การเปรียบเทียบประสิทธิภาพของอัลกอริทึมสำหรับการรู้จำใบหน้าบนคอมพิวเตอร์แบบราสเบอร์รี่พาย

Performance Comparison of Face Recognition Algorithms on Raspberry Pi 3 Model B+

เกรียงไกร ลิ่มทอง และ ญาใจ เจริญสุข

ภาควิชาวิศวกรรมคอมพิวเตอร์และหุ่นยนต์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยกรุงเทพ kriangkrai.l@bu.ac.th

บทคัดย่อ

การเลือกอัลกอริทึมที่เหมาะสมสำหรับระบบตรวจจับและรู้จำใบหน้าเป็นปัญหาใหญ่สำหรับการนำไปประยุกต์ใช้ในเครื่องคอมพิวเตอร์ขนาดเล็ก เช่น ราสเบอร์รี่พาย เนื่องจากคอมพิวเตอร์ขนาดเล็กดังกล่าวมีข้อจำกัดหลายด้าน ข้อจำกัดหลักคือหน่วยการประมวลผลและขนาดของหน่วยความจำ จึงทำให้อัลกอริทึมที่เลือกนำมาประยุกต์ใช้งานมีผลกระทบต่อประสิทธิภาพการทำงานโดยรวมของระบบ โดยเฉพาะอย่างยิ่งในระบบที่มีการตรวจจับและรู้จำใบหน้าบนวิดีโอแบบเรียลไทม์ ในบทความนี้เราจึงได้ทำการเปรียบเทียบประสิทธิภาพของอัลกอริทึมที่ใช้ในการตรวจจับและรู้จำใบหน้าที่มักนำมาใช้ประมวลผลบนวิดีโอแบบเรียลไทม์ในเครื่องคอมพิวเตอร์แบบราสเบอร์รี่พาย ซึ่งได้แก่อัลกอริทึม Eigenfaces, Fisherfaces, และ Local Binary Pattern Histogram (LBPH) ผลที่ได้จากการทดลองในบทความนี้จะช่วยให้ผู้พัฒนาระบบตรวจจับและรู้จำใบหน้าสามารถเลือกอัลกอริทึมที่เหมาะสมกับการนำไปประยุกต์ใช้งานบนเครื่องคอมพิวเตอร์ขนาดเล็กที่มีข้อจำกัดเรื่องหน่วยประมวลผลและหน่วยความจำ

คำสำคัญ: ไอเกนเฟซ, ฟิชเชอร์เฟซ, โลคอลไบนารีแพทเทิร์นฮิสโตแกรม, อัลกอริทึมตรวจจับและรู้จำใบหน้า, ราสเบอร์รี่พาย

Abstract

Deciding for the appropriate algorithm is one of the key issues in face recognition systems, especially for an embedded system as the Raspberry Pi. As we know, the Raspberry Pi functions as a small computer that contains limits of computation and memory capacity. Therefore, face recognition algorithms mainly affect operational efficiency of overall system, especially in real-time video detection. In this paper, we evaluated efficiency of three famous algorithms in both training and test phases. The face recognition algorithms are Eigenfaces, Fisherfaces, and Local Binary Patterns Histogram (LBPH). Our experimental results of these three algorithms will help you to select the face recognition algorithm, which eminently suitable for your system on a Raspberry Pi.

Keywords: Eigenfaces, Fisherfaces, Local Binary Pattern Histogram, Face Recognition Algorithms, Raspberry Pi

