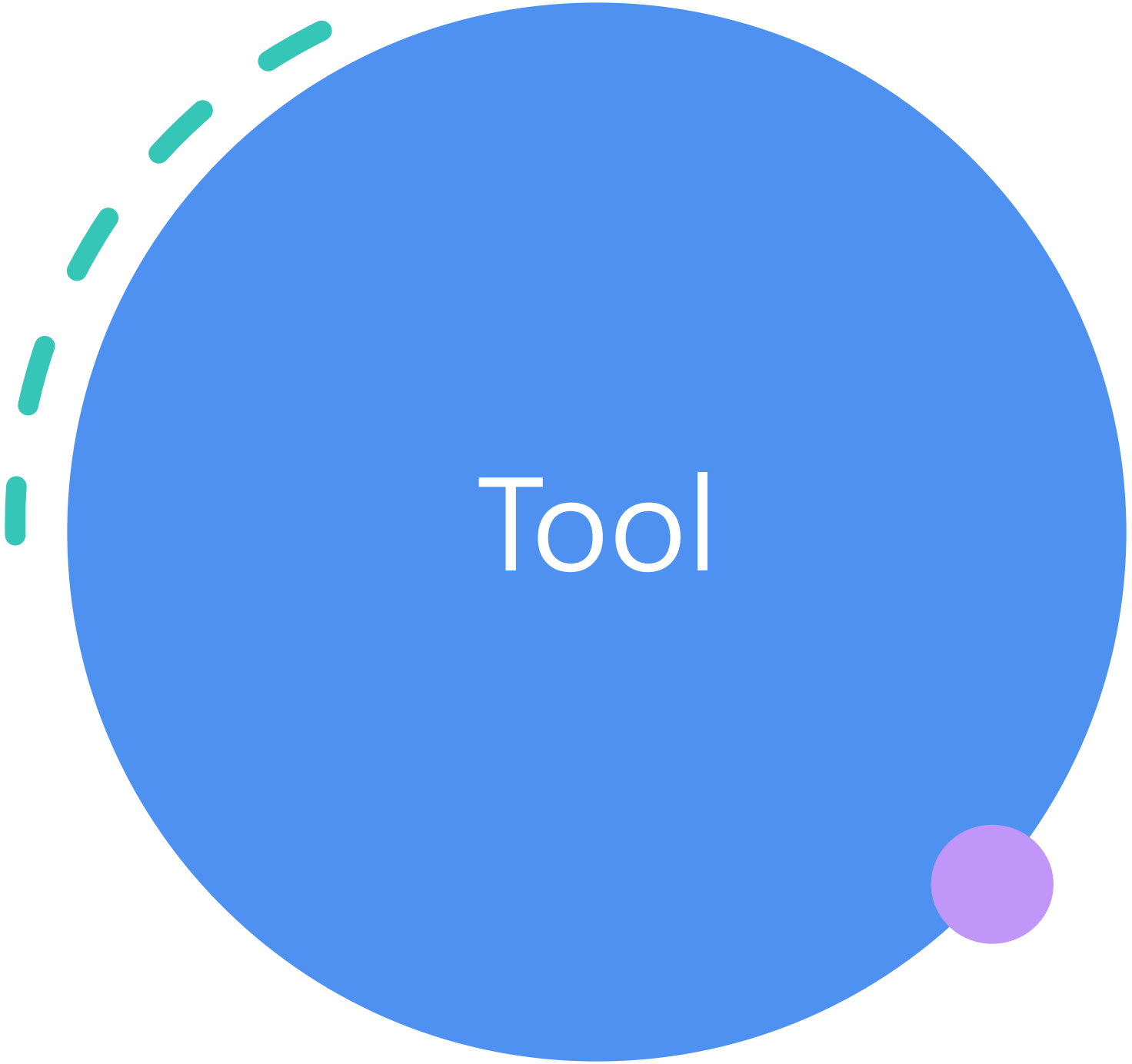
Clock recognition

Professor : Chi-Yu Lin (林其禹)
Student：WEI YEN CHEN

Outline

1. Tool
2. Program Design
3. Results and Discussion


Tool

Tool


OpenCV


NumPy


scikit-image
image processing in python


SPYDER

Program Design

Workflow

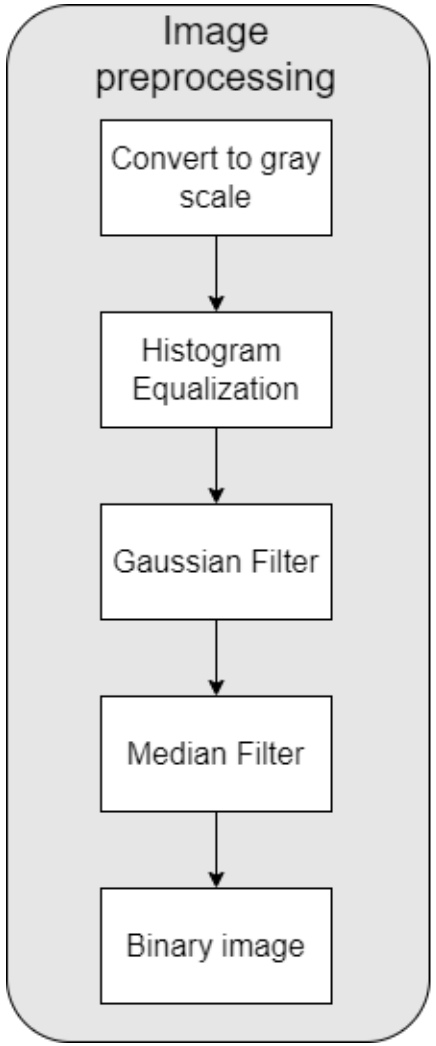
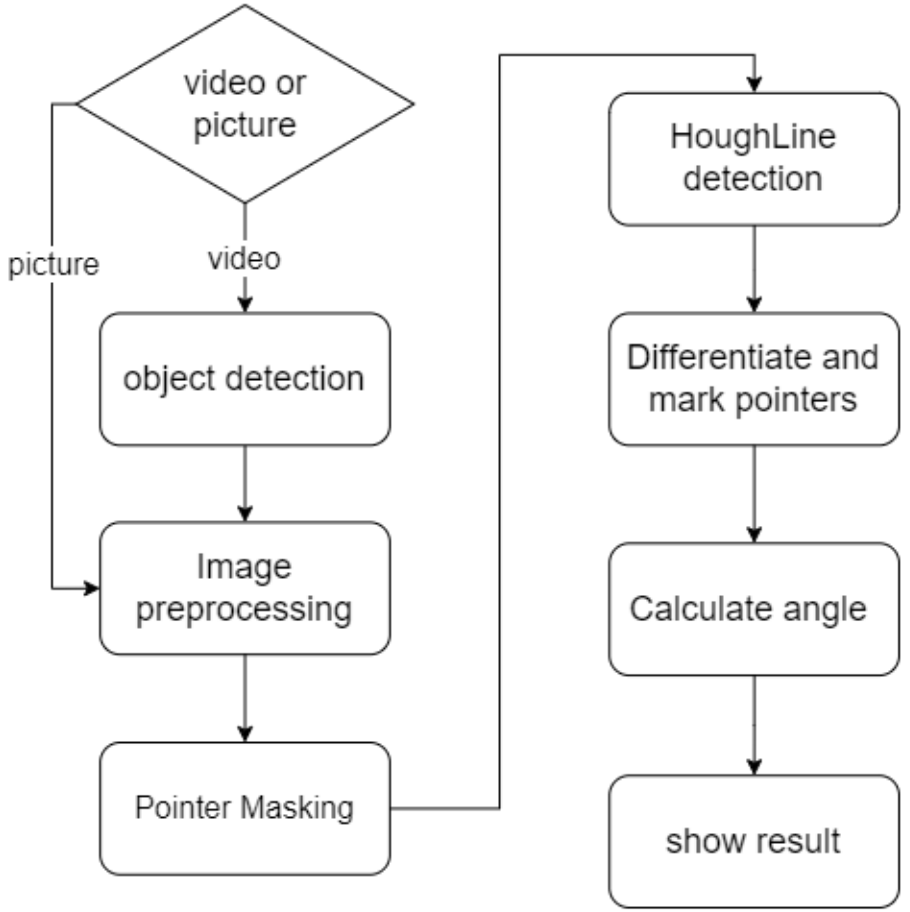


Image preprocessing
Convert to gray scale
Histogram Equalization
Gaussian Filter
Median Filter
Binary image


video or picture
picture
video
object detection
Image preprocessing
Pointer Masking
HoughLine detection
Differentiate and mark pointers
Calculate angle
show result

Steps

- Step1 Load picture or video
- Step2 Object detection (for video)
- Step3 Image preprocessing
- Step4 Pointer masking
- Step5 Hough Line detection
- Step6 Differentiate and mask pointers
- Step7 Calculate angle
- Step8 Show result

Step1 Load picture or video

```python
#Step1 影像讀取
mode=input("video or picture?")
if mode =="picture":
    img = cv2.imread( "./data/1.jpg", -1 )
    threshod = 40
    minLineLength = 10
    MaxLineGap = 25
elif mode == "video":
    cap = cv2.VideoCapture(0)
    #cap = cv2.VideoCapture('./data/v1.mp4')
    cap.set(3,1280)     #width
    cap.set(4,720)      #high
    cap.set(10,70)
#********************model setup**************************
    thres = 0.45 # Threshold to detect object
    classNames= []
    classFile = "coco.names"
    with open(classFile,'rt') as f:
        classNames = f.read().rstrip('\n').split('\n')

    configPath = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
    weightsPath = 'frozen_inference_graph.pb'

    net = cv2.dnn_DetectionModel(weightsPath,configPath)
    net.setInputSize(320,320)
    net.setInputScale(1.0/ 127.5)
    net.setInputMean((127.5, 127.5, 127.5))
    net.setInputSwapRB(True)
#***********************************************************
    threshod = 40
    minLineLength = 10
    MaxLineGap = 20
    if not cap.isOpened():
        print("Cannot open camera")
        exit()
```

Img:

