

ระบบเซ็นเซอร์วิดีโออัตโนมัติ
AUTOMATIC VIDEO CENSORING SYSTEM

พงศธร เลิศรุ่งวิเชียร
พลากร โตมอญ
รัฐชิตา นายประเทศ

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

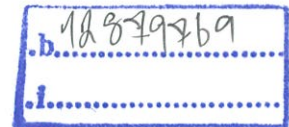
ระบบเซ็นเซอร์วิดีโออัตโนมัติ
AUTOMATIC VIDEO CENSORING SYSTEM



T149035

พงศธร เลิศรุ่งวิเชียร
พลากร โตมอญ
รัฐชิตา นายประเทศ

เลขหมู่.....
เลขทะเบียน **149035**
วัน,เดือน,ปี... **27.S.A. 2560**



ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

AUTOMATIC VIDEO CENSORING SYSTEM

PONGSATRON LERTRUNGWICHEAN
PALAKORN TOMORN
RUNCHIDA NAOYAPRATED

SPECIAL PROBLEM SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2015

หัวข้อปัญหาพิเศษ

ระบบเซ็นเซอร์วิดีโออัตโนมัติ

Automatic Video Censoring System

ชื่อนักศึกษา

นายพงศธร เลิศรุ่งวิเชียร รหัสนักศึกษา 55050382

นายพลากร ไทมอญ รหัสนักศึกษา 55050390

นางสาวรัญชิตา เนายประเทศ รหัสนักศึกษา 55050431

ปริญญา

วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)

ภาควิชา

วิทยาการคอมพิวเตอร์




ปีการศึกษา

2558

อาจารย์ที่ปรึกษา

ผศ.ธีระ ศิริธีรากล

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง(สจล.) อนุมัติให้
ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา วิทยาศาสตร์บัณฑิต
(วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2558

คณะกรรมการสอบ	ลายมือชื่อ
อ.สันธนะ อุ่อตมยิ่ง ประธานกรรมการ	
รศ.ธีรวัฒน์ ประกอบผล กรรมการ	
ผศ.ธีระ ศิริธีรากล กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อปัญหาพิเศษ	ระบบเซ็นเซอร์วิดีโออัตโนมัติ Automatic Video Censoring System
ชื่อนักศึกษา	นายพงศธร เลิศรุ่งวิเชียร รหัสนักศึกษา 55050382 นายพลากร โทมอญ รหัสนักศึกษา 55050390 นางสาวรัญชิตา นายประเทศ รหัสนักศึกษา 55050431
ปริญญา	วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง(สจล.)
ปีการศึกษา	2558
อาจารย์ที่ปรึกษา	ผศ.ธีระ ศิริธีรากล

บทคัดย่อ

โปรแกรมระบบเซ็นเซอร์วิดีโออัตโนมัติ (Automatic Video Censoring System) เป็นระบบที่ทำงานโดยใช้ภาพตัวอย่างมาค้นหาภาพที่ต้องการเซ็นเซอร์ในวิดีโอต่างๆ ซึ่งการใช้งานจะสะดวกกว่าการทำงานปกติมากขึ้น เพราะโปรแกรมนี้อาจทำการค้นหาและติดตามภาพ ภาพนั้นโดยอัตโนมัติพร้อมทั้งทำการเซ็นเซอร์ให้ โดยหลักการทำงานของโปรแกรมนี้อาจแบ่งเป็น 3 ส่วน คือ ส่วนเริ่มต้นเป็นการนำวิดีโอเกี่ยวกับภาพตัวอย่างที่ต้องการมาใส่ลงในระบบ ส่วนต่อมาเป็นการทำงานของระบบ ระบบจะประมวลผลค้นหาภาพตัวอย่างในวิดีโอเพื่อมาเซ็นเซอร์ ในการค้นหาตำแหน่งภาพตัวอย่างสามารถปรับแต่งค่าความเหมือนของบริเวณที่ตรวจจับ เพื่อประสิทธิภาพที่ดีที่สุดของการทำงาน และหากมีภาพนั้นอยู่หลายตำแหน่งในวิดีโอระบบก็สามารถตรวจจับให้ได้ทั้งหมด ส่วนสุดท้ายคือการแสดงผล ผลลัพธ์ที่ออกมาได้เป็นวิดีโอที่ทำการเซ็นเซอร์ภาพเรียบร้อยแล้ว ซึ่งผลลัพธ์อาจจะไม่ได้ตรงตามที่ต้องการทั้งหมดแต่สามารถนำมาแก้ไขในส่วนที่ต้องการในภายหลังได้

Title	Automatic Video Censoring System	
Students	Mr. Pongsatron Lertrungwichean	Student ID 55050382
	Mr. Palakorn Tomorn	Student ID 55050390
	Miss Runchida Naoyaprated	Student ID 55050431
Degree	Bachelor of Science (Computer Science)	
Department	Computer Science	
Faculty	Science	
University	King Mongkut's Institute of Technology Ladkrabang(KMITL)	
Academic Year	2015	
Advisor	Asst.Prof.Teera Siriteerakul	

Abstract

Automatic Video Censoring Application works by using an input image to find out the image. That you want to censor in video. It will work more convenient than people have worked before, because this program will detect and censor it automatically. The principle of this program is divided into three parts, starting with inserting the clip video and the input image that you want into our application. Next our application searches for the input image on each frame using OpenCV detect API. Manual Threshod configuration feature is available for allowing the user to determine the degree of similarity of the input image to the capture area. If there are multiple locations in the video, system also can detect them all. The last part is the display, Output video will censored. Manual video censoring feature is implemented in our application.

กิตติกรรมประกาศ

ในการจัดทำปัญหาพิเศษในหัวข้อเรื่อง ระบบเซ็นเซอร์วิดีโออัตโนมัติ (Automatic Video Censoring System) ทำให้เกิดการเสริมสร้างประสบการณ์ในการทำงานต่างๆเช่น การทำงานร่วมกัน การนำความรู้ที่ได้เรียนมาต่อยอด และการประยุกต์ใช้เพื่อปฏิบัติให้บรรลุตามจุดประสงค์ของปัญหาพิเศษให้สามารถดำเนินการเสร็จลุล่วงได้ ทางคณะผู้จัดทำขอขอบพระคุณ ผศ.ธีระ ศิริธีรากล อาจารย์ที่ปรึกษาปัญหาพิเศษที่ได้มอบแนวคิดต่างๆ และคำแนะนำติชมในการตรวจสอบผลงาน รวมถึงคณะกรรมการสอบปัญหาพิเศษ ซึ่งประกอบด้วย อ.สันธนะ อุ่อดมยิ่ง และ รศ.ธีรวัฒน์ ประกอบผล ที่ได้เสียสละเวลาให้คำแนะนำในสิ่งที่ควรปรับปรุงทำให้ปัญหาพิเศษนี้มีความสมบูรณ์มากยิ่งขึ้น และขอขอบคุณบุคคลต่างๆที่ได้เกี่ยวข้องให้ความช่วยเหลืออุปกรณ์การทำงานในการทำปัญหาพิเศษนี้

สุดท้ายนี้ทางคณะผู้จัดทำขอขอบพระคุณทุกท่าน และครอบครัวที่ให้ความสนับสนุนกำลังใจ เพื่อนๆของคณะผู้จัดทำที่ให้ความช่วยเหลือทำให้คณะผู้จัดทำสามารถดำเนินงานพัฒนาปัญหาพิเศษได้จนสำเร็จลุล่วงไปด้วยดี

พงศธร เลิศรุ่งวิเชียร

พลากร โทมอญ

รัญชิตา นายประเทศ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ฉ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของการทำปัญหาพิเศษ	1
1.2 วัตถุประสงค์ของการทำปัญหาพิเศษ	1
1.3 ขอบเขตของปัญหาพิเศษ	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ	2
บทที่ 2 ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง	3
2.1 ภาพ	3
2.1.1 ภาพ	3
2.1.2 ภาพดิจิทัล	4
2.1.3 พิกเซล (Pixel)	4
2.1.4 ประเภทของภาพ	5
2.1.5 จุดรบกวนในภาพ (Image Noise)	6
2.2 การประมวลผลภาพ	7
2.2.1 การแปลงภาพสีเป็นภาพขาว-ดำ	8
2.3 โอเพ่นซีวี (OpenCV)	8
2.3.1 ความหมายของโอเพ่นซีวี	8
2.3.2 โครงสร้างและข้อมูลเนื้อหาในโอเพ่นซีวี	9
2.3.3 ความหมายของคอมพิวเตอร์วิชัน	9
2.4 ฟังก์ชันของโอเพ่นซีวีที่ถูกนำมาใช้งาน	10
2.4.1 ฟังก์ชันหลัก	10
2.4.2 ฟังก์ชันเสริม	13
2.5 ระบบงานที่เกี่ยวข้อง	15
2.6 งานวิจัยที่เกี่ยวข้อง	17

สารบัญ(ต่อ)

	หน้า
บทที่ 3 วิธีการดำเนินงาน	18
3.1 ภาพรวมของระบบ	18
3.1.1 ความสามารถของระบบ	18
3.2 การวิเคราะห์ระบบ	19
3.2.1 แผนภาพยูสเคส (Use case Diagram)	19
3.3 อธิบายการทำงานของโปรแกรม	20
3.3.1 ฟังก์ชันหลักของการทำงาน	20
3.3.2 ฟังก์ชันเสริมในการทำงาน	22
3.4 ส่วนแสดงผลของระบบ	24
บทที่ 4 ผลการดำเนินงานและการอภิปรายผล	25
4.1 ทดสอบการทำงานของระบบ	25
4.2 กรณีทดสอบอื่นๆ	30
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ	34
5.1 สรุปผล	33
5.2 ข้อจำกัดของปัญหาพิเศษ	33
5.3 ข้อเสนอแนะ	33
เอกสารอ้างอิง	34
ภาคผนวก	35
ภาคผนวก ก วิธีการติดตั้งไลบรารี OpenCV สำหรับใช้งานในภาษาซีพลัสพลัส	36

สารบัญรูป

รูปที่	หน้า
รูปที่ 2.1 จุดพิกัดของแต่ละพิกเซลของภาพดิจิทัล	5
รูปที่ 2.2 ภาพแบบปกติ	6
รูปที่ 2.3 ภาพเมื่อเกิดจุดรบกวน	6
รูปที่ 2.4 ภาพแสดงลำดับค่า noise	7
รูปที่ 2.5 แสดงโครงสร้างพื้นฐานของโอเพนซีวี	9
รูปที่ 2.6 แสดงการทำงานของฟังก์ชัน matchTemplate ในการจับคู่ภาพ	11
รูปที่ 2.7 แสดงการบอกตำแหน่งของภาพด้วยฟังก์ชัน minMaxLoc ร่วมกับ matchTemplate	12
รูปที่ 2.8 แสดง Source image (ในที่นี้หมายถึงภาพในวิดีโอ)	16
รูปที่ 2.9 แสดง Template image (ในที่นี้หมายถึงภาพตัวอย่างที่ต้องการเซ็นเซอร์)	16
รูปที่ 2.10 แสดงตำแหน่งภาพที่เหมือนกับ Template image ใน Source image	17
รูปที่ 2.11 การเปรียบเทียบภาพถ่ายทั้งสองภาพ และจุดที่ระบบตรวจพบ	17
รูปที่ 3.1 แสดงการออกแบบขั้นตอนการทำงานของระบบเซ็นเซอร์ภาพอัตโนมัติ	18
รูปที่ 3.2 แสดงแผนภาพยูสเคส (Use case Diagram) ของระบบ	19
รูปที่ 3.3 ตัวอย่างรูปที่จะนำมาเก็บในตัวแปร imgo	20
รูปที่ 3.4 ตัวอย่างวิดีโอที่จะนำมาเก็บในตัวแปร	20
รูปที่ 3.5 ภาพผลลัพธ์จากฟังก์ชัน matchTemplate	21
รูปที่ 3.6 แสดงการลากสีเหลี่ยมเพิ่มส่วนที่ต้องการเซ็นเซอร์	22
รูปที่ 3.7 ส่วนแสดงผลของระบบ	24
รูปที่ 4.1 การใส่ชื่อไฟล์รูปภาพ	25
รูปที่ 4.2 การใส่ชื่อไฟล์วิดีโอ	26
รูปที่ 4.3 การใส่ชื่อไฟล์ภาพที่จะใช้ในการทำงาน	26
รูปที่ 4.4 แสดงภาพตัวอย่าง	26
รูปที่ 4.5 แสดงวิดีโอที่เซ็นเซอร์เรียบร้อยแล้ว	26
รูปที่ 4.6 แสดงภาพในวิดีโอที่เอาส่วนเซ็นเซอร์ที่ไม่ต้องการออก	27
รูปที่ 4.7 แสดงการนำภาพอื่นมาแทนที่การเซ็นเซอร์	27
รูปที่ 4.8 การลากเส้นตีกรอบสีเหลี่ยมเพิ่ม	28
รูปที่ 4.9 เซ็นเซอร์ตำแหน่งที่ตีกรอบสีเหลี่ยมเพิ่ม	28
รูปที่ 4.10 ส่วนควบคุมการทำงานของระบบ	29
รูปที่ 4.11 แสดงหน้าต่างการทำงานของระบบ	29
รูปที่ 4.12 ภาพตัวอย่าง	30
รูปที่ 4.13 กรณียภาพเอียงที่ไม่สามารถเซ็นเซอร์ได้	30
รูปที่ 4.14 การเซ็นเซอร์ของระบบ	31
รูปที่ 4.15 ภาพตัวอย่าง	31
รูปที่ 4.16 ภาพเซ็นเซอร์ขนาดปกติ	32

สารบัญรูป

รูปที่	หน้า
รูปที่ 4.17 ภาพในวิดีโอมีการขยายใหญ่ขึ้น	32
รูปที่ 4.18 ภาพในวิดีโอลดขนาดลง	32

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของการทำปัญหาพิเศษ

ปัจจุบันวิดีโอมีอิทธิพลต่อผู้คนในสังคมเป็นอย่างมาก สามารถเข้าถึงผู้คนทุกเพศทุกวัย ทุกสถานที่ทุกเวลาอย่างไม่จำกัด จึงต้องตระหนักว่าวิดีโอเป็นสื่อสาธารณะที่จำเป็นต้องมีความรับผิดชอบต่อสังคมเพราะมีผู้คน และเด็กๆจำนวนมากที่เฝ้าดูสิ่งต่างๆสำหรับเด็กที่ยังมีวิจารณญาณไม่เพียงพอจึงมักคล้อยตามไปกับภาพที่เห็นว่าเป็นบรรทัดฐานของสังคมที่ควรทำตาม วิดีโอจึงไม่ได้เป็นไปเพื่อความบันเทิงเท่านั้น แต่ต้องตระหนักถึงผลกระทบที่มีต่อผู้ชมโดยเฉพาะเด็กๆ ซึ่งวิดีโอ นั้น มีหลากหลายเรื่องราวให้ได้รับชมทั้งเนื้อหาเหมาะสม และไม่เหมาะสม จึงต้องมีการเซ็นเซอร์ หรือทำภาพเบลอในส่วนของเนื้อหาที่ไม่เหมาะสมออกเพื่อป้องกันไม่ให้ออกสู่สายตาสาธารณะชน ซึ่งการเซ็นเซอร์ หรือทำภาพเบลอ นั้นถ้ามีส่วนของเนื้อหาที่ไม่เหมาะสมอยู่หลายจุดการเซ็นเซอร์แบบเดิม (การทำเซ็นเซอร์โดยใช้วิธีการแบบ manual) ค่อยข้างใช้เวลาในการทำงานยิ่งปริมาณของงานมากขึ้นเท่าไร เวลาในการทำงานยิ่งมากขึ้นเท่านั้นเพราะต้องทำงานซ้ำเดิม และประสบปัญหาที่เหมือนกัน จึงทำให้เกิดปัญหาพิเศษนี้ขึ้นเพื่อเซ็นเซอร์เนื้อหาไม่เหมาะสมในวิดีโอให้แสดงผลออกมาอัตโนมัติทำให้เกิดความสะดวก และรวดเร็วขึ้นในการทำงาน

1.2 วัตถุประสงค์ของการทำปัญหาพิเศษ

- 1.2.1) เพื่อพัฒนาให้โปรแกรมคอมพิวเตอร์สามารถค้นหาทุกตำแหน่งของภาพที่ต้องการในวิดีโอได้ ทำให้ลดปัญหาการทำงานซ้ำหากภาพที่ต้องเซ็นเซอร์เป็นภาพเดียวกันแต่มีหลายตำแหน่ง
- 1.2.2) เพื่อทำเซ็นเซอร์ภาพในวิดีโอได้ครบตรงตามที่กำหนดก่อนจะเผยแพร่ออกไป
- 1.2.3) เพื่อเซ็นเซอร์วิดีโอให้แสดงผลออกมาอัตโนมัติ
- 1.2.4) เพื่อพัฒนาให้มีการนำภาพอื่นมาใส่แทนการเซ็นเซอร์ในวิดีโอ
- 1.2.5) เพื่อความสะดวกในการทำงาน และลดระยะเวลาในการทำงานให้มีความรวดเร็วขึ้นในกรณีที่มีปริมาณงานจำนวนมาก

1.3 ขอบเขตของปัญหาพิเศษ

- 1.3.1) การค้นหาตำแหน่งภาพที่ต้องการในวิดีโอ ระบบสามารถค้นหาตำแหน่งของภาพได้ทุกภาพในวิดีโอ โดยภาพที่ค้นหามาได้นั้นอาจตรงตามที่ต้องการหรือมีความใกล้เคียงกัน
- 1.3.2) การเลือกภาพตัวอย่างที่จะนำมาเซ็นเซอร์ในวิดีโอ สามารถเลือกภาพตัวอย่างได้เพียงภาพเดียวต่อการทำงาน 1 ครั้ง
- 1.3.3) ภาพตัวอย่าง สามารถเลือกใช้ได้เฉพาะนามสกุลไฟล์ .jpg, .png เท่านั้น
- 1.3.4) วิดีโอ สามารถเลือกใช้ได้เฉพาะนามสกุลไฟล์ .mp4, .avi เท่านั้น
- 1.3.5) สามารถทำเซ็นเซอร์ภาพที่ต้องการในวิดีโอต่างๆได้ ในการทำงานครั้งเดียวแม้ภาพนั้นมีอยู่หลายตำแหน่งในวิดีโอ

1.3.6) สามารถทำงานได้เฉพาะระบบปฏิบัติการ windows 7 ขึ้นไป

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1) สามารถนำไปใช้งานการทำเซ็นเซอร์วิดีโอต่างๆ เพื่อให้มีความสะดวก และรวดเร็วขึ้นกว่าแบบเดิม (การทำเซ็นเซอร์โดยใช้วิธีการแบบ manual)
- 1.4.2) สามารถนำไปประยุกต์ใช้ในการป้องกันภาพที่ไม่เหมาะสมในวิดีโอต่างๆ ไม่ให้เผยแพร่ออกไป
- 1.4.3) สามารถเป็นแนวทางในการศึกษาต่อสำหรับผู้ที่สนใจการทำงานตรวจจับวัตถุภาพในเฟรมวิดีโอ และการทำเซ็นเซอร์ได้

1.5 อุปกรณ์ที่ใช้ในการทำปัญหาพิเศษ

- 1.5.1) คอมพิวเตอร์ตั้งโต๊ะ หรือคอมพิวเตอร์พกพา
- 1.5.2) ซอฟต์แวร์ที่เกี่ยวข้อง ดังนี้
 - ระบบปฏิบัติการ Windows
 - Visual studio เครื่องมือที่ใช้ร่วมกับ OpenCV ใช้ในการวิเคราะห์ภาพ