1. บทนำ

การรู้จำใบหน้าเป็นเทคนิคที่ใช้สำหรับการตรวจจับและระบุยืนยันตัวบุคคลจากภาพถ่ายหรือจากภาพจากกล้องวิดีโอที่บันทึกแบบเรียลไทม์ เทคโนโลยีดังกล่าวนี้ได้มีการพัฒนาและนำไปประยุกต์ใช้งานอย่างกว้างขวางในช่วงระยะเวลาไม่กี่ปีที่ผ่านมา การรู้จำใบหน้าได้ถูกนำไปใช้ในธุรกิจต่างๆ เช่น การชำระเงิน, การเข้าใช้งานและความมั่นคงปลอดภัย, การระบุประวัติอาชญากร, การโฆษณา, และด้านสาธารณสุข เป็นต้น ในปี 2016 บริษัท MasterCard ได้พัฒนาระบบการชำระเงินด้วยการถ่ายภาพตนเอง (selfie) ที่เรียกว่า MasterCard Identify Check โดยลูกค้าจะต้องเปิดโปรแกรมดังกล่าวเพื่อยืนยันการชำระเงินผ่านกล้องของสมาร์ตโฟน ในอนาคตอันใกล้เราอาจจะเห็นผู้คนเปิดล็อครถยนต์ ประตูบ้าน หรือการเข้าใช้งานในสถานที่ต่างๆ โดยใช้เพียงแค่การมองไปยังอุปกรณ์ดังกล่าว นอกจากนี้ระบบดังกล่าวยังสามารถป้องกันบุคคลที่ไม่พึงประสงค์ได้ เช่นเดียวกัน ตัวอย่างเช่นสำนักงานสอบสวนกลางสหรัฐได้มีการนำระบบรู้จำใบหน้ามาตรวจหาบุคคลที่น่าสงสัยในการก่ออาชญากรรมจากรูปถ่ายในใบขับขี่

Internet of Things (IoT) หรือ อินเตอร์เน็ตในทุกสิ่ง หมายถึงอุปกรณ์ต่างๆ สิ่งต่างๆ ได้ถูกเชื่อมต่อกับอินเตอร์เน็ตและเป็นอุปกรณ์หนึ่งที่ถูกนำมาใช้กับการรู้จำใบหน้าในช่วงไม่กี่ปีที่ผ่านมา อุปกรณ์ IoT รูปแบบต่างๆ ได้เพิ่มขึ้นและนำมาใช้งานเป็นจำนวนมาก เช่น เครื่องใช้ไฟฟ้าภายในบ้าน, อุปกรณ์สวมใส่ คุณสมบัติที่สำคัญของ IoT คือ ใช้พลังงานจากแบตเตอร์รี่และมีทรัพยากรณ์สำหรับการประมวลผลอย่างจำกัด ซึ่งหมายความว่ายากที่เราจะนำอัลกอริทึมซับซ้อน เช่น deep learning network หรือ convolutional neural network มาใช้กับการรู้จำใบหน้าที่ติดตั้งบนอุปกรณ์ IoT ดังนั้นเราจึงจำเป็นต้องใช้อัลกอริทึมการรู้จำใบหน้าที่มีขนาดเล็กและประมวลผลได้เร็วสำหรับอุปกรณ์ IoT ตัวอย่างอุปกรณ์ IoT ที่นำการรู้จำใบหน้ามาใช้คือ Raspberry Pi ซึ่งเป็นคอมพิวเตอร์ที่มีขนาดเล็กมาก, ราคาถูก, และมีความสามารถในการประมวลผลการรู้จำใบหน้า โดยเฉพาะอย่างยิ่ง Raspberry Pi 4 ซึ่งมีการปรับปรุงพัฒนาประสิทธภาพในการประมวลผลและมีทางเลือกในการจัดเก็บข้อมูลใน SD RAM ที่มีความจุถึง 8GB แต่ก็ยังไม่เพียงพอที่จะประยุกต์ใช้อัลกอริทึมรู้จำใบหน้าที่ซับซ้อนสำหรับการรู้จำใบหน้าจากวิดีโอแบบเรียลไทม์ ดังนั้นเราจึงมีความจำเป็นที่ต้องเปรียบเทียบประสิทธิภาพของอัลกอริทึมรู้จำใบหน้าที่ เพื่อเลือกให้ทำงานเหมาะสมกับอุปกรณ์ IoT ที่มีข้อจำกัดอย่าง Raspberry Pi เป็นต้น

ในงานวิจัยนี้เราได้ทำการทดลองการรู้จำใบหน้าโดยใช้ทั้งหมด 3 อัลกอริทึมที่ประมวลผลเร็วและเหมาะสมกับอุปกรณ์ IoT นั่นคือ Eigenfaces, Fisherfaces, และ Local Binary Patterns Histogram (LBPH) อุปกรณ์ IoT ที่ใช้คือ Raspberry Pi 3 model B+ ที่เชื่อมต่อกับ Raspberry Pi camera module สำหรับการเก็บข้อมูลใบหน้าและการตรวจจับใบหน้าจากข้อมูลวิดีโอแบบเรียลไทม์ ภาษาโปรแกรมที่ใช้คือ Python ร่วมกับ Open CV และ DLib ซึ่งเป็นไลบรารีที่สำคัญสำหรับการประมวลผลภาพและวิดีโอ ในการทดลองเราได้ทำการวัดช่วงเวลาในการประมวลผลทั้งขั้นตอนเรียนรู้ (training) และขั้นตอนการทดสอบ (test) หลังจากนั้นจึงทำการเปรียบเทียบความแม่นยำของทั้ง 3 อัลกอริทึมดังกล่าว

