**บทที่ 2**

**ทฤษฎีที่เกี่ยวข้อง**

การนำเทคโนโลยีปัญญาประดิษฐ์นำมาใช้ในการรู้จำใบหน้า และสามารถรู้จำใบหน้าบุคคลเป้าหมายได้นั้น ต้องอาศัยความรู้หลายแขนงนำมารวมกัน เพื่อให้ได้ความแม่นยำที่น่าเชื่อถือ และมีประสิทธิภาพ จึงได้นำทฤษฎีที่เกี่ยวข้องในการพัฒนาดังนี้

2.1 โมเดลการเรียนรู้เชิงลึก Visual Geometry Group 16-layer: VGG-16

2.1.1 สถาปัตยกรรมคอนโวลูชัน VGG-16

2.1.2 การแบ่งชั้นเลเยอร์ของ VGG-16

2.1.3 หลักการทำงานของ VGG-16

2.2. กระบวนการปรับแต่งภาพและปรับปรุงภาพ (Image Preprocessing)

2.2.1 การปรับขนาดภาพ (Image Resizing)

2.2.2 การปรับความเข้มสีเทา (Gray Scale)

2.3. สถาปัตยกรรม CycleGAN Architecture

2.4. Haar Cascade

2.4.1 การคํานวณรูปแบบการจําลองด้วย Integral Image

2.4.2 การค้นหารูปแบบการจําลองด้วย Adaboost

2.4.3 การรวมตัวจําแนกกลุ่มแบบต่อเรียง (Cascaded Classifier)

2.5 OpenVC ในไลบรารี่ haarcascade

2.5.1 เลเยอร์คอนโวลูชัน (Convolutional Layer)

2.5.2 การทำสัญญาณรบกวน (Noise Reduction)

2.5.3 การแยกส่วน (Segmentation)

2.5.4 การหาเส้นขอบ (Edge Detection)

2.5.5 การทำฟีเจอร์ (Feature Extraction)

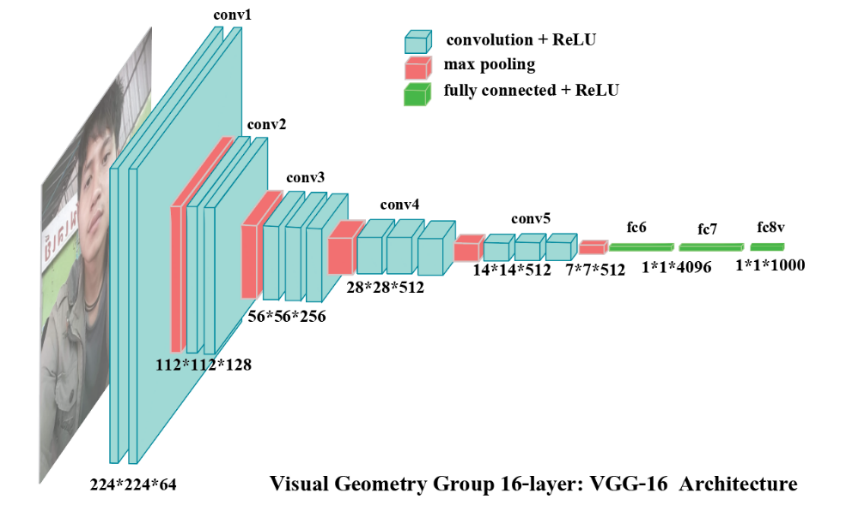
2.5.6 การเรียนรู้เชิงลึก (Deep Learning)

2.5.7 การจัดการภาพ (Image Manipulation)

**2.1 โมเดลการเรียนรู้เชิงลึก Visual Geometry Group 16-layer (VGG-16)**

**2.1.1 สถาปัตยกรรมคอนโวลูชัน VGG-16**

คือ แบบจำลอง VGG-16 เป็นสถาปัตยกรรมโครงข่ายประสาทเทียมแบบ Convolutional (CNN) มีลักษณะเด่นคือความลึก ประกอบด้วย 16 เลเยอร์ รวมถึงเลเยอร์ Convolutional 13 เลเยอร์และเลเยอร์ที่เชื่อมต่อกันอย่างสมบูรณ์ 3 เลเยอร์ VGG-16 มีจุดเด่นในเรื่องความเรียบง่ายและประสิทธิภาพ รวมถึงความสามารถในการบรรลุประสิทธิภาพที่สุดในงานคอมพิวเตอร์วิชันต่างๆ รวมถึงการจำแนกภาพและการจดจำวัตถุ สถาปัตยกรรมของแบบจำลองประกอบด้วยเลเยอร์ Convolutional ซ้อนกันตามด้วยเลเยอร์ max-pooling โดยมีความลึกที่เพิ่มขึ้นเรื่อย ๆ การออกแบบนี้ทำให้แบบจำลองสามารถเรียนรู้การแสดงลำดับชั้นที่ซับซ้อนของคุณลักษณะภาพ นำไปสู่การทำนายที่มั่นคงและแม่นยำ



**ภาพที่ 2.1** สถาปัตยกรรม VGG-16

**2.1.2 การแบ่งชั้นของ VGG-16**

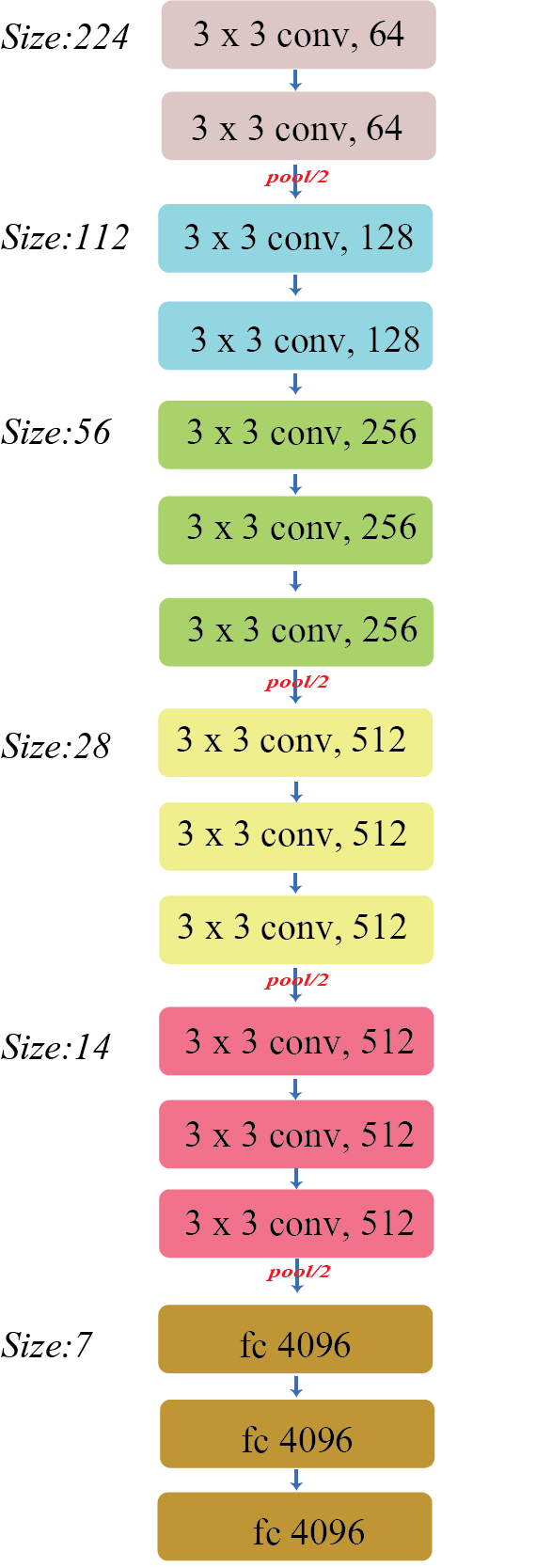
สถาปัตยกรรม VGG-16 แต่ละชั้นมีการแบ่งออกดังนี้