บทที่ 2

ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีและความรู้พื้นฐานที่เกี่ยวข้องกับการพัฒนาโปรแกรม โดยเกิดขึ้นจากการรวบรวมข้อมูลและนำความรู้เหล่านี้ไปประยุกต์ใช้ ซึ่งทฤษฎีและความรู้พื้นฐานที่เกี่ยวข้องได้รวบรวมไว้มีดังนี้

2.1 ภาพ

2.1.1 ภาพ

[1] โดยปกติแล้วข้อมูลภาพต่างๆ ไปนั้นได้มาจากการที่แสงตกกระทบกับวัตถุแล้วเกิดการสะท้อนผ่านเลนส์เข้าสู่ตัวบันทึกภาพ อาจอยู่ในรูปแบบของตัวตรวจจับ (Sensor) หรือฟิล์ม (Film) หากนึกถึงเวลาที่ได้ถ่ายภาพด้วยกล้องดิจิทัล วัตถุหรือภาพที่เราสามารถเห็นได้ด้วยตา นั้นเป็นข้อมูลสามมิติ (Three-dimension) ที่ประกอบไปด้วยความลึก ความสูง และความกว้าง แต่หากแปลงข้อมูลภาพออกมาเป็นข้อมูลภาพดิจิทัล (Digital Image) ข้อมูลของภาพนั้นจะประกอบไปด้วยความกว้างและความสูงของภาพ (Width and Height) เท่านั้น ทั้งนี้ในปัจจุบันมีอุปกรณ์อิเล็กทรอนิกส์หลายประเภทที่สามารถสร้างสัญญาณภาพดิจิทัลได้สะดวกและง่ายขึ้น เช่น กล้องถ่ายภาพดิจิทัล กล้องวิดีโอที่บันทึกเป็นต้น

โดยทั่วไปแล้วสามารถแบ่งรูปภาพที่ปรากฏและใช้งานบนเครื่องคอมพิวเตอร์ ออกเป็น 2 ประเภทคือ รูปภาพแบบบิตแมป (Bitmap Image) และรูปภาพแบบเวกเตอร์ (Vector Image) โดยรูปภาพแบบบิตแมปจะถูกแบ่งออกเป็นส่วนย่อยเล็กๆ หลายๆ ส่วนหรือที่เรียกว่า พิกเซล (Pixels) ที่ถูกนำมารวมกันและใช้แสดงผลภาพ ส่วนรูปภาพแบบเวกเตอร์จะประกอบด้วยเส้นสายต่างๆ ที่สร้างขึ้นจากการคำนวณทางคณิตศาสตร์ของลักษณะทางเรขาคณิตเพื่อสร้างรูปทรงต่างๆ ที่เราเห็น ซึ่งเรียกว่าเวกเตอร์ (Vector)

2.1.1.1 ภาพแบบบิตแมป (Bitmap Image)

ภาพบิตแมป หรือเรียกอีกอย่างหนึ่งว่าภาพแบบราสเตอร์ (Raster Image) เป็นภาพที่ประกอบขึ้นจากจุดขนาดเล็กๆ หรือที่เรียกว่า พิกเซลจำนวนมากมาเรียงต่อกันจนเป็นภาพภาพหนึ่ง เพื่อให้เห็นภาพลักษณะนี้ชัดเจนยิ่งขึ้น ให้นึกถึงการสร้างภาพบนตารางสี่เหลี่ยมเล็กๆ จะใช้สีแต้มในช่องสี่เหลี่ยมแต่ละช่องจนกลายเป็นภาพสมบูรณ์ที่มีขนาดใหญ่ ซึ่งเหมือนกับการป้ายสีแปดอักษรบนจอจันท์ ภาพแบบบิตแมปนี้จะมีจำนวนจุดเล็กๆ นั้นได้เมื่อลองขยายภาพดูจะเห็นเป็นรูปตาราง ยิ่งขยายใหญ่เท่าไรตารางสี่เหลี่ยมก็ยังมีขนาดใหญ่ขึ้นจนทำให้มองเห็นจุดของภาพหรือพิกเซล จึงมีผลทำให้ไม่สามารถบอกได้ว่าภาพนั้นเป็นภาพอะไร สิ่งนั้นส่งผลเช่นเดียวกับหากขยายภาพบิตแมปบนหน้าจอคอมพิวเตอร์ จะมีผลทำให้รายละเอียดความคมชัดของภาพลดลง โดยทั่วไปแล้วภาพบิตแมปเป็นภาพประเภทที่นิยมใช้กันมากในภาพถ่ายหรือภาพวาด เนื่องจากภาพบิตแมปสามารถไล่โทนสีและแสงเงาได้เหมือนจริงที่สุด

2.1.1.2 ภาพแบบเวกเตอร์ (Vector Image)

ภาพแบบเวกเตอร์มีคุณสมบัติที่แตกต่างกับภาพแบบบิตแมป คือภาพประเภทนี้ไม่ว่าจะขยายให้ใหญ่เล็กน้อยเพียงใด ก็ยังคงรายละเอียดและความคมชัดไว้ได้เหมือนเช่นเดิมโดยไม่ผิดเพี้ยน เนื่องจากภาพแบบเวกเตอร์นั้นประกอบด้วยเส้นตรง เส้นโค้ง และรูปทรงต่างๆ ภาพที่ได้จะสร้างขึ้นจากคำสั่งที่บอกถึงลักษณะของในรูปร่างทางเรขาคณิตด้วยสมการทางคณิตศาสตร์ ดังนั้นโปรแกรมที่ต้องการเปิดรูปภาพจะต้องนำสมการต่างๆ ที่บันทึกเอาไว้มาคำนวณและสร้างรูปทรงของภาพขึ้นมาใหม่ จุดเด่นคือไม่ว่าจะขยายภาพให้ใหญ่เล็กน้อยเพียงใด คอมพิวเตอร์จะคำนวณค่าต่างๆ ให้ใหม่ทุกครั้ง ทำให้ภาพที่ได้มีความคมชัด ภาพแบบเวกเตอร์จึงเหมาะกับการใช้งานที่ต้องการความแม่นยำสูง และต้องการความละเอียดสูง เช่น การสร้างภาพโลโก้ การสร้างภาพสามมิติ การสร้างแบบร่างทางวิศวกรรม

2.1.2 ภาพดิจิทัล

ข้อมูลภาพดิจิทัลระดับความเข้มเทา สามารถเขียนให้อยู่ในรูปของฟังก์ชันสองมิติของค่าระดับความเข้มแสง โดยค่าของฟังก์ชัน $f(x,y)$ จะแสดงถึงค่าความเข้มแสงที่จุดพิกัดสเปเชียล (Spatial) x,y ใดๆ โดยทั่วไปนิยมเขียนแทนภาพดิจิทัลสองมิติ $f(x,y)$ ขนาด $M \times N$ ให้อยู่ในรูปของเมทริกซ์ (Matrix) สองมิติได้ดังสมการที่ 2.1 ส่วนรูปที่ 2.1 แสดงตำแหน่งของจุดพิกัดใดๆ ของภาพดิจิทัลในแต่ละจุดพิกเซลของภาพ

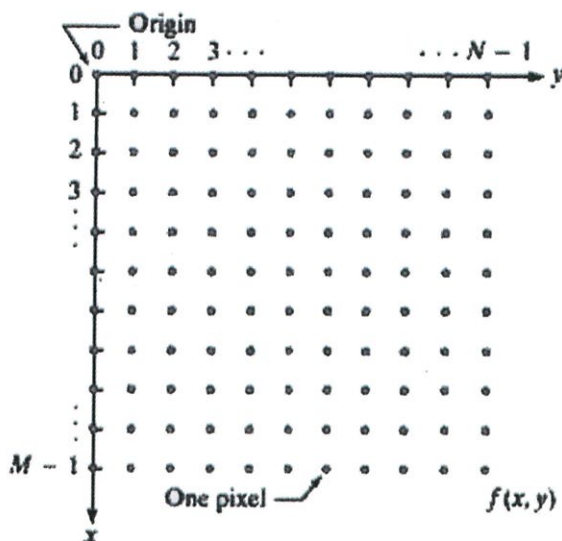
$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

สมการที่ 2.1

โดยที่ $f(x,y)$ แทนค่าระดับความเข้มเทาที่จุดพิกัด x,y ใดๆ x,y คือจุดพิกัดของลำดับแถวและหลักของภาพ ตามลำดับ

2.1.3 พิกเซล (Pixel)

คือ ความเข้มแสงที่รวมกันทำให้เกิดเป็นภาพ ภาพหนึ่งๆ จะประกอบด้วยพิกเซล[2] มากมาย ซึ่งภาพแต่ละภาพที่สร้างขึ้นจะมีความหนาแน่นของพิกเซล เหล่านี้แตกต่างออกไป ความหนาแน่นนี้เป็นตัวบอกถึงความละเอียด (Resolution) ของภาพซึ่งมีหน่วยเป็น ppi (Pixel Per Inch)



รูปที่ 2.1 จุดพิกัดของแต่ละพิกเซลของภาพดิจิทัล

คุณภาพของรูปภาพดิจิทัลนั้นจะขึ้นอยู่กับจำนวนพิกเซลและจำนวนค่าระดับความเข้มเทา กล่าวคือ จำนวนพิกเซลในแต่ละแถวและแต่ละหลักของรูปภาพดิจิทัลเป็นจำนวนค่าสุ่มตัวอย่างทั้งหมด หรือเป็นความละเอียดของสัญญาณภาพดิจิทัลนั่นเอง ดังนั้นการลดจำนวนพิกเซลลงจะเป็นการลดความละเอียดของรูปภาพดิจิทัล ทั้งนี้จึงมีผลทำให้รูปภาพผลลัพธ์ที่ได้อาจผิดเพี้ยนไปจากรูปภาพต้นฉบับได้ ในทำนองเดียวกัน จำนวนค่าระดับของสัญญาณดิจิทัลจะถูกกำหนดด้วยค่าระดับความเข้มเทาที่ใช้ในการแสดงผลภาพดิจิทัล นั่นคือคุณภาพของรูปภาพดิจิทัลจะขึ้นอยู่กับจำนวนค่าระดับความเข้มเทาที่ใช้ อย่างไรก็ตาม การเพิ่มพิกเซลหรือจำนวนค่าระดับความเข้มเทาในบางครั้ง อาจจะได้ไม่ได้เพิ่มคุณภาพของรูปภาพดิจิทัลหรือเพิ่มน้อยมาก โดยปกติแล้วภาพจะดูคมชัดเท่าใด จะขึ้นอยู่กับภาพนั้นมีจำนวนพิกเซลอยู่มากเท่าใด ส่วนใหญ่มักจะบอกความละเอียดของภาพ (Resolution) โดยดูจากจำนวนพิกเซลต่อนิ้ว (Pixel/Inch:PPI) หรือพิกเซลต่อตารางนิ้ว (Pixels/Inch²) หากภาพมีจำนวนพิกเซลต่อหนึ่งตารางนิ้วมาก ภาพก็จะมีรายละเอียดเหมือนจริงมากขึ้น โดยภาพที่มีความละเอียดมากๆ จะเรียกภาพนี้ว่ามี Resolution สูง ส่วนภาพที่มีความละเอียดน้อยก็เรียกว่าภาพนี้ว่ามี Resolution ต่ำ เช่น ภาพขนาดกว้าง 1 นิ้ว ยาว 1 นิ้ว ที่มีความละเอียด 72 PPI จะมีจำนวนพิกเซลทั้งหมด 5184 จุดพิกเซล ($72 \times 72 = 5184$)

2.1.4 ประเภทของภาพ

โดยทั่วไปแล้ว สามารถแบ่งประเภทของภาพบิตแมปตามคุณสมบัติการแสดงผลของสีภาพเป็น 4 ประเภท ดังนี้คือ

2.1.4.1 ภาพระดับความเข้มเทา (Intensity Image or Gray Scale Image)

ลักษณะของภาพชนิดนี้ ในแต่ละพิกเซลจะมีค่าความเข้มแสงในแต่ละระดับที่แตกต่างกันไป ตั้งแต่ระดับเทาดำไปยั้งระดับสีขาว เราสามารถกำหนดระดับความเข้มแสงได้โดยใช้ค่าระดับความเข้มเทา (Gray Scale หรือ Gray Level) โดยปกติทั่วไป ภาพแบบระดับสีเทาจะมีค่าระดับความเข้มเทาเท่ากับ 8 บิต ดังนั้นค่าความเข้มแสงจะถูกแบ่งออกเป็น 256 ระดับ เมื่อค่าระดับความเข้มเทามีค่าเป็น 0 จะหมายถึงจุดภาพนั้นมีค่าความเข้มของแสงต่ำ จะทำให้จุดภาพเป็นสีดำ ในทางกลับกัน หากค่าระดับความเข้มเทาเป็น 255 จะหมายถึงจุดภาพนั้นมีค่าความเข้มของแสงมาก จะทำให้

จุดภาพเป็นสีขาว ซึ่งสีขาวจะถูกแทนด้วยค่าความเข้มเทาเท่ากับ 255 (1111111) และสีดำจะถูกแทนด้วยค่าความเข้มเทาเท่ากับ 0 (0000000) ส่วนค่าระหว่าง 0-255 ก็จะมีค่าไล่ระดับเฉดสีจากสีดำไปหาสีขาวนั่นเอง

2.1.4.2 ภาพสี (Color Image)

ภาพชนิดนี้ แต่ละจุดภาพหรือพิกเซลของภาพจะเก็บค่าระดับความเข้มเทาของแต่ละแถบแสงของแม่สีหลัก 3 สีที่ซ้อนกันอยู่คือ สีแดง (Red) สีเขียว (Green) สีน้ำเงิน (Blue) ซึ่งในแต่ละพิกเซลจะไล่ตามระดับความเข้มในแต่ละแถบนั้น

2.1.4.3 ภาพไบนารี (Binary Image)

ภาพไบนารีจะแสดงลักษณะของข้อมูลภาพในรูปแบบขาวดำ กล่าวคือในแต่ละพิกเซลของภาพจะถูกแสดงด้วยค่าแบบไบนารี (Binary) คือมี 1 บิต ซึ่งประกอบไปด้วยค่า 1 และ 0 โดยที่ 1 หมายถึงจุดภาพสีดำ ภาพประเภทนี้เหมาะสำหรับภาพที่เกี่ยวกับตัวอักษร (Text) ภาพลายนิ้วมือ (Finger Print) เป็นต้น

2.1.4.4 ภาพแบบดัชนี (Index Image)

ภาพประเภทนี้ ในแต่ละพิกเซลของภาพจะเก็บค่าดัชนี (Index Number) ซึ่งเป็นตัวเลขจำนวนเต็ม จะถูกนำค่าดัชนีดังกล่าวไปเทียบกับตารางสี (Color Tabel) ซึ่งเป็นตารางแสดงค่าแสงสีแดง สีเขียว และสีน้ำเงิน ซึ่งค่าดัชนีนี้จะเป็นตัวบ่งชี้ให้เห็นว่าภาพในแต่ละตำแหน่งพิกเซลใดๆ มีค่าอัตราส่วนของแม่สี 3 สีในอัตราส่วนละเท่าใด

2.1.5 จุดรบกวนในภาพ (Image Noise)

เป็นสัญญาณรบกวนที่เกิดขึ้นในภาพ[3] ทำให้เกิดความไม่สม่ำเสมอของความสว่างและสี

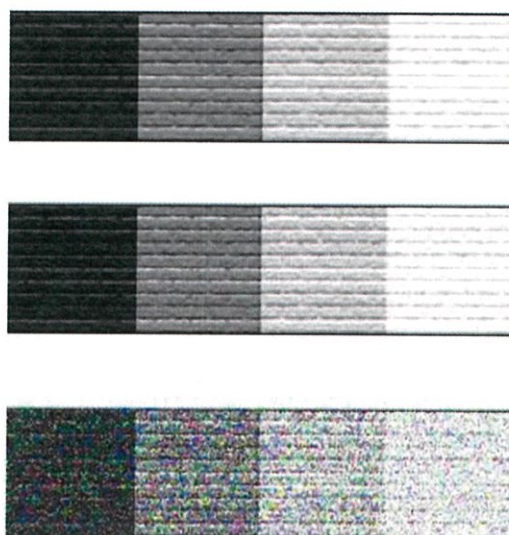


รูปที่ 2.2 ภาพแบบปกติ



รูปที่ 2.3 ภาพเมื่อเกิดจุดรบกวน

จากภาพจะเห็นว่าภาพที่มีสีครบถ้วนสวยงามกลับมีคุณภาพลดต่ำไปมากจากรูปที่ 2.2 และ 2.3



รูปที่ 2.4 ภาพแสดงลำดับค่า noise

จากรูปที่ 2.4 ระดับการเกิด noise ที่มากขึ้นจะทำให้รายละเอียดของภาพหายไป ระดับ noise ที่เท่ากันแต่จะมองเห็น Grain ต่างกันขึ้นอยู่กับระดับของสัญญาณภาพส่วนสว่างและส่วนเงา รายละเอียดจะน้อยกว่าส่วนน้ำหนักสีกลาง

2.2 การประมวลผลภาพ

การประมวลผลภาพ คือ การเอาภาพมาประมวลผล โดยคำนวณด้วยคอมพิวเตอร์เพื่อให้คอมพิวเตอร์รู้ว่าภาพนั้นคือภาพอะไรใช้สิ่งที่เราต้องการหรือไม่

การมองเห็นของมนุษย์เป็นสิ่งที่สำคัญและเป็นกลไกการรับรู้ภาพที่ซับซ้อนอย่างหนึ่ง ซึ่งจะให้ข้อมูลที่มีความจำเป็นสำหรับใช้งานง่าย ๆ (ตัวอย่างเช่น การจดจำวัตถุ) และสำหรับงานที่มีความซับซ้อน (ได้แก่ การวางแผน การตัดสินใจ การค้นคว้าทางวิทยาศาสตร์ การพัฒนาทางด้านความคิด) รูปภาพมีบทบาทมากสำหรับองค์กรต่างๆ เช่น หนังสือพิมพ์ โทรทัศน์ ภาพยนตร์ ซึ่งได้ใช้ภาพ (ภาพนิ่ง ภาพเคลื่อนไหว) เป็นสื่อนำเสนอข้อมูลข่าวสารต่างๆ สิ่งที่น่าสนใจของข้อมูลที่เกี่ยวข้องกับการมองเห็นหรือข้อมูลภาพนั้นก็คือกระบวนการประมวลผลภาพ (Image Processing) โดยใช้ดีจิตอลคอมพิวเตอร์

Digital image processing จะเกี่ยวกับการแปลงข้อมูลภาพให้อยู่ในรูปแบบข้อมูลดิจิตอล (Digital format) ซึ่งสามารถที่จะนำเอาข้อมูลนี้จัดผ่านการบวนการต่างๆ ด้วยดีจิตอลคอมพิวเตอร์ได้ในระบบของดีจิตอล อินพุตและเอาต์พุตของระบบจะอยู่ในรูปแบบดิจิตอลเท่านั้น