2. งานวิจัยที่เกี่ยวข้อง

ขั้นตอนของการรู้จำใบหน้าสามารถแบ่งออกได้เป็น 2 ขั้นตอนหลัก คือ การตรวจจับใบหน้า (detection) และการรู้จำใบหน้า (recognition) ในหัวข้อนี้จะกล่าวถึงอัลกอริทึม Haar cascade ซึ่งถูกนำไปใช้ในขั้นตอนการตรวจจับใบหน้า ในขั้นตอนการรู้จำใบหน้าเราจะใช้อัลกอริทึม Eigenfaces, Fisherfaces, และ LBPH

2.1 Haar cascade algorithm

อัลกอริทึม Haar cascade [1] ถูกนำไปเป็นพื้นฐานของเทคนิคการตรวจจับใบหน้าของ Viola-Jones ซึ่งเป็นการนำภาพใบหน้ามาแบ่งเป็นภาพย่อย (sub-window) กระบวนการตรวจหาใบหน้าโดยเทคนิคทั่วไปจะทำการปรับขนาดของภาพย่อยแตกต่างกันหลายๆ ขนาด และใช้ตัวตรวจจับ (detector) ที่มีขนาดคงที่ ซึ่งการตรวจจับแบบนี้จะใช้ระยะเวลาในการคำนวณไม่คงที่ ดังนั้น Viola-Jones จึงเสนอเทคนิคการตรวจจับใบหน้าแบบใหม่ โดยการใช้การจำลองรูปแบบ Haar-like เป็นตัวตรวจจับ และทำการปรับขนาดของตัวตรวจจับแทนการปรับขนาดของภาพย่อย นอกจากนี้ยังใช้ตัวตรวจจับทำการตรวจจับใบหน้าหลายๆ รอบ ในแต่ละรอบจะใช้ขนาดของตัวตรวจจับที่แตกต่างกันอีกด้วย เมื่อนำมาเปรียบเทียบกับวิธีเดิมพบว่าเวลาที่ใช้ในการคำนวณไม่มีความแตกต่างกันมากนัก และพบว่าวิธีดังกล่าวใช้เวลาในการคำนวณการตรวจจับภาพใบหน้าแต่ละรอบมีค่าคงที่ ถึงแม้ขนาดของตัวตรวจจับจะแตกต่างกัน

เทคนิคนี้นำคุณลักษณะ (features) ที่ถูกคัดแยกโดย Adaboots มาทำการคำนวณอีกครั้ง โดยในขั้นตอนแรกจะใช้ features ไม่มากนักในการคำนวณ หากขั้นตอนใดถูกพิจารณา sub-windows แล้วไม่ผ่านก็จะทำการทิ้ง sub-windows นั้นไปแล้วไม่นำ features ที่เหลือมาพิจารณา ตัวตรวจจับในลำดับแรกๆ มักจะมีความซับซ้อนที่น้อยกว่า อาจสร้างมาจากตัวตรวจจับไม่กี่ตัวเพื่อลดปริมาณการคำนวณและลดระดับของค่าที่กำหนด (threshold) ลง ทำให้ช่วยลดการความผิดพลาดในการตรวจจับไม่พบใบหน้า ซึ่งเมื่อคำนวณแล้วไม่ใช่ใบหน้าจะถูกทิ้งไปเป็นจำนวนมากในลำดับแรกๆ แต่ถ้าภาพย่อยนั้นถูกจำแนกว่ามีโอกาสเป็นภาพใบหน้าก็จะถูกส่งต่อไปยังตัวตรวจจับถัดไปซึ่งมีความซับซ้อนสูงขึ้นตามลำดับ อาจบอกได้ว่ายิ่งใช้ตัวตรวจจับขั้นสูงมากเท่าไหร่ ยิ่งมีโอกาสที่ภาพย่อยนั้นจะเป็นภาพใบหน้าได้มากขึ้น