Step2 Object detection

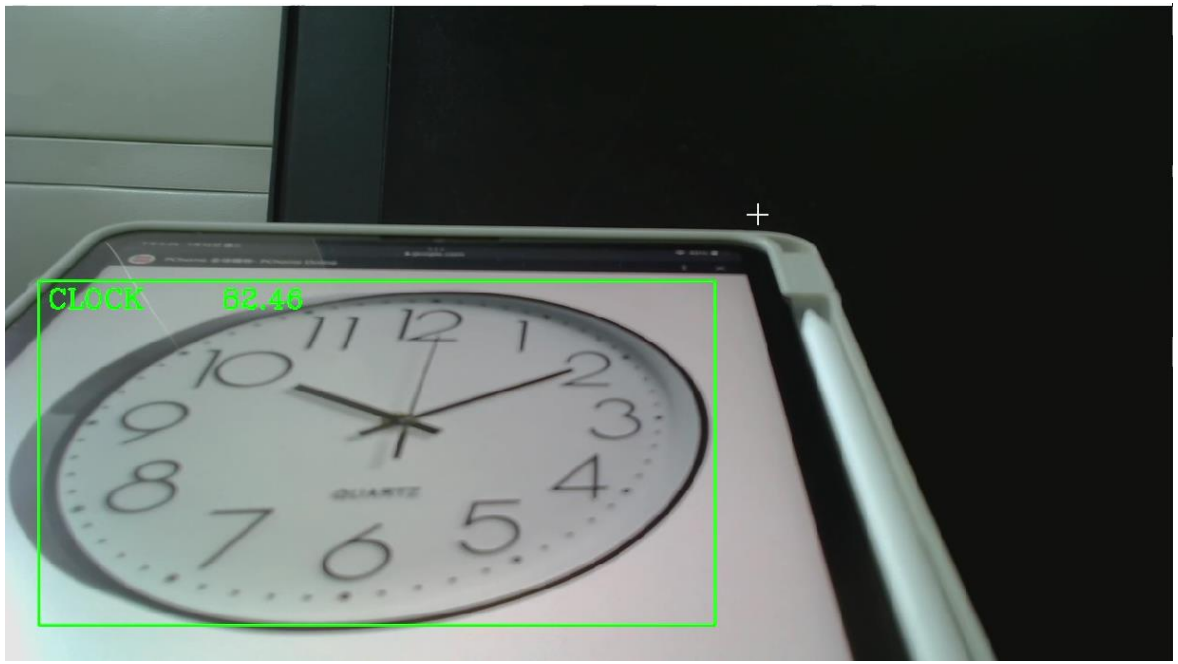
```
189         #物件偵測
190         classIds, confs, bbox = net.detect(img,confThreshold=thres)
191         if len(classIds) != 0 :
192             for classId, confidence,box in zip(classIds.flatten(),confs.flatten(),bbox):
193                 if classId == 85:
194                     cv2.rectangle(img,box,color=(0,255,0),thickness=2)
195                     cv2.putText(img,classNames[classId-1].upper(),(box[0]+10,box[1]+30),
196                                 cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)
197                     cv2.putText(img,str(round(confidence*100,2)),(box[0]+200,box[1]+30),
198                                 cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)
199         cv2.imshow("detection",img)
200         if classId == 85:
201             img0 = img0[box[1]:box[1]+box[3],box[0]:box[0]+box[2]]
202             img1 = cv2.resize(img0,(600,600))
203         elif classId != 85:
204             img1 = cv2.resize(img,(600,600))
205     elif mode == "picture":
206         img1 = cv2.resize(img, (600, 600))
207     img2 = img1.copy()
208     img3 = img1.copy()
209     img4 = img1.copy()
```

Select the ROI

Network : mobilenet_v3

```
19  #*********************model setup*************************
20      thres = 0.45 # Threshold to detect object
21      classNames= []
22      classFile = "coco.names"
23      with open(classFile,'rt') as f:
24          classNames = f.read().rstrip('\n').split('\n')
25
26      configPath = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
27      weightsPath = 'frozen_inference_graph.pb'
28
29      net = cv2.dnn_DetectionModel(weightsPath,configPath)
30      net.setInputSize(320,320)
31      net.setInputScale(1.0/ 127.5)
32      net.setInputMean((127.5, 127.5, 127.5))
33      net.setInputSwapRB(True)
34  #*************************************************************
```