Digital image analysis จะเกี่ยวกับวิธีการอธิบายและการจดจำข้อมูลภาพดิจิตอลเหล่านั้น ในการวิเคราะห์ภาพมีอยู่หลายวิธีด้วยกันที่ได้นำมาจากการทำงานของตามนุษย์ (human vision) นั่นก็คืองานด้าน Computer Vision เป็นลักษณะเดียวกับ Digital image analysis นั่นเอง การมองเห็นของมนุษย์นับว่าเป็นกระบวนการที่ซับซ้อนซึ่งลักษณะเทคนิคต่างๆไปในกระบวนการ Digital image analysis และ Computer Vision จะค่อนข้างซับซ้อนเช่นกัน

การคำนวณนั้นมีหลายวิธี เช่น การนำเอาจุดสีแต่ละ Pixel มาคำนวณหรือการนำหลายๆจุดมารวมกันเป็นบริเวณกว้างขึ้น เช่น การดู Pattern ของวัตถุ การวิเคราะห์หารูปร่างของวัตถุเพื่อหาค่าอะไรบางอย่างที่ช่วยให้เข้าใจได้ว่าภาพนั้นมีลักษณะอย่างไร

2.2.1 การแปลงภาพสีเป็นภาพขาว-ดำ

การแปลงภาพสีให้เป็นภาพขาว-ดำ (Thresholding) เป็นกระบวนการแปลงภาพสีให้มีการแสดงผลได้แค่ 2 ระดับ คือ ขาว และดำ โดยจะแปลงข้อมูลภาพให้เป็นภาพ binary (Binary Image) มีกระบวนการแปลงภาพที่มีความเข้มหลายระดับ (Multilevel Image) ให้เป็นภาพที่มีความเข้มเพียง 2 ระดับ หรือ 1 บิต (bit) คือ 0 และ 1 โดย 0 แทนด้วยจุดที่มีภาพสีขาว และ 1 แทนด้วยจุดที่มีภาพสีดำ Threshold Technique คือการพิจารณาจุด pixel ในภาพว่าจุดใดควรจะเป็นจุดขาว หรือจุดใดควรจะเป็นจุดดำที่มีค่าเท่ากับ 1 โดยจะทำการเปรียบเทียบค่าของแต่ละ pixel ($f(x,y)$) กับค่าคงที่ ที่เรียกว่า Threshold (Threshold Value) เทคนิคนี้นิยมใช้กันมากในกรณีที่มีความแตกต่างระหว่างวัตถุ และพื้นหลัง ค่า pixel ในภาพมีค่าน้อยกว่าค่า Threshold จะถูกกำหนดเป็น 1 และถ้าค่าของ pixel ใดๆในภาพมีค่ามากกว่าหรือเท่ากับค่า Threshold จะถูกกำหนดเป็น 0 ในการทำภาพ Binary โดยการ Thresholding ให้ได้ภาพดี และคมชัด ต้องเกิดจากการเลือกค่า Threshold ที่ถูกต้องและเหมาะสม ถ้าเลือกค่าไม่เหมาะสม ภาพที่ได้จะขาดความคมชัดหรืออาจทำให้รายละเอียดของภาพขาดหายไป หรือภาพที่ได้อาจมืดเกินไป หรือสว่างเกินไป หรืออาจจะเป็นภาพที่มีสิ่งรบกวน (Noise) เกิดขึ้นทำให้ภาพผลลัพธ์ที่ไม่ชัดเจน

2.3 โอเพ่นซีวี (OpenCV)

2.3.1 ความหมายของโอเพ่นซีวี

OpenCV[4] เป็น Library ของ Programming Functions ที่ใช้ช่วยในการเขียนโปรแกรมเกี่ยวกับ Computer Vision เป็นหลักสามารถนำไปประยุกต์หรือใช้งานได้เลยโดยไม่ต้องทำการคิด Algorithm หรือทำการโค้ดขึ้นเอง โดยมี source code และตัวอย่าง ถูกพัฒนาจากทีมพัฒนาของ Intel และ optimize code สำหรับ CPU 586 ของ Intel ทำให้เป็น library ที่มีความเร็วในการประมวลผลอยู่ในระดับต้นๆ เร็วกว่าที่มีขายบางตัว โดยเปิดให้บริการฟรีสำหรับการใช้งานภายใต้ใบอนุญาต open source BSD ซึ่ง library เป็นระบบหลายระบบปฏิบัติการ ที่จะเน้นหลักในการประมวลผลภาพแบบเรียลไทม์

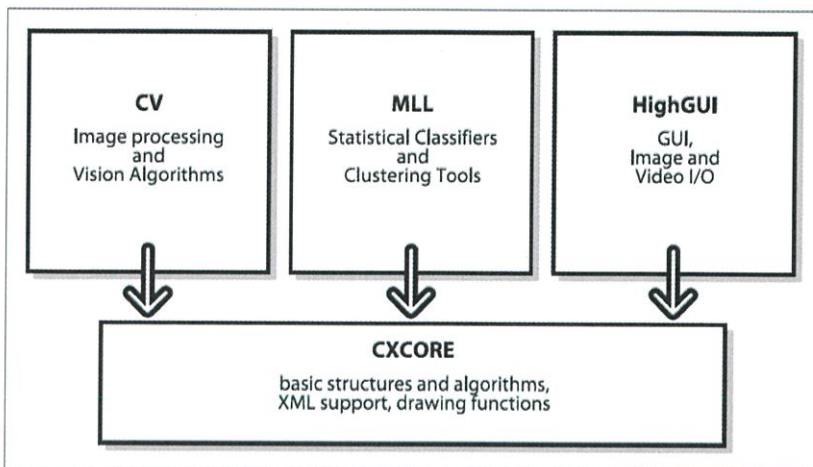
เพื่อให้การพัฒนาโปรแกรมทางด้านการมองเห็นของคอมพิวเตอร์ (Computer Vision) คือสามารถประมวลผลภาพดิจิทัลได้ทั้งภาพนิ่ง และภาพเคลื่อนไหว เช่น ภาพจากกล้อง VDO หรือ VDO File เป็นไปได้อย่างสะดวก มีฟังก์ชันสำเร็จรูปสำหรับจัดการข้อมูลภาพ และการประมวลผลภาพพื้นฐาน เช่น การหาขอบภาพ การกรองข้อมูลภาพ การประมวลผลเกี่ยวกับภาพและสัญญาณ (Image and Signal Processing), การตรวจสอบลักษณะวัตถุจากภาพหรือวิดีโอ (Object Identification), การจดจำใบหน้า (Face Recognition), ม่านตา (iris Recognition), ตรวจสอบขอบหรือด้านของวัตถุ (Edge Detection), ตรวจสอบความเคลื่อนไหว (Motion Detection) และอื่นๆ

เป้าหมายหนึ่งของโอเพ่นซีวี คือ การให้บริการโครงสร้างสำหรับคอมพิวเตอร์วิชั่นที่ใช้งานง่าย ทำให้ผู้ใช้งานสามารถสร้างแอปพลิเคชันที่มีความซับซ้อนได้อย่างรวดเร็ว ไลบรารีโอเพ่นซีวีบรรจุฟังก์ชันสำหรับงานด้านการประมวลผลภาพประมาณ 500 ฟังก์ชัน ที่ครอบคลุมการใช้งาน

หลายด้าน เช่น การตรวจสอบผลิตภัณฑ์ การถ่ายภาพทางการแพทย์ ความปลอดภัยส่วนติดต่อกับ
 ผู้ใช้การปรับเทียบกล้อง ระบบสเตอริโอ และหุ่นยนต์

2.3.2 โครงสร้างและข้อมูลเนื้อหาในโอเพ่นซีวี

โอเพ่นซีวีประกอบด้วย 4 ส่วนประกอบหลัก และส่วนประกอบ 4 ส่วนถูกแสดง
 ดังรูปที่ 2.5[5]



รูปที่ 2.5 แสดงโครงสร้างพื้นฐานของโอเพ่นซีวี

ความหมายของส่วนประกอบ 4 ส่วน ดังรูปที่ 2.5 แสดงได้ดังนี้

- 1) CXCORE เป็นฟังก์ชันเบื้องต้นที่ใช้จัดการเกี่ยวกับจุดขนาดอาร์เรย์ หน่วยความจำคำสั่งในการวาดภาพ และการประกาศตัวแปรภาพ
- 2) CV ใช้ในการประมวลผลและการวิเคราะห์รูปภาพ
- 3) Machine Learning เป็นไลบรารีที่รวมคลาสและฟังก์ชันทางสถิติ (Statistical) การแยกคลาส และการแบ่งกลุ่มของข้อมูล (Clustering)
- 4) HighGUI เป็นไลบรารีที่ใช้ในการดึงภาพ การบันทึกภาพ การเปลี่ยนขนาดและเคลื่อนย้ายหน้าต่าง รวมไปถึงการตรวจสอบเมาส์ และแป้นพิมพ์

2.3.3 ความหมายของคอมพิวเตอร์วิชัน

[6]คอมพิวเตอร์วิทัศน์ (Computer Vision) เป็นสาขาหนึ่งของวิทยาการคอมพิวเตอร์ ว่าด้วยเรื่องเกี่ยวกับการดึงสารสนเทศจากรูปภาพหรือวิดีโอ เครื่องมือที่ใช้ในคอมพิวเตอร์วิทัศน์ ได้แก่ คณิตศาสตร์โดยเฉพาะ เรขาคณิต พีชคณิตเชิงเส้น สถิติ และการวิจัยดำเนินงาน (การหาค่าเหมาะที่สุด) และการวิเคราะห์เชิงฟังก์ชัน โดยเครื่องมือเหล่านี้ใช้ในการสร้างขั้นตอนวิธี หรือขั้นตอนวิธีในการแยกส่วนภาพ และการจัดกลุ่มภาพเพื่อให้คอมพิวเตอร์สามารถเข้าใจทัศนียภาพหรือคุณลักษณะต่างๆ ในภาพเป้าหมายโดยทั่วไปของคอมพิวเตอร์วิทัศน์ ได้แก่

- 1) การตรวจจับตัดแบ่งขอบเขต ระบุตำแหน่ง และจำวัตถุที่ต้องการในภาพ เช่น หน้าคน
- 2) การประเมินผล สำหรับการตัดแบ่งขอบเขตวัตถุในภาพ หรือการวางทาบเทียบ เป็นต้น
- 3) การวางทาบเทียบของมุมมองต่างๆของทัศนียภาพ หรือวัตถุหนึ่งๆ
- 4) การติดตามวัตถุหนึ่งในภาพต่อเนื่อง
- 5) การเชื่อมโยงมุมมองต่างๆของทัศนียภาพหนึ่งๆเพื่อสร้างแบบจำลองสามมิติ ของทัศนียภาพนั้น แบบจำลองดังกล่าวอาจนำมาใช้เพื่อนำทางหุ่นยนต์ ในทัศนียภาพจริง

6) การกะประมาณ ท่าทางต่างๆของมนุษย์ และส่วนต่างๆของร่างกาย เช่น แขน ขา นิ้วมือ ฯลฯ ในสามมิติ

7) การค้นหารูปภาพด้วยเนื้อหาของภาพในฐานข้อมูลภาพขนาดใหญ่

เพื่อที่จะบรรลุซึ่งเป้าหมายเหล่านี้ ระบบคอมพิวเตอร์วิทัศน์จะต้องใช้กระบวนการต่างๆ เช่น การรู้จำแบบ การเรียนรู้เชิงสถิติ เรขาคณิตเชิงภาพฉาย การประมวลผลภาพ ทฤษฎีกราฟ เป็นต้น คอมพิวเตอร์วิทัศน์การรับรู้มันเกี่ยวข้องกับอย่างใกล้ชิดกับจิตวิทยาการรับรู้ และการคำนวณทางชีวภาพ คอมพิวเตอร์วิทัศน์ที่น่าสนใจ ได้แก่ การสร้างปรากฏการณ์ลวงตาต่างๆในภาพยนตร์ ปัจจุบันได้พบการประยุกต์ใช้คอมพิวเตอร์วิทัศน์ในสาขาต่างๆ เช่น การแพทย์ การทหาร ระบบตรวจตรา และรักษาความปลอดภัย การตรวจสอบและควบคุมคุณภาพ ระบบหุ่นยนต์ รถยนต์

2.4 ฟังก์ชันของโอเพ่นซีวีที่ถูกนำมาใช้งาน

ฟังก์ชันในโอเพ่นซีวีนั้นมีหลากหลาย หากต้องการใช้ฟังก์ชันต่างๆให้เข้ากับงานของเรามากที่สุด ต้องประยุกต์ใช้ฟังก์ชันหลายๆอย่างให้เข้ากับระบบที่จะทำให้มากที่สุด โดยฟังก์ชันที่ถูกนำมาใช้งานในระบบนี้แบ่งออกเป็น 2 ส่วนใหญ่คือ

2.4.1 ฟังก์ชันหลัก

เป็นฟังก์ชันที่ถูกเรียกใช้งานทุกครั้งที่ในการทำงาน เพื่อให้ระบบทำงานตามเป้าหมายได้อย่างครบถ้วน

2.4.1.1 Mat::imread (const string& filename, int flags=1)

คือ ฟังก์ชันที่ใช้งานในการอ่านไฟล์รูปจากข้อมูลในคอมพิวเตอร์เก็บใส่ตัวแปรประเภท Mat โดย Filename คือ path ไฟล์ของรูปนั้น Flags คือ กำหนดรูปที่โหลดมาจะให้เป็นรูปเป็นประเภทอะไร ในระบบงานนี้ได้ใช้ CV_LOAD_IMAGE_UNCHANGED นั่นคือไม่มีการเปลี่ยนแปลงใดอาจใช้เลข 1 ก็ได้เช่นกัน

2.4.1.2 VideoCapture::VideoCapture (const string& filename)

คือ ฟังก์ชันที่ใช้งานในการอ่านไฟล์วิดีโอจากข้อมูลในคอมพิวเตอร์เก็บใส่ตัวแปรประเภท VideoCapture โดย Filename คือ path ไฟล์ของวิดีโอ

2.4.1.3 VideoCapture::read (Mat& Image)

เป็นฟังก์ชันเฉพาะของตัวแปรประเภท VideoCapture ใช้ในการอ่านเฟรมวิดีโอ เก็บใส่ตัวแปรประเภทรูปภาพ Image คือ ตัวแปรประเภท Mat ไว้เก็บรูปแต่ละเฟรม

2.4.1.4 cvtColor (InputArray src, OutputArray dst, int code, int dstCn=0)

คือฟังก์ชันที่ใช้เพื่อแปลงภาพสีประเภทหนึ่งเป็นอีกประเภทหนึ่ง โดยในระบบได้ใช้ฟังก์ชันนี้ในการแปลงภาพสี 3 มิติ (BGR น้ำเงิน,เขียว,แดง) เป็นภาพสี 1 มิติ (สีขาว-ดำ) เพื่อให้ง่ายต่อการคำนวณในการทำงานฟังก์ชันถัดไป

2.4.1.5 Mat::create (int rows, int cols, int type)

ฟังก์ชันเฉพาะของตัวแปรประเภท Mat ใช้ในการสร้างรูปร่างๆ มา 1 รูป

2.4.1.6 matchTemplate (InputArray image, InputArray templ, OutputArray result, int method)

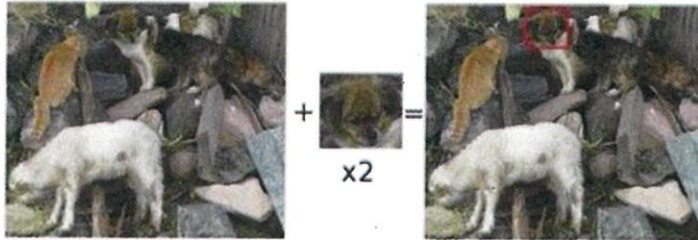
ฟังก์ชันที่ใช้ในการคำนวณเปรียบเทียบรูป 2รูป โดยรูปแรกเป็นรูปที่ต้องค้นหาในวิดีโอ รูปที่สองเป็นรูปภาพตัวอย่าง เมื่อนำรูปทั้งสองมาคำนวณได้ผลลัพธ์เก็บค่านั้นลงในรูปผลลัพธ์ในแต่ละตำแหน่งที่ค้นหา

Image เป็นตัวแปร Mat ที่ต้องการค้นหาข้างในรูป (ในระบบนี้ใช้เป็นรูปในเฟรมวิดีโอ)

Templ เป็นตัวแปร Mat เป็นรูปตัวอย่างที่จะนำมาค้นหา

Result เป็นตัวแปร Mat รับค่า มาเป็นรูปผลลัพธ์ที่มาจากจากการคำนวณของภาพทั้งสอง

Method คือ สูตรที่ใช้ในการคำนวณการจับคู่ภาพในระบบนี้ใช้สูตร `CV_TM_CCORR_NORMED` ในการคำนวณ[7]



รูปที่ 2.6 แสดงการทำงานของฟังก์ชัน `matchTemplate` ในการจับคู่ภาพ

2.4.1.7 `threshold` (InputArray src, OutputArray dst, double thresh, double maxval, int type)

เป็นฟังก์ชันกำหนดขอบเขตของค่าสีเพื่อตัดช่วงค่าสีที่ไม่ต้องการออกไป ในระบบนี้ใช้ในการหาตำแหน่งภาพที่เหมือนกันให้มีความถูกต้องมากขึ้น

Src คือ ตัวแปร Mat ภาพที่รับเข้า

Dst คือ ตัวแปร Mat ภาพที่แสดงผลออกมา

Thresh คือ ค่าthreshold ที่กำหนดขอบเขต

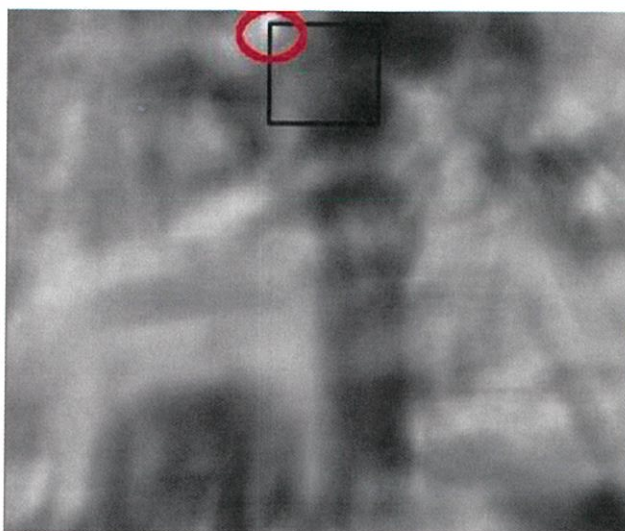
Maxval คือ กรณีที่ใช้ threshold ประเภท binary ค่าค่าสูงสุด (ในระบบนี้เป็น 0)

Type คือ ประเภท Threshold ที่จะใช้ในการค้นหา (ในระบบนี้ใช้ `CV_THRESH_TOZERO`)

2.4.1.8 minMaxLoc (InputArray src, double*minVal, double*maxVal=0, Point*minLoc=0, Point*maxLoc=0)

เป็นฟังก์ชันที่ใช้ในการหาค่าสีที่สูงสุด-ต่ำสุดของภาพพร้อมทั้งหาค่าตำแหน่งนั้นให้ (ในระบบนี้ใช้ในการหาค่าในภาพผลลัพธ์)[7]

Src	ตัวแปร Mat ภาพที่รับเข้า
minVal	ค่าต่ำสุดที่หาได้ในภาพ
maxVal	ค่าสูงสุดที่หาได้ในภาพ
minLoc	ตำแหน่งภาพที่มีค่าต่ำสุด
maxLoc	ตำแหน่งภาพที่มีค่าสูงสุด