2.2 Eigenfaces algorithm

Eigenfaces Recognition [2] ใช้การวิเคราะห์องค์ประกอบหลักของข้อมูล (principal component analysis: PCA) เพื่อลดขนาดแมทริกซ์ของตัวแปรหรือใช้หาความสัมพันธ์ของข้อมูล แบ่งการทำงานออกเป็น 2 ขั้นตอน คือ ขั้นตอนก่อนการประมวลผล (preprocess) และขั้นตอนทดสอบ (test) ในขั้นตอนก่อนการประมวลผลจะเป็นการนำชุดภาพข้อมูลมาทำการเรียนรู้หา features ของภาพดังกล่าว และในขั้นตอนการทดสอบจะทำการคำนวณหาค่า Eigenface ของรูปภาพที่ต้องการทดสอบ ถึงแม้ว่าอัลกอริทึมนี้เหมาะสำหรับการตรวจจับใบหน้าจากรูปภาพหน้าตรง แต่ในช่วงเวลาหลายปีที่ผ่านมานักวิจัยได้ทำการเสนอเทคนิคต่างๆ ที่นำมาช่วยให้ Eigenfaces สามารถรู้จำจากภาพใบหน้าจากหลายๆ มุมได้ และหวังว่าจะพัฒนาอัลกอริทึมดังกล่าวให้มีประสิทธิภาพและประสิทธิผลมากขึ้นในการรู้จำใบหน้าในเวลาอันใกล้นี้

อัลกอริทึมนี้มีข้อดีเมื่อเปรียบเทียบกับอัลกอริทึมอื่นๆ คือ เรื่องความเร็วและประสิทธิภาพ เพราะพื้นฐานการทำงานเป็นการลดมิติของข้อมูลทำให้ระบบสามารถจัดเก็บข้อมูลใบหน้าได้เป็นจำนวนมากแต่ใช้พื้นที่ในการจัดเก็บข้อมูลน้อยและมีขนาดไม่เปลี่ยนแปลงมากแม้จะเพิ่มจำนวนข้อมูลใบหน้าอีกเป็นจำนวนมาก อย่างไรก็ดี Eigenfaces มีความไวต่อแสง, ขนาด, และการหมุน ดังนั้นถ้าจะนำไปใช้จำเป็นต้องมีการควบคุมสภาพแวดล้อมที่มีผลต่อข้อมูลใบหน้า อีกประการหนึ่งคือไม่สามารถตรวจจับการเปลี่ยนแปลงของสีหน้าได้ ดังนั้นนี่คือสิ่งสำคัญที่จะต้องนำไปพิจารณาเลือกใช้งาน

2.3 Fisherfaces algorithm

อัลกอริทึม Fisherfaces [3] ได้ถูกพัฒนามาจาก Eigenfaces และเป็นอีกหนึ่งวิธีที่มีหลักการทำงานคล้ายกันกับอัลกอริทึม Eigenfaces แต่จะใช้การวิเคราะห์จำแนกประเภทเชิงเส้น (linear discrimination analysis) ในการรู้จำใบหน้า ซึ่งทำให้วิธีนี้มีความไวต่อแสงและมุมของภาพใบหน้าน้อยกว่า Eigenfaces ระบบรู้จำใบหน้าที่ใช้ Fisherfaces จำเป็นต้องใช้จำนวนใบหน้าสำหรับการเรียนรู้จำนวนมากในขั้นตอนการลดมิติของข้อมูล โดยอัลกอริทึมนี้มีการประยุกต์ใช้ PCA ต่อด้วยการใช้ Fisher's linear discriminant (FLD) หรือที่บางคนรู้จักและเรียกกันในชื่อ linear discriminant analysis (LDA) นอกจากนี้ยังมีการประยุกต์ใช้ค่าต่ำสุดของ Euclidean ในการเปรียบเทียบระบุตัวตนจากภาพใบหน้า อย่างไรก็ดี Fisherfaces ยังมีปัญหาเรื่องเวลาที่ใช้ในการประมวลผลค่อนข้างนานและปัญหาเรื่องเงื่อนไขของภาพใบหน้าที่จะถูกนำมาใช้ในขั้นตอนการทดสอบ เนื่องจาก Fisherfaces มีขั้นตอนในการประมวลผลมีความยุ่งยากและมีความซับซ้อน และเมื่อจำนวนใบหน้าในการรู้จำมากขึ้นก็จะทำให้ยิ่งใช้เวลานานในการประมวลผลในขั้นตอนการตรวจจับ ภาพรวมของ Fisherfaces จึงไม่ได้มีข้อได้เปรียบมากไปกว่าอัลกอริทึม Eigenfaces

