JIST Journal of Information Science and Technology
Volume 13, NO 1 | JAN – JUN 2023 | 1-9
ISSN: 2651-1053 (Online)
ISSN: 1906-9553 (Print)

แอนดรอยด์แอปพลิเคชันสำหรับจดจำและตรวจสอบใบหน้าโดยใช้โมบายเฟซเน็ต

Android Application for Face Recognition and Verification using MobileFaceNets

เชี่ยวชาญ ยางศิลา

Chiawchan Yangsila

สาขาวิชาคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏร้อยเอ็ด

Computer and Information Technology Program, Faculty of Information Technology, Roi Et Rajabhat University

Received: November 11, 2022; Revised: April 18, 2023; Accepted: June 08, 2023; Published: June 30, 2023

ABSTRACT – This paper presents the development of Android application for face recognition and verification using MobileFaceNets: a case study of the activity attendance check of staff at Roi Et Rajabhat University (RERU). The four objectives and their results in the study are as follows. 1) The developed Android application for face recognition and verification using MobileFaceNets was applicable to the activity attendance check of the RERU staff. 2) The Manhattan function had the fastest processing speed. 3) The researcher did a distance analysis for 100% accuracy and found that image A was identical to image B if the distance was <= 0.752 (Euclidean distance) or <= 5.796 (Manhattan distance) or >= 0.717 (Cosine Coefficient). 4) The efficiency analysis revealed that the application processing was 100% accurate.

KEYWORDS: Android Application, Face Recognition and Verification, MobileFaceNets, TensorFlow Lite, Convolutional Neural Network

บทคัดย่อ -- **บทความนี้นำเสนอการพัฒนาแอนดรอยด์แอปพลิเคชันสำหรับจดจำและตรวจสอบใบหน้าโดยใช้โมบายเฟซเน็ต กรณีศึกษา การเช็คชื่อเข้าร่วมกิจกรรมมหาวิทยาลัยราชภัฏร้อยเอ็ด ซึ่งมีวัตถุประสงค์ 4 ข้อ ขอสรุปผลการทดลองตามวัตถุประสงค์ดังนี้ 1) การพัฒนาแอนดรอยด์แอปพลิเคชันสำหรับจดจำและตรวจสอบใบหน้าโดยใช้โมบายเฟซเน็ต กรณีศึกษา การเช็คชื่อเข้าร่วมกิจกรรมมหาวิทยาลัยราชภัฏร้อยเอ็ดสามารถนำไปใช้งานได้จริง 2) การทดลอง พบว่า ฟังก์ชันแมนฮัตตัน มีความเร็วในการประมวลผลมากที่สุด 3) การวิเคราะห์ระยะห่างที่ให้คำตอบถูกต้อง 100% สรุปว่า ภาพ ก จะเหมือนกันกับภาพ ข ถ้าระยะห่าง <= 0.752 (ยูคลิเดียน) หรือ <= 5.796 (แมนฮัตตัน) หรือ >= 0.717 (สหสัมพันธ์โคไซน์) วิเคราะห์โดยใช้ผู้วิจัยเอง และ 4) ผลการหาประสิทธิภาพของแอปพลิเคชัน พบว่า แอปพลิเคชันประมวลผลถูกต้อง 100%**

คำสำคัญ: แอนดรอยด์แอปพลิเคชัน, จดจำและตรวจสอบใบหน้า, โมบายเฟซเน็ต, เทนเซอร์โฟลลิต, โครงข่ายประสาทเทียมแบบสังวัตนาการ

Corresponding Author: cheawchang2@gmail.com

1. บทนำ

โทรศัพท์มือถือแบบสมาร์ทโฟนได้มีบทบาทมากขึ้นในชีวิตประจำวันซึ่งโทรศัพท์มือถือนั้นมีประสิทธิภาพมากพอที่จะประมวลผลได้เทียบเท่ากับคอมพิวเตอร์ขนาดเล็ก ในขณะเดียวกันเทคโนโลยีการเรียนรู้ของเครื่อง (Machine Learning) ก็มีบทบาทสำคัญมากในการเข้ามาจำแนกชนิดหรือประเภทของวัตถุโดยการใช้เทคนิคโครงข่ายประสาทเทียมแบบสังวัตนาการ (Convolutional Neural Network: CNN) ในการจำแนกรูปภาพ [1,2,3,4] อีกทั้งโทรศัพท์มือถือยังมีกล้องถ่ายรูปในตัวและสามารถนำไปใช้งานร่วมกับเทคโนโลยีที่กล่าวไว้ข้างต้นอีกด้วย