รูปที่ 2.7 แสดงการบอกตำแหน่งของภาพด้วยฟังก์ชัน minMaxLoc ร่วมกับ matchTemplate

2.4.1.9 floodFill (InputOutputArray image, Point seedPoint, Scalar newVal, Rect* rect=0, Scalar loDiff=Scalar(), Scalar upDiff=Scalar())

เป็นฟังก์ชันที่ใช้สำหรับเติมค่าสีใหม่ลงในสีเก่า ลงในพื้นที่ที่มีสีเหมือนกัน (ในระบบนี้ใช้เพื่อกำจัดตำแหน่งที่พบแล้ว)

Image	ตัวแปรประเภท Mat ภาพรับเข้า
seedPoint	ตำแหน่งที่เริ่มแทนที่สี
newVal	สีใหม่ที่จะนำมาแทน
rect	ในระบบไม่ได้ถูกใช้งาน มีค่าเป็น 0
loDiff	ในระบบไม่ได้ถูกใช้งาน มีค่า Scalar เป็น 1
upDiff	ในระบบไม่ได้ถูกใช้งาน มีค่า Scalar เป็น 1

2.4.1.10 Mat::Mat (Rect(int x,int y,int width,int height))

Region of interest (ROI)

Region of interest (ROI) คือ บริเวณที่เราสนใจซึ่งอาจจะเป็นบริเวณใดภาพในภาพก็ได้ โดยการติกรอบล้อมรอบบริเวณที่สนใจด้วยวงกลม กรอบสี่เหลี่ยม หรือกรอบรูปเหลี่ยมใดๆ เพื่อนำภาพเฉพาะส่วนดังกล่าวมาประมวลผลหรือเปลี่ยนแปลงภาพตามต้องการ

โดยไม่มีผลกระทบต่อส่วนอื่นๆ ซึ่งภายในหนึ่งภาพสามารถกำหนดได้หลายๆบริเวณที่สนใจ การตัดเฉพาะส่วนของภาพ เพื่อนำมาประมวลผลทำให้การประมวลผลภาพเร็วขึ้นเนื่องจาก Resolution ของภาพที่นำมาประมวลผลมีขนาดเล็กลง ซึ่งในการประมวลผลภาพขนาดใหญ่ จะทำให้เห็นความแตกต่างของประสิทธิภาพในการประมวลผลมาก

เป็นฟังก์ชันเพื่อกำหนดขอบเขตในภาพที่เราสนใจ ซึ่งการแก้ไขอะไรบางอย่างในภาพที่กำหนดจะมีผลต่อภาพหลักด้วย

x	กำหนดตำแหน่ง ในแกน x
y	กำหนดตำแหน่ง ในแกน y
width	กำหนดความกว้าง
hight	กำหนดความสูง

2.4.1.11 GaussianBlur (InputArray src, OutputArray dst, Size ksize, double sigmaX, double sigmaY=0)

เป็นฟังก์ชันใช้ในการเบลอภาพ

Src	ตัวแปร Mat รับภาพเข้า
Dst	ตัวแปร Mat ภาพหลังจากประมวลผล
Ksize	เป็นค่ากำหนดความเบลอ
sigmaX	ในระบบไม่ได้ใช้งานมีค่าเป็น 0
sigmaY	ในระบบไม่ได้ใช้งานมีค่าเป็น 0

2.4.1.12 goodFeaturesToTrack (InputArray image, OutputArray corners, int maxCorners, double qualityLevel, double minDistance)

เป็นฟังก์ชันที่ใช้กำหนด feature (มุมของภาพ) ที่เหมาะสม

Image	ตัวแปร Mat ภาพที่รับเข้า
Corners	เป็น vector array ที่เก็บตำแหน่งของมุมในภาพที่เจอ
maxCorners	กำหนดจำนวนที่มากที่สุดที่หามุมเจอได้
qualityLevel	กำหนดระดับความละเอียดของมุมที่เจอ
minDistance	ระยะทาง Euclidean ที่น้อยที่สุดที่เป็นไปได้ในมุมกลับ

2.4.1.13 calcOpticalFlowPyrLK(InputArray prevImg, InputArray nextImg, InputArray prevPts, InputOutputArray nextPts)

เป็นฟังก์ชันที่ใช้ในการค้นหาตำแหน่งวัตถุถัดไปของเฟรมวิดีโอที่เหมาะสม

prevImg	ภาพจากเฟรมวิดีโอก่อนหน้า
nextImg	ภาพเฟรมวิดีโอถัดไป หรือเฟรมปัจจุบัน
prevPts	ตำแหน่ง feature ที่กำหนดไว้ในเฟรมก่อนหน้า
nextPts	ตำแหน่ง feature ของเฟรมถัดไป หรือเฟรมปัจจุบัน

2.4.2 ฟังก์ชันเสริม

เป็นฟังก์ชันที่ช่วยในการใช้งานของระบบให้สามารถใช้งานได้ง่ายขึ้น หรือเพื่อใช้ในการทำงานบางอย่างของระบบที่ฟังก์ชันหลักไม่สามารถทำได้

2.4.2.1 VideoWriter::VideoWriter (string& filename, int fourcc, double fps, Size frameSize, bool isColor=true)

ฟังก์ชันที่ไว้ใช้ในการบันทึกไฟล์วิดีโอหลังจากการทำงานโดยนามสกุลไฟล์ที่บันทึกเป็น .avi

Filename	กำหนด path ไฟล์ และชื่อไฟล์ที่ต้องการจะบันทึก
Fourcc	เป็นตัวกำหนด code ที่จะเลือกรูปแบบการอัดเป็นแบบไหน (ในระบบนี้ผู้ใช้เลือก CV_FOURCC('P', 'I', 'M', '1'))
Fps	อัตราส่วนเฟรมต่อวินาที ที่ต้องการบันทึก ลงไปในวิดีโอ
frameSize	ขนาดเฟรมวิดีโอที่ต้องการบันทึก
isColor	กำหนดการบันทึกเป็นเฟรมสี หรือขาวดำ

2.4.2.2 setMouseCallback (string& winname, MouseCallback onMouse, void*userdata=0)

ฟังก์ชันตัวกำหนดในการใช้งานเมาส์ เพื่อให้ผู้ใช้งานใช้งานในระบบง่ายขึ้น

Winname	ชื่อของ windows (หน้าต่างแสดงผล) ที่ต้องการควบคุม
onMouse	method ที่กำหนดการใช้งานของเมาส์
userdata	กำหนดตัวแปรที่ต้องการเปลี่ยนแปลงค่าตามการใช้งานของเมาส์ หากไม่ใช้งานให้เป็น 0

2.4.2.3 VideoCapture::get (int propld)

เป็นฟังก์ชันเฉพาะของตัวแปลประเภท VideoCapture ใช้ในการรับข้อมูลบางอย่างจากวิดีโอ

propld	code ที่กำหนดสิ่งต้องการรับข้อมูลจากวิดีโอ (ในระบบนี้ผู้ใช้เลือก CV_CAP_PROP_POS_FRAMES ในการรับตำแหน่งเฟรมปัจจุบัน)
--------	--

2.4.2.4 bool VideoCapture::set (int propld, double value)

เป็นฟังก์ชันเฉพาะของตัวแปลประเภท VideoCapture ใช้ในการกำหนดบางอย่างในวิดีโอ

propld	code ที่กำหนดสิ่งต้องการกำหนดในวิดีโอ (ในระบบนี้ผู้ใช้เลือก CV_CAP_PROP_POS_FRAMES คือการกำหนด ตำแหน่งเฟรมใหม่ที่ต้องการอ่าน)
value	ค่าที่เราต้องการกำหนดบางอย่างในวิดีโอ

2.4.2.5 waitKey (int delay=0)

เป็นฟังก์ชันหน่วงเวลาเพื่อรับข้อมูลจากคีย์บอร์ด หรืออาจจะใช้เพื่อหน่วงเวลาอย่างเดียวก็ได้

Delay	กำหนดเวลาที่หน่วงหน่วยเป็น milliseconds
-------	---

2.4.2.6 createTrackbar (string& trackbarname, string& winname, int*value, int count)

เป็นฟังก์ชันสร้างแท็กรีกบาร์ขึ้นมาเพื่อใช้กำหนดค่าบางอย่างในการทำงาน

Trackbarname	กำหนดชื่อแท็กรีกบาร์
Winname	ชื่อของ windows (หน้าต่างแสดงผล) ที่จะนำมาแสดงแท็กรีกบาร์
Value	ตัวแปรที่ต้องการเปลี่ยนแปลงตามค่าที่กำหนดในแท็กรีกบาร์

Count ขอบเขตค่าที่กำหนดได้ในแท็กรับ

2.4.2.7 rectangle (Mat& img, Point pt1, Point pt2, const Scalar& color, int thickness=1)

ฟังก์ชันที่ใช้ในการสร้างกรอบสีเหลี่ยม

img ตัวแปร Mat รูปที่ต้องการสร้าง สีเหลี่ยมไว้ข้างใน
 pt1 ตำแหน่งสีเหลี่ยม มุมบนซ้าย
 pt2 ตำแหน่งสีเหลี่ยม มุมล่างขวา
 color ค่าสีของกรอบสีเหลี่ยม
 thickness ค่าความหนาของกรอบสีเหลี่ยม

2.5 ระบบงานที่เกี่ยวข้อง

ศึกษาการทดสอบโปรแกรมกับภาพที่นำเข้าโค้ดตัวอย่างที่ใช้ทดสอบในการศึกษาครั้งนี้ คือ

```
Mat img; Mat templ; Mat result;
char* image_window = "Source Image";
char* result_window = "Result window";
int match_method;
int max_Trackbar = 5;
void MatchingMethod( int, void* );
/* function main */
int main( int argc, char** argv )
{
    img = imread( argv[1], 1 );
    templ = imread( argv[2], 1 );
    namedWindow( image_window, CV_WINDOW_AUTOSIZE );
    namedWindow( result_window, CV_WINDOW_AUTOSIZE );
    char* trackbar_label = "Method: \n 0: SQDIFF \n 1: SQDIFF NORMED \n
2: TM_CCORR \n 3: TM_CCORR NORMED \n 4: TM_COEFF \n 5: TM
COEFF NORMED";
    createTrackbar( trackbar_label, image_window, &match_method,
max_Trackbar, MatchingMethod );
    MatchingMethod( 0, 0 );
    waitKey(0);
    return 0;
}
void MatchingMethod ( int, void* )
{
    Mat img_display;
    img.copyTo( img_display );
```

```

int result_cols = img.cols - templ.cols + 1;
int result_rows = img.rows - templ.rows + 1;
result.create( result_rows, result_cols, CV_32FC1 );
matchTemplate( img, templ, result, match_method );
normalize( result, result, 0, 1, NORM_MINMAX, -1, Mat() );
double minVal; double maxVal; Point minLoc; Point maxLoc;
Point matchLoc;
minMaxLoc( result, &minVal, &maxVal, &minLoc, &maxLoc, Mat() );
if( match_method == CV_TM_SQDIFF || match_method ==
CV_TM_SQDIFF_NORMED )
{ matchLoc = minLoc; }
else
{ matchLoc = maxLoc; }
rectangle( img_display, matchLoc, Point( matchLoc.x + templ.cols ,
matchLoc.y + templ.rows ), Scalar::all(0), 2, 8, 0 );
rectangle( result, matchLoc, Point( matchLoc.x + templ.cols ,
matchLoc.y + templ.rows ), Scalar::all(0), 2, 8, 0 );
imshow( image_window, img_display );
imshow( result_window, result );
return;
}

```

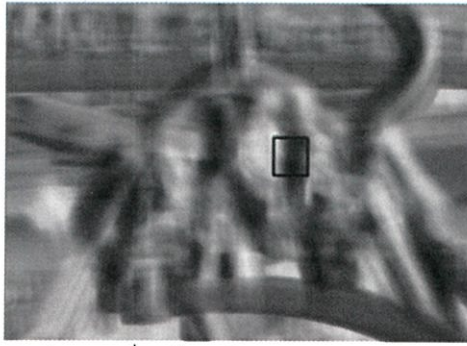
ได้ผลลัพธ์จากการทดสอบโปรแกรมดังรูปที่ 2.8, 2.9 และ 2.10[7]



รูปที่ 2.8 แสดง Source image (ในที่นี้หมายถึงภาพในวิดีโอ)



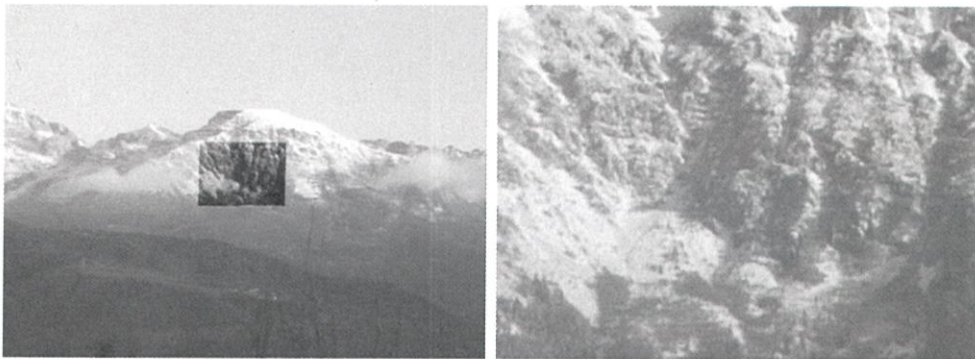
รูปที่ 2.9 แสดง Template image (ในที่นี้หมายถึงภาพตัวอย่างที่ต้องการเซ็นเซอร์)



รูปที่ 2.10 แสดงตำแหน่งภาพที่เหมือนกับ Template image ใน Source image

2.6 งานวิจัยที่เกี่ยวข้อง

[8]งานวิจัย Matching Images with Different Resolutions พ.ศ. 2543 โดย C. Schmid, R. Horaud เสนอวิธีการแก้ไขปัญหาการจับคู่ของสองภาพที่มีความละเอียดแตกต่างกัน ระหว่างภาพความละเอียดสูง และความละเอียดต่ำ บนสมมติฐานที่มีการเปลี่ยนแปลงมิติที่ต่างกัน วิธีการคือ ใช้ภาพถ่ายจากกล้องภาพแรกถ่ายปกติ ภาพที่สองซูมขยายภาพขณะถ่าย การแก้ปัญหามีวิธีการที่จะจัดการกับการเปลี่ยนแปลงได้ โดยใช้ขนาดพื้นที่ตรวจสอบจุดที่น่าสนใจ และส่งจุดหนึ่งไปยังอีกจุดหนึ่ง แสดงภาพถ่ายทั้งสองดังรูปที่ 2.11



รูปที่ 2.11 การเปรียบเทียบระหว่างภาพถ่ายทั้งสองภาพ และจุดที่ระบบตรวจพบ

บทที่ 3

การวิเคราะห์และการออกแบบระบบ

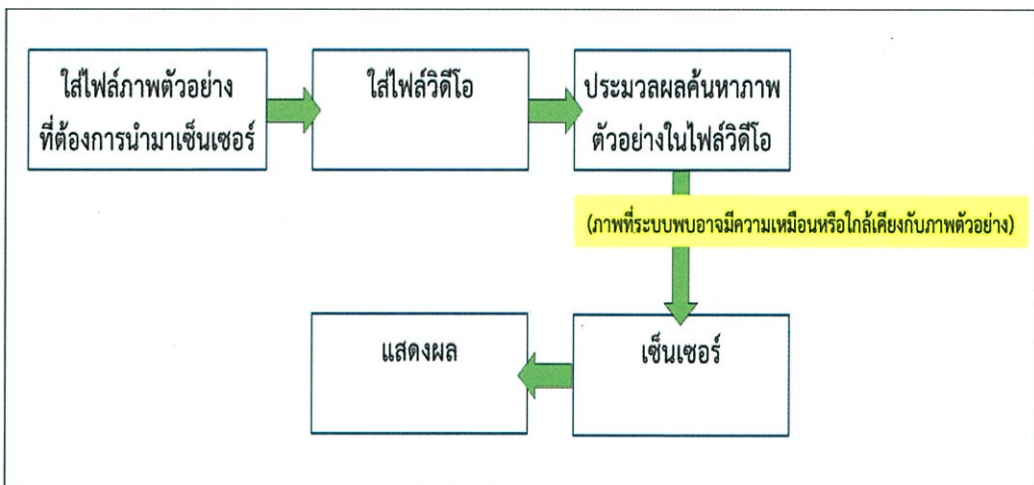
ในส่วนของบทที่ 3 นี้ กล่าวถึงรายละเอียดเชิงลึกของการดำเนินงานโดยจะประกอบด้วย ขั้นตอนวิธีการดำเนินงานรวมถึงการออกแบบโคแอมแกรมต่างๆ และแสดงถึงการออกแบบส่วนติดต่อกับการทำงานกับผู้ใช้ (User interface) โปรแกรมโดยคร่าวๆ

3.1 ภาพรวมของระบบ

ระบบเซ็นเซอร์วิดีโออัตโนมัติเป็นโปรแกรมที่พัฒนาขึ้นมาเพื่อเซ็นเซอร์ภาพที่ต้องการในวิดีโอให้แสดงผลออกมาอัตโนมัติ โดยทำการใส่ภาพตัวอย่างที่ต้องการเซ็นเซอร์ เพื่อไปค้นหาภาพที่เหมือนหรือใกล้เคียงกันในวิดีโอ เมื่อโปรแกรมประมวลผลค้นหาภาพนั้นในวิดีโอแล้วทำการเซ็นเซอร์ โดยจะการเซ็นเซอร์ภาพที่เจอทั้งหมด โดยมีการออกแบบขั้นตอนการทำงานของระบบเซ็นเซอร์วิดีโออัตโนมัติ ดังที่แสดงในรูปที่ 3.1

3.1.1 ความสามารถของระบบ

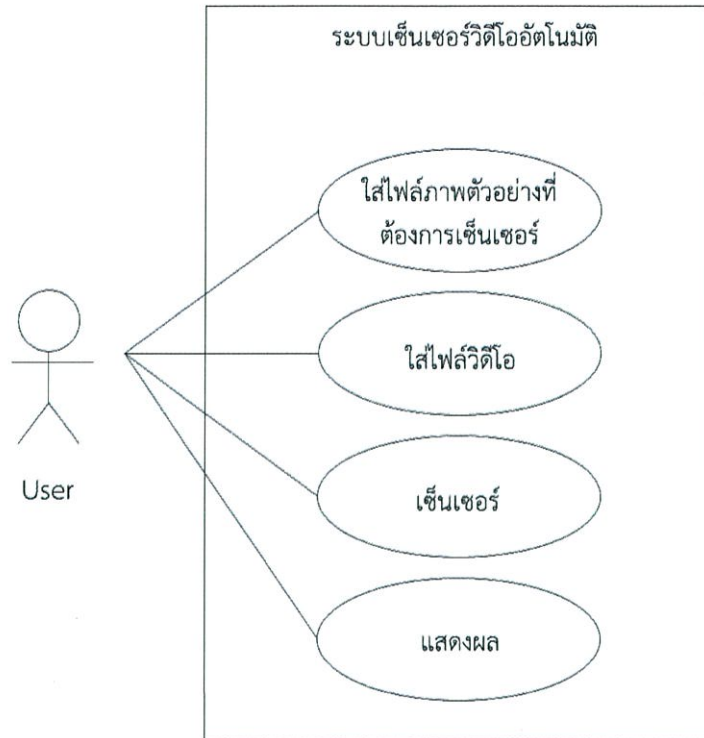
- สามารถประมวลผลค้นหาภาพตัวอย่าง ที่จะนำมาเซ็นเซอร์ในวิดีโอได้ โดยภาพที่ได้ อาจมีความเหมือนหรือใกล้เคียงกับภาพตัวอย่าง
- เซ็นเซอร์ภาพตัวอย่าง หรือภาพที่ใกล้เคียงที่พบในวิดีโอ
- แสดงผลลัพธ์