2.1.2.1 การนำเข้าข้อมูล โครงข่าย VGG-16 รับอินพุตรูปภาพขนาด 224×224 พิกเซล โมเดลนี้ครอปภาพขนาด 224 x 224 พิกเซล จากศูนย์กลางแต่ละภาพ

2.1.2.2 ชั้นคอนโวลูชัน (Convolutional layers) ตัวกรองคอนโวลูชันของ VGG-16 ใช้พื้นที่รับข้อมูลขนาด 3 × 3 พิกเซลและยังใช้ตัวกรองคอนโวลูชันขนาด 1×1 เป็นการแปลงเชิงเส้นของอินพุต

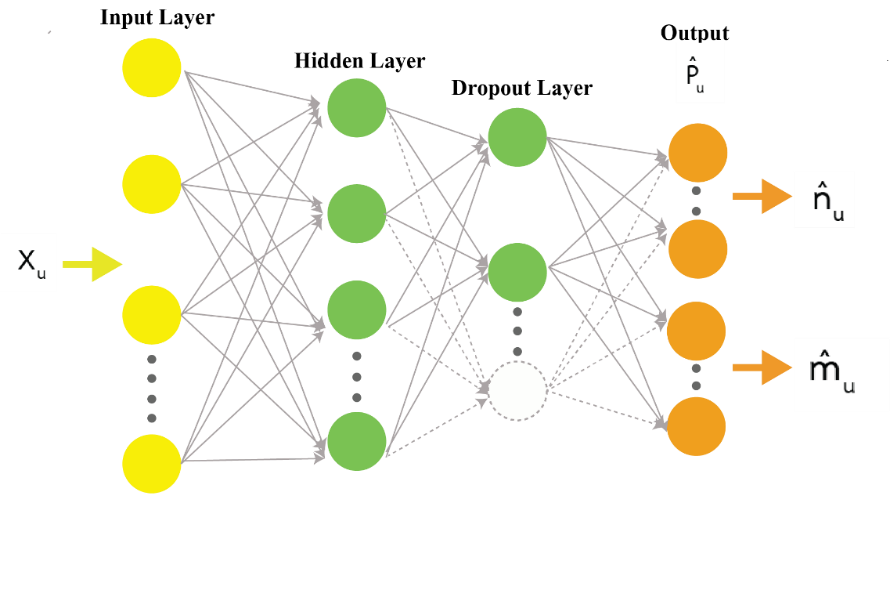
2.1.2.3 ฟังก์ชันการกระตุ้น ReLU คือฟังก์ชันการกระตุ้นเชิงเส้นที่นวัตกรรมสำคัญ ReLU เป็นฟังก์ชันเส้นตรงที่ให้ผลลัพธ์ที่สอดคล้องสำหรับอินพุตที่เป็นบวกและให้ผลลัพธ์เป็นศูนย์สำหรับอินพุตที่เป็นลบ ทำให้ VGG-16 พิกเซล เพื่อรักษาความละเอียดของพื้นที่หลังจากการคอนโวลูชัน

2.1.2.4 เลเยอร์ซ่อน (Hidden layers) ทุกเลเยอร์ซ่อนของเครือข่าย VGG-16 แม้จะมีการเพิ่มระยะเวลาการเทรนและใช้ความจำที่มากขึ้น แต่ก็ได้ปรับค่าความถูกต้องที่มากขึ้นด้วยเช่นกัน

2.1.2.5 ชั้นการรวมข้อมูล (Pooling layers) เป็นการรวมข้อมูลตามหลังชั้นคอนโวลูชันหลายๆ ชั้นนี้ช่วยลดมิติและจำนวนพารามิเตอร์ของแผนที่ลักษณะที่สร้างขึ้นโดยขั้นตอนคอนโวลูชันแต่ละขั้นตอน ชั้นนี้มีความสำคัญในการเพิ่มจำนวนตัวกรองที่มีการจำนวนที่มากขึ้นจาก 64 ไปยัง 128, 256 จนถึง 512 ในชั้นสุดท้าย

2.1.2.6 ชั้นที่เชื่อมต่อเต็มรูปแบบ (Fully connected layers) ในโครงข่าย VGG-16 รวมถึงชั้นที่เชื่อมต่อเต็มรูปแบบสามชั้น ชั้นแรกและชั้นที่สองมีแต่ละชั้นมี 4,096 ช่อง และชั้นที่สามมี 1,000 ช่อง หนึ่งช่องสำหรับแต่ละคลาส

**ภาพที่ 2.2** แสดงขนาด (Size) ของแต่ละชั้นของ VGG-16



**ภาพที่ 2.3** แสดงชั้น Hidden Layer

**2.1.3 หลักการทำงานของ VGG-16**

2.1.3.1 โครงสร้างแบบลำดับชั้น (Hierarchical Structure)

VGG-16 มีโครงสร้างที่ประกอบด้วย 16 เลเยอร์ที่สามารถเรียนรู้ได้ ซึ่งประกอบไปด้วยชั้นคอนโวลูชัน (Convolutional layers) และชั้น Fully Connected (FC layers) โดยหลักแล้วประกอบด้วย Convolutional layers และ Fully Connected layers

2.1.3.2 ชั้นคอนโวลูชัน (Convolutional Layers)

ใช้ฟิลเตอร์ขนาด 3x3 สำหรับทุก ๆ Convolutional layer เพื่อจับรายละเอียดของภาพในระดับต่างๆ การใช้ Padding แบบ “same” เพื่อให้ขนาดของภาพออกมาเท่ากับภาพเข้า หลังจากแต่ละ Convolutional block จะมีการใช้ฟังก์ชัน ReLU (Rectified Linear Unit) เป็น Activation function

2.1.3.3 การลดขนาดของภาพ (Pooling)