มหาวิทยาลัยราชภัฏร้อยเอ็ดเป็นสถาบันอุดมศึกษาเพื่อการพัฒนาท้องถิ่นซึ่งมีพันธกิจหลายพันธกิจ พันธกิจที่สำคัญคือการผลิตบัณฑิตซึ่งการผลิตบัณฑิตนั้นนอกจากจะพัฒนานักศึกษาตามโครงสร้างหลักสูตรแล้วยังมีการเสริมประสบการณ์ให้กับนักศึกษาด้วยกิจกรรมต่าง ๆ ที่ดูแลโดยสำนักกิจการนักศึกษา ในการจัดกิจกรรมแต่ละครั้งจะต้องมีการเช็คชื่อนักศึกษาที่เข้าร่วมกิจกรรม กระบวนการเช็คชื่อจะกระทำเมื่อสิ้นสุดการจัดกิจกรรมโดยให้นักศึกษาเข้าแถวรอรับคูปองสำหรับใช้ในการลงทะเบียนเข้าร่วมกิจกรรม ในการจัดกิจกรรมแต่ละครั้งมีนักศึกษาเข้าร่วมจำนวนมาก เจ้าหน้าที่ต้องจัดเตรียมคูปองให้เพียงพอและภายหลังจากนักศึกษาได้รับคูปองแล้วนักศึกษาจะต้องเข้าไปกรอกข้อมูลในระบบสารสนเทศด้วยตนเอง ผู้วิจัยจึงมีแนวคิดในการนำโทรศัพท์มือถือที่ใช้ระบบปฏิบัติการแอนดรอยด์ซึ่งเชื่อว่านักศึกษาส่วนใหญ่ใช้กันอยู่มาใช้สำหรับการเช็คชื่อการเข้าร่วมกิจกรรม กระบวนการเช็คชื่อก็คือให้ศึกษาแต่ละคนใช้โทรศัพท์มือถือของตัวเองถ่ายภาพใบหน้าของตัวเองแล้วแอปพลิเคชันที่พัฒนาขึ้นจะประมวลผล โดยถ้าภาพที่ถ่ายตรงกับภาพที่เคยลงทะเบียนไว้ในโทรศัพท์มือถือ แอปพลิเคชันก็จะยินยอมให้สามารถบันทึกข้อมูลการเข้าร่วมกิจกรรมไปเก็บในฐานข้อมูลได้ทันที โดยแอปพลิเคชันดังกล่าวจะใช้บริการ GPS เพื่อให้ทราบตำแหน่งที่นักศึกษาอยู่เพื่อให้ทราบว่าขณะนั้นนักศึกษาอยู่ในบริเวณจัดกิจกรรมหรือไม่ ผู้วิจัยได้ศึกษาความเป็นไปได้ในการเลือกเครื่องมือในการพัฒนาแอปพลิเคชันจากงานวิจัยต่าง ๆ ดังนี้

1.1 การค้นหาใบหน้าในรูปภาพ (Face detection) Salinda Hettiarachchi [5] กล่าวว่า Google ML Kit เป็นเครื่องมือที่ดีที่สุด ใช้เวลาเฉลี่ย 68 มิลลิวินาที

1.2 การตรวจสอบใบหน้า (Face Verification)มีหลายงานวิจัยดังนี้ Salinda Hettiarachchi [5] กล่าวว่าโมเดล FaceNet [6] เป็นอัลกอริทึมที่ให้ค่าความถูกต้องมากที่สุดประมาณ 95% ในขณะเดียวกันโมเดล MobileFaceNet [7] เป็นอัลกอริทึมที่ประมวลผลเร็วที่สุดประมาณ 24 มิลลิวินาที Sheng Chen และคณะ [7] สรุปว่า โมเดล MobileFaceNet มีขนาด 4 MB ประมวลผลได้เร็วเหมาะสำหรับการใช้งานในโทรศัพท์มือถือ มีค่าความถูกต้อง 99.55% ในขณะที่โมเดล MobileNetV2 ให้ค่าความถูกต้อง 98.58% ความเร็ว 49 มิลลิวินาที ขนาดของโมเดลอยู่ระหว่าง 1.7 MB – 6.9 MB และโมเดล FaceNet ค่าความถูกต้อง 99.63% ขนาดของโมเดล 30 MB ซึ่งถือว่ามีขนาดใหญ่ไม่เหมาะสำหรับใช้งานในโทรศัพท์มือถือ Adrian Rosebrock [8] ได้แก้ปัญหาเดียวกันนี้โดยใช้ OpenCV's face_recognition library ในภาษาไพทอน ด้วย nn4.small2 pre-trained โมเดล (ขนาดมากกว่า 30 MB) รันบน MacBook Pro ซีพียู 8 คอร์ 3.70 GHz พบว่า ใช้เวลาประมวลผลประมาณ 58.9 มิลลิวินาทีต่อเฟรม ค่าความถูกต้อง 93% ซึ่งถ้านำเทคนิคนี้มาใช้ในโทรศัพท์มือถือซึ่งมีความสามารถในการประมวลผลน้อยกว่าคอมพิวเตอร์ก็คงจะทำให้ระยะเวลาในการประมวลผลมากขึ้น Esteban Uri [9] ได้ทดลองนำโมเดล FaceNet แปลงไปเป็นโมเดลที่สามารถรันใน TensorFlow Lite จากนั้นได้พัฒนาแอนดรอยด์แอปพลิเคชันแล้วเรียกใช้งานโมเดลดังกล่าว พบว่าโมเดลสามารถทำงานได้ดีใช้เวลาในการประมวลผล 3.5 วินาทีรันใน Google Pixel 3 ซึ่งถือว่าใช้เวลาค่อนข้างมาก

