抽個空又把《OpenCV-Python-Tutorial-中文版》這本電子書看了一遍，這次看的時候帶著一個心思去看，就是整理每個章節的主要函數，查了每個函數的文檔

**函數名 函數**

**Chapter one：圖片**

讀入圖像 cv2.imread(filename[, flags]) → retval   
顯示圖像 cv2.imshow(winname, mat) → None   
cv2.waitKey([delay]) → retval   
創建窗口 cv2.namedWindow(winname[, flags]) → None   
摧毀窗口 cv2.destroyWindow(winname) → None   
cv2.destroyAllWindows()→ None   
保存圖像 cv2.imwrite(filename,img[, params]) → retval

**Chapter two: 視頻**

讀取視頻檔、攝像頭獲取視頻 cv2.VideoCapture(filename/device) →   
.read([image]) → retval,image   
.open()→ retval   
.isOpened()→ retval   
視頻檔案屬性、設置 .get(prodId) → retval   
.set(prodId,value) → retval   
保存視頻檔 cv2.VideoWriter([filename,fourcc,fps,framesize[,isColor]]) →   
.write(image) → None   
關閉視頻檔 .release()→ None

**Chapter three: 繪圖**

<http://wiki.opencv.org.cn/index.php/Cxcore%E7%BB%98%E5%9B%BE%E5%87%BD%E6%95%B0#GetTextSize>   
只有C++：InitFont、arrowedLine、

畫圓 cv2.circle(img,center,radius,color[,thickness[,lineType[,shift]]]) → None   
畫矩陣 cv2.rectangle(img,pt1,pt2,color[,thickness[,lineType[,shift]]]) → None   
畫線段 cv2.Line(img,pt1,pt2,color[,thickness[,lineType[,shift]]]) → None   
畫折線 cv2.polyLines(img,pts,isClosed,color[,thickness[,linrType[,shift]]]) → None   
剪切矩形內部的直線 cv2.clipLine(imgRect,pt1,pt2) → retval,pt1,pt2   
畫橢圓 cv2.ellipse(img,center,axes,angle,startAngle,endAngle,color[,thickness[,lineType[,shift]]]) → None   
cv2.ellipse(img,box,color[,thickness[,lineType]]) → None   
用折線逼近橢圓弧 cv2.ellipse2Poly(center,axes,angle,arcStart,arcEnd,delta) → pts   
填充多邊形 cv2.fillConvexPoly(img,points,color[,lineType[,shift]]) → None   
填充多邊形內部 cv2.fillPoly(img,pts,color[,lineType[,shift[,offset]]]) → None   
在圖像中顯示文本字串 cv2.putText(img,text,org,fontFace,fontScale,color[,thickness[,lineType[,bottomLeftOrigin]]]) → None   
獲得文本字串的寬度和高度 cv2.getTextSize(text,fontFace,fontScale,thickness) → retval, baseLine   
畫外部和內部的輪廓 cv2.drawContours(image,contours,contourIdx,color[,thickness[,lineType[,hierarchy[,maxLevel[,offset]]]]]) → None

**Chapter four: 處理滑鼠事件**

處理滑鼠事件 cv2.SetMouseCallback(winname,onMouse,param=None) → None

**Chapter five: 滑動條**

創建滑動條 cv2.createTrackbar(trackname,winname,value,count,onChange) → None   
return 滑動條屬性 cv2.getTrackbarPos(trackname,winname) → retval   
設置滑動條屬性 cv2.setTrackbarPos(trackname,winname,pos) → None

**Chapter six:圖像操作**

獲取圖像形狀 .shape   
獲取圖元數目 .size   
獲取圖像資料類型 .dtype   
拆分、合併通道 cv2.split(m[,mv]) → mv   
cv2.merge(mv[,dst]) → dst   
圖像擴邊 cv2.copyMakeBorder(src,top,bottom,left,right,borderType[,dst[,value]]) → dst   
圖像減法 cv2.subtract(src1,src2[,dst[,mask[,dtype]]])→dst   
圖像加法 cv2.add(src1,src2[,dst[,mask[,dtype]]]) → dst   
圖像混合 cv2.addWeighted(src1,alpha,sec2,beta,gamma[,dst[,dtype]]) → dst   
圖像位元運算 cv2.bitwise\_and(src1,src2[,dst[,mask]]) → dst   
cv2.bitwise\_not(src[,dst[,mask]]) → dst   
cv2.bitwise\_or(src1,src2[,dst[,mask]]) → dst   
cv2.bitwise\_xor(src1,src2[,dst[,mask]]) → dst