รูปที่ 3.1 แสดงการออกแบบขั้นตอนการทำงานของระบบเซ็นเซอร์วิดีโออัตโนมัติ

3.2 การวิเคราะห์ระบบ

3.2.1 แผนภาพยูสเคส (Use case Diagram)



รูปที่ 3.2 แสดงแผนภาพยูสเคส (Use case Diagram) ของระบบ

สามารถอธิบาย Process ของแผนภาพยูสเคสได้ ดังนี้

Process

ใส่ไฟล์ภาพตัวอย่างที่ต้องการเซ็นเซอร์
ใส่ไฟล์วิดีโอ
เซ็นเซอร์
แสดงผล

Process Description

ทำหน้าที่เลือกไฟล์ภาพที่ต้องการ
ทำหน้าที่นำไฟล์วิดีโอมาเก็บไว้ในระบบ
ทำหน้าที่เรียกดูภาพที่ระบบประมวลผลค้นหาพบ และเซ็นเซอร์
ทำหน้าที่แสดงผลวิดีโอที่เซ็นเซอร์เรียบร้อยแล้ว

3.3 อธิบายการทำงานของโปรแกรม

ในด้านการพัฒนาโปรแกรมนั้นสามารถอธิบายการทำงานของระบบได้ ดังนี้

3.3.1 ฟังก์ชันหลักของการทำงาน

3.3.1.1 ฟังก์ชันสำหรับการอ่านรูปจากเครื่อง

```
Mat imgo = imread("&Path file", CV_LOAD_IMAGE_UNCHANGED);
```

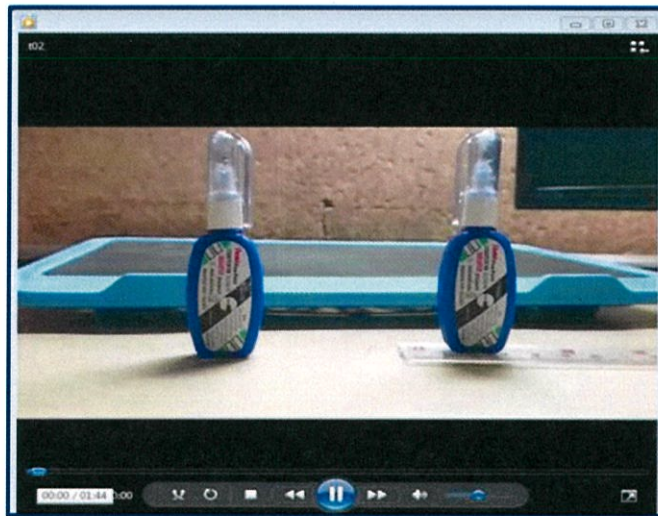


รูปที่ 3.3 ตัวอย่างรูปที่จะนำมาเก็บในตัวแปร Imgo

ในการทำงานของระบบได้ใช้ฟังก์ชันนี้ในการรับรูปตัวอย่าง และรูปที่ต้องการนำมาแทนเซ็นเซอร์ภาพ เพื่อใช้ในการทำงานขั้นตอนต่อไป โดยนามสกุลไฟล์ภาพต้องเป็น .jpg, .png และ .bmp เท่านั้น

3.3.1.2 ฟังก์ชันสำหรับการอ่านวิดีโอจากเครื่อง

```
VideoCapture cap("&Path file" + j + ".mp4");
```



รูปที่ 3.4 ตัวอย่างวิดีโอที่จะนำมาเก็บในตัวแปร

ใช้ในการรับไฟล์วิดีโอ ที่ต้องการนำมาใช้งานกับระบบ โดยนามสกุลไฟล์ต้องเป็น .mp4 หรือ .avi เท่านั้น

3.3.1.3 ฟังก์ชันคำนวณเพื่อให้ได้ภาพผลลัพธ์

```
bool bSuccess = cap.read(frame);
cvtColor(frame, gframe, CV_BGR2GRAY);
cvtColor(imgo, gimgo, CV_BGR2GRAY);
matchTemplate(gframe, gimgo, result, CV_TM_CCOEFF_NORMED);
```

ใช้ในการหาภาพผลลัพธ์ เพื่อนำภาพผลลัพธ์ไปค้นหาตำแหน่งที่ต้องการเซ็นเซอร์ในขั้นตอนต่อไป



รูปที่ 3.5 ภาพผลลัพธ์จากฟังก์ชัน matchTemplate

ภาพผลลัพธ์จากรูปที่ 3.5 เป็นการนำภาพตัวอย่างมาค้นหาในวิดีโอโดยการแปลงภาพให้เป็นภาพขาว-ดำก่อน จากรูปแสดงให้เห็นว่าจุดที่มีสีขาวสว่างมากที่สุดจะมีความเหมือนกับภาพตัวอย่างมากที่สุด

3.3.1.4 ฟังก์ชันปรับค่าสีของภาพ

```
threshold(result, result, 0.2, 0., CV_THRESH_TOZERO);
```

โดยสีในตำแหน่งภาพที่มีค่าต่ำกว่า 0.2 ทั้งหมดจะถูกปรับเป็นสีดำ และค่าที่มากกว่า 0.2 ทั้งหมดจะคงเดิม การทำงานนี้เพื่อลดการคำนวณในฟังก์ชันถัดไป

3.3.1.5 ฟังก์ชันหาค่าสูงสุด ต่ำสุดในภาพ

```
Point minloc, maxloc;
```

```
minMaxLoc(result, &minval, &maxval, &minloc, &maxloc);
```

เพื่อหาค่าความสว่างที่สุดในภาพผลลัพธ์ โดยตำแหน่งที่สว่างที่สุดแสดงว่าในตำแหน่งนั้นมีความเหมือนมากที่สุด

3.3.1.6 ฟังก์ชันเซ็นเซอร์ภาพในวิดีโอ

```
roi = Rect(maxloc.x, maxloc.y, imgo.cols, imgo.rows);
```

```
cen = frame(roi);
```

```
for (int i = 1; i < 40; i = i + 2)
```

```
    GaussianBlur(cen, cen, Size(i, i), 0, 0);
```

จากตำแหน่งที่พบของฟังก์ชันที่ 3.3.5 จะทำการเซ็นเซอร์ภาพตามขนาดภาพตัวอย่าง

3.3.1.7 ฟังก์ชันการเติมสี

```
if (maxval >= &threshold)
```

```
{
```

```
    floodFill(result, maxloc, Scalar(0), 0, Scalar(.1), Scalar(1));
```

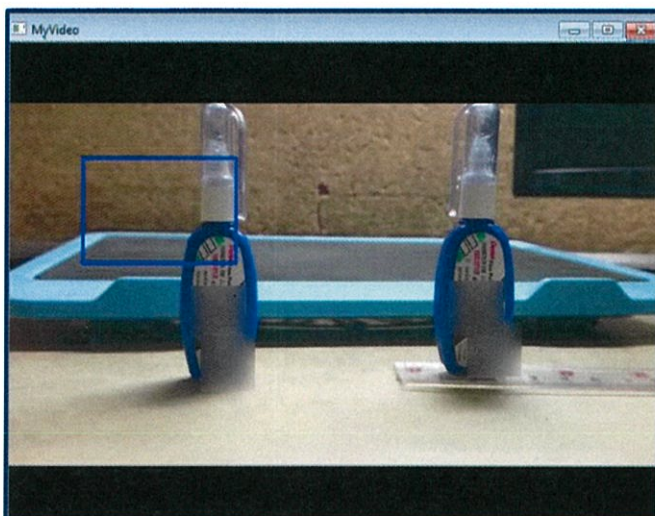
```
}
```

การทำงานของฟังก์ชัน floodFill จะนำตำแหน่งที่สว่างที่สุดที่ระบบค้นหาได้มาเปลี่ยนค่าสีเป็นสีดำ จากนั้นระบบจะไปวนไปทำงานในฟังก์ชันที่ 3.3.4 ใหม่ เพื่อค้นหาตำแหน่งที่สว่างที่สุดใหม่ แล้วเปลี่ยนค่า ทำซ้ำไปจนกว่าจะมีค่าต่ำกว่า ค่า threshold ที่กำหนดจึงหยุดทำ

3.3.2 ฟังก์ชันเสริมในการทำงาน

3.3.2.1 ฟังก์ชันกำหนดตำแหน่งเบลอที่ต้องการ

```
setMouseCallback("MyVideo", CallBackFunc, &rect);
rectangle(img_display, rect, Scalar(255, 0, 0), 2);
```



รูปที่ 3.6 แสดงการลากสี่เหลี่ยมเพิ่มส่วนที่ต้องการเซ็นเซอร์

ทำได้โดยคลิกซ้ายแล้วลากสี่เหลี่ยมขึ้นมาในตำแหน่งที่ต้องการเบลอ ดังแสดงในรูปที่ 3.6 เพื่อให้สามารถเพิ่มจุดที่จะเซ็นเซอร์ได้ความต้องการ

3.3.2.2 ฟังก์ชันยกเลิกการเบลอผิดจุดของระบบ

```
setMouseCallback("MyVideo", CallBackFunc, &rect);
if ((ganx[1] == NULL && gany[1] == NULL) || (ganx[1] == 0 && gany[1]
== 0))
{
for (int i = 1; i < 40; i = i + 2)
GaussianBlur(cen, cen, Size(i, i), 0, 0);
}
```

ทำงานโดยการคลิกขวาในกรอบที่ระบบเบลอผิดจุด ส่วนที่เบลอผิดนั้นก็จะกลับมาเหมือนเดิม

3.3.2.3 ฟังก์ชันควบคุมการทำงานของระบบ

```
createTrackbar("threshold", "TrackBar Controll", &controll, 20);
สามารถเพิ่มระดับความเหมือนของภาพตัวอย่างในวิดีโอ
createTrackbar("Next Frame", "TrackBar Controll", &Nframe, 20);
สามารถเลื่อนดูไปยังเฟรมถัดไปในวิดีโอ
createTrackbar("Previous Frame", "TrackBar Controll", &Pframe, 20);
สามารถย้อนกลับไปดูเฟรมก่อนหน้าในวิดีโอ
```

3.3.2.4 ฟังก์ชันที่ใช้คีย์บอร์ดควบคุมการทำงานของระบบ

```
if (e == 'p')
```

```

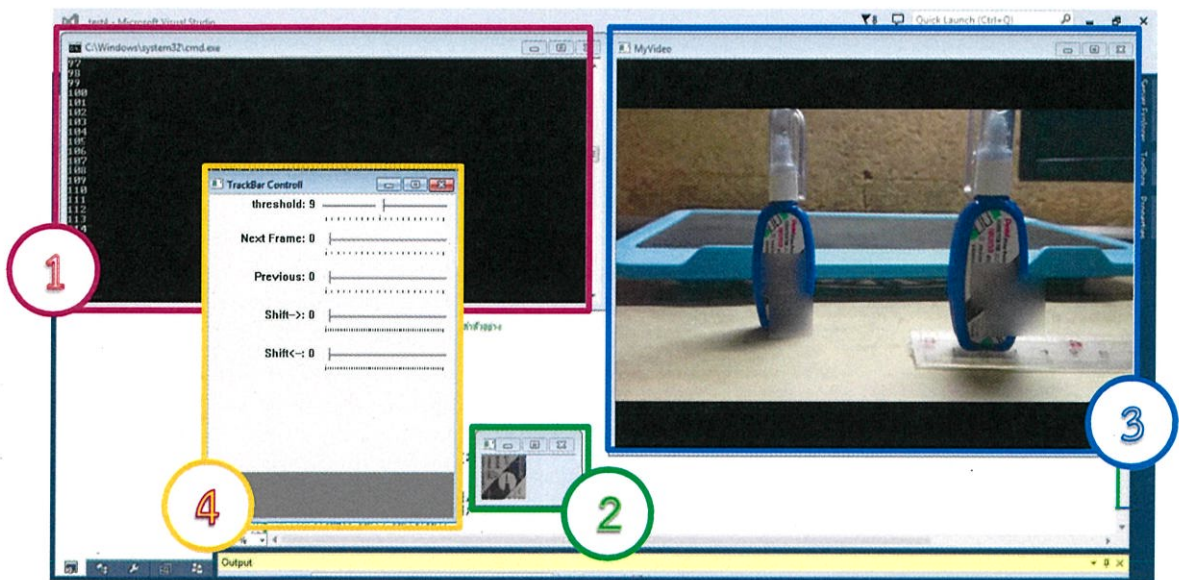
{ pause = !pause; }
else if (e == 's')
{ Save = !Save; }
  else if (e == 'n')
{ Nmatch = !Nmatch; }
  else if (c == 'f')
{ manl = false; fsm = false; cen2.release(); }
  else if (e == 'z')
{ CoP = !CoP; }

```

อธิบายการใช้คีย์บอร์ดควบคุมการทำงานได้ ดังนี้

- 1) กด p เพื่อหยุดการเล่นของวิดีโอ และกด p ซ้ำอีกครั้งเพื่อให้วิดีโอเล่นต่อ
- 2) กด s เพื่อบันทึกวิดีโอ หรือกด s ซ้ำอีกครั้งเพื่อหยุดบันทึกวิดีโอ
- 3) กด n เพื่อเปลี่ยนโหมดการเล่นของวิดีโอ จากโหมดเซ็นเซอร์เป็นโหมดธรรมดา หรือกด n ซ้ำอีกครั้ง เพื่อเปลี่ยนจากโหมดธรรมดาเป็นโหมดเซ็นเซอร์
- 4) กด f เพื่อยกเลิกตำแหน่งที่เราทำการเซ็นเซอร์ขึ้นมาเอง
- 5) กด z เพื่อนำภาพอื่นมาแสดงแทนการเซ็นเซอร์

3.4 ส่วนแสดงผลของระบบ



รูปที่ 3.7 ส่วนแสดงผลของระบบ

อธิบายส่วนต่างๆ ของรูปที่ 3.7 ได้ดังนี้

1) ส่วนรับและแสดงผลในการทำงาน

รับค่า Input

- ใส่ชื่อไฟล์ภาพ
- ใส่ชื่อไฟล์วิดีโอ
- ใส่ชื่อไฟล์รูปภาพที่จะแสดงแทนส่วนเซ็นเซอร์

แสดงผล Output

- แสดงหมายเลขเฟรมที่ได้บันทึกไว้แล้ว
- แสดงตำแหน่งที่ผู้ใช้งานลบส่วนเซ็นเซอร์ที่ไม่ต้องการออก

2) แสดงภาพตัวอย่าง

ภาพตัวอย่างที่ผู้ใช้งานต้องการเซ็นเซอร์ในวิดีโอ (ใช้ภาพตัวอย่างได้ 1 ภาพ ต่อการทำงาน 1 ครั้ง)

3) แสดงวิดีโอ และควบคุมการทำงานของวิดีโอ

แสดงส่วนของวิดีโอที่ใช้ในการทำงานรวมถึงควบคุมการใช้งานบางอย่าง เช่น ลบส่วนเซ็นเซอร์ที่ไม่ต้องการออกด้วยการคลิกขวา และวาดส่วนที่ต้องการเซ็นเซอร์เพิ่ม

4) ส่วนควบคุม

- threshold : เพิ่ม ลดระดับความแม่นยำในการเซ็นเซอร์ของภาพ
- Next Frame : เลื่อนดูวิดีโอไปข้างหน้า เป็นวินาที
- Previous : เลื่อนดูวิดีโอย้อนหลัง เป็นวินาที
- Shift→ : เลื่อนดูเฟรมวิดีโอไปข้างหน้า
- Shift← : เลื่อนดูเฟรมวิดีโอย้อนหลัง

* ตั้งแต่ฟังก์ชัน Next Frame ลงมาจะสามารถใช้งานได้นั้นก็ต่อเมื่อทำการกด pause “p”

บทที่ 4

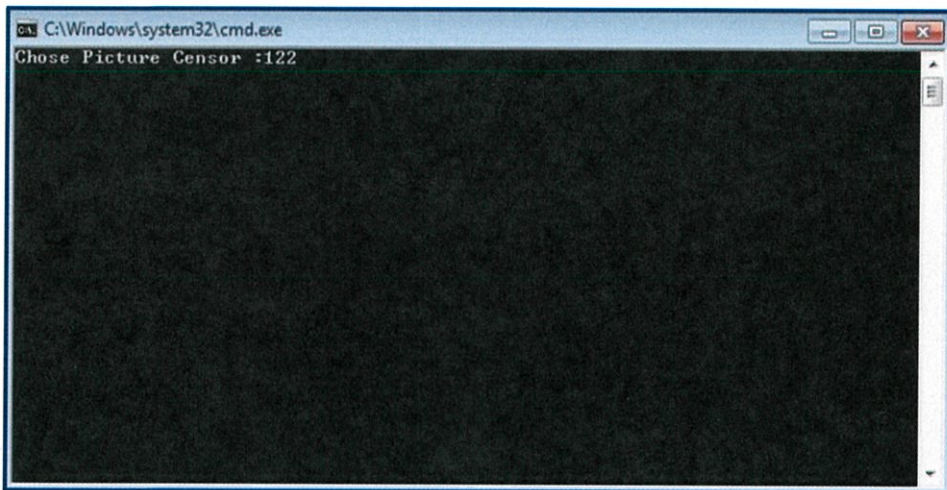
ผลการดำเนินงาน

หลังจากศึกษา และพัฒนาระบบพร้อมกับสร้างกรณีทดสอบที่ครอบคลุมการทำงานโดยรวมของระบบเป็นอันเสร็จสิ้น ซึ่งผลลัพธ์ที่ได้จากการทดสอบพบว่าสามารถช่วยให้ตรวจพบข้อผิดพลาดของระบบที่เกิดขึ้นในระหว่างการพัฒนา ช่วยให้ทีมพัฒนาสามารถแก้ไขปัญหาและข้อผิดพลาดที่เกิดขึ้นได้รวดเร็วและทันท่วงที

