**ชื่อหัวข้อ**  License Plate Recognition โปรแกรมอ่านป้ายทะเบียนรถยนต์

**สมาชิก**

นายคุณากร กัลยาวุฒิพงศ์ รหัสนิสิต 6621600143

นางสาวอัสวาณี อารง รหัสนิสิต 6621604700

**วัตถุประสงค์**

1. เรียนรู้การพัฒนาโปรแกรมเพื่อการตรวจจับป้ายทะเบียนรถยนต์
2. เป็นการฝึกออกแบบกระบวนการแปลงภาพเป็นข้อความตัวอักษร
3. เป็นการฝึกการเขียนโปรแกรมด้วยภาษา Python

**อธิบายขั้นตอนวิธีการดำเนินงานโดยละเอียด**

1. แปลงภาพเป็น binary ด้วยฟังก์ชั่น binarize\_image

* แปลงภาพเป็นสีเทา ใช้ grayscale\_image = image.convert('L') (L หมายถึงโหมด Grayscale)
* เบลอภาพด้วย Gaussian Blur ทำให้ภาพเนียนขึ้น ลด noise ที่ภาพ   
  Gaussian Blur blurred\_image = cv2.GaussianBlur(gray\_array, (5, 5), 0)
* แปลงเป็นภาพไบนารี binary\_image = Image.fromarray(binary\_array.astype(np.uint8) \* 255) โดยการคูณ 255 จะมีแค่ภาพสีขาวดำเท่านั้นในภาพ

1. ทำ vertical\_projection ด้วยฟังก์ชั่น vertical\_projection

* แปลงภาพไบนารีเป็น Array ทำให้ไปคำนวณได้ง่ายขึ้นและเร็วด้วย numpy

img\_array = np.array(binary\_image)

* คำนวณผลรวมของพิกเซลสีดำในแต่ละคอลัมน์ใช้เงื่อนไข img\_array == 0 เอาสีดำ และรวมค่าของแต่ละคอลัมน์ใช้ np.sum(..., axis=0) จะได้ array ที่เก็บ pixel  สีดำในแต่ละคอลัมน์ของภาพ
* ผลลัพธ์ของ Vertical Projection เป็น Array แสดงจำนวน pixel สีดำในแต่ละ คอลัมน์ คอลัมน์ที่มีค่าพิกเซลสีดำเป็น 0 อาจจะเป็นช่องว่างระหว่างตัวอักษร ข้อมูลนี้สามารถนำไปใช้ในการบอกขอบเขตตัวอักษรในภาพได้

1. แยกตัวอักษรแต่ละตัวจากการทำ vertical\_projection ด้วยฟังก์ชั่น segment\_characters\_modified

* function segment\_characters\_modified สร้าง list เพื่อเก็บตัวอักษร
* เดินดูทุกคอลัมน์ เริ่มต้นให้เป็น false ถ้าเจอ pixel สีดำ ถือว่าเจอขอบตัวอักษรแล้ว ให้เก็บไว้เป็น start\_index  ให้เป็น true เพื่อบอกว่าอยู่ในช่วงตัวอักษรแล้ว ถ้าค่า pixel ต่ำกว่าค่า T ให้เป็น end\_index ไปเป็น false แล้วนำ start กับ end ไปเก็บไว้ใน list

1. ตัดตัวอักษรออกมาทีละตัว ด้วยช้ฟังก์ชั่น extract\_characters โดยฟังก์ชั่นมีการทำงานดังนี้ binary\_image คือ ภาพขาว-ดำ segments คือ ตำแหน่งเริ่มต้นและสิ้นสุดของแต่ละตัวอักษร

* ตัดแยกตัวอักษรแต่ละตัวโดย การวนลูปผ่านแต่ละ segment
* ตัดเฉพาะส่วนของภาพตามความกว้างที่กำหนด ครอบตัดให้เหลือแค่ pixel ที่มีสีดำอยู่ return เป็นตัวอักษรเเต่ละตัว

1. สร้าง pattern ตัวอักษร ด้วยฟังก์ชั่น get\_character\_pattern และ load\_character\_patterns

get\_character\_pattern คือ การสร้าง pattern ของตัวอักษร โดยฟังก์ชั่นมีการทำงานดังนี้

* แปลงภาพเป็น grayscale ปรับขนาดให้เท่ากัน เเละ แปลงเป็นภาพขาว-ดำ
* คำนวณ vertical projection รวมค่าพิกเซลตามแนวตั้งของภาพ เพราะทำการครปตัดเฉพาะส่วนที่มีสีดำเเล้ว พร้อม normalize ค่า projection

load\_character\_patterns คือ สร้างฐานข้อมูลรูปแบบของตัวอักษรแต่ละตัว

* กำหนด dictionary ของตัวอักษรและ path ของไฟล์รูปภาพ สร้าง pattern สำหรับทุกตัวอักษร วนลูปผ่านทุกตัวอักษร ใช้ get\_character\_pattern เพื่อสร้าง pattern

1. เทียบตัวอักษร ด้วยฟังก์ชั่น recognize\_characters

* วนลูปผ่านแต่ละ segment ตัดเอาเฉพาะส่วน projection ของตัวอักษรนั้นๆ normalize ค่าให้อยู่ในช่วง 0-1
* เปรียบเทียบ projection กับ pattern ของทุกตัวอักษร คำนวณค่าความแตกต่าง (ยิ่งน้อยยิ่งเหมือน) np.abs(normalized\_projection - pattern)
* ตรวจสอบความเหมือน ถ้าค่าความแตกต่างน้อยกว่า threshold (0.5) ถือว่าเป็นตัวอักษรที่ถูกต้อง ส่งคืนรายการของตัวอักษรที่เหมือนที่สุด