**Chapter seven: 顏色空間轉換**

轉換顏色空間 cv2.cvtColor(src,code[,dst[,dstcn]]) → dst   
檢查陣列元素是否在兩個數量之間 cv2.inRange(src,lowerb,upperb[,dst]) → dst

**Chapter eight: 幾何變換**

<http://www.cnblogs.com/dupuleng/articles/4055020.html>   
擴展縮放 cv2.resize(src,dsize[,dst[,fx[,fy[,interpolation]]]]) → dst   
仿射變換 cv2.getAffineTransform(src,dst) → retval   
cv2.warpAffine(src,M,dsize[,dst[,flags[,borderMode[,borderValue]]]]) → dst   
旋轉 cv2.getRotationMatrix2D(center,angle,scale)→ret   
透視變換 cv2.getPerspectiveTransform(src,dst) → retval   
cv2.warpPerspective(src,M,dsize[,dst[,flags[,borderMod[,borderValue]]]]) → dst

**Chapter nine:圖像閾值**

簡單閾值 cv2.threshold(src,thresh,maxvalue,type[,dst])→retval,dst   
自我調整閾值 cv2.adaptiveThreshold(src,maxvalue,adaptiveMethod,thresholdType,blockSize,[,dst]) → dst

**Chapter ten: 圖像平滑**

2D卷積 cv2.filter2D(src,ddepth,kernel[,dst[,anchor[,delta[,borderType]]]]) → dst   
平均 cv2.blur(src,ksize,[,dst[,anchor[,borderType]]])→dst   
cv2.boxFilter(src,ddepth,ksize[,dst[,anchor[,normalize[,borderType]]]]) → dst   
高斯模糊 cv2.GuassianBlur(src,ksize,sigmaX[,dst[,sigmaY[,borderType]]]) → dst   
cv2.getGuassianKernel(ksize,sigma[,kType]) → retval   
中值模糊 cv2.medianBlur(src,ksize[,dst]) → dst   
雙邊濾波 cv2.bilateralFilter(src,d,sigmaColor,sigmaSpace[,dst[,borderType]]) → dst

**Chapter eleven: 形態學轉換**

腐蝕 cv2.erode(src,kernel[,dst[,anchor[,iterations[,borderType[,borderType]]]]]) → dst   
膨脹 cv2.dilate(src,kernel[,dst[,anchor[,iterations[,borderType[,borderType]]]]]) → dst   
形態學操作（開運算，閉運算，梯度，禮帽，黑帽） cv2.morphologyEx(src,op,kernel[,dst[,anchor[,iterations[,borderType[,borderValue]]]]])→ dst   
結構化元素 cv2.getStructuringElement(shape,ksize[,anchor])→retval

**Chapter twelve: 圖像梯度**

Sobel運算元 cv2.Sobel(src,ddepth,dx,dy[,dst[,ksize[,scale[,delta[,borderType]]]]]) →dst   
Scharr運算元 cv2.Scharr(src,ddepth,dx,dy[,dst[,scale[,delta[,borderType]]]]) → dst   
Laplacian運算元 cv2.Laplacian(src,ddepth[,dst[,ksize[,scale[,delta[,borderType]]]]])→dst

**Chapter thirteen: canny邊緣檢測**

cv2.Canny(image,threshold1,threshold2[,edges[,apertureSize[,L2gradient]]]) → edges

**Chapter fourteen: 輪廓**

查找輪廓 cv2.findContours(image,mode,method[,contours[,hierarchy[,offset]]]) → contours,hierarchy   
繪製輪廓 cv2.drawContours(image,contours,contourIdx,color[,thickness[,lineType[,hierarchy[,maxLevel[,offset]]]]]) → None   
圖像的矩 cv2.moments(array[,binaryImage]) → retval   
輪廓面積 cv2.contourArea(contour[,oriented]) → retval   
輪廓周長 cv2.arcLength(curve,closed) → retval   
輪廓近似 cv2.approxPolyDP(curve,epsilon,closed[,approxCurvel])→approxCurvel   
凸包 cv2.convexHull(point[,hull[,clockwise[,returnPoints]]]) → hull   
凸缺陷 cv2.convexityDefects(contour,convexhull[,convexityDefects])→convexityDefects   
凸性檢測 cv2.isContourConvex(contour) → retval   
邊界矩形 cv2.boundingRect(points) → retval   
cv2.minAreaRect(points) → retval   
cv2.boxPoints()   
最小外接圓 cv2.minEnclosingCircle(points) → center,radius   
點到輪廓距離 cv2.pointPolygonTest(contour,pt,measureDist) → retval   
橢圓擬合 cv2.fitEllipse(points) → retval   
直線擬合 cv2.fitLine(points,distType,param,reps,aeps[,line]) → line   
形狀匹配 cv2.matchShapes(contour1,contour2,method,paramter) → retval   
模組匹配 cv2.matchTemplate(image,temp1,method[,result]) → result   
最值及其位置 cv2.minMaxLoc(src[,mask]) → minVal,maxVal,minLoc,maxLoc   
平均顏色/灰度 cv2.mean(src[,mask]) → retval

