### [HW04: Color Detection](http://hwshow-vp.blogspot.tw/2010/01/hw04-color-detection.html)

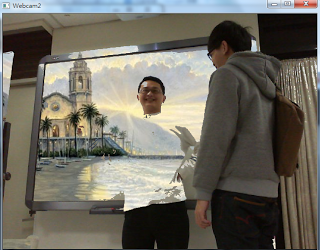
One of the most popular methods used for color detection is using the vector distance.  
  
Let's say you are searching for "red" pixels. Pure red exists when the values of the pixels are: [B, G, R]=[0, 0, 255]. When searching for red you may want to have a "tolerance" meaning that you don't only search for pure red, but also for some other red-like combinations.  
  
Suppose that each pixel is a vector: pixel = [B, G, R]  
Also, the color combination you are looking for is another vector: red = [0, 0, 255] (reference color).  
  
The eucledian distance between those 2 vectors is:  
D = sqrt((B-0)^2 + (G-0)^2 + (R-255)^2)  
  
So the algorithm goes:  
  
for each pixel {  
1. compute D between that pixel and the reference color  
2. if D < Threshold  
 then current pixel is accepted  
 else current pixel is NOT accepted  
}

* Download 41\_Color\_Detection\_in\_Image\_by\_Pixel\_Processing.py.
* Execute and trace the above program.
* Extend your program to a video captured by a webcam. 42\_Color\_Detection\_in\_Video\_by\_Pixel\_Processing.py
* 43\_Color\_Detection\_by\_Frame\_Processing.py
* 44\_\_Dilate\_and\_Erode\_to\_Reduce\_Noise.py
* 45\_Skin\_Detection.py
* 46\_Red\_Color\_Detection.py
* 47\_Black\_Color\_Detection.py
* 48\_Color\_Exchange.py
* 49\_Color\_Tracking.py

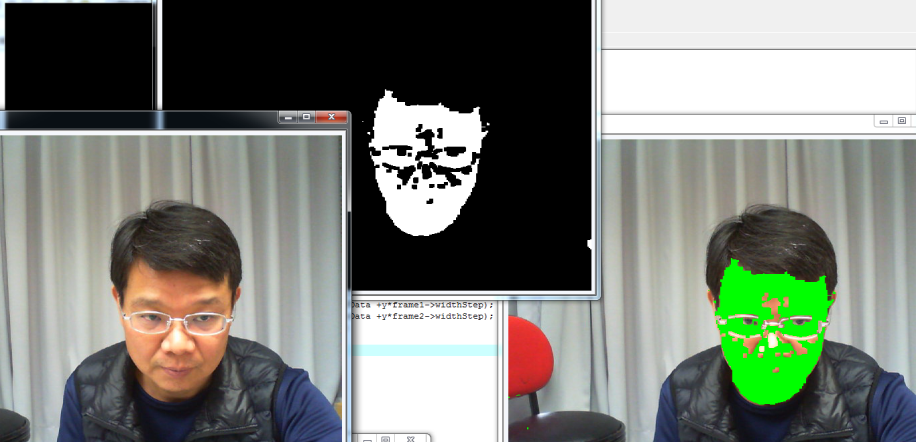
[](http://3.bp.blogspot.com/_jwZ3P1V1xNI/TBTPT-3Zr0I/AAAAAAAAGBE/_sjNN6gXvkA/s1600/hw4.jpg)

隱形人

[](http://4.bp.blogspot.com/-mMvvhJ5IsGg/UUKHfqM7UQI/AAAAAAAAGIY/XieoJIabhVY/s1600/hw04.jpg)

[](http://4.bp.blogspot.com/-GAqyjchsm2k/UVFUmeMge8I/AAAAAAAAGI4/2NR5yLIEO2I/s1600/%E9%80%8F%E6%98%8E%E4%BA%BA2.png)

面膜

[](http://2.bp.blogspot.com/-yJqYajjngMU/VTWO2PvYugI/AAAAAAAAGSA/bdCVWe8FIOw/s1600/01.png)

鋼鐵人:            面具自選   眼睛須去背 

[](http://4.bp.blogspot.com/-yp_097p-Iqk/UZWEbLljdmI/AAAAAAAAGLk/nmy2Esm2HoE/s1600/01.jpg)

[](http://1.bp.blogspot.com/-kRpr6e66uUU/UZWZCbCPlZI/AAAAAAAAGL0/_0nECdP74e0/s1600/001.png)

### [HW05: Face Detection](http://hwshow-vp.blogspot.tw/2010/01/hw04-color-detection.html)

51\_Face\_Eye\_Detection\_in\_Image.py

52\_Face\_Eye\_Detection\_in\_Webcam.py

53\_Face\_Off\_in\_Image.py

54\_Face\_Off\_in\_Webcam.py