Result



ROI

Step3 Image preprocessing

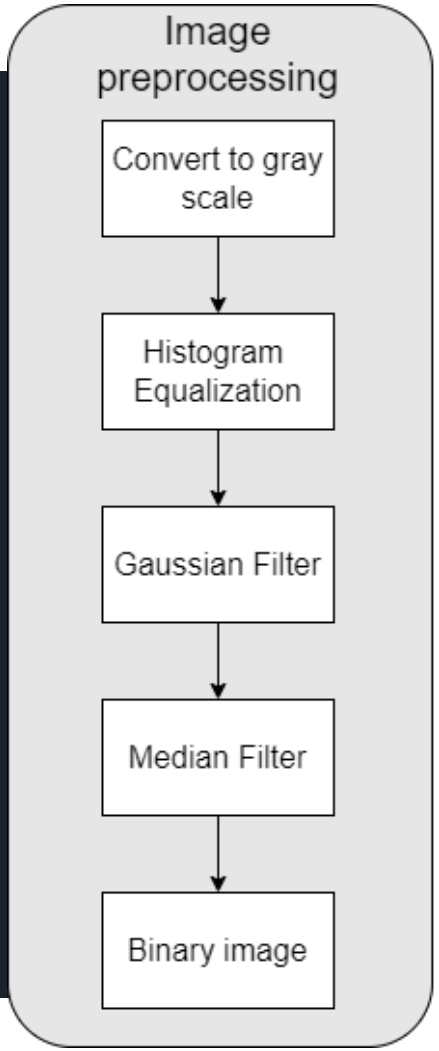
```python
def preprocessing(img):
    #灰階
    gray = cv2.cvtColor( img, cv2.COLOR_BGR2GRAY )

    #質方圖等化
    clahe = cv2.createCLAHE(clipLimit=2)
    clahe_img = clahe.apply(gray)

    #高斯濾波 & 中值濾波器 (如果是網路照片就不使用)
    if mode == "video":
        gaussian_img = cv2.GaussianBlur(clahe_img,(5,5),5)
        median_img = cv2.medianBlur(gaussian_img,5)
    elif mode == "picture":
        gaussian_img = cv2.GaussianBlur(clahe_img,(5,5),0)
        median_img = gaussian_img

    #二進值
    ret, bin_img = cv2.threshold(median_img, 90, 255, cv2.THRESH_BINARY_INV+ cv2.THRESH_OTSU)

    return bin_img
```



Result

Original image

Gray scale +
Histogram Equalization

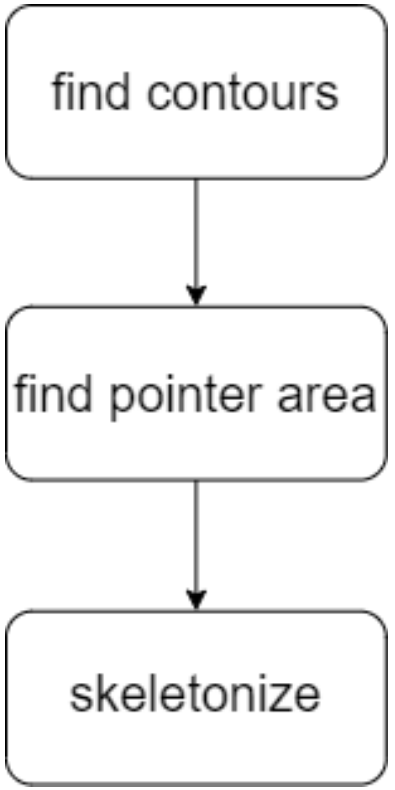
Gaussian Filter + Median Filter + convert to Binary image

Step4 Pointer masking

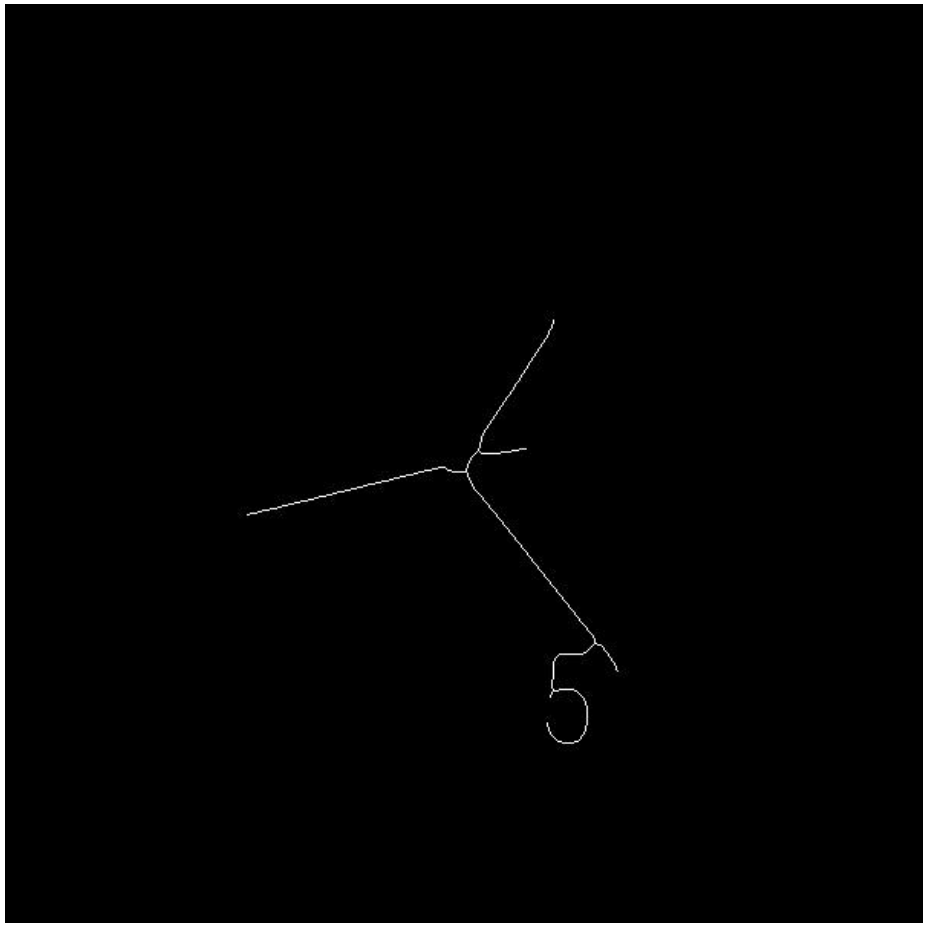
```python
def get_contours(img):
    #尋找輪廓
    contours, hierarchy = cv2.findContours(bin_img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    #計算面積，界在自訂的值內(2000<area<7000)才畫出輪廓
    mask_img = np.zeros_like(bin_img)
    for i, j in enumerate(contours):
        area = int(cv2.contourArea(contours[i]))
        #print("area=",area)
        if area>1500 and area<7000:
            cv2.drawContours(mask_img, contours, i, (255,255,255), -1)
    return mask_img
```

```python
#Step3 遮罩，提取指針
    mask_img = get_contours(bin_img)
    #骨架提取
    mask_img[mask_img==255] = 1
    arms_thin = skeletonize(mask_img)
    arms_thin = (255*arms_thin).clip(0,255).astype(np.uint8)
```