2.4 Local Binary Pattern Histogram (LBPH)

หลักการทำงานของอัลกอริทึม LBPH แบบดั้งเดิม [4] คือทำการติดป้ายในจุดของรูปด้วยตัวเลขที่เรียกกันว่ารหัส LBP ซึ่งใช้การเข้ารหัสภายในรอบๆ จุดของรูป ตัวอย่างเช่นมีการเปรียบเทียบทั้งหมด 8 ค่าจากจำนวนพิเซลขนาด 3x3 โดยนำค่าตัวเลขในแต่ละจุดเปรียบเทียบกับค่าตัวเลขรอบจุดดังกล่าวด้วยการลบกัน ถ้าได้ค่าลบเข้ารหัสด้วย 0, ค่าบวกเข้ารหัสด้วย 1 ทำให้ได้ค่าตัวเลขฐาน 2 จากการนำรหัสไบนารีมาเรียงกันตามเข็มนาฬิกาเริ่มจากมุมบนซ้ายแล้วแปลงเป็นเลขฐาน 10 การทำงานดังกล่าวเรียกว่า Local Binary Patterns (LBP)

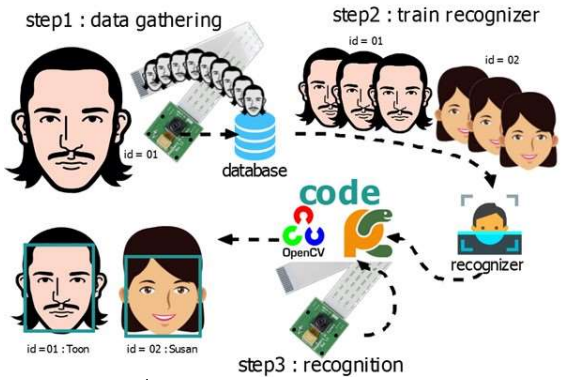
ข้อจำกัดของ LBP คือเป็นการทำงานด้วยจำนวนพิกเซลขนาด 3x3 ซึ่งมีขนาดเล็กมาก ไม่สามารถตรวจจับคุณลักษณะหลักได้ในกรณีที่รูปภาพมีโครงสร้างขนาดใหญ่ จึงจำเป็นต้องมีการปรับเปลี่ยนใช้ตัวดำเนินการที่มีขนาดเปลี่ยนไปตามขนาดของโครงสร้างและลวดลายของรูปภาพ โดยตัวดำเนินการคือจำนวนจุดรอบๆ ของจุดที่ต้องการติดป้ายนั่นเอง

3. หลักการทำงานและขั้นตอนการทดลอง

หัวข้อนี้เรานำเสนอภาพการทำงานโดยรวมของระบบ, การเก็บข้อมูลใบหน้าจากผู้ใช้งานจริงภายในมหาวิทยาลัย, และขั้นตอนทดลองหาประสิทธิภาพทั้งในส่วนของการเรียนรู้และส่วนของการทดสอบ

3.1 ระบบรู้จำใบหน้า

เราได้พัฒนาระบบรู้จำใบหน้าด้วยภาษา Python ร่วมกับ OpenCV ซึ่งเป็นไลบรารีสำหรับงานรับภาพของคอมพิวเตอร์ โดยนำไปติดตั้งใน Raspberry Pi 3 model B+ ที่มีกล้อง Pi camera module สำหรับการถ่ายภาพและวิดีโอ ระบบดังกล่าวสามารถแบ่งออกได้ 3 ขั้นตอนคือ การรวบรวมข้อมูลใบหน้า (data gathering), การฝึกสอนตัวรู้จำ (train recognize), และการระบุตัวบุคคล (recognition) ดังแสดงในรูปที่ 1



