QRcode檔案轉換器

4110060006-鄭皓

核心概念

Youtube平台沒有上傳大小限制

只要想辦法將檔案轉為影片檔，就可以上傳到Youtube

從而得到無限雲端容量

需要時將影片轉回原先檔案

大綱

上傳

　　將zip檔轉為QRcode

　　將QRcode寫入影片

　　上傳到Youtube

下載

　　下載影片

　　逐幀解碼

　　轉回zip檔

Part 1-輸出影片

檔案前處理

每一張QRcode大約可以儲存2500位元的資料

所有檔案都能夠被轉換為64進位的字串

將檔案分割為每2048位元一份

```python
self.zip = open(file,'rb')
self.zip = self.zip.read()
binary = base64.b64encode(self.zip)
```

```python
binary = self.split(binary, 2048)
```

64bit to QRcode

將分割好的檔案個別轉換為一張QRcode

```python
def to_QR(self, data):
    qr = qrcode.QRCode(version=4,
            error_correction=qrcode.constants.ERROR_CORRECT_L, box_size=10, border=4)
    qr.add_data(data)
    qr.make(fit=True)
    img = qr.make_image(fill_color="black", back_color="white")
    return np.array(img).astype(np.uint8)*255
```
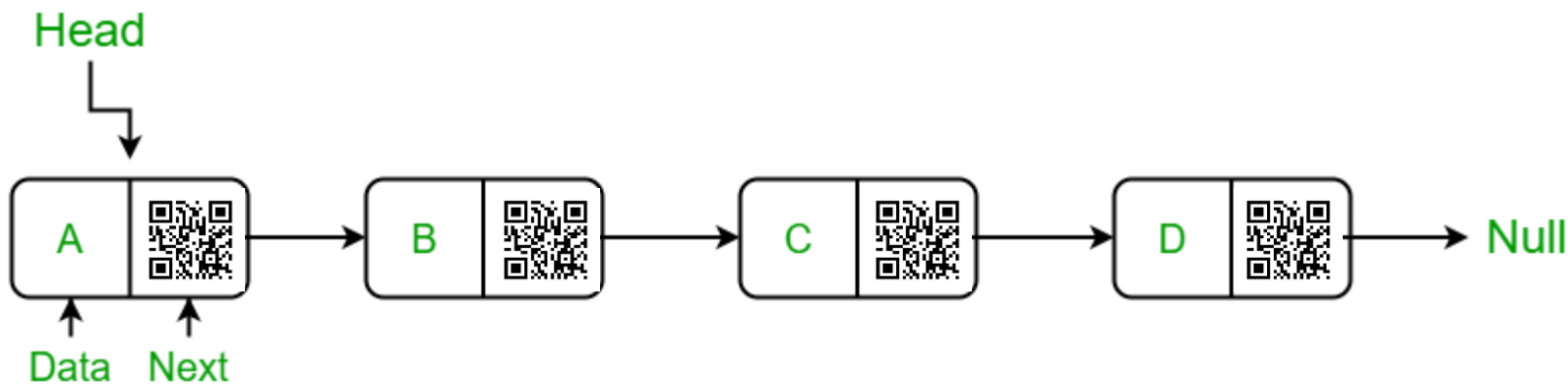
資料儲存方法-Linked List

QRcode是有順序的，不需要搜尋特定元素

用Linked list可以減少記憶體

Insert也較快

將QRcode存入Linked list

```python
for i in binary:
    i = self.to_QR(i)
    self.append(i)
```

輸出影片

一部60FPS的影片

每一幀分別是一個QRcode

```python
def to_video(self):
    out = cv2.VideoWriter('output.mp4', cv2.VideoWriter_fourcc(*'XVID'), 60, (900,900))

    if self.head == None:
        print("Empty list, please input zip file")
        return

    current = self.head
    while current:
        current.data = cv2.cvtColor(current.data, cv2.COLOR_GRAY2BGR)
        current.data = cv2.resize(current.data, (900,900))
        out.write(current.data)
        current = current.next

    out.release()
    print("Video Finished")
```

影片範例

Part 2-影片轉zip

將影片逐幀放入Linked list

```python
cap = cv2.VideoCapture(video)
count = 0

while cap.isOpened():
    rep, frame = cap.read()
    if rep == False:
        break
    self.append(frame)
    count+=1
```

相片解碼

```python
def decode(self, img):
    byte = pyzbar.decode(img)
    byte = base64.b64decode(byte[0].data)
    return byte
```

```python
current = self.head
count=0
while current != None:
    current.data = cv2.cvtColor(current.data, cv2.COLOR_BGR2GRAY)
    current.data = self.decode(current.data)
    current = current.next
    count+=1

    if count%100==0:
        print(int((count/self.length)*100),"%")

print(count, "Frame decoded")
```

將所有Linked list寫入zip檔

```python
def to_zip(self, filename):
    self.bit = bytes()
    with open(filename, 'wb') as z:
        current = self.head

        while current != None:
            self.bit+=current.data
            current = current.next
        z.write(self.bit)
```

```
In [15]: runfile('C:/Code/work/Data Structure/output.py', wdir='C:/Code/work/Data Structure')
81 Frame decoded
File complete!
```

解壓縮的結果會跟input一模一樣

Thank you!