Result



Step5 Hough Line detection

```
76  def Houghline_detection():
77      linesP = cv2.HoughLinesP(arms_thin , 1, np.pi / 180, threshod, None, minLineLength, MaxLineGap)
78      print(linesP)
79      lines_length=[]
80      p1 = []
81      p2 = []
82      if linesP is not None:
83          for i in range(0, len(linesP)):
84              l = linesP[i][0]
85              cv2.line(img3, (l[0], l[1]), (l[2], l[3]), (0,255,255), 2, cv2.LINE_AA)
86              p1.append(l[0])
87              p1.append(l[1])
88              p2.append(l[2])
89              p2.append(l[3])
90              length = int(math.sqrt((l[0]-l[2])**2+(l[1]-l[3])**2))
91              #length = int(math.dist(p1,p2))
92              lines_length.append(length)
93      return lines_length,linesP
```

Result

Detected line

```
[[[158 333 325 292]]

 [[298 305 396 427]]

 [[295 305 344 232]]]
```

Step6 Differentiate and mask pointers

```python
def classify_pointer(lines_length,linesP):
    #比較直線長度
    t = lines_length
    max_number = []
    max_index = []
    if linesP is not None:
        for _ in range(3):
            number = max(t)
            index = t.index(number)
            t[index] = 0
            max_number.append(number)
            max_index.append(index)
        print(max_number)
        print(max_index)

    #劃出分針 時針
        minute = []
        hour = []
        second = []
        for i in range(4):
            second.append(linesP[max_index[0]][0][i])
            minute.append(linesP[max_index[1]][0][i])
            hour.append(linesP[max_index[2]][0][i])

        cv2.line(img4, (second[0], second[1]), (second[2], second[3]), (0,255,0), 3, cv2.LINE_AA)
        cv2.line(img4, (minute[0], minute[1]), (minute[2], minute[3]), (0,255,255), 3, cv2.LINE_AA)
        cv2.line(img4, (hour[0], hour[1]), (hour[2], hour[3]), (0,0,255), 3, cv2.LINE_AA)
        print("second(G):",second)
        print("minute(Y):",minute)
        print("hour(R):",hour)
        return second,minute,hour
    else:
        return 0,0,0
```

The longest : second head
The second: minute head
The third: hour head

Result

```
[[[158 333 325 292]] 171
 [[298 305 396 427]] 156
 [[295 305 344 232]]] 87
```


```
[171, 156, 87]    Length
[0, 1, 2]         Index of line
second(G): [158, 333, 325, 292]
minute(Y): [298, 305, 396, 427]
hour(R): [295, 305, 344, 232]
```

Step7 Calculate angle

```python
def claculate_time(a,mode):
    #print(a)
    a[0] = a[0]-300
    a[1] = -a[1]+300
    a[2] = a[2]-300
    a[3] = -a[3]+300
    #print("校正:",a)
    delta_x = a[2]-a[0]
    delta_y = a[3]-a[1]
    print("delta_x=",delta_x)
    print("delta_y=",delta_y)
    #判斷象限
    if delta_y/delta_x > 0 and a[0]<-45:
        pi=math.pi
    elif delta_y/delta_x < 0 and a[0]<-50 and a[2]<35:
        pi=math.pi
    else:
        pi=0
    print("pi=",pi)
    #計算角度
    angle = math.degrees(math.atan2(delta_y,delta_x) + pi)
    if angle <0 :
        angle=angle+360
    print("arctan=",angle)
    #轉換角度座標
    angle_CCW90 = angle - 90
    if angle_CCW90 < 0:
        angle_CCW90 = angle_CCW90 + 360
    angle_clock = -angle_CCW90 + 360
    #如果最後角度>360，扣一圈(360
    if angle_clock>360:
        angle_clock=angle_clock-360
    print("angle_clock = ",angle_clock)
    #判斷3,6,9,0
    if angle_clock % 90 == 0:
        if delta_y > 0:
            angle_clock = 0
        elif delta_y < 0:
            angle_clock = 180
        elif delta_x < 0:
            angle_clock = 90
        elif delta_x > 0:
            angle_clock =270
    #計算時間
    if mode==1:
        time = math.floor(angle_clock / 6)
    elif mode ==0:
        time = math.floor(angle_clock / 30)
        print("time=",time)
    return time
```

