1. 上课代码以及常用函数练习一遍。

```python
# 画火柴人

import cv2
import numpy as np

img = cv2.imread('./01.png')

cv2.circle(img, (263, 75), 50, (255, 0, 0))

# cv2.putText(img, 'hello', (100,100),fontFace=cv2.FONT_HERSHEY_SIMPLEX,
fontScale=5,(255, 0, 0))

pts = np.array([(126, 154), (178, 173), (226, 130), (359, 131), (411, 173),
(451, 149)])

pts_sec = np.array([(243, 256), (336, 255), (359, 131), (226, 130), (243, 256),
(220, 339)])




cv2.polylines(img, [pts], isClosed=False, color=(255, 0, 0), )

cv2.polylines(img, [pts_sec], isClosed=False, color=(255, 0, 0), )




cv2.imshow('img', img)
cv2.waitKey(0)




"""
级联分类器
"""

import cv2

class FaceDetect:
    def __init__(self):
        self.faceImg = cv2.imread('./media/sy.png')
        self.classifier = cv2.CascadeClassifier()
        self.classifier.load('./model/caseharden_frontalface_alt.xml')
    def faceDetect(self):
        faceImg = self.faceImg
        classifier = self.classifier
        faceRects = classifier.detectMultiScale(faceImg)
        for x,y,w,h in faceRects:
            cv2.rectangle(faceImg,(x,y),(x+w,y+h),color=(0,0,255),thickness=2)
        cv2.imshow('faceImg',faceImg)
        cv2.waitKey(0)
        cv2.destroyAllWindows()
detect = FaceDetect()
detect.faceDetect()




# 加logo

import cv2
class FaceDetect:
    def __init__(self):
        self.faceImg = cv2.imread('./media/sy.png')
        self.fansLogo = cv2.imread('./media/fans.jpg')
        self.classifier = cv2.CascadeClassifier()
        self.classifier.load('./model/haarcascade_frontalface_alt.xml')
    def faceDetect(self):
        faceImg = self.faceImg
        classifier = self.classifier
        faceRects = classifier.detectMultiScale(faceImg)

        for x, y, w, h in faceRects:
            cv2.rectangle(faceImg, (x, y), (x+w, y+h), color=(0, 0, 255),
thickness=2)
            rect = (x, y, w, h)
            self.drawLogo(self.fansLogo, rect)

    def drawLogo(self, logo, faceRect):
        ratio = logo.shape[0] / logo.shape[1]
        faceX = faceRect[0]
        faceY = faceRect[1]
        faceW = faceRect[2]
        faceH = int(faceW * ratio)
        smallLogo = cv2.resize(logo, dsize=(faceW, faceH))
        smallLogoH = smallLogo.shape[0]
        smallLogoW = smallLogo.shape[1]

        for row in range(smallLogoH):
            for col in range(smallLogoW):
                self.faceImg[faceY+row-smallLogoH, faceX+col] = smallLogo[row,
col]

detect = FaceDetect()
detect.faceDetect()
cv2.imshow('faceImg', detect.faceImg)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

3. 绘制线、矩形、圆， 火柴人，能够看见绘制的过程

```python
import cv2
import numpy as np
```

```python
img = np.zeros((512, 512, 3), np.uint8)

# cv2.circle(img, (263, 75), 50, (255, 0, 0))

# pts = np.array([(126, 154), (178, 173), (226, 130), (359, 131), (411, 173),
(451, 149)])
# pts_sec = np.array([(243, 256), (336, 255), (359, 131), (226, 130), (243, 256),
(220, 339),(243, 433)])
# pts_thr = np.array([(336, 255), (361, 340), (339, 432)])

# cv2.polylines(img, [pts], isClosed=False, color=(255, 0, 0), thickness=2)
# cv2.polylines(img, [pts_sec], isClosed=False, color=(255, 0, 0), thickness=2)
# cv2.polylines(img, [pts_thr], isClosed=False, color=(255, 0, 0), thickness=2)

drawing = False
prev_point = None
# 在回调函数中记录鼠标移动的轨迹，并将轨迹画在图像上，以实时看到绘制轨迹
def draw(event, x, y, flags, param):
    global drawing, prev_point

    if event == cv2.EVENT_LBUTTONDOWN:
        drawing = True
        prev_point = (x, y)
    elif event == cv2.EVENT_MOUSEMOVE:
        if drawing:
            cv2.line(img, prev_point, (x, y), (0, 255, 0), 2)
            prev_point = (x, y)
    elif event == cv2.EVENT_LBUTTONUP:
        drawing = False


cv2.namedWindow('image')
cv2.setMouseCallback('image', draw)