4.1 ทดสอบการทำงานของระบบ

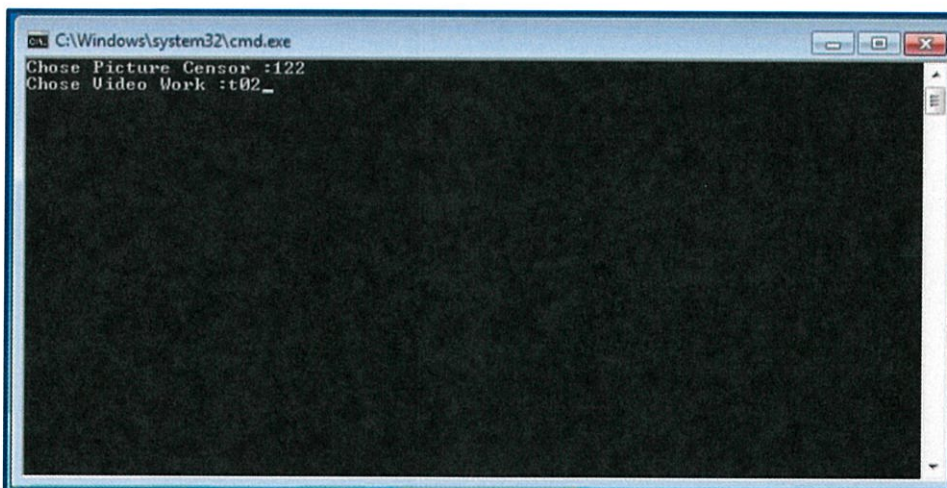
ทำการเปิดไฟล์ CATS.exe ขึ้นมา

- เลือกไฟล์ภาพตัวอย่างที่ต้องการใช้งาน แสดงดังรูปที่ 4.1



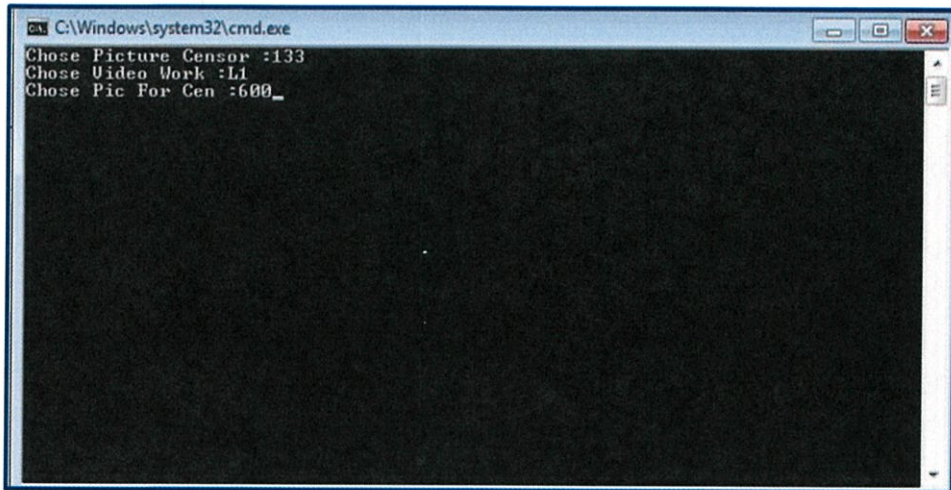
รูปที่ 4.1 การใส่ชื่อไฟล์รูปภาพ

- เลือกไฟล์วิดีโอที่จะเซ็นเซอร์ แสดงดังรูปที่ 4.2



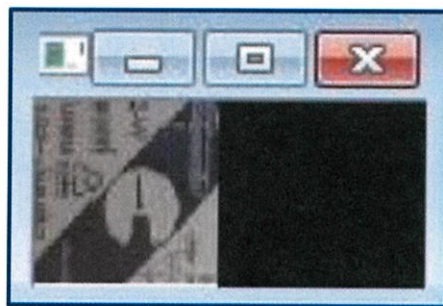
รูปที่ 4.2 การใส่ชื่อไฟล์วิดีโอ

- เลือกไฟล์ภาพอื่นๆ ที่จะนำมาใช้แทนการเซ็นเซอร์ แสดงดังรูปที่ 4.3



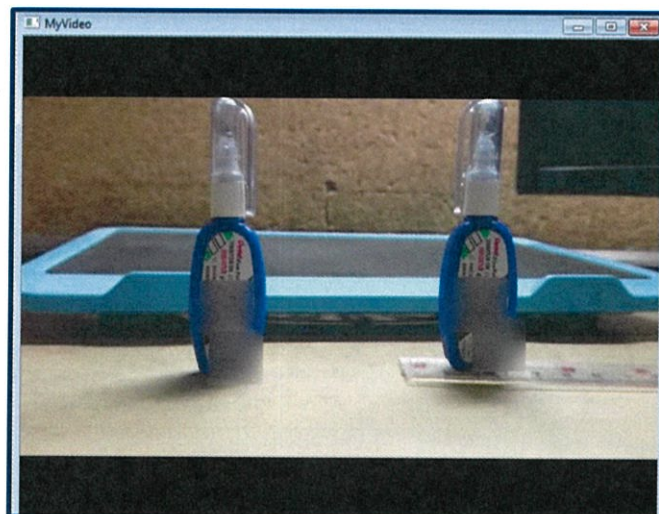
รูปที่ 4.3 การใส่ชื่อไฟล์ภาพที่จะใช้ในการทำงาน

ระบบแสดงภาพตัวอย่างที่ใช้ในการทำงาน แสดงดังรูปที่ 4.4



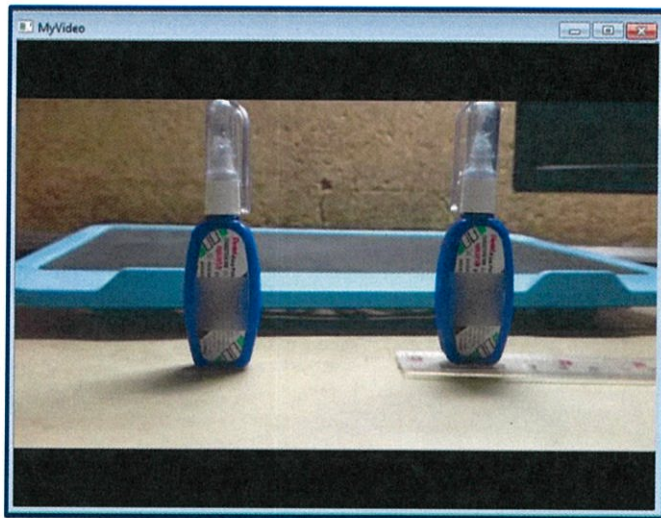
รูปที่ 4.4 แสดงภาพตัวอย่าง

จากนั้นเปิดวิดีโอที่ค้นหาภาพตัวอย่างพบและเซ็นเซอร์เรียบร้อยแล้วขึ้นมา แสดงดังรูปที่ 4.5



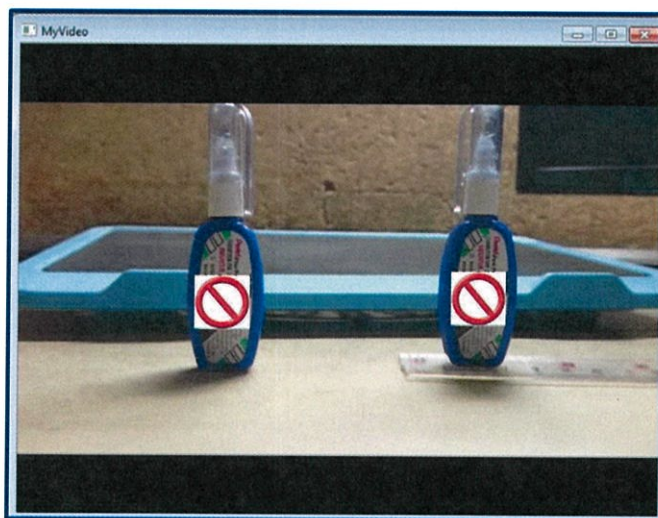
รูปที่ 4.5 แสดงวิดีโอที่เซ็นเซอร์เรียบร้อยแล้ว

สามารถเลือกตำแหน่งที่ไม่ต้องการให้เซ็นเซอร์ออกได้โดย คลิกขวาตำแหน่งที่เซ็นเซอร์ แสดงดังรูปที่ 4.6



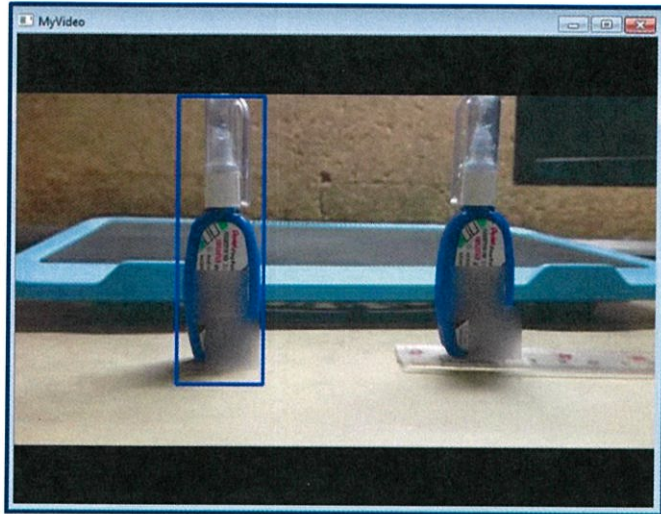
รูปที่ 4.6 แสดงภาพในวิดีโอที่เอาส่วนเซ็นเซอร์ที่ไม่ต้องการออก

และการเซ็นเซอร์สามารถนำภาพอื่นมาแทนที่ได้โดยการกด z ในการรันระบบได้ผลลัพธ์ แสดงดังรูปที่ 4.7

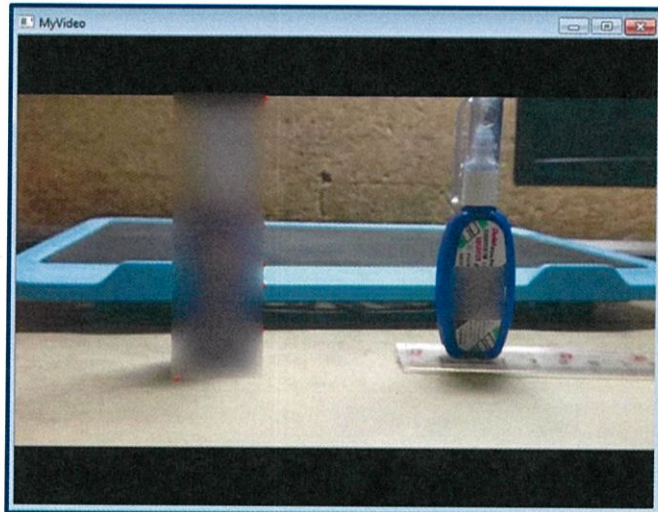


รูปที่ 4.7 แสดงการนำภาพอื่นมาแทนที่การเซ็นเซอร์

นอกจากนี้ระบบยังสามารถเลือกส่วนที่จะเซ็นเซอร์เองได้ด้วยการคลิกซ้ายลากเส้นติกรอบสี่เหลี่ยม ส่วนที่ต้องการเมื่อได้กรอบสี่เหลี่ยมระบบจะเซ็นเซอร์ให้ทันที แสดงดังรูปที่ 4.8 และรูปที่ 4.9



รูปที่ 4.8 การลากเส้นตีกรอบสีเหลี่ยมเพิ่ม

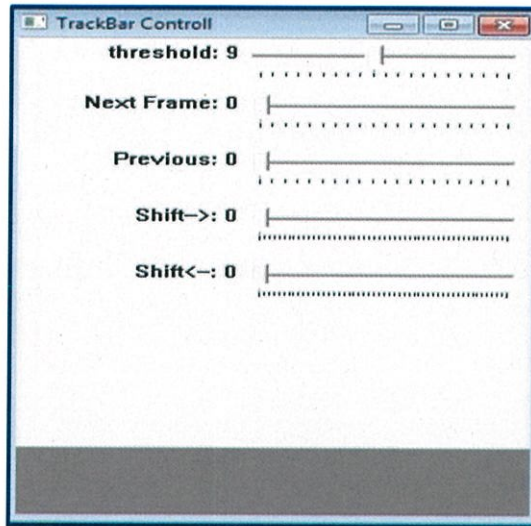


รูปที่ 4.9 เซ็นเซอร์ตำแหน่งที่ตีกรอบสีเหลี่ยมเพิ่ม

ส่วนควบคุมการทำงานของระบบมีดังนี้

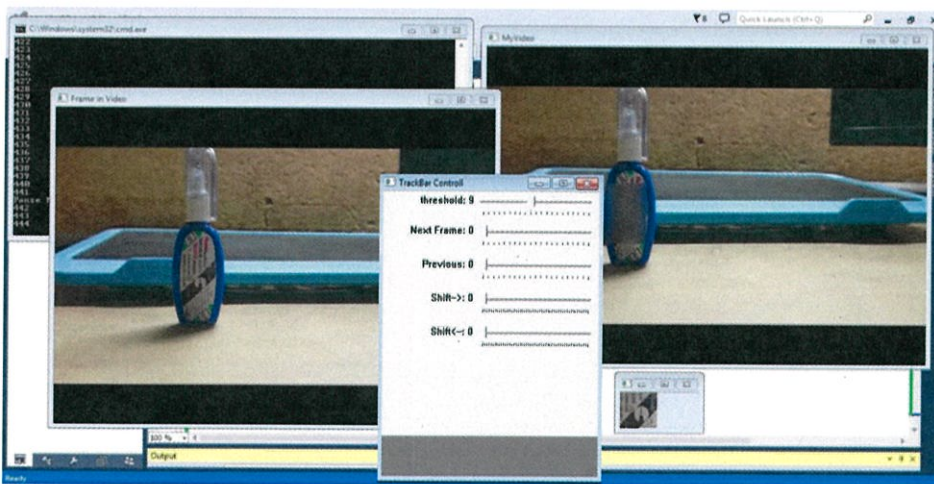
- threshold : เพิ่ม ลดระดับความแม่นยำในการเซ็นเซอร์ของภาพ
- Next Frame : เลื่อนดูวิดีโอไปข้างหน้า เป็นวินาที
- Previous : เลื่อนดูวิดีโอย้อนหลัง เป็นวินาที
- Shift→ : เลื่อนดูเฟรมวิดีโอไปข้างหน้า
- Shift← : เลื่อนดูเฟรมวิดีโอย้อนหลัง

แสดงส่วนควบคุมได้ดังรูปที่ 4.10



รูปที่ 4.10 ส่วนควบคุมการทำงานของระบบ

การใช้งานในส่วนควบคุมนั้นจะสามารถทำงานได้ก็ต่อเมื่อหยุดเล่นวิดีโอ แล้วจะปรากฏหน้าต่างต่าง Frame in Video เพื่อแสดงผลของการควบคุมการทำงาน ดังรูปที่ 4.11



รูปที่ 4.11 แสดงหน้าต่างการทำงานของระบบ

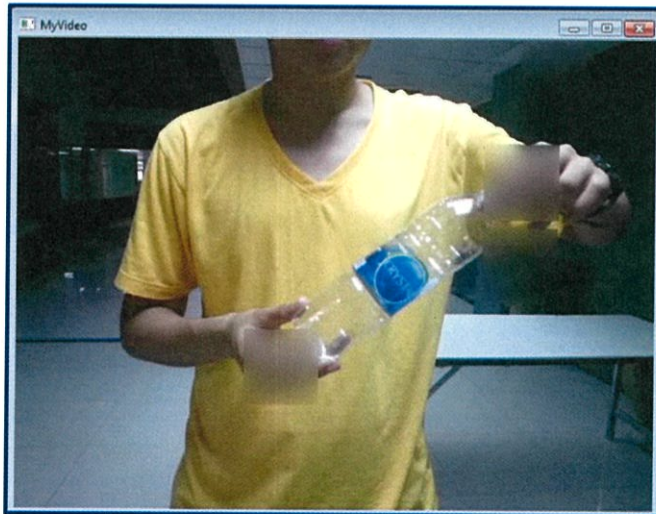
4.2 กรณีทดสอบอื่นๆ

4.2.1 ภาพแรกในวิดีโอเอียง

ในการค้นหาตำแหน่งภาพที่เหมือนกับภาพตัวอย่างในวิดีโอ ถ้าภาพแรกที่ระบบค้นหาพบเป็นภาพเอียงจะไม่สามารถเซ็นเซอร์ให้ได้ แต่ถ้าภาพต่อไปสามารถค้นหาได้ระบบจะเซ็นเซอร์ให้ในภาพนั้น

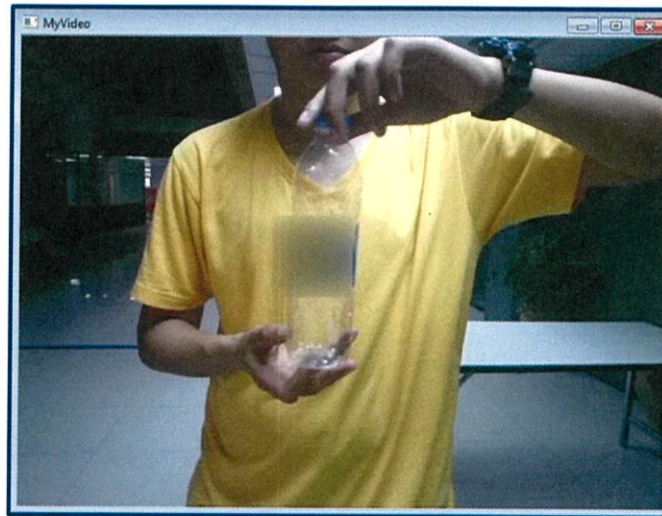


รูปที่ 4.12 ภาพตัวอย่าง



รูปที่ 4.13 กรณีภาพเอียงที่ไม่สามารถเซ็นเซอร์ได้

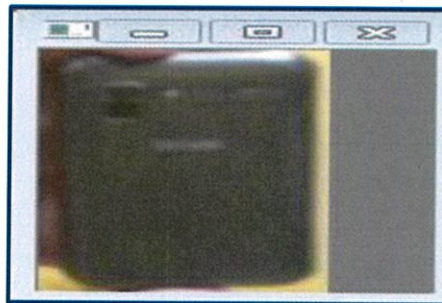
เมื่อวิดีโอเล่นไปจนค้นหาภาพได้ในเฟรมในเฟรมหนึ่ง ระบบจะทำการเซ็นเซอร์ให้ แสดงดังรูปที่ 4.14



รูปที่ 4.14 การเซ็นเซอร์ของระบบ

4.2.2 การซูมเข้า-ออกของภาพในวิดีโอ

ในวิดีโอหากภาพที่ค้นหาได้จากภาพตัวอย่างมีการขยายใหญ่ และลดขนาดลงระบบสามารถเซ็นเซอร์ให้ได้

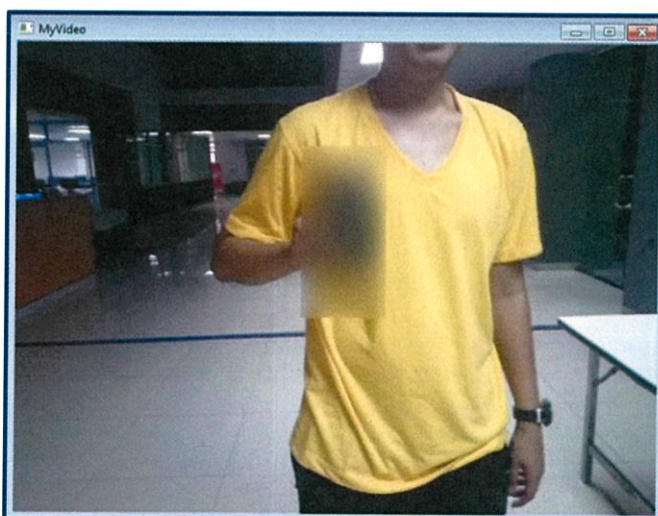


รูปที่ 4.15 ภาพตัวอย่าง

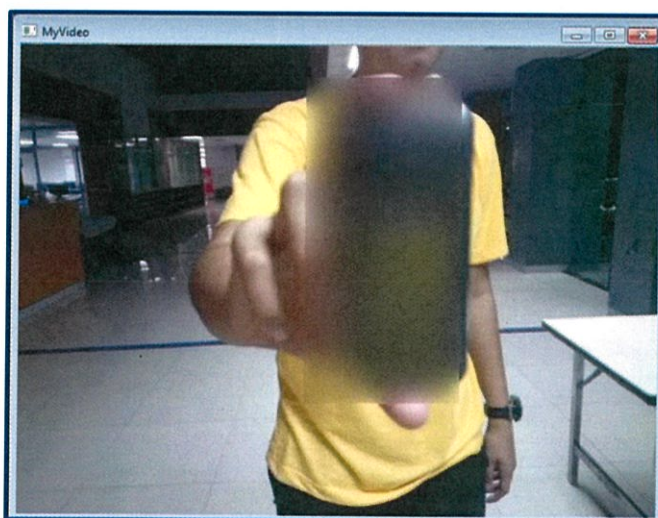
เมื่อระบบค้นหาตำแหน่งภาพในวิดีโอที่เหมือนกับภาพตัวอย่างได้แล้ว ในวิดีโอมีการขยาย หรือลดขนาดภาพลง แสดงผล ดังรูปที่ 4.16, รูปที่ 4.17 และรูปที่ 4.18