Step8 Show result

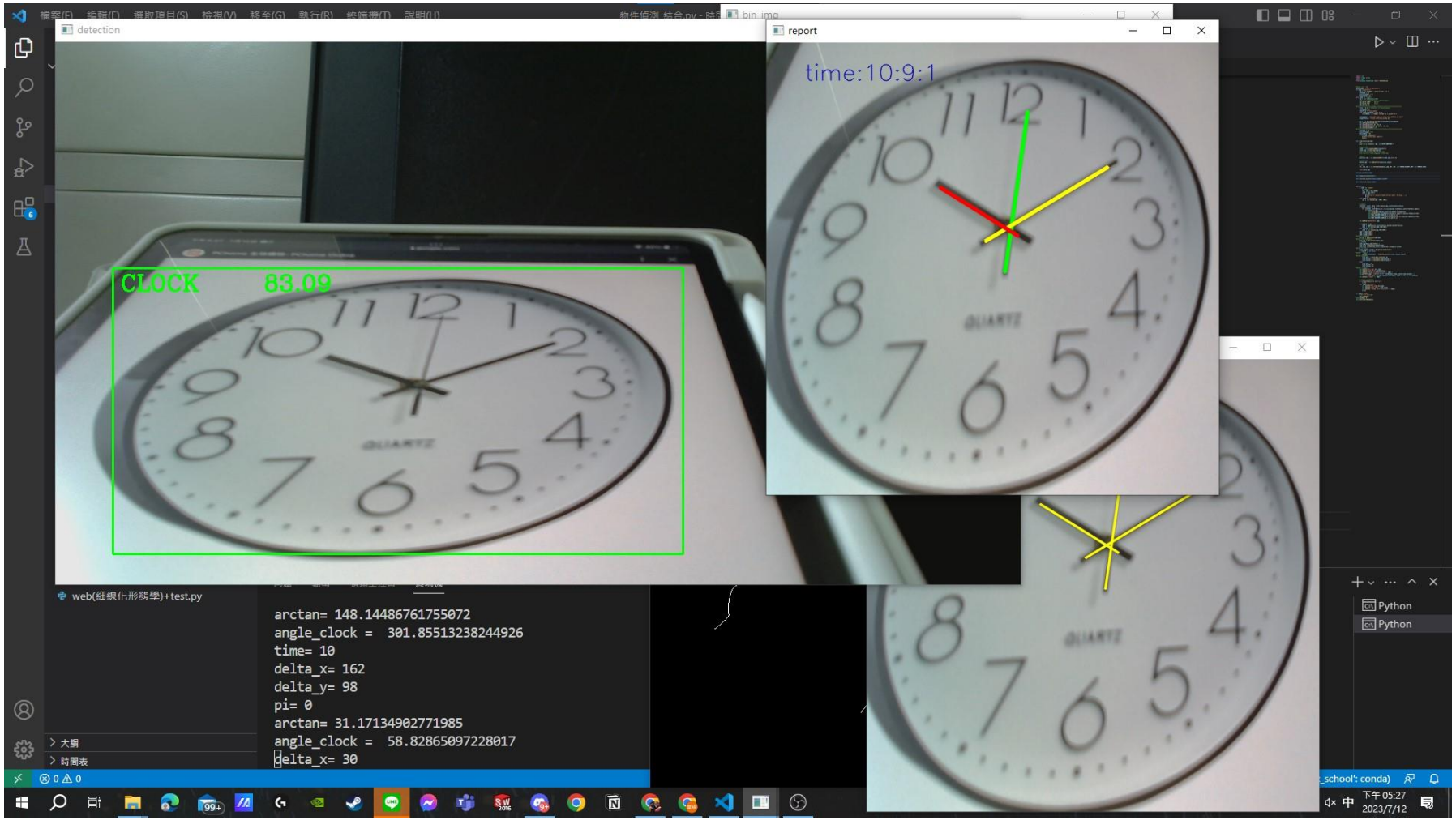
```
#Step7 顯示結果
    cv2.imshow("bin_img",bin_img)
    cv2.imshow('arms_thin', arms_thin)
    cv2.imshow( "Hough Line DetectionP ", img3 )
    cv2.putText(img4, "time:%d:%d:%d" %(time_hour,time_minute,time_second), \
                (50, 50), cv2.FONT_HERSHEY_SIMPLEX,1, (180, 0, 0), 1, cv2.LINE_AA)
    cv2.imshow( "report ", img4 )
```

Results and Discussion

Success case

Success case

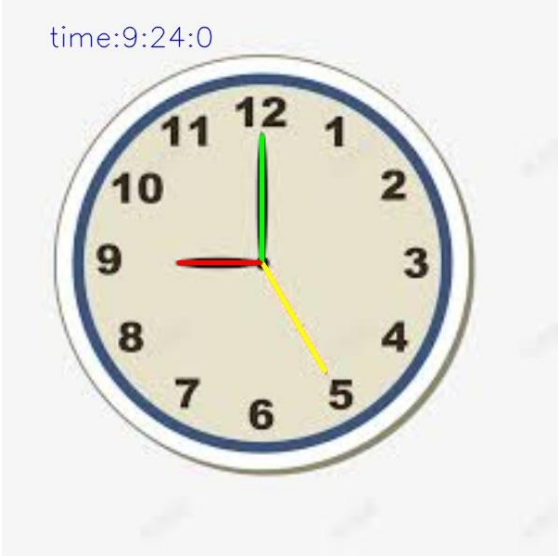
Success case of video



Failure case

The hour hand and minute hand are reversed

Square clocks have different angles



The hour hand and minute hand have similar lengths.