**Chapter fifteen: 長條圖**

繪製長條圖 cv2.calcHist(image,channels,mask,histSize,ranges[,hist[,accumulate]]) → hist   
長條圖均衡化 cv2.equalizeHist(src[,dst]) → dst   
cv2.createCLAHE([,clipLimit[,tileGridSize]]) → retval   
長條圖反向投影 cv2.calcBackProject(image,channels,hist,range,scale[,dst]) → dst   
歸一化 cv2.normalize(src[,dst[,alpha[,beta[,norm\_type[.dtype[,mask]]]]]])→dst

**Chapter sixteen: 圖像變換**

傅立葉轉換 cv2.dft(src[,dst[,flags[,nonzeroRows]]]) → dst   
cv2.getOptimalDFTSize(vecsize) → retvach

**Chapter seventeen: 霍夫變換**

霍夫直線變換 cv2.HoughLines(image,rho,theta,threshold[,lines[,srn[,stn]]]) → lines   
cv2.HoughLinesP(image,rho,theta,threshold[,lines[,minLineLength[,maxLineGap]]]) → lines   
霍夫圓環變換 cv2.HoughCircles(image,method,dp,minDist[,circles[,param1[,param2[,minRadius[,maxRadius]]]]]) → lines

**Chapter eighteen: 分水嶺演算法圖像分割**

分水嶺演算法 cv2.watershed(image,maskers) → None   
連通分支 cv2.connectedComponents(image[,labels[,connectivity[,ltype]]]) → retval,labels

**Chapter nineteen: grabcut演算法互動式前景提取**

cv2.grabcut(image,mask,rect,bgdModel,fgdModel,iterCount[,mode]) → None

**Chapter twenty: 角點檢測**

C++:DescriptorExtractor\_create()   
Harris角點檢測 cv2.cornerHarris(src,blockSize,ksize,k[,dst[,borderType]]) → dst   
提煉角位置 cv2.cornerSubPix(image,corners,winSize,zeroZone,criteria) → None   
Shi-Tomasi角點檢測 cv2.goodFeaturesToTrack(image,maxCorners,qualityLevel,minDistance[,corners[,mask[,blockSize[,useHarrisDetect[,k]]]]]) → corners   
SURF cv2.SURF([hessianThreshold[,n0ctaves[,n0ctaveLayers[,extend[,upright]]]]]) →   
SURF檢測和計算要點 .detect(image[,mask]) → keypoints   
.detectAndCompute(image,mask[,descriptors[,useProvidedKeypoints]]) → keypoints,descriptors   
FAST特徵檢測 cv2.FastFeatureDetector\_create([,threshold[,nonmaxSuppression[,type]]]) → retval   
繪製要點 cv2.drawKeypoints(image,keypoints,outImage[,color[,flags]])→outImg

**Chapter twenty-one: 特徵匹配**

C++: FLANN   
Brute-Force cv2.BFMatcher([,normType[,crossCheck]]) →   
對象透視圖變換 cv2.findHomography(srcPoints,dstPoints[,method[,ransacReprojThreshold[,mask]]]) → retval,mask

**Chapter twenty-two: 在視頻中找到並跟蹤目標物件**

Meanshift cv2.meanShift(proImage,window,criteria) → retval,window   
camshift cv2.camShift(proImage,window,criteria) → retval,window

**Chapter twenty-three: 光流**

Lucas-Kanade光流 cv2.calcOpticalFlowPyrLK(prevImg,nextImg,prevPts[,nextPts[,status[,err[,winSize[,maxLevel[,criteria[,derivLambda[,flags]]]]]]]])   
→nextPts,status,err   
cv2.goodFeaturesToTrack(image,maxCorners,qualityLevel,minCorners[,corners[,mask[,blockSize[,useHarrisDetector[,k]]]]]) → corners   
GF稠密光流 cv2.calcOpticalFlowFarneback(prevImg,nextImg,pyr\_scale,levels,winsize,iterations,poly\_n,poly\_sigma,flags[,flow]) → flow