Max-Pooling ขนาด 2x2 ถูกใช้งานในแต่ละชั้นเพื่อลดขนาดของภาพระหว่างการประมวลผล แต่ยังคงรักษาคุณสมบัติสำคัญของภาพไว้

2.1.3.4 Fully Connected Layers

เมื่อผ่านชั้นคอนโวลูชันทั้งหมด ภาพจะถูก Flatten (แปลงเป็นเวกเตอร์หนึ่งมิติ) และผ่านเข้าสู่ชั้นที่เชื่อมต่อแบบเต็ม (Fully Connected) จำนวน 3 ชั้น ซึ่งชั้นสุดท้ายเป็น Softmax layer ที่ทำหน้าที่จำแนกประเภท โดยให้ค่าความน่าจะเป็นสำหรับแต่ละคลาส

2.1.3.5 การเรียนรู้จากภาพ (Feature Learning)

VGG-16 สามารถเรียนรู้คุณสมบัติต่าง ๆ ของภาพจากระดับต่ำ (เช่น ขอบ รูปทรงพื้นฐาน) ไปจนถึงระดับสูง (เช่น วัตถุ ใบหน้า) โดยผ่านชั้นคอนโวลูชันหลายชั้น

2.1.3.6 ขนาดภาพอินพุต

ขนาดของภาพที่ VGG-16 ต้องการให้เป็นอินพุตคือ 224 x 224 พิกเซล และเป็นภาพที่มี 3 ช่องสี (RGB)

2.1.3.7 สมการที่ใช้งานในสถาปัตยกรรมมีดังนี้

การคอนโวลูชันเป็นกระบวนการหลักที่ใช้ในการดึงฟีเจอร์จากภาพ สมการที่ใช้อธิบายกระบวนการคอนโวลูชันระหว่างฟิลเตอร์กับข้อมูลอินพุตเป็นดังนี้



(1)

โดยกำหนดให้

 คือ ค่าสัญญาณผลลัพธ์ที่พิกเซลตำแหน่ง(i, j)

 คือ ค่าอินพุตที่ตำแหน่ง

 คือ ค่าในฟิลเตอร์หรือ kernel ที่ใช้คูณกับอินพุต

 คือ ขนาดของฟิลเตอร์

ฟังก์ชัน ReLU (Rectified Linear Unit) ถูกใช้เป็นฟังก์ชันกระตุ้นในแต่ละเลเยอร์คอนโวลูชัน เพื่อแปลงค่าสัญญาณที่ไม่เป็นเชิงเส้น (Non-linearity) ดังนี้

 (2)

โดยกำหนดให้

 คือ ผลลัพธ์ของ ReLU

 คือ ค่าที่ได้จากการคอนโวลูชัน

ฟังก์ชันนี้จะให้ค่าเป็นศูนย์หากอินพุตน้อยกว่าศูนย์ และให้ค่าเป็นอินพุตเดิมหากมากกว่าศูนย์ การทำ Max-Pooling ถูกใช้เพื่อลดขนาด โดยเลือกค่าสูงสุงจากบริเวณย่อย

(Sub-region) 2X2

 (3)

โดยกำหนดให้

 คือ ค่าผลลัพธ์จาก Max-Pooling ที่ตำแหน่ง (I, j)

 คือค่าภายใน sub-region ขนาด 2x2 ของภาพอินพุตการทำ Max-Pooling จะลดขนาดของภาพลงเหลือครึ่งของขนาดเดิม

Fully Connected Layer ในชั้นที่เชื่อมต่อกันอย่างสมบุรณ์ (Fully Connected Layer) ภาพที่ผ่านการ Flatten (เปลี่ยนเป็นเวกเตอร์หนึ่งมิติ) จะถูกคำนวณโดยสมการต่อไปนี้

  (4)

โดยกำหนดให้

*y* คือ ผลลัพธ์ของเลเยอร์ Fully Connected

*x* คือ ค่าอินพุตจากเลเยอร์ก่อนหน้า (เวกเตอร์จากการ Flatten)

*W* คือ เมตริกน้ำหนักที่ถูกเรียนรู้

*b* คือ bias ที่ถูกเพิ่มเพื่อความยึดหยุ่นในการเรียนรู้

Softmax Functionชั้นสุดท้ายของ VGG-16 ใช้ฟังก์ชัน Softmax เพื่อแปลงผลลัพธ์เป็นค่าความน่าจะเป็นในการจำแนกประเภทโดยสมการของ Softmax คือ

 (5)

โดยกำหนดให้

 คือ ความน่าจะเป็นที่ข้อมูลจะถูกจัดให้อยู่ในคลาส i

 คือ ค่าอินพุตที่ผ่านมาจากเลเยอร์ Fully Connected

*N*คือ จำนวนคลาสทั้งหมด

**2.2. กระบวนการปรับแต่งภาพและปรับปรุงภาพ (Image Preprocessing)**

**2.2.1 การปรับขนาดภาพ (Image Resizing)**

การปรับขนาดภาพเป็นกระบวนการที่ใช้เพื่อเปลี่ยนขนาดของภาพให้มีขนาดที่เหมาะสมกับการวิเคราะห์ที่จะทำต่อไป โดยใช้เทคนิค เช่นการใช้การโยงข้อมูลโดยใช้ภาพตัวอย่างเพื่อปรับขนาดภาพนำเข้าให้มีเหมาะสมกับอินพุตที่ต้องการ

**2.2.2 การปรับความเข้มสีเทา (Gray Scale)**

โดยใช้วิธีหาความเข้มแสงด้วยวิธีการประมาณค่าจากตำแหน่งใกล้เคียงด้วยพหุนามกำลังศูนย์โดยจะคำนวณเพื่อหาว่าตำแหน่งที่คำนวณได้นั้นใกล้เคียงกับตำแหน่งใดในภาพอินพุตมากที่สุด จากนั้นจะใช้ความเข้มแสงจากจุดนั้นเป็นความเข้มแสงของตำแหน่งในภาพใหม่ โดยใช้สมการ

โดยกำหนดให้

 คือ ค่าความเข้มแสงที่ตำแหน่งของภาพอินพุต

**** คือ ค่าความเข้มแสงที่ตำแหน่ง ของภาพผลลัพธ์

**ภาพที่ 2.4** แสดงภาพต้นฉบับ และรูปที่แปลงเป็นสีเทา

**2.2.3 การใช้ภาพสีในการแปลง (RGB to Grayscale)**

การปรับภาพสีธรรมให้เป็นรูปภาพสีเทาโดยใช้หลักการ RGB แล้วแปลงค่าตามรูปแบบสูตรดังนี้