The second hand has been eliminated due to preprocessing

Future work

- Improve the preprocessing
- Improve the elimination of the second hand in some images
- Solve image rotation

Thanks for listening

Code

```python
import cv2
import numpy as np
import math
from skimage.morphology import skeletonize

#Step1 影像讀取
mode=input("video or picture?")
if mode =="picture":
    img = cv2.imread( "./data/14.jpg", -1 )
    threshod = 40
    minLineLength = 10
    MaxLineGap = 25

elif mode == "video":
    cap = cv2.VideoCapture(0)
    #cap = cv2.VideoCapture('./data/v1.mp4')
    cap.set(3,1280)     #width 3 = cv2.CAP_PROP_FRAME_WIDTH
    cap.set(4,720)      #high  4 = cv2.CAP_PROP_FRAME_HEIGHT
    cap.set(10,70)
#********************model setup*************************
    thres = 0.45 # Threshold to detect object
    classNames= []
    classFile = "coco.names"
    with open(classFile,'rt') as f:
        classNames = f.read().rstrip('\n').split('\n')

    configPath = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
    weightsPath = 'frozen_inference_graph.pb'

    net = cv2.dnn_DetectionModel(weightsPath,configPath)
    net.setInputSize(320,320)
    net.setInputScale(1.0/ 127.5)
    net.setInputMean((127.5, 127.5, 127.5))
    net.setInputSwapRB(True)
#***********************************************************
    threshod = 40       #40
    minLineLength = 40  #10
    MaxLineGap = 100     #20
    if not cap.isOpened():
        print("Cannot open camera")
        exit()
```

Code

```python
def preprocessing(img):
    #灰階
    gray = cv2.cvtColor( img, cv2.COLOR_BGR2GRAY )

    #質方圖等化
    clahe = cv2.createCLAHE(clipLimit=2)
    clahe_img = clahe.apply(gray)

    #高斯濾波 & 中值濾波器 (如果是網路照片就不使用)
    if mode == "video":
        sigma = 5
        kernel = 5
    elif mode == "picture":
        sigma = 0
        kernel = 3
    gaussian_img = cv2.GaussianBlur(clahe_img,(5,5),sigma)
    median_img = cv2.medianBlur(gaussian_img,kernel)

    #二進值
    ret, bin_img = cv2.threshold(median_img, 90, 255, cv2.THRESH_BINARY_INV+ cv2.THRESH_OTSU)

    return bin_img

def get_contours(img):
    #尋找輪廓
    contours, hierarchy = cv2.findContours(bin_img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    #計算面積，界在自訂的值內(2000<area<7000)才畫出輪廓
    mask_img = np.zeros_like(bin_img)
    for i, j in enumerate(contours):
        area = int(cv2.contourArea(contours[i]))
        #print("area=",area)
        if area>2200 and area<7000:    #**********************************7000
            cv2.drawContours(mask_img, contours, i, (255,255,255), -1)
    return mask_img
```

Code

```python
def get_contours(img):
    #尋找輪廓
    contours, hierarchy = cv2.findContours(bin_img, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    #計算面積，界在自訂的值內(2000<area<7000)才畫出輪廓
    mask_img = np.zeros_like(bin_img)
    for i, j in enumerate(contours):
        area = int(cv2.contourArea(contours[i]))
        #print("area=",area)
        if area>2200 and area<7000:    #**********************************7000
            cv2.drawContours(mask_img, contours, i, (255,255,255), -1)
    return mask_img

def Houghline_detection(img):
    linesP = cv2.HoughLinesP(img , 1, np.pi / 180, threshod, None, minLineLength, MaxLineGap) #40 10 20' 40 10 25
    print("minLineLength=",minLineLength)
    #print(linesP)
    lines_length=[]
    p1 = []
    p2 = []
    if linesP is not None:
        for i in range(0, len(linesP)):
            l = linesP[i][0]
            cv2.line(img3, (l[0], l[1]), (l[2], l[3]), (0,255,255), 2, cv2.LINE_AA)
            p1.append(l[0])
            p1.append(l[1])
            p2.append(l[2])
            p2.append(l[3])
            length = int(math.sqrt((l[0]-l[2])**2+(l[1]-l[3])**2))
            #length = int(math.dist(p1,p2))
            lines_length.append(length)
    return lines_length,linesP
```

Code

```python
def classify_pointer(lines_length,linesP):
    #比較直線長度
    t = lines_length
    max_number = []
    max_index = []
    if linesP is not None:
        for _ in range(3):
            number = max(t)
            index = t.index(number)
            t[index] = 0
            max_number.append(number)
            max_index.append(index)
        print(max_number)
        print(max_index)

    #劃出分針 時針
        minute = []
        hour = []
        second = []
        for i in range(4):
            second.append(linesP[max_index[0]][0][i])
            minute.append(linesP[max_index[1]][0][i])
            hour.append(linesP[max_index[2]][0][i])

        cv2.line(img4, (second[0], second[1]), (second[2], second[3]), (0,255,0), 3, cv2.LINE_AA)
        cv2.line(img4, (minute[0], minute[1]), (minute[2], minute[3]), (0,255,255), 3, cv2.LINE_AA)
        cv2.line(img4, (hour[0], hour[1]), (hour[2], hour[3]), (0,0,255), 3, cv2.LINE_AA)
        print("second(G):",second)
        print("minute(Y):",minute)
        print("hour(R):",hour)
        return second,minute,hour
    else:
        return 0,0,0
```