รูปที่ 1 การทำงานโดยรวมของระบบรู้จำใบหน้า

ในขั้นตอน 1 การรวบรวมข้อมูลใบหน้า เราได้เก็บภาพข้อมูลใบหน้าจริงของนักศึกษาจำนวน 41 คนและบันทึกลงไปในระบบไฟล์ของ Raspberry Pi เราบันทึกรูปภาพใบหน้าจากโปรแกรมรับภาพวิดีโอหลังจากนั้นก็กำหนดหมายเลขจากรหัสนักศึกษาแต่ละคน ระยะห่างจากบุคคลกับกล้องประมาณ 1 เมตรในสภาพแวดล้อมที่มีการควบคุมความเข้มของแสงและภาพพื้นหลัง ซึ่งใช้เวลาน้อยกว่า 1 นาทีสำหรับการบันทึกรูปภาพใบหน้าของแต่ละคนจำนวน 100 รูปและอยู่ในรูปแบบ jpeg โดยแยกรูปภาพของแต่ละคนในโฟลเดอร์ที่ต่างกัน ในโปรแกรมของเราสามารถกำหนดได้ว่าจะเก็บรูปภาพใบหน้าเป็นจำนวนกี่รูปโดยทำการแก้ไขในรหัสต้นฉบับของโปรแกรมดังกล่าว

ขั้นตอนถัดมา การฝึกสอนตัวรู้จำ เราได้ฝึกสอนให้โปรแกรมเรียนรู้และจดจำใบหน้าโดยใช้อัลกอริทึมทั้ง 3 แบบเพื่อนำมาเปรียบเทียบกันในเชิงความถูกต้อง (accuracy) ซึ่งอัลกอริทึมดังกล่าวมีอยู่ในชุดคำสั่งของ OpenCV โดยจะนำข้อมูลใบหน้าจากขั้นตอนที่ 1 มาใช้ฝึกสอน (training) ขั้นตอนนี้ใช้เวลาน้อยกว่า 1 นาทีต่อการฝึกสอนให้โปรแกรมจดจำใบหน้าจำนวน 1 คน

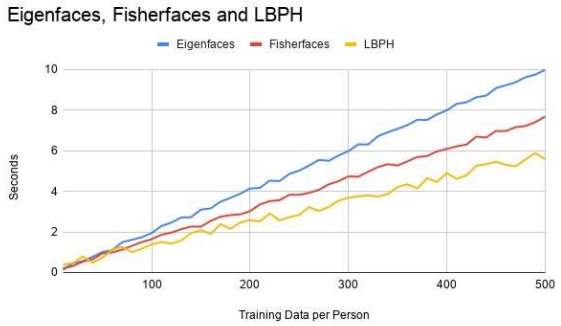
ขั้นตอนสุดท้าย การระบุตัวบุคคล เรียกใช้งานโปรแกรมเฉพาะสำหรับการตรวจจับและระบุตัวบุคคล ในขั้นตอนนี้โปรแกรมจะถูกเรียกใช้งานและทำงานตลอดเวลา เมื่อมีคนเดินเข้ามาภายในระยะที่กล้องสามารถตรวจจับได้ โปรแกรมทำการเรียกอัลกอริทึม Haar cascade เพื่อตรวจจับตำแหน่งของใบหน้าจากเฟรมของวิดีโอ จากนั้นจึงเรียกใช้อัลกอริทึมเดียวกันกับการทำงานในขั้นตอนที่ 2 เพื่อระบุตัวบุคคลว่าตรงกันกับฐานข้อมูลที่ได้จัดเก็บในขั้นตอนที่ 2 หรือไม่ ถ้าตรงกันก็จะทำการบันทึกเวลาเข้า-ออกของบุคคลดังกล่าวและจัดเก็บลงในฐานข้อมูล MySQL แต่ถ้าไม่ตรงกันก็จะทำการเก็บภาพข้อมูลใบหน้าของบุคคลดังกล่าวพร้อมทั้งบันทึกเวลาเข้า-ออกลงในฐานข้อมูลเช่นกัน

3.2 ขั้นตอนการทดลอง

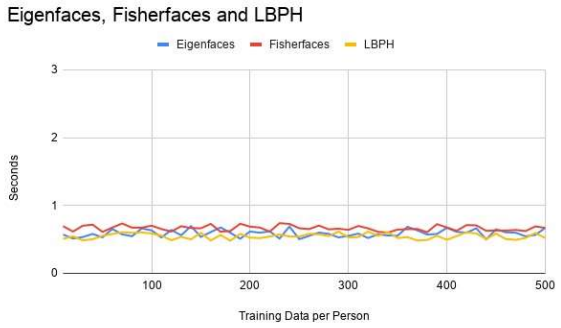
เราสามารถแบ่งการทดลองออกเป็น 3 ส่วนหลักคือ การวัดช่วงเวลาการฝึกสอนตัวรู้จำ, การวัดช่วงเวลาการระบุตัวบุคคล, และการวัดความแม่นยำในการรู้จำใบหน้า