Grayscale = 0.299R + 0.587G + 0.114B

โดยค่า Gray ค่าความเข้มของสีเทา มีค่า 0–255

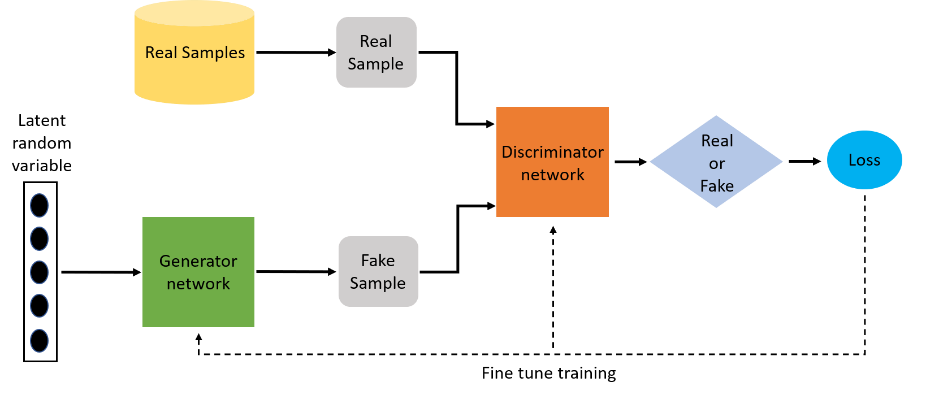
R มีค่า 0–255, G มีค่า 0–255, B มีค่า 0–255

**2.3 สถาปัตยกรรม CycleGAN Architecture**

โดยได้พัฒนามาจาก Generative Adversarial Networks (GAN) โดยต้นฉบับเป็นของ Ian Goodfellow ที่ได้คิดค้นเป็นโมเดลในการสร้างรูปภาพให้มีความเหมือนหรือใกล้เคียงตัวอย่างมากที่สุด ซึ่งประกอบด้วย ปัญญาประดิษฐ์ 2 ตัวที่ทำงานร่วมกันคือ

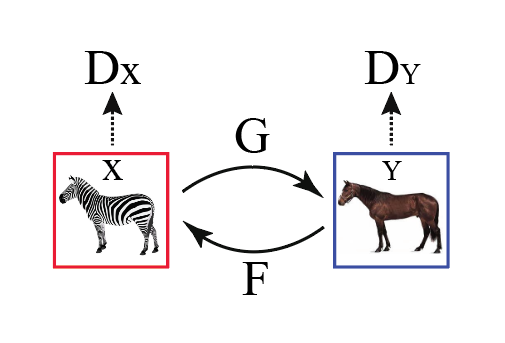
Generator ซึ่งเป็นตัวสร้างข้อมูลรูปภาพจาก noise / latent random variable ที่เป็นค่าอินพุตเป็นตัวเริ่มต้น แล้วต่อจากนั้นจะมีการเรียนรู้ autoencoder จากค่า loss (Backpropagation) วนกลับมา ที่ส่งมาจาก Discriminator ที่ส่งมาหลังจากแยกแยะว่าเป็นภาพที่สร้างขี้นแต่แยกแยะออก (Fake) มีการเรียนรู้วนไป จนกว่าจะมีการแยกแยะว่าเป็นภาพที่สร้างขึ้นแต่แยะไม่ออก (Real)

Discriminator ทำหน้าที่แยกแยะระหว่าง ภาพจริงที่มีการอินพุต (Real image) ไว้เพื่อใช้แยกแยะ กับภาพที่สร้างขึ้นจากตัว Generator และจะส่งค่า loss (Backpropagation) กลับมายัง Generator เพื่อให้มีการเรียนรู้และสร้างจนกว่า จะแยกแยะไม่ออก



**ภาพที่ 2.5** ไดอะแกรมการทำงานของโมเดล GAN

CycleGAN เป็นโมเดลสถาปัตยกรรมที่ถูกพัฒนาต่อจาก GAN โดย Jun-Yan Zhu และคณะ จากมหาวิทยาลัยแคลิฟอร์เนีย เบิร์กลีย์ (University of California, Berkeley) จะเน้นการเรียนรู้และการแปลงข้อมูลระหว่างสองโดเมนที่ไม่เกี่ยวข้องกัน โดยไม่ต้องมีข้อมูลเฉพาะในการฝึก ซึ่งเป็นการแปลงจากโดเมน A ไปยัง โดเมน B แล้วมีการแปลงกลับ จากโดเมน B ไปยัง โดเมน A อีกทีได้อย่างสมบูรณ์เป็นวงรอบ(Cycle) โดยไม่ต้องมีการใช้ข้อมูลที่เฉพาะเจาะจงในกระบวนการฝึก ทำได้มีการเพิ่มประสิทธิ์ภาพด้านกราฟิก การสร้างภาพที่มีความซับซ้อน การแปลงรูปภาพหนึ่งเพื่อให้ได้อีกรูปภาพต้องการ โดยไม่มีการจับคู่หรือให้ข้อมูลรูปภาพอะไรแก่ตัวสร้าง (Generator) เมื่อความต้องการ (Goal) สร้างภาพ (Image X) เซตค่าโดนเมนคือจากแหล่งต้นทาง แล้วมีการเรียนรู้ การแปลคุณลักษณะทางโดเมน ไปยังโดเมนของภาพเป้าหมาย(Image Y) เซตค่าโดนเมนคือ จะได้ว่า (G(x)) โดยใช้ ค่าสูญเสียจากการแข่งขัน (Adversarial loss) สำหรับใช้การแข่งขันกันภายในระหว่าง Generator และ Discriminator โดยใช้โดเมน X ของภาพต้นแบบ ไปยังเป้าหมายโดเมน Y ซึ่งเป็นรูปแบบการฝึกโมเดลแบบ GAN และสร้างโมเดลอีกตัว ซึ่งบังคับให้ย้อนกลับ (Enforce)  *(F(y))* เป็นวงรอบการเรียนรู้ โดยมี Discriminator (Dx) แล้วนำต่อกันทำให้เกิดเป็นวงรอบการเรียนรู้ เรียกว่า CycleGAN