**Chapter twenty-four: 背景減除**

BackgroundSubtractorMOG   
BackgroundSubtractorGMG

BackgroundSubtractorMOG cv2.BackgroundSubtractorMOG(history,nmixtures,backgroundRatio[,noiseSigma]) →   
createBackgroundSubtractorMOG2([, history[, varThreshold[, detectShadows]]]) → retval

**Chapter twenty-five: 攝像機設定和3D重構**

將物件點轉換到圖像點 cv2.projectPoints(objectPoints,rvec,tvec,cameraMatrix,distCoeffs[,imagePoints[,jacobian[,aspectRatio]]])   
→ imagePoints,jacobian   
攝像機標定 cv2.calibrateCamera(objectPoints,imagePoints,imageSize[,cameraMatrix[,distCoeffs[,rvecs[,tvecs[,flags]]]]])   
→ retval,cameraMatrix,distCoeffs,rvecs,tvecs   
畸變校正 cv2.getOptimalNewCameraMatrix(cameraMatrix,distCoeffs,imageSize,alpha[,newImgSize[,centerPrincipalPoint]])   
→retval,validPixROI   
cv2.undistort(src,cameraMatrix,distCoeffs[,dst[,newCameraMatrix]]) → dst   
cv2.remap(src,map1,map2,interpolation[,dst[,borderMode[,borderValue]]]) → dst   
計算旋轉和變換 cv2.solvePnPRansac(objectPoints,imagePoints,cameraMatrix,distCoeffs[,rvec[,tvec[,useExtrinsicGuess[,iterationsCount[,reprojectionError[,minInliersCount[,inliers[,flags]]]]]]]])   
→ rvec,tvec,inliers

**Chapter twenty-six: 機器學習**

K近鄰 cv2.KNearest.train(trainData,responses[,sampleIdx[,isRegression[,maxK[,updateBase]]]]) → retval   
cv2.KNearest.find\_nearest(samples,k[,results[,neighborResponses[,dists]]]) → retval,results,neighborResponses,dist   
SVM機 cv2.SVM(trainData,responses[,varIdx[,sampleIdx[,params]]]) →   
.train(traindata,responses[,varIdx[,sampleIdx[,params]]]) → retval   
.train\_auto(trainData,responses,varIdx,sampleIdx,params[,k\_fold[,Cgrid[,gammaGrid[,pGrid[,nuGrid[,coeffGrid[,degreeGrid[,balanced]]]]]]]]) → retval   
.predict(sample[, returnDFVal]) → retval   
.save()   
K值聚類 cv2.kmeans(data,K,criteria,attempts,flags[,bestLabels[,centers]]) → retval,bestLabels,centers

**Chapter twenty-seven: 計算攝影學**

圖像去噪 cv2.fastNlMeansDenoising(src[,dst[,h[,templateWindowSize[,searchWindowSize]]]])→dst   
cv2.fastNlMeansDenoising(src,h[,dst[,templateWindowSize[,searchWindowSize[,normType]]]])→dst   
fastNlMeansDenoisingMulti(srcImgs,imgToDenoiseIndex,temporalWindowSize[,dst[,h[,templateWindowSize[,searchWindowSize]]]])→dst   
cv2.fastNlMeansDenoisingMulti(srcImgs,imgToDenoiseIndex,temporalWindowSize,h[,dst[,templateWindowSize[,searchWindowSize[,normType]]]])→dst   
cv2.fastNlMeansDenoisingColored(src[,dst[,h[,hColor[,templateWindowSize[,searchWindowSize]]]]])→dst   
fastNlMeansDenoisingColoredMulti(srcImgs,imgToDenoiseIndex,temporalWindowSize[,dst[,h[,hColor[,templateWindowSize[,searchWindowSize]]]]])→dst   
圖像修補 cv2.inpaint(src,inpaintMask,inpaintRange,flags[,dst]) →dst

**Chapter twenty-eight: 使用Haar 分類器進行面部檢測**

XML分類器 cv2.CascadeClassifier(filename)→   
.empty()→retval   
.load(filename) → retval   
.detectMultiScale(image,rejectLevels,levelWeights[,scaleFactor[,minNeighbors[,flags[,minSize[,maxSize[,outputRejectLevels]]]]]]) → objects