Code

```python
def claculate_time(a,mode):
    #print(a)
    a[0] =  a[0]-300
    a[1] = -a[1]+300
    a[2] =  a[2]-300
    a[3] = -a[3]+300
    #print("校正:",a)
    delta_x = a[2]-a[0]
    delta_y = a[3]-a[1]
    print("delta_x=",delta_x)
    print("delta_y=",delta_y)
    #判斷象限
    if delta_y/delta_x > 0 and a[0]<-45:
        pi=math.pi
    elif delta_y/delta_x < 0 and a[0]<-50 and a[2]<35:
        pi=math.pi
    else:
        pi=0
    print("pi=",pi)
    #計算角度
    angle = math.degrees(math.atan2(delta_y,delta_x) + pi)
    if angle <0 :
        angle=angle+360
    print("arctan=",angle)
    #轉換角度座標
    angle_CCW90 = angle - 90
    if angle_CCW90 < 0:
        angle_CCW90 = angle_CCW90 + 360
    angle_clock = -angle_CCW90 + 360
    #如果最後角度>360，扣一圈(360
    if angle_clock>360:
        angle_clock=angle_clock-360
    print("angle_clock = ",angle_clock)
    #判斷3,6,9,0
    if angle_clock % 90 == 0:
        if delta_y > 0:
            angle_clock = 0
        elif delta_y < 0:
            angle_clock = 180
        elif delta_x < 0:
            angle_clock = 90
        elif delta_x > 0:
            angle_clock =270
    #計算時間
    if mode==1:
        time = math.floor(angle_clock / 6)
    elif mode ==0:
        time = math.floor(angle_clock / 30)
        print("time=",time)
    return time
```

Code

```
while(True):
    if mode == "video":
        # 擷取影像
        ret, img = cap.read()
        img0 = img.copy()
        if not ret:
            print("Can't receive frame (stream end?). Exiting ...")
            break
        #物件偵測
        classIds, confs, bbox = net.detect(img,confThreshold=thres)
        if len(classIds) != 0 :
            for classId, confidence,box in zip(classIds.flatten(),confs.flatten(),bbox):
                if classId == 85:
                    cv2.rectangle(img,box,color=(0,255,0),thickness=2)
                    cv2.putText(img,classNames[classId-1].upper(),(box[0]+10,box[1]+30),
                                cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)
                    cv2.putText(img,str(round(confidence*100,2)),(box[0]+200,box[1]+30),
                                cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)
        cv2.imshow("detection",img)
        if classId == 85:
            img0 = img0[box[1]:box[1]+box[3],box[0]:box[0]+box[2]]
            img1 = cv2.resize(img0,(600,600))
        elif classId != 85:
            img1 = cv2.resize(img,(600,600))
    elif mode == "picture":
        img1 = cv2.resize(img, (600, 600))
    img2 = img1.copy()
    img3 = img1.copy()
    img4 = img1.copy()
#Step2 圖像預處理
    bin_img = preprocessing(img1)
#Step3 遮罩，提取指針
    mask_img = get_contours(bin_img)
    #骨架提取
    mask_img[mask_img==255] = 1
    arms_thin = skeletonize(mask_img)
    arms_thin = (255*arms_thin).clip(0,255).astype(np.uint8)
#Step4 直線偵測
    lines_length,linesP = Houghline_detection(arms_thin)
    if linesP is not None:
#Step5 區分指針
        second,minute,hour = classify_pointer(lines_length,linesP)
```

Code

```
227 #Step6 判斷時間
228         time_hour = claculate_time(hour,0)
229         time_minute = claculate_time(minute,1)
230         time_second = claculate_time(second,1)
231     else:
232         time_hour = 0
233         time_minute = 0
234         time_second = 0
235 #Step7 顯示結果
236     cv2.imshow("bin_img",bin_img)
237     cv2.imshow('arms_thin', arms_thin)
238     cv2.imshow( "Hough Line DetectionP ", img3 )
239     cv2.putText(img4, "time:%d:%d:%d" %(time_hour,time_minute,time_second), \
240                 (50, 50), cv2.FONT_HERSHEY_SIMPLEX,1, (180, 0, 0), 1, cv2.LINE_AA)
241     cv2.imshow( "report ", img4 )
242 
243     # 按下 q 鍵離開迴圈
244     if cv2.waitKey(1) == ord('q'):
245         break
246     elif mode=="picture":
247         cv2.imshow("bin_img",bin_img)
248         cv2.imshow('arms_thin', arms_thin)
249         cv2.imshow( "Hough Line DetectionP ", img3 )
250         break
251 
252 if mode=="video":
253     # 釋放該攝影機裝置
254     cap.release()
255 cv2.waitKey( 0 )
256 cv2.destroyAllWindows()
```