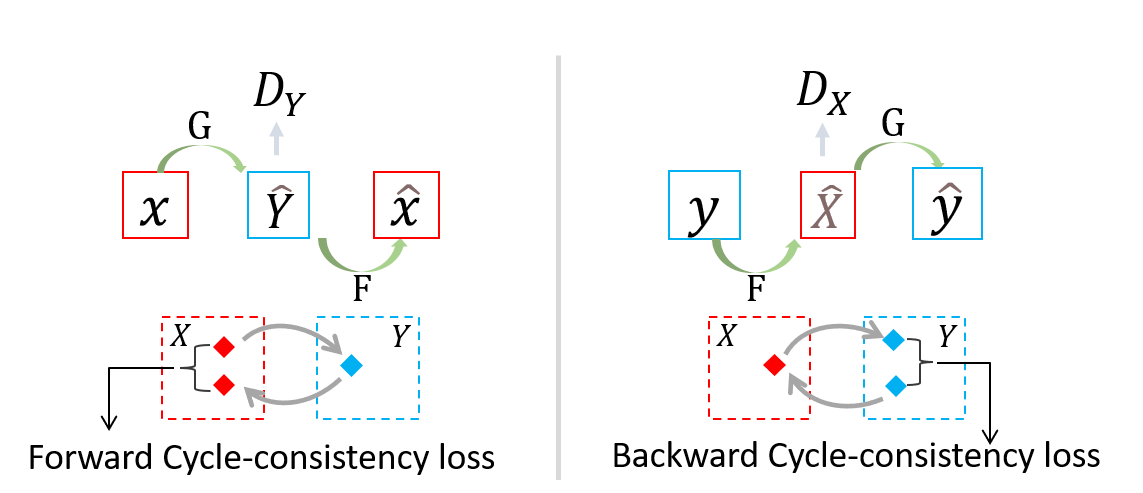


**ภาพที่ 2.6**  โมเดล CycleGAN

แต่ด้วยการค่าอินพุตในการสร้างวนรอบแข่งขันระหว่าง Generator และ Discriminator นั้นมีการโต้ตอบกันหลายรอบทำให้ค่าความถูกต้องจากภาพต้นฉบับนั้นลดลง โดยมักจะเกิด Mod Collapse ซึ่งเน้นการสร้างโมเดลรูปในลักษณะเด่นหรือลักษณะบางอย่างข้อมูลเท่าโดยไม่สร้างลักษณะอื่นที่เป็นบริบทรอบๆ ภาพ เช่น เมื่อในรูปภาพมีม้าอยู่ หลายตัว ก็จะมีการสร้างแต่ภาพม้าที่อยู่ข้างหน้าสุด โดยไม่สนใจรายละเอียดของม้าที่อยู่ไกล หรือวัตถุภาพอื่นๆ ทำให้ได้ภาพที่ไม่ค่อยถูกต้องในแบบต้นฉบับ ในแง่ของคุณภาพการสร้างภาพทั้งหมดก็จะดูลดลงไปเพื่อแก้ปัญหาดังกล่าวไม่ให้ภาพเสียคุณภาพไปนั้น จึงได้มีการทำการสร้างภาพ Reconstruction ไปยังภาพต้นฉบับอีกครั้งเพื่อตรวจสอบความถูกต้อง

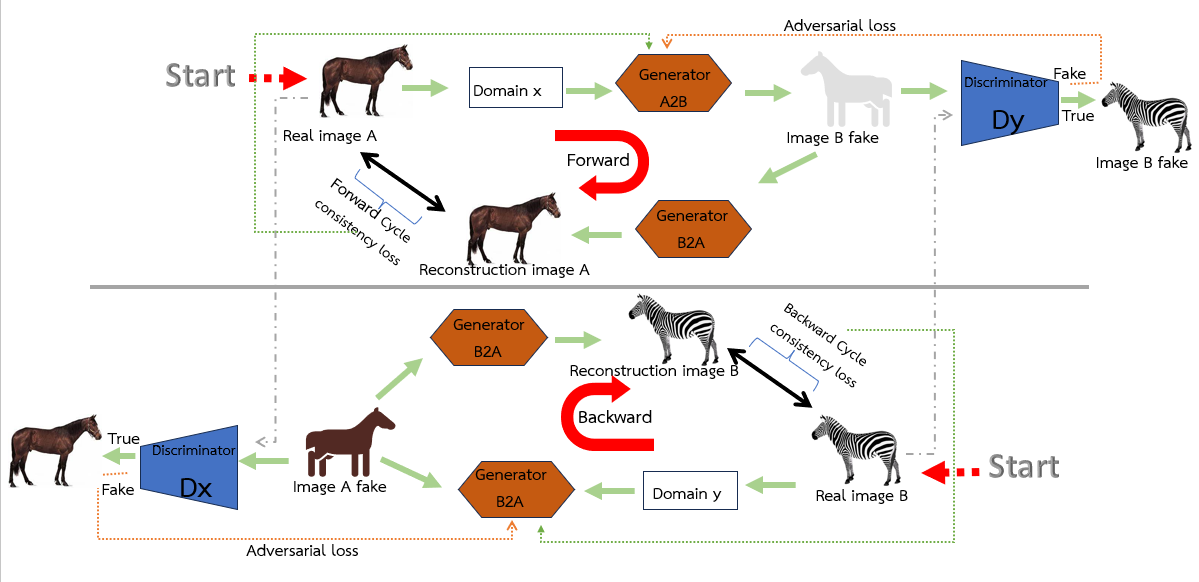
CycleGAN จะมีการโดยการแปลงภาพจากโดเมน *x* ไปยังโดเมนเป้าหมาย  แล้วแปลงกลับไปยังโครงสร้างเหมือนเดิมเพื่อตรวจสอบความถูกต้องแบบย้อนกลับ และอีกด้านจะมีการแปลงภาพจากโดเมน *y* ไปยังโดเมนเป้าหมาย  แล้วแปลงกลับไปยังโครงสร้าง  เดิมเหมือนกัน โดยมีค่าสูญเสียความสมดุลตามรอบ (cycle-consistency loss) หรือ reconstruction error  และ  ทำการตรวจสอบว่าการแปลงรูปภาพจากโดเมนไปยังอีกโดเมนหนึ่งแล้วกลับมาเป็นรูปภาพเดิมหรือไม่ ด้วยหลักการที่ว่า หากข้อมูลมีความสัมพันธ์ของข้อมูลที่ถูกต้องสมบูรณ์แล้ว รูปภาพที่ถูกแปลงควรมีคุณลักษณะเดียวกับรูปภาพเดิมโดยไม่มีการสูญเสียข้อมูล โดยแบ่งออกเป็น

Adversarial loss เป็นเหมือน loss function ของการเรียนรู้เชิงลึกในระหว่างโดเมน X และ โดเมน Y ในการปรับค่าการเรียนรู้แสดงดังสมการ



**ภาพที่ 2.7** ค่าสูญเสียความสมดุลตามรอบแบบไปข้างหน้า (Cycle-Consistency Loss)

จากภาพที่ 2.7 เพื่อให้เข้าใจ แนวคิด Cycle-consistency loss ใน CycleGAN โดยใช้ตัวสร้าง (G และ F) และตัวแยกแยะ (Dx และ Dy) เพื่อแปลงข้อมูลระหว่างสองโดเมน (X และ Y) โดยไม่มีข้อมูลจับคู่กัน Forward Cycle-consistency Loss ตัวอย่างจาก X ถูกสร้างเป็น Y ผ่าน G และแปลงกลับเป็น X ผ่าน F ความต่างระหว่าง X ต้นฉบับกับ X ที่สร้างขึ้นเป็น forward loss Backward Cycle-consistency Loss ตัวอย่างจาก Y ถูกสร้างเป็น X ผ่าน F และแปลงกลับเป็น Y ผ่าน G ความต่างระหว่าง Y ต้นฉบับกับ Y ที่สร้างขึ้นเป็น backward loss ซึ่งค่า Cycle-consistency loss เพื่อทำให้มั่นใจว่าตัวอย่างที่ถูกสร้างสามารถแปลงกลับมาเป็นต้นฉบับได้ ช่วยให้ CycleGAN เรียนรู้การแปลงที่สมจริงและสอดคล้อง