ในการทดลองวัดช่วงเวลาการฝึกสอนตัวรู้จำ เราจะทำการดัดแปลงรูปภาพใบหน้าที่ได้จากขั้นตอนรวบรวมข้อมูลใบหน้า 100 ภาพต่อ 1 คน เพื่อสร้างรูปภาพประดิษฐ์ให้ได้ 500 ภาพต่อ 1 คน โดยใช้การเลื่อนภาพ (transition), การหมุนภาพ (rotation), การปรับความเข้มแสง (contrast), การเพิ่มตัวกรอง (filter) เป็นต้น จากนั้นจึงให้โปรแกรมฝึกสอนตัวรู้จำจาก 1 ภาพไปจนถึง 500 ภาพต่อ 1 คนและทำการวัดช่วงเวลาที่ใช้ในการฝึกสอน เมื่อเสร็จเรียบร้อยก็ทำการเปลี่ยนอัลกอริทึมและดำเนินการวัดช่วงเวลาที่ใช้ในการฝึกสอนใหม่จาก 1 จนถึง 500 ภาพ ทำเช่นนี้จนครบทั้ง 3 อัลกอริทึม

ในการทดลองวัดช่วงเวลาการระบุตัวบุคคล เราจะทำการวัดช่วงเวลาที่ใช้ตั้งแต่ระบบตรวจจับเจอใบหน้าในเฟรมของวิดีโอจนกระทั่งสามารถระบุตัวบุคคลได้ตรงกันกับฐานข้อมูลที่จัดเก็บในขั้นตอนการฝึกสอนตัวรู้จำ การทดลองในขั้นตอนนี้จำเป็นต้องใช้อัลกอริทึมเดียวกันกับอัลกอริทึมที่ใช้ในขั้นตอนการฝึกสอนตัวรู้จำ



รูปที่ 2 เปรียบเทียบเวลาที่ใช้ประมวลผลในส่วนการเรียนรู้ (training)



รูปที่ 3 เปรียบเทียบเวลาที่ใช้ประมวลผลในส่วนการทดสอบ (test)

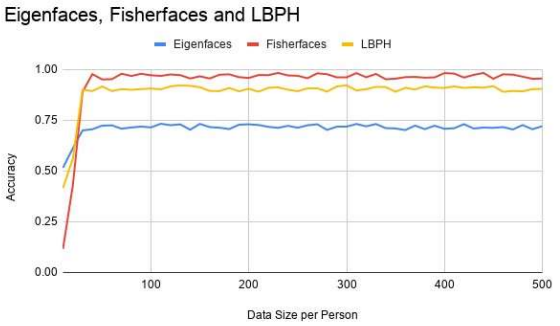
ในการทดลองวัดความแม่นยำในการรู้จำใบหน้า เราจะทำการทดลองระบุตัวบุคคลที่มีอยู่ในฐานข้อมูลทั้งหมด 41 คนและไม่มีอยู่ในฐานข้อมูลอีก 9 คน ถ้าระบุตัวบุคคลในฐานข้อมูลถูกต้องนับเป็น true positive (TP), ถ้าระบุตัวบุคคลไม่อยู่ในฐานข้อมูลถูกต้องนับเป็น true negative (TN), ถ้าระบุตัวบุคคลในฐานข้อมูลไม่ถูกต้องนับเป็น false negative (FN), และถ้าระบุตัวบุคคลไม่อยู่ในฐานไม่ถูกต้องนับเป็น false positive (FP) และคำนวนแม่นยำ (ACC) จากสมการที่ (1)

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \qquad (1)$$

4. ผลการทดลอง

เราสามารถแสดงผลออกมาได้เป็น 3 รูปดังนี้ รูปที่ 2 และ 3 แกน x คือจำนวนรูปภาพใบหน้าต่อ 1 คนและแกน y คือเวลาที่ใช้ในการประมวลผล รูปที่ 4 แกน x คือจำนวนรูปภาพใบหน้าต่อ 1 คนและแกน y คือความแม่นยำในการระบุตัวบุคคล