รูปที่ 4.16 ภาพเซ็นเซอร์ขนาดปกติ



รูปที่ 4.17 ภาพในวิดีโอมีการขยายใหญ่ขึ้น



รูปที่ 4.18 ภาพในวิดีโอลดขนาดลง

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

- 5.1.1) โปรแกรมระบบเซ็นเซอร์วิดีโออัตโนมัติ สามารถค้นหาตำแหน่งภาพที่ต้องการในวิดีโอได้หลายตำแหน่งที่พบ โดยการค้นหาภาพของระบบ อาจได้ภาพที่มีความเหมือนหรือใกล้เคียงกัน และในบางกรณีระบบไม่สามารถค้นหาตำแหน่งภาพให้ได้
- 5.1.2) การทำเซ็นเซอร์ของระบบ ถ้าระบบไม่สามารถทำเซ็นเซอร์ภาพในวิดีโอให้ได้ตรงตามที่ต้องการทั้งหมด ผู้ใช้สามารถแก้ไขหรือเพิ่มเติมได้เอง
- 5.1.3) เมื่อพบตำแหน่งภาพในวิดีโอ ระบบสามารถเซ็นเซอร์ให้อัตโนมัติ และแสดงผลออกมาเป็นวิดีโอที่เซ็นเซอร์ภาพเรียบร้อยแล้ว
- 5.1.4) การเซ็นเซอร์ภาพในวิดีโอสามารถนำภาพอื่นๆ มาแทนที่การทำเซ็นเซอร์ได้
- 5.1.5) การทำงานของระบบหากใช้ไฟล์วิดีโอที่มีขนาดใหญ่มาก ระบบจะสามารถประมวลผลในการทำงานได้ช้าลง ซึ่งจะส่งผลต่อความสะดวกในการทำงาน เพราะไม่สามารถลดระยะเวลาให้มีความรวดเร็วขึ้นได้ในกรณีที่มีปริมาณงานจำนวนมาก

5.2 ข้อจำกัดของปัญหาพิเศษ

- 5.2.1) ในการเลือกภาพตัวอย่างที่จะนำมาทำเซ็นเซอร์ สามารถจะเลือกภาพตัวอย่างได้เพียงภาพเดียวต่อการประมวลผล 1 ครั้งในระบบ
- 5.2.2) ถ้าภาพในวิดีโอที่พบครั้งแรกมีการ rotate หรือภาพมีขนาดที่แตกต่างกับภาพตัวอย่าง ระบบจะไม่สามารถค้นหาตำแหน่งภาพนั้นได้
- 5.2.3) หากภาพในวิดีโอมีความโค้งงอ ระบบจะไม่สามารถค้นหาตำแหน่งของภาพนั้นได้
- 5.2.4) ความเร็วในการทำงานของระบบขึ้นอยู่กับขนาดไฟล์ภาพ ไฟล์วิดีโอ และคุณสมบัติของเครื่องคอมพิวเตอร์ที่ผู้ใช้นำมาใช้งาน เพราะจะส่งผลให้ระบบประมวลผลได้ช้าลง

5.3 ข้อเสนอแนะ

- 5.3.1) หากมีการพัฒนาโปรแกรม ให้มีประสิทธิภาพเพิ่มมากขึ้นแล้ว ควรพัฒนาโปรแกรมให้สามารถใช้งานได้บนระบบปฏิบัติการแอนดรอยด์ และระบบปฏิบัติการไอโอเอส เพื่อตอบสนองต่อการใช้งานของผู้ใช้งานได้มากขึ้น

เอกสารอ้างอิง

- [1] 30 ตุลาคม 2558. Motion Detection By Background Substraction. [Online]. Available : <http://fivedots.coe.psu.ac.th/~kom/wp-content/uploads/>
- [2] 30 ตุลาคม 2558. AUTOMATIC VEHICLE MODEL DETECTION SYSTEM. [Online]. Available : <http://www.researchsystem.siam.edu/images/thesistee/>
- [3] 1 ธันวาคม 2558. วิธีลด Noise ในภาพ. [Online]. Available : <https://www.ilovetogo.com/Article/102/3131/>
- [4] 1 ธันวาคม 2558. OpenCV คือ อะไร ใช้ประโยชน์อะไรได้บ้าง. [Online]. Available : <https://th.answers.yahoo.com/question/index?qid=20090523084409AAgq3Rw>
- [5] Gary, B. and Adrian, K. 2008. O'Reilly Learning OpenCV. United States of America : O'Reilly Media.
- [6] 4 ธันวาคม 2558. Computer vision. [Online]. Available : http://en.wikipedia.org/wiki/Computer_vision
- [7] 7 ธันวาคม 2558. Template Matching. [Online]. Available : http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching.html
- [8] C. Schmid and R. Horaud. 2000. Matching Images with Different Resolutions. Europe Montbonnot, France : 612-618 vol.1.

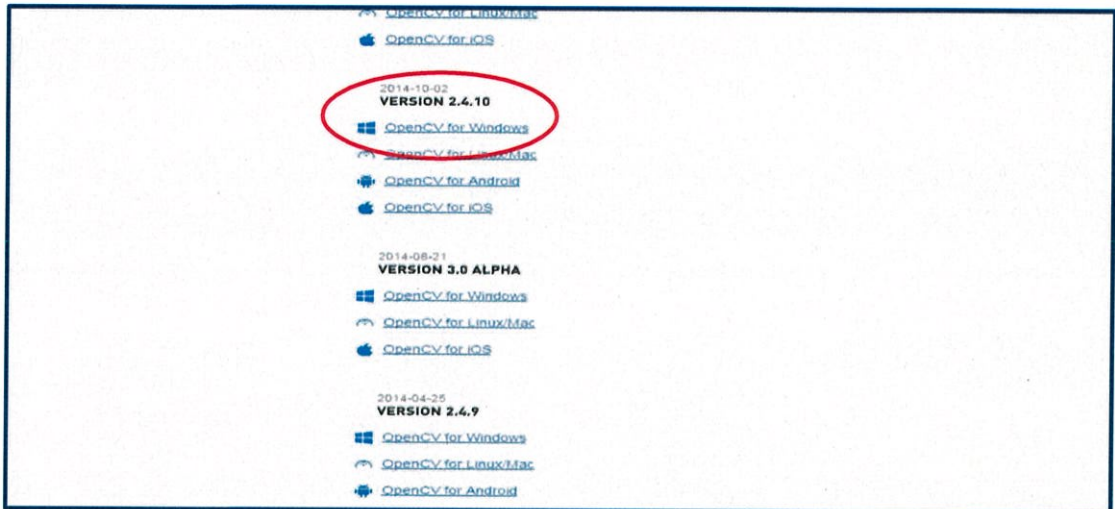
ภาคผนวก

ภาคผนวก ก

วิธีการติดตั้งไลบรารี OpenCV สำหรับใช้งานในภาษาซีพลัสพลัส

วิธีการติดตั้ง OpenCV 2.4.10 เพื่อพัฒนาโปรแกรมด้านคอมพิวเตอร์วิชั่นและการประมวลผลภาพดิจิทัลบนระบบปฏิบัติการ windows มีขั้นตอนดังนี้

- 1) ดาวน์โหลด OpenCV จาก www.opencv.org/downloads โดยให้คลิกที่ OpenCV for Windows

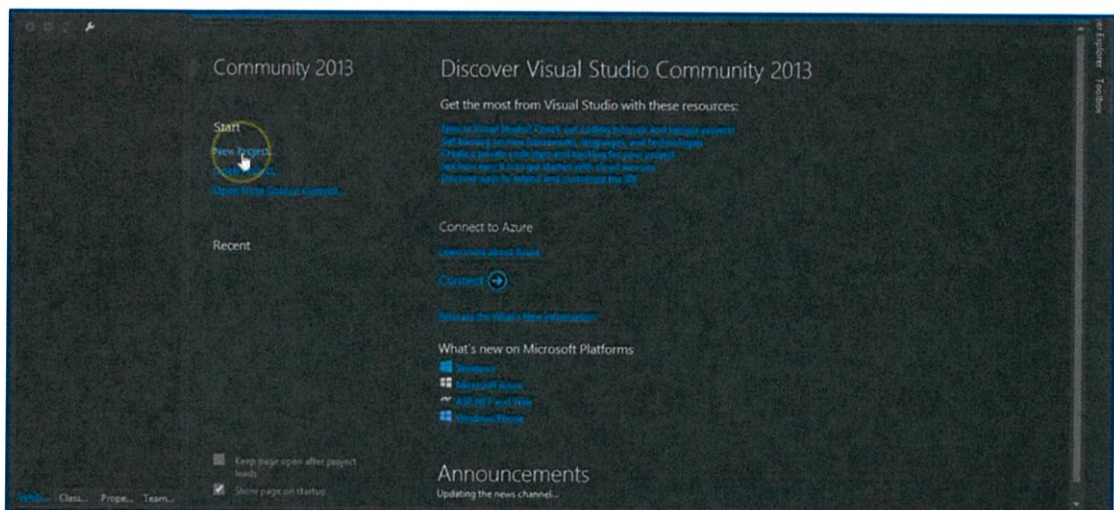


รูปที่ ก.1 หน้าต่างของเว็บไซต์ www.opencv.org/downloads

2) บันทึกไฟล์โดยสร้างโฟลเดอร์ OpenCV2410 ในไดรฟ์ C

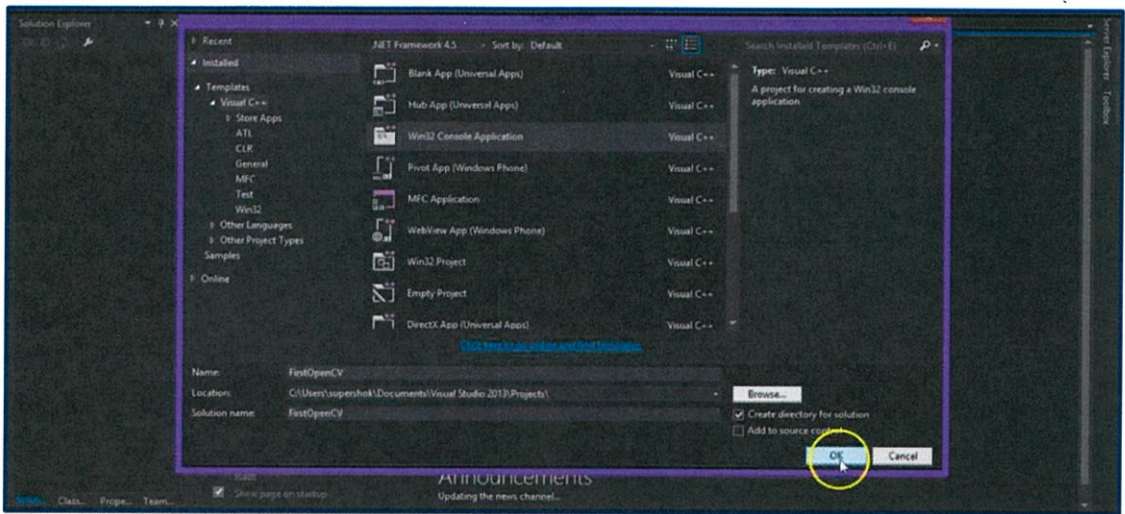
3) เมื่อทำการติดตั้งเสร็จเรียบร้อยแล้วโปรแกรม Visual Studio จำเป็นที่จะต้องสร้างโปรเจค และตั้งค่าดังนี้

3.1) New -> Project



รูปที่ ก.2 แสดงการสร้างโปรเจค

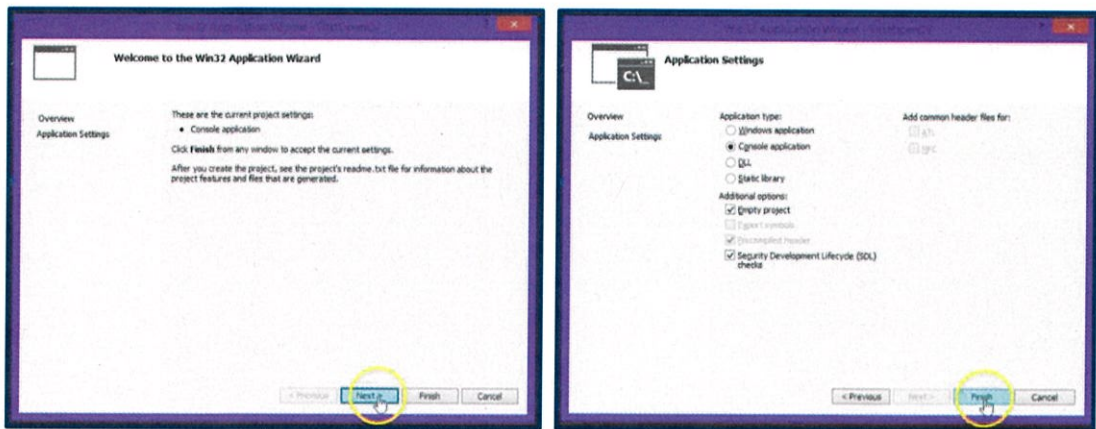
3.2) เลือก Win32 Console Application ตั้งชื่อโปรเจค และกดปุ่ม OK



รูปที่ ก.3 แสดงหน้าต่างการตั้งค่า

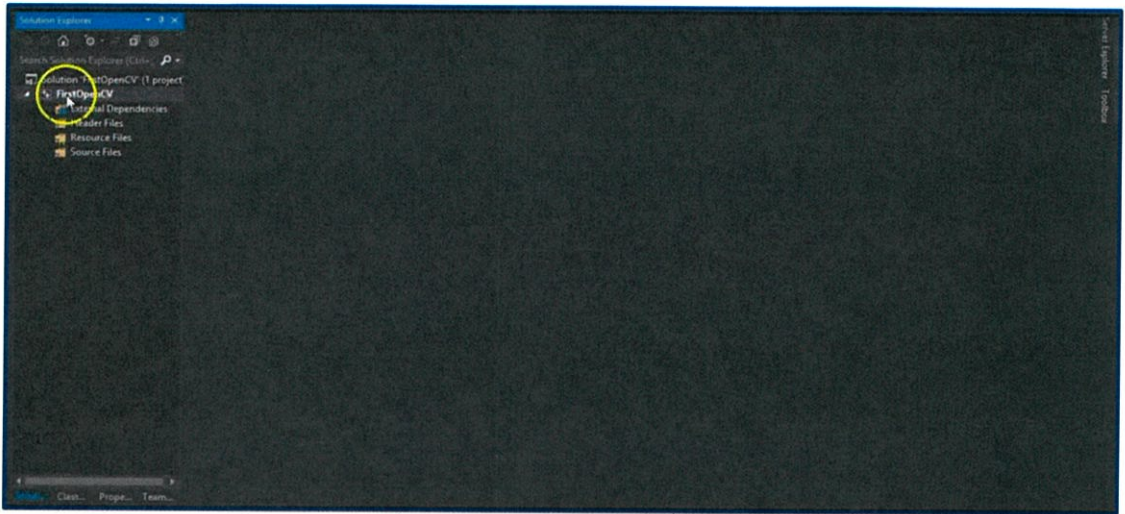
3.3) Application Settings

Application type เลือก console Application และ Additional options เลือก Empty Project

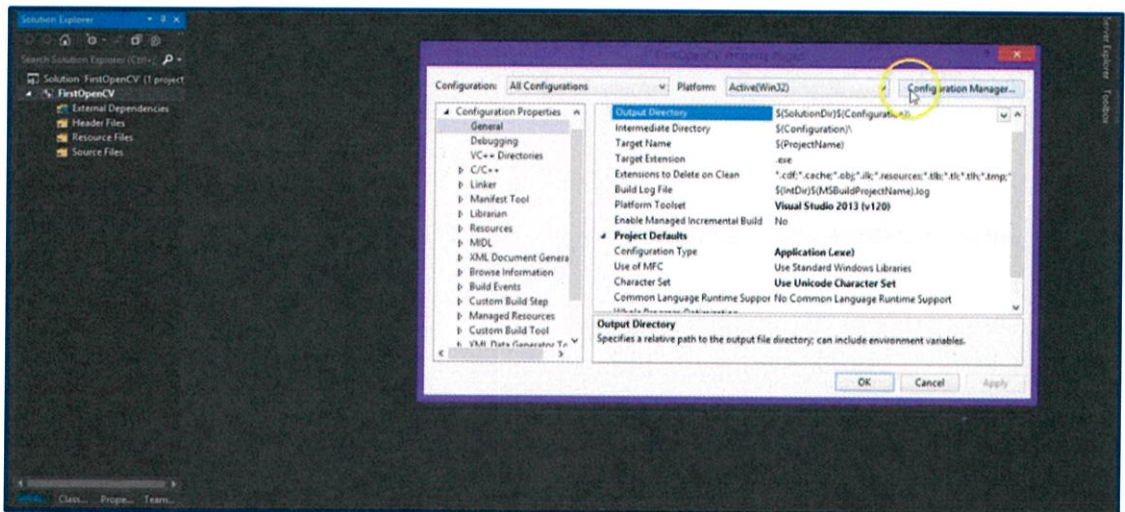


รูปที่ ก.4 แสดงหน้าต่างการตั้งค่า

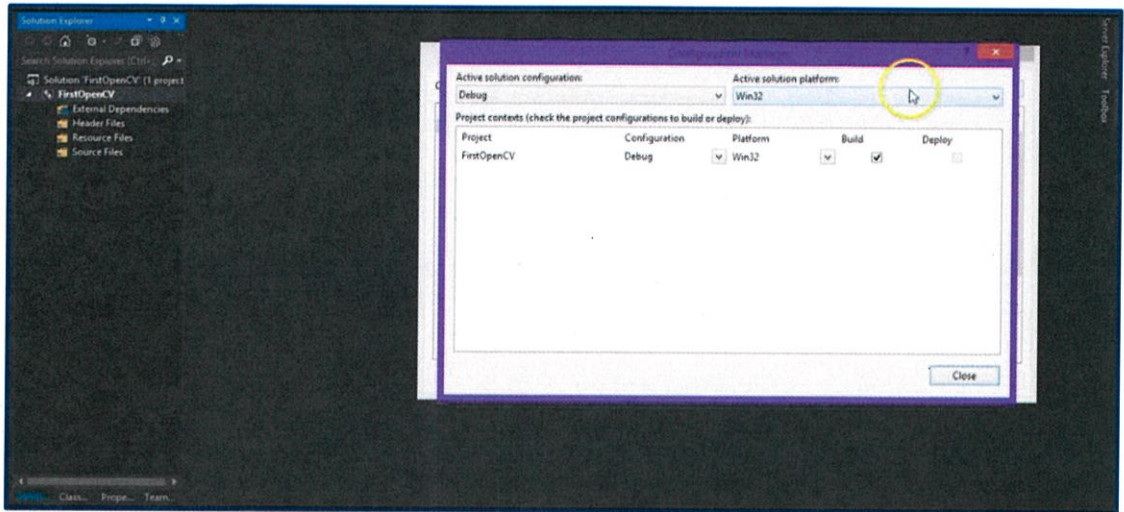
3.4) คลิกที่ชื่อโปรเจกต์เลือก property และ Configuration Manager จะปรากฏหน้าต่างดังรูปที่ ก.7