**ภาพที่ 2.8**  แสดงการทำงานโดยรวมของโมเดล CycleGAN

**2.4. Haar Cascade**

วิธีการตรวจจับใบหน้า Haar Detection ของ Viola-Jones ที่มีประสิทธิภาพสูงโดยนําเสนอเทคนิค ในการตรวจจับใบ้หน้านั้นต้องใช้เทคนิคสเกล เชดสีแรงเงา ความเข้ม มีตำแหน่ง ระยะห่าง ที่มีผลต่อการวิเคราะห์ใบหน้าบุคคลจากรูปภาพ ที่ส่งเข้ามาเพื่อให้ความแม่นยำรวดเร็วและมีประสิทธิภาพ

ที่สูงด้วยเทคนิค 3 ขั้นตอน คือ

2.4.1. การคํานวณรูปแบบการจําลองด้วย Integral Image

2.4.2 การค้นหารูปแบบการจําลองด้วย Adaboost

2.4.3. การรวมตัวจําแนกกลุ่มแบบต่อเรียง (Cascaded Classifier)

หลักการพื้นฐานของเทคนิคการตรวจจับใบหน้าของ Viola-Jones คือ การนําภาพที่ต้องการตรวจหาใบหน้ามาแบ่งเป็นภาพยอย (Sub-window) จากนั้นภาพยอยดังกล่าวจะถูกพิจารณาเป็นภาพอินพุตของกระบวนการตรวจหาใบหน้า เทคนิคทั่วไปในการตรวจหาใบหน้าจะทำการปรับขนาดของภาพอินพุตแตกตางกันหลาย ๆ ขนาด และใช้ตัวตรวจจับ (Detector) ที่มีขนาดคงที่ค้นหาวัตถุ

**2.4.1 ขั้นตอนการ Integral image**

**ตัวอย่างของการจําลองรูปแบบ Haar-like ที่ใช้ในเทคนิคการตรวจจับใบหน้าของ Viola-Jones แสดงในภาพที่ 2.9 (ก) โดยแต่ละลักษณะเดนจะประกอบด้วยพื้นที่สี่เหลี่ยม 2 ประเภทคือ สวนที่แรงเงา และสวนที่ไม่ได้แรงเงา การหาคาการจําลองรูปแบบ Haar-like คือการหาผลตางระหวางความเข้มในสวนที่แรงเงากับ สวนที่ไม่ได้แรงเงา จากนั้นนําผลลัพธ์ที่ได้ไปเปรียบเทียบกับคาขีดแบง (Threshold) กับขั้ว (Polarity) ที่ใช้ในการตัดสินใจวาภาพที่รับเข้ามาควรถูกจัดให้เป็นบวก (ภาพใบหน้า) หรือเป็นลบ (ไม่ใชภาพใบหน้า) ผลลัพธ์ที่ได้จะถูกนําไปพิจารณาในขั้นตอนต่อไปตัวอย่างการจําลองรูปแบบ Harr-like ที่แสดงใน**

**ภาพที่ 2.9 มี 3 แบบคือ การจําลองรูปแบบ Haar-like ที่ประกอบด้วยพื้นที่สี่เหลี่ยมสองภาพหรือ Two-rectanglefeature (ภาพ a และ ภาพ b) การจําลองรูปแบบ Haar-like ที่ประกอบด้วยพื้นที่สี่เหลี่ยมสามภาพ หรือ Three-rectangle feature (ภาพ c) และการจําลองรูปแบบ Haar-like ที่ประกอบด้วยพื้นที่สี่เหลี่ยมสี่ภาพหรือ Four-rectangle feature (ภาพ d) ในภาพที่ 2.9 (ข) เป็นวิธีการทั่วไปในการหาคาผลรวมความเข้มของพื้นที่สี่เหลี่ยมใดๆ สามารถเขียนแทนได้ด้วยสมการต่อไปนี้**

 (8)

โดยกำหนดให้

 แทน ค่าความเข้มที่จุด *x* และ *y* ในรูปภาพ

 แทน ผลรวมความเข้มภายพื้นที่สี่เหลี่ยม

 และ แทน พิกัดมุมทั้งสี่ของรูปสี่เหลี่ยม

จะเห็นได้วาเวลาที่ใช้ในการคํานวณนั้นจะขึ้นอยู่กับขนาดของรูปสี่เหลี่ยมในการเพิ่มประสิทธิภาพ Viola และ Jones ได้นําเสนอให้ใช้เทคนิคที่เรียกวา Integral Image ในการคํานวณการจําลองรูปแบบ Haar-like เทคนิค Integral image คือ การรวมความเข้ม (Intensity) ของแต่ละพิกเซลเข้าด้วยกัน การคํานวณหา Integral image ที่จุด สามารถเขียนแทนได้ด้วยสมการตอไปนี้

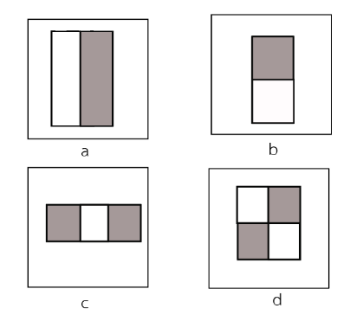
 (9)

โดยกำหนดให้

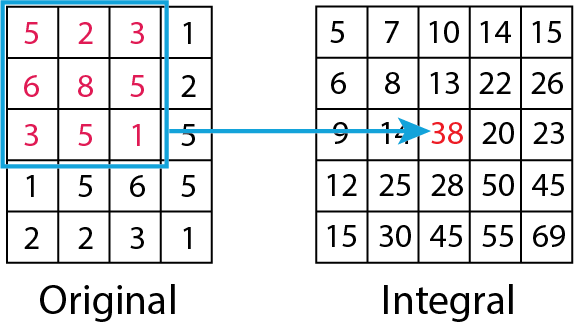
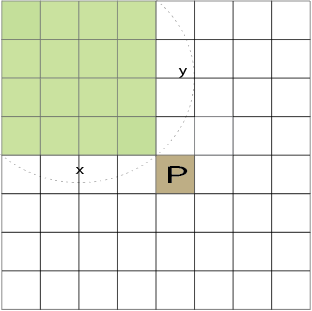
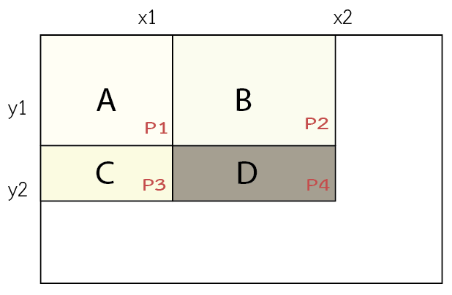
 คือ ค่าของ Integra image

