we're going to be covering how to reduce the background of images, by detecting motion. This is going to require us to re-visit the use of video, or to have two images, one with the absense of people/objects you want to track, and another with the objects/people there. You can use your webcam if you like, or use a video like:

[**People walking sample video**.](https://pythonprogramming.net/static/images/opencv/people-walking.mp4)

The code here is actually quite simple with what we know up to this point:

import numpy as np

import cv2

cap = cv2.VideoCapture('people-walking.mp4')

fgbg = cv2.createBackgroundSubtractorMOG2()

while(1):

ret, frame = cap.read()

fgmask = fgbg.apply(frame)

cv2.imshow('fgmask',frame)

cv2.imshow('frame',fgmask)

k = cv2.waitKey(30) & 0xff

if k == 27:

break

cap.release()

cv2.destroyAllWindows()

Result:

The idea here is to extract the moving forground from the static background. You can also use this to compare two similar images, and immediately extract the differences between them.

In our case, we can see we've definitely detected some people, but we do have some "noise" The noise is actually tree leaves moving a bit in the ambient wind. If only we knew of a way to diminish noise. Oh wait! we do! A wild challenge has appeared for you +=1 folks!

The next tutorial begins to move us away from applying filters or transforms and gets us detecting general objects using Haar Cascades for things like face detection and more.