รูปที่ 2 คือผลการทดลองวัดช่วงเวลาการฝึกสอนตัวรู้จำแสดงให้เห็นว่าถ้าจำนวนรูปภาพใบหน้าไม่เกิน 50 รูปต่อคนทั้ง 3 อัลกอริทึมใช้เวลาประมวลผลไม่ต่างกันมาก แต่ถ้าเกิน 50 รูปต่อคน LBPH ใช้เวลาในการประมวลผลน้อยที่สุด ตามด้วย Fisherfaces และ Eigenfaces ตามลำดับ ผลการทดลองวัดช่วงเวลาการระบุตัวบุคคลแสดงในรูปที่ 3 เห็นได้ชัดว่าทั้ง 3 อัลกอริทึมใช้เวลาในการประมวลผลเพื่อระบุตัวบุคคลไม่ต่างกัน และการใช้จำนวนรูปภาพใบหน้าต่อคนไม่มีผลต่อเวลาในการระบุตัวบุคคล ซึ่งทั้ง 3 อัลกอริทึมใช้เวลาน้อยกว่า 1 วินาที



รูปที่ 4 เปรียบเทียบความแม่นยำ (accuracy) ในการรู้จำใบหน้า

ผลการทดลองเรื่องความแม่นยำในการรู้จำใบหน้าแสดงในรูปที่ 4 เห็นชัดว่าจำนวนรูปภาพใบหน้าต่อ 1 คนมีผลต่อความแม่นยำในทุกอัลกอริทึมโดยเฉพาะระหว่าง 1 ถึง 30~40 รูปต่อคน ถ้ามากกว่า 30~40 รูปต่อคนจะไม่ค่อยมีผลกระทบต่อความแม่นยำมากนัก จากกราฟแสดงให้ทราบว่าถ้าใช้จำนวนรูปภาพใบหน้าต่อคนจำนวนน้อย (1~30 รูปต่อคน) Eigenfaces มีความแม่นยำมากที่สุด ตามด้วย LBPH และ Fisherfaces ตามลำดับ แต่ถ้าใช้จำนวนรูปภาพใบหน้าต่อคนจำนวนมาก (มากกว่า 30 รูปต่อคน) Fisherfaces จะมีความแม่นยำมากที่สุดตามด้วย LBPH และ Eigenfaces ตามลำดับ โดยที่ความแม่นยำระหว่าง Fisherfaces กับ LBPH มีความแตกต่างกันไม่มากนัก

5. สรุปและแนวทางการพัฒนาต่อไป

จากการทดลองทั้งหมดได้แก่ การวัดช่วงเวลาการฝึกสอนตัวรู้จำ, การวัดช่วงเวลาการระบุตัวบุคคล, และการวัดความแม่นยำในการรู้จำใบหน้า จากทั้ง 3 อัลกอริทึม Eigenfaces, Fisherfaces, และ LBPH แสดงให้เห็นว่าอัลกอริทึม LBPH ใช้เวลาในการประมวลผลน้อยกว่าและให้ความแม่นยำในระดับที่สามารถนำไปใช้งานจริงได้ เหมาะสำหรับอุปกรณ์ IoT ที่มีหน่วยประมวลผลกลางความเร็วจำกัดได้เป็นอย่างดี แต่ถ้าระบบต้องการความแม่นยำสูงและยอมรับกับการใช้เวลาประมวลผลนานในขั้นตอนการฝึกสอนตัวรู้จำได้ แนะนำให้ใช้ Fisherfaces เนื่องจากมีความแม่นยำสูงกว่า LBPH หากนำไปใช้การจริงแนะนำว่าควรทดสอบกับจำนวนคนที่มากกว่านี้ เนื่องจากการทดลองนี้ใช้เพียง 50 คนเท่านั้น

เอกสารอ้างอิง

[1] P. Viola and M. Jones., “Rapid object detection using a boosted cascade of simple features”, *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*., pp.511-518, 2001.

[2] L. Sirovich and M. Kirby., “Low-dimensional procedure for the characterization of human faces”, *Journal of the Optical Society of America A4*., pp.519-524, 1987

[3] Klecka and William R., *Discriminant analysis. Quantitative Applications in the Social Sciences Series*., No.19., Thousand Oaks, CA: Sage Publications., 1980

[4] DC. He and L. Wang., “Texture Unit, Texture Spectrum, And Texture Analysis”., *IEEE Transactions on Geoscience and Remote Sensing.*, vol.28, pp.509-512, 1990