 และ คือ จุดพิกัดตำแหน่ง ค่าความเข้มในแต่ละพิกเซล ของภาพต้นฉบับ

ตัวอย่างการคํานวณหาค่าผลรวมความเข้มของพื้นที่สี่เหลี่ยม D โดยใช้ค่า Integral image จำนวน 4 จุด ผลลัพธ์ที่ได้คือ 4+1-(2+3) ดังแสดงในภาพที่ 2.9 ซึ่งจะเห็นได้ว่า ขนาดของกรอบสี่เหลี่ยมจะมีขนาดเท่าใดก็จะใช้เวลาในการคํานวณค่าผลรวมความเข้มคงที่

****

1. การแบ่งช่องแรงเงา สำหรับ**แบบจำลอง Haar-like**



(ข) ตัวอย่างการคำนวณ Integral image

**ภาพที่ 2.9** ตัวอย่างการจำลองรูปแบบ Haar-like

จากภาพที่ 2.9 (ก) เป็นการการหาค่าส่วนต่าง integral ที่จัดกลุ่มกันเป็นสัดส่วนแบ่ง 2ส่วน อย่างเห็นได้ชัดเพื่อใช้ในการจำแนกเป็นตำแหน่งดวงตา คิว จมูก แก้ม ต่อไป (ข) จากภาพที่ 2.10 แสดงค่าผลรวมภาพ Integral image ที่ได้จากค่าของแถวในช่อง *x1, y1, x2* และ *y2* โดยทำการรวมภาพในสมการดังสมการด้านบน คือเงื่อนไขการอินทรีกรอลภาพเพื่อใช้การหาค่า *(x1, y1), (x1, y2),* *(x2, y1)* และ *(x2, y2)* ตามลำดับ

**2.4.2. ขั้นตอนการเรียนรูปด้วยวิธี Adaboost**

เทคนิคการตรวจจับใบหน้าของ Viola-Jones ในขั้นตอนนี้จะนําการจําลองรูปแบบ Haar-like ที่ได้จากขั้นตอนแรกมาเข้ากระบวนการเรียนรูปของเครื่อง (Machine Learning) ที่เรียกว่า “Adaptive Boost” หรือ “Adaboost” ซึ่งกระบวนการนี้ใช้การเรงหาตัวจําแนก แบบอ่อนแอ และกำหนดค่าถ่วงน้ำหนักที่ทำให้ค่าผิดพลาดน้อยที่สุดในแต่ละรอบของกระบวนการ เพื่อสร้างตัวจําแนกแบบแข็งแรง (Strong Classifier) ขั้นตอนการเรียนรูปด้วย Adaboost มีดังต่อไปนี้

2.4.2.1 พิจารณาเลือกกลุ่มภาพตัวอย่างจำนวน nภาพ เพื่อใช้ในเรียงรูป กลุ่มตัวอย่างภาพสามารถเขียนได้ดังสมการข้างล่าง

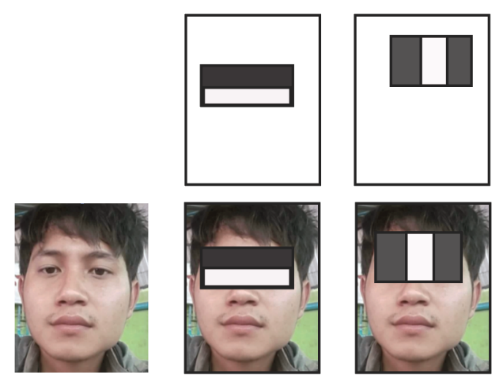
 (6)

โดยกำหนดให้

เมื่อ  คือ ภาพตัวอย่างที่ **i** และ  คือ ค่าที่ระบุวาดภาพตัวอย่างภาพที่เป็นใบหน้าคน 

หรือไม่ใชใบหน้าคน 

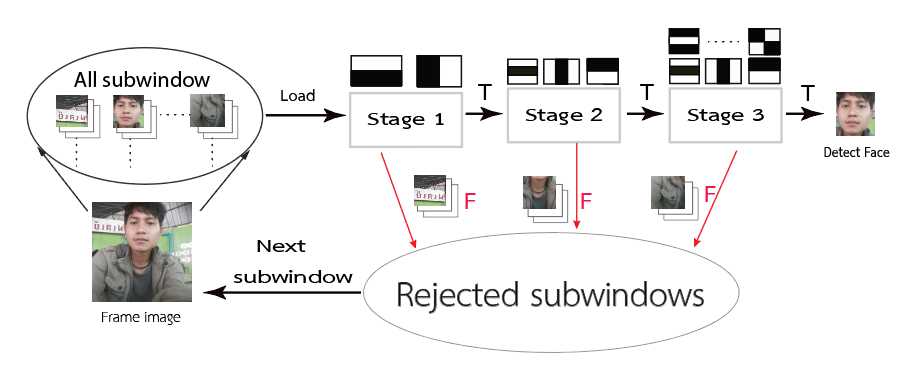
*ภาพที่ 2.10 แสดงตัวอย่างของการจำลองรูปแบบ* Haar-like *สองลำดับแรกที่ได้จากกระบวนการเรียนรู้ด้วยวิธี* Adahoost *จากภาพที่* 2.1*0 สังเกตเห็นว่าการจำลองรูปแบบ* Haar-like *แรกเป็นการจำลองรูปแบบ* Haar-like *ที่ได้จากความแตกต่างบริเวณดวงตาซึ่งจะมืดกว่าบริเวณส่วนบนของโหนกแก้ม สำหรับการจำลองรูปแบบ* Harr-like *ตัวที่สองได้จากความแตกต่างบริเวณดวงตาทั้งสองข้างซึ่งจะมืดกว่าบริเวณสันจมูก*

**

**ภาพที่ 2.10** แสดงตัวอย่างของตัวจำแนกแข็งแรงที่ได้จากกระบวนการเรียนรู้ด้วย Adaboots

**2.4.3 ขั้นตอนการรวมตัดจำแนกกลุ่มแบบต่อเรียง (Cascade structure for Haar classifiers)**

ขั้นตอนที่ 3 ที่ Viola-Jones นำเสนอเป็นเทคนิคที่ใช้ในการเพิ่มประสิทธิภาพในการตรวจจับให้มีความถูกต้อง และยังใช้เวลาในการคำนวณลดลงด้วย โดยแนวความคิดของขั้นตอนนี้ คือ การสร้างการรวมตัว จำแนกกลุ่มแบบต่อเรียง (Cascaded Classifier) โดยเทคนิคนี้จำนตัวจำแนก (Classifier) หลายๆ ตัวต่อกัน เป็นลำดับ ดังแสดงในภาพที่ 2.11 โดยที่ตัวจำแนกในลำดับต้นๆ มักจะมีความซับซ้อนที่น้อยกว่า อาจสร้างมาจากตัวจำแนกแบบอ่อนแอไม่กี่ตัวเพื่อลดประมาณการคำนวณและลดระดับของค่าแบ่งขีด (Threshold) ลงเพื่อเพิ่มลดการความผิดพลาดในการตรวจจับไม่พบใบหน้า ซึ่งภาพย่อยที่ถูกจัดประเภทเป็นไม่ใช่ใบหน้า (Non-face) จะถูกส่งต่อไปยังตัวจำแนกว่ามีโอกาสเป็นภาพใบหน้า จะถูกส่งต่อไปยังตัวแนกตัวถัดไปซึ่งมีความซับซ้อนสูงขึ้นไปตามลำดับ และกล่วาได้ว่ายิ่งมีจำนวนชั้นของตัวจำแนกมากเท่าใด โอกาสที่ภาพย่อยจะเป็นใบหน้าจะยิ่งมีมากขึ้น