รูปที่ ก.5 แสดงหน้าต่างการคลิกเลือก

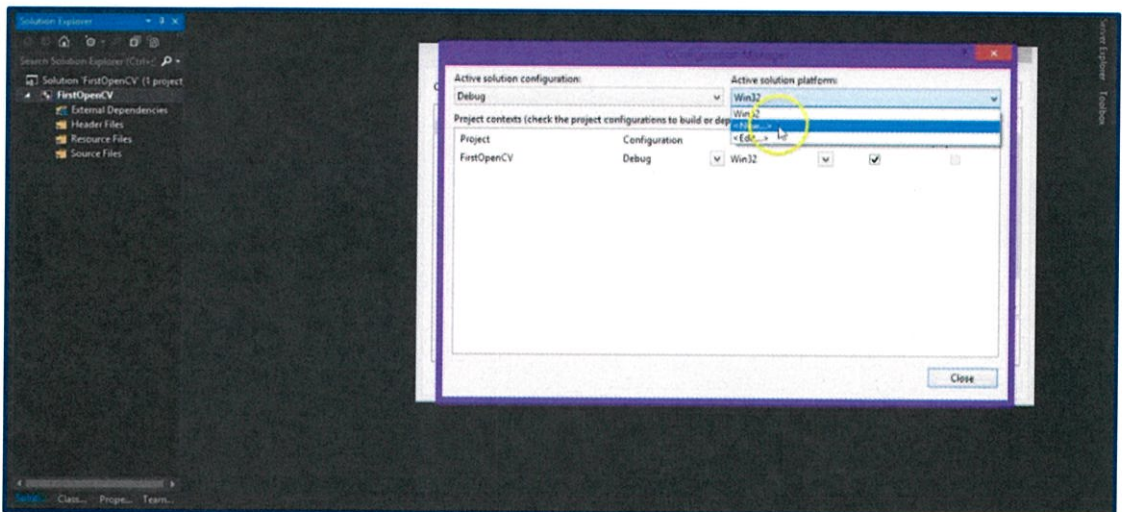


รูปที่ ก.6 แสดงหน้าต่างคลิกเลือก Configuration Manager

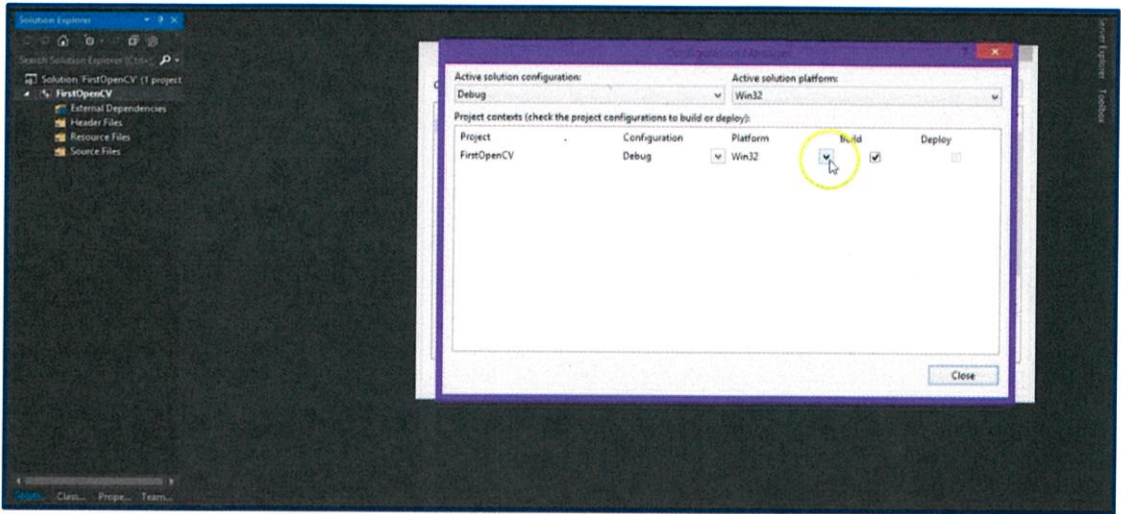


รูปที่ ก.7 แสดงหน้าต่าง Configuration Manager

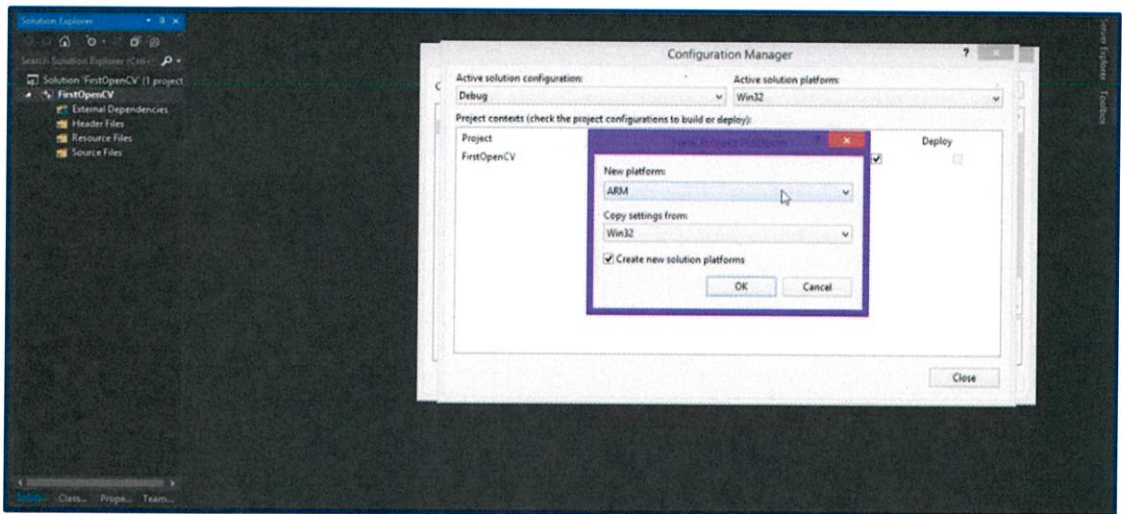
3.5) เลือก Win32 คลิกที่ new และเลือก Win32 ตรง platform คลิก new จะได้หน้าต่างแสดงดังรูปที่ ก.8 และก.9



รูปที่ ก.8 แสดงการตั้งค่าใน Configuration Manager

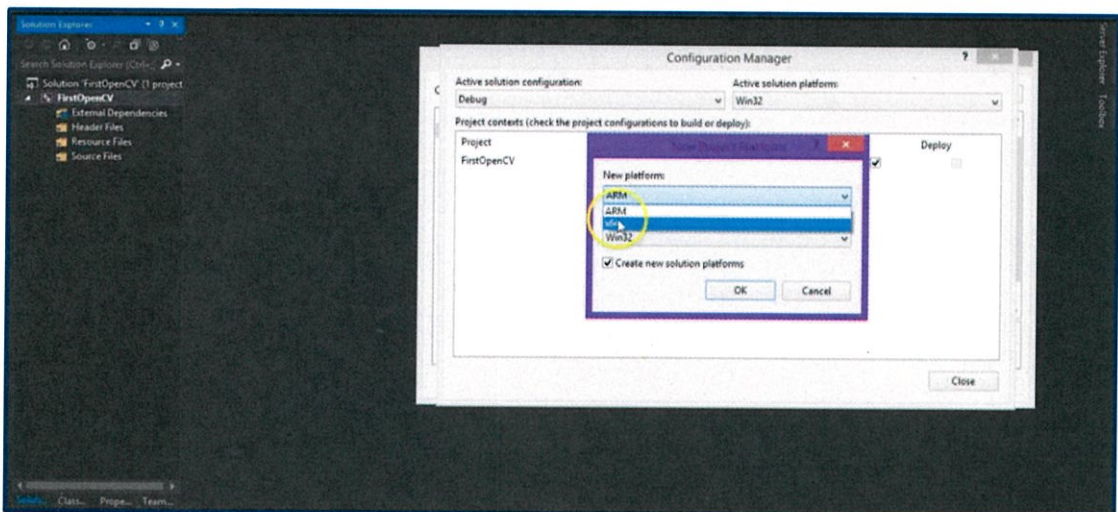


รูปที่ ก.9 แสดงการตั้งค่าใน Configuration Manager



รูปที่ ก.10 แสดงหน้าต่างเมื่อคลิก new ที่ Win32

3.6) จากนั้นคลิกที่ ARM เลือก x64 และกด OK



รูปที่ ก.11 แสดงการตั้งค่าใน Configuration Manager

3.7) เลือก property คลิ๊กที่ C/C++ -> General -> Additional Include Directories กด Edit ใส่

C:\OpenCV2410\opencv\build\include

C:\OpenCV2410\opencv\build\include\opencv

C:\OpenCV2410\opencv\build\include\opencv2

จากนั้นเลือก Linker -> General -> Additional Library Directories

กด Edit ใส่ C:\OpenCV2410\opencv\build\x64\vc12\lib

และเลือก Linker -> Input -> Additional Dependencies

กด Edit ใส่

opencv_calib3d2410d.lib

opencv_contrib2410d.lib

opencv_core2410d.lib

opencv_features2d2410d.lib

opencv_flann2410d.lib

opencv_gpu2410d.lib

opencv_highgui2410d.lib

opencv_imgproc2410d.lib

opencv_legacy2410d.lib

opencv_ml2410d.lib

opencv_nonfree2410d.lib

opencv_objdetect2410d.lib

opencv_ocl2410d.lib

opencv_photo2410d.lib

opencv_stitching2410d.lib

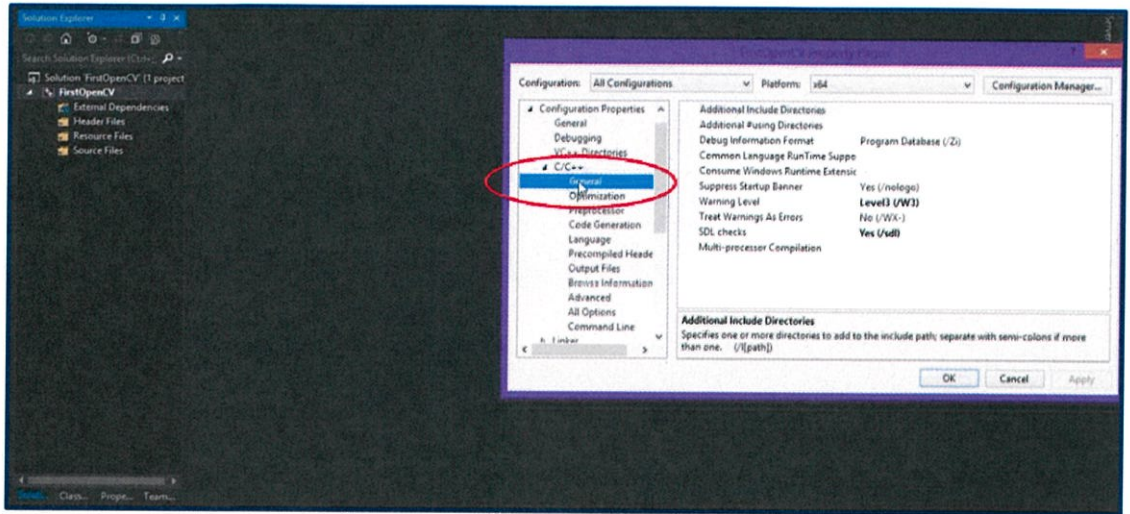
opencv_superres2410d.lib

opencv_ts2410d.lib

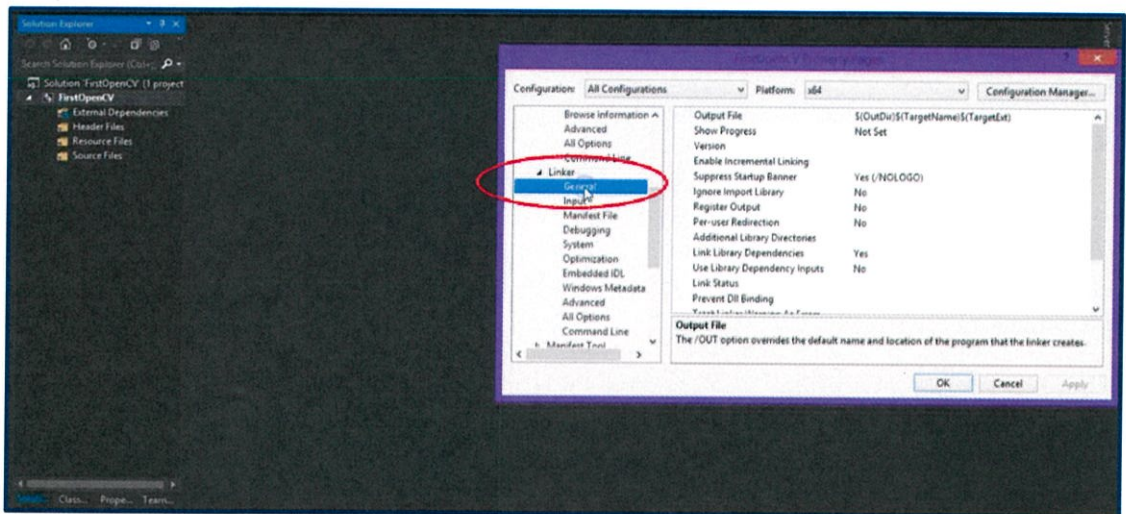
opencv_video2410d.lib

opencv_videostab2410d.lib

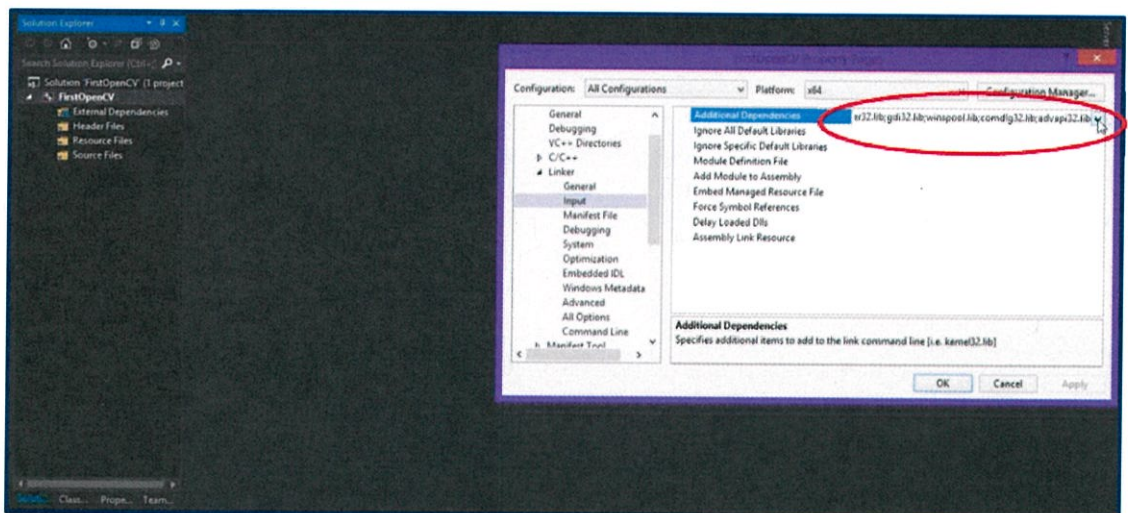
แสดงขั้นตอนที่กล่าวมาดังรูปที่ ก.12, ก.13, ก.14 และก.15



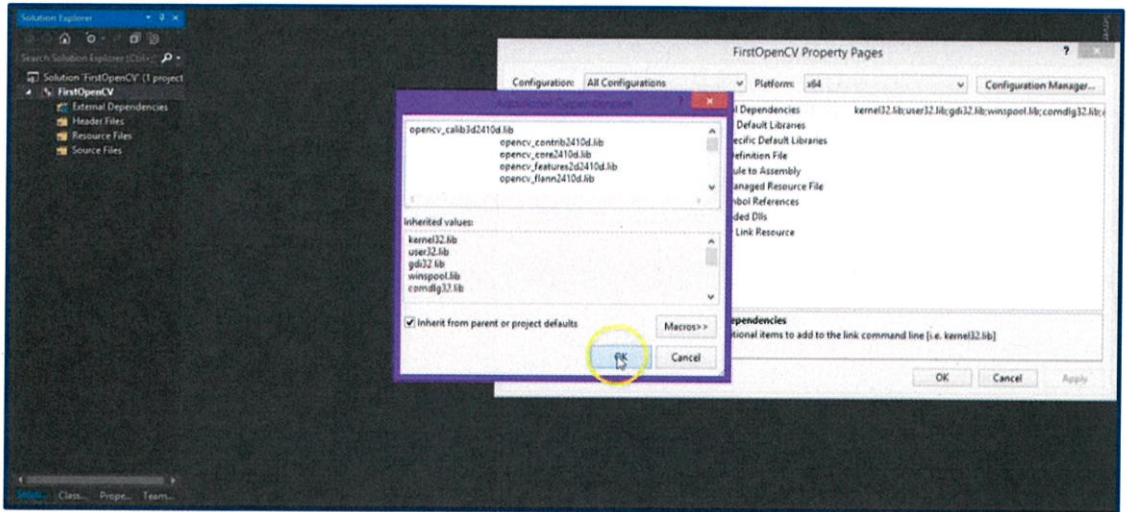
รูปที่ ก.12 แสดงขั้นตอนการตั้งค่าใน property



รูปที่ ก.13 แสดงขั้นตอนการตั้งค่าใน property



รูปที่ ก.14 แสดงขั้นตอนการตั้งค่าใน property



รูปที่ ก.15 แสดงขั้นตอนสุดท้ายของการตั้งค่าใน property

4) เพิ่ม opencv path ลงในระบบ

วิธีการ set path เข้าไปที่

Control Panel -> System -> Advanced System Setting -> Environment Variables จะปรากฏหน้าต่างนี้ขึ้นตรง System variables เลือก path -> edit

Variable value ให้ add C:\OpenCV2410\opencv\build\x64\vc12\bin

จากขั้นตอนที่กล่าวมาข้างต้นทำให้สามารถใช้งานฟังก์ชัน ตัวแปร และประเภทข้อมูลต่างๆ ของไลบรารี OpenCV ได้

ทดสอบการใช้งาน ดังนี้

นำโค้ดตัวอย่างที่ใช้งานไลบรารี OpenCV มารันในระบบ

โค้ดแสดงรูปภาพ

```
#include "opencv2/highgui/highgui.hpp"
#include <iostream>
using namespace cv;
using namespace std;
int main( int argc, const char** argv )
{
    Mat img = imread("MyPic.JPG", CV_LOAD_IMAGE_UNCHANGED);
    if (img.empty())
    {
        system("pause");
        return -1;
    }
    namedWindow("MyWindow", CV_WINDOW_AUTOSIZE);
    imshow("MyWindow",img);
    waitKey(0);
    destroyWindow("MyWindow");
    return 0;
}
```

รูปที่ ก.16 ตัวอย่างการใช้งานไลบรารี OpenCV เพื่อแสดงรูปภาพ