cv2.imshow('img', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

4.按下不同的按键 绘制不同的图形 l(线) r(矩形) c(圆)

```python
import cv2

drawing = False  # 是否正在绘制
mode = 'l'  # 默认为绘制线段

# 回调函数，用于绘制不同的图形
def draw_shape(event, x, y, flags, param):
    global drawing, mode
    if event == cv2.EVENT_LBUTTONDOWN:
        drawing = True
        if mode == 'l':
            cv2.line(img, (x, y), (x, y), (0, 255, 0), thickness=2)
        elif mode == 'r':
                cv2.rectangle(img, (x, y), (x, y), (0, 0, 255), thickness=2)
            elif mode == 'c':
                cv2.circle(img, (x, y), radius=10, color=(255, 0, 0), thickness=2)
    elif event == cv2.EVENT_MOUSEMOVE:
        if drawing:
            if mode == 'l':
                cv2.line(img, (x, y), (x, y), (0, 255, 0), thickness=2)
            elif mode == 'r':
                cv2.rectangle(img, (x, y), (x, y), (0, 0, 255), thickness=2)
            elif mode == 'c':
                cv2.circle(img, (x, y), radius=10, color=(255, 0, 0),
thickness=2)
    elif event == cv2.EVENT_LBUTTONUP:
        drawing = False

# 创建一个黑色的图像窗口
img = np.zeros((512, 512, 3), np.uint8)
cv2.namedWindow('image')

# 绑定回调函数
cv2.setMouseCallback('image', draw_shape)

while True:
    cv2.imshow('image', img)
    k = cv2.waitKey(1) & 0xFF
    if k == ord('l'):  # 切换到绘制线段模式
        mode = 'l'
    elif k == ord('r'):  # 切换到绘制矩形模式
        mode = 'r'
    elif k == ord('c'):  # 切换到绘制圆形模式
        mode = 'c'
    elif k == 27:  # 按下ESC键退出
        break

cv2.destroyAllWindows()
```

5.人脸绘制Logo 使用切片的方式

```
import cv2
class FaceDetect:
    def __init__(self):
        self.faceImg = cv2.imread('./media/sy.png')
        self.fansLogo = cv2.imread('./media/fans.jpg')
        self.classifier = cv2.CascadeClassifier()
        self.classifier.load('./model/haarcascade_frontalface_alt.xml')
    def faceDetect(self):
        faceImg = self.faceImg
        classifier = self.classifier
        faceRects = classifier.detectMultiScale(faceImg)

        for x, y, w, h in faceRects:
            cv2.rectangle(faceImg, (x, y), (x+w, y+h), color=(0, 0, 255),
thickness=2)
```

```
            rect = (x, y, w, h)
            self.drawLogo(self.fansLogo, rect)

    def drawLogo(self, logo, faceRect):
        ratio = logo.shape[0] / logo.shape[1]
        faceX = faceRect[0]
        faceY = faceRect[1]
        faceW = faceRect[2]
        faceH = int(faceW * ratio)
        smallLogo = cv2.resize(logo, dsize=(faceW, faceH))
        smallLogoH = smallLogo.shape[0]
        smallLogoW = smallLogo.shape[1]

        # for row in range(smallLogoH):
        #     for col in range(smallLogoW):
        #         self.faceImg[faceY+row-smallLogoH, faceX+col] = smallLogo[row,
col]
        self.faceImg[faceY - smallLogoH:faceY,faceX:faceX + faceW] = smallLogo


detect = FaceDetect()
detect.faceDetect()
cv2.imshow('faceImg', detect.faceImg)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

6. 使用摄像头或者视频显示粉丝灯牌

```
import cv2

class FaceDetect:
    def __init__(self):
        self.faceImg = cv2.imread('./media/sy.png')
        self.fansLogo = cv2.imread('./media/fans.jpg')
        self.classifier = cv2.CascadeClassifier()
        self.classifier.load('./model/caseharden_frontal face_alt.xml')

    def faceDetect(self):
        faceImg = self.faceImg
        classifier = self.classifier
        faceRects = classifier.detectMultiScale(faceImg)

        for x, y, w, h in faceRects:
            cv2.rectangle(faceImg, (x, y), (x + w, y + h), color=(0, 0, 255),
thickness=2)
            rect = (x, y, w, h)
            self.drawLogo(self.fansLogo, rect)

    def drawLogo(self, logo, faceRect):
        ratio = logo.shape[0] / logo.shape[1]
        faceX = faceRect[0]
        faceY = faceRect[1]
        faceW = faceRect[2]
        faceH = int(faceW * ratio)
        smallLogo = cv2.resize(logo, dsize=(faceW, faceH))
        smallLogoH = smallLogo.shape[0]
        smallLogoW = smallLogo.shape[1]

        # for row in range(smallLogoH):
        #     for col in range(smallLogoW):
        #         self.faceImg[faceY+row-smallLogoH, faceX+col] = smallLogo[row, 
col]
        self.faceImg[faceY - smallLogoH:faceY, faceX:faceX + faceW] = smallLogo



class VideoCap:
    def __init__(self):
        self.cap = cv2.VideoCapture(r'C:\Users\Administrator\Desktop\123.mp4')
        self.faceDetect = FaceDetect()

    def handleVideoCapture(self):
        cap = self.cap
        while cap.isOpened():
            retval, img = cap.read()
            if not retval:
                print('can not read capture')
                break
            self.faceDetect.faceDetect()
            cv2.imshow('img', img)
            key = cv2.waitKey(25)
            if key == ord('q'):
                break
        cv2.destroyAllWindows()
videoCap = VideoCap()

videoCap.handleVideoCapture()
```