**ภาพที่ 2.11** แสดงขั้นตอนการรวมตัวจำแนกกลุ่มแบบต่อเรียง

จากภาพที่ 2.11 เป็นภาพการทำงานของ Cascade structure for Haar classifiers ในการจัดเรียงผ่านไปตามขั้นตอนหากทำการตรวจสอบไม่พบลักษณะตามเงื่อนไขก็จะทำการปฏิเสธ ไม่พบใบหน้าตั้งแต่เริ่มต้นในทันที ไม่ต้องตรวจสอบทั้งหมด

**2.5. OpenVC ในไลบรารี่ Haar Cascade**

OpenCV เป็นซอฟต์แวร์โอเพนซอร์ซที่เอาไว้ใช้จัดการรูปภาพเริ่มพัฒนาขึ้นโดยบริษัทอินเทล (intel) ในปี 1999 คำว่า OpenCV ย่อมาจาก open source computer vision ขอบเขตการใช้งานของ OpenCV ค่อนข้างกว้าง มีความสามารถหลากหลายหลากหลาย นอกจากรูปภาพธรรมดาแล้วยังใช้จัดการกับวีดีโอภาพเคลื่อนไหวอัลกอริทึมที่ใช้มีตั้งแต่แบบง่ายๆไปจนถึงระดับสูงซึ่งรวมถึงการใช้เทคนิคการเรียนรู้ของเครื่อง ซึ่งในที่นี้ได้นำหลักการ haar cascadeมาประยุกต์ใช้งานกับ OpenCV ซึ่งได้ถูกสร้างขึ้นเพื่อทำงานทางวิทยาการคอมพิวเตอร์และการมองเห็นที่เกี่ยวข้องกับการประมวลผลภาพโดยเน้นไปที่การวิเคราะห์ ประมวลผลวัตถุลักษณะที่เป็นใบหน้าบุคคล โดยหลักการทำงานของ OpenCV มีดังนี้

2.5.1 เลเยอร์คอนโวลูชัน (Convolutional Layer) เป็นหลักการที่สำคัญในการประมวลผลภาพที่ใช้เครื่องกรอง (Filter) เพื่อหาคุณลักษณะที่สำคัญในภาพ เช่น การตรวจจับขอบ (Edge detection) หรือการทำความเข้มของภาพ (Sharpening)

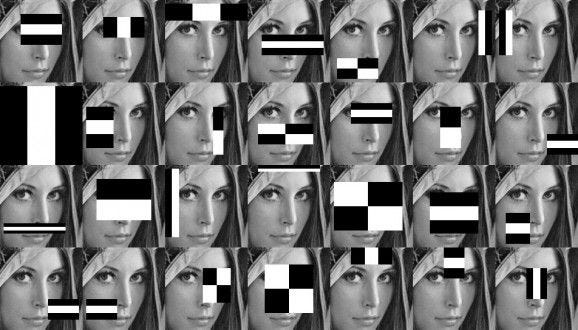
2.5.2 การทำสัญญาณรบกวน (Noise Reduction) การลดระดับของสัญญาณรบกวนในภาพ เพื่อปรับปรุงคุณภาพและเพิ่มความชัดเจนของภาพ ซึ่งสามารถทำได้โดยใช้เทคนิคต่างๆ เช่น การกรองเซพเพิล (Gaussian filtering) หรือการกรองมังกรี (Median filtering)

2.5.3 การแยกส่วน (Segmentation) การแบ่งภาพออกเป็นส่วนๆ หรือการจำแนกวัตถุที่มีความเชื่อถือได้ในภาพ ซึ่งสามารถใช้เพื่อวิเคราะห์วัตถุแต่ละอย่างในภาพ หรือใช้เพื่อทำการตรวจจับวัตถุ

2.5.4 การหาเส้นขอบ (Edge Detection) การค้นหาเส้นขอบหรือขอบเขตของวัตถุในภาพ เพื่อใช้เป็นข้อมูลสำคัญในการวิเคราะห์หรือประมวลผลต่อไป

2.5.5 การทำฟีเจอร์ (Feature Extraction) การหาคุณลักษณะเฉพาะ (Feature) ในภาพที่สามารถใช้เพื่อระบุหรือแยกแยะวัตถุหรือลักษณะต่างๆ ของภาพ เช่น การใช้เทคนิคที่เกี่ยวกับการหาจุดเด่น (key point) หรือการหาเวกเตอร์ลักษณะ (Feature vector)

2.5.6 การเรียนรู้เชิงลึก (Deep Learning) การใช้โครงข่ายประสาทเทียมเพื่อการวิเคราะห์และประมวลผลภาพทีซับซ้อน ทำให้สามารถทำงานได้อย่างมีประสิทธิภาพและแม่นยำมากขึ้น

2.5.7 การจัดการภาพ (Image Manipulation) การปรับแต่งหรือปรับเปลี่ยนคุณลักษณะต่างๆ ของภาพ เช่น การเปลี่ยนขนาดภาพ การปรับแต่งสี หรือการลบวัตถุที่ไม่ต้องการออกจากภาพ และนำไปใช้ในการวิเคราะห์วิดีโอ (Video Analysis) การประมวลผลภาพที่เกี่ยวข้องกับการสร้างและวิเคราะห์วิดีโอ เช่น การตรวจจับวัตถุในวิดีโอ การติดตามวัตถุ หรือการคัดแยกภาพเฟรม

**ภาพที่ 2.12** แสดงการตรวจจับใบหน้าของ OpenCV Haar Cascade

จากภาพที่ 2.12 แสดงการทำงานของ OpenCV ที่ได้นำแนวคิดของ Haar Cascade นำมาประยุกต์ใช้ในทางซอฟต์แวร์เพื่อให้สามารถนำมาตรวจจับใบหน้าได้อย่างมีประสิทธิภาพ ตามแนวคิดทฤษฎี Haar Cascade ที่ตรวจจับใบหน้าได้รวดเร็ว และไม่สิ้นเปลืองทรัพยากรหากไม่มีการค้นพบจุดสำคัญก่อน เลื่อนไปยังช่องต่อไป