จากองค์ความรู้ที่กล่าวมาข้างต้นผู้วิจัยจึงเลือกใช้ Google ML Kit เป็นเครื่องมือสำหรับการค้นหาใบหน้าภายในภาพและใช้โมเดล MobileFaceNet ซึ่งมีความเร็วในการประมวลผลมากที่สุดและให้ค่าความถูกต้อง 99.55% เป็นโมเดลสำหรับการตรวจสอบใบหน้า

2. วัตถุประสงค์

2.1 เพื่อพัฒนาแอนดรอยด์แอปพลิเคชันสำหรับจดจำและตรวจสอบใบหน้าโดยใช้โมบายเฟซเน็ต กรณีศึกษาการเช็คชื่อเข้าร่วมกิจกรรมมหาวิทยาลัยราชภัฏร้อยเอ็ด

2.2 เพื่อเปรียบเทียบความเร็วของฟังก์ชันความเหมือน ได้แก่ ฟังก์ชันระยะห่างยูคลิเดียน (Euclidean Distance) ฟังก์ชันระยะห่างแมนฮัตตัน (Manhattan Distance) และฟังก์ชันสหสัมพันธ์โคไซน์ (Cosine Coefficient)

2.3 เพื่อหาค่า threshold ที่เหมาะสมของแต่ละฟังก์ชัน

2.4 เพื่อหาประสิทธิภาพของแอปพลิเคชันที่พัฒนาขึ้น

3. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

3.1 ชนาธิป หมั่นเพียรสุขและสุพจน์ เฮงพระพรหม [10] ได้ทำวิจัยพบว่า ฟังก์ชันแมนฮัตตันให้ประสิทธิภาพที่ดีสำหรับข้อมูลที่มีจำนวนคุณลักษณะน้อย ๆ แต่เมื่อจำนวนคุณลักษณะเพิ่มมากขึ้น ฟังก์ชันยูคลิเดียนจะให้ประสิทธิภาพที่ดีกว่า แต่โดยภาพรวมแล้วฟังก์ชันแมนฮัตตันให้ประสิทธิภาพที่ดี ในขณะที่ฟังก์ชันสหสัมพันธ์ ถ้าข้อมูลมีจำนวนคุณลักษณะน้อย ๆ ทั้งเพียร์สันและโคไซน์ให้ประสิทธิภาพที่ไม่ต่างกัน แต่เมื่อจำนวนคุณลักษณะเพิ่มมากขึ้น โคไซน์จะให้ประสิทธิภาพที่ดีกว่า

1) ฟังก์ชันยูคลิเดียน (Euclidean Distance) เป็นการวัดระยะห่างปกติระหว่างจุด 2 จุดในแนวเส้นตรง ซึ่งอาจวัดได้ด้วยไม้บรรทัด ที่ได้มาจากทฤษฎีพีทาโกรัส ระยะห่างยูคลิเดียนระหว่างจุด p และ จุด q แสดงด้วย d(p,q) คำนวณได้ดังสมการ (1)

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \qquad (1)$$

โดยที่ p_i และ q_i คือ จุด 2 จุดที่ต้องการคำนวณระยะห่าง n คือ จำนวนมิติ (จุด) ของข้อมูล

ค่า d(p,q) น้อยแสดงว่า 2 จุด p และ q มีความใกล้เคียงกันมาก (หากมีค่าเป็นศูนย์ หมายถึง ทั้ง 2 จุด คือจุดจุดเดียวกัน) แต่หากมีค่ามาก แสดงว่า 2 จุดนี้ มีความห่างกันหรือแตกต่างกันมาก

2) ฟังก์ชันแมนฮัตตัน (Manhattan Distance) เป็นการวัดระยะทางระหว่างจุดสองจุดตามแกนวัดมุมขวา ซึ่งลอกเลียนมาจากตารางเค้าโครงของถนนในแมนฮัตตัน ซึ่งทำให้สามารถใช้เส้นทางที่สั้นที่สุดระหว่างจุดสองจุดในเมือง คำนวณได้ดังสมการ (2)

$$d = \sum_{i}^{n} \left| X_i - Y_i \right| \qquad (2)$$

โดยที่ X_i และ Y_i คือจุด 2 จุดที่ต้องการคำนวณระยะห่าง n คือ จำนวนมิติ (จุด) ของข้อมูล

3) ฟังก์ชันสหสัมพันธ์โคไซน์ (Cosine Coefficient) [11] หรือบางครั้งเรียกว่า ความคล้ายคลึงโคไซน์ (CosineSimilarity) เป็นการวัดความคล้ายคลึงระหว่าง 2 เวคเตอร์ โดยการวัดมุมโคไซน์ของเวคเตอร์ทั้งสอง ซึ่งคำนวณได้จากสมการ (3)

$$\cos(q,r) = \frac{\sum_v q(v)r(v)}{\sqrt{\sum_v q(v)^2}\sqrt{\sum_v r(v)^2}} \qquad (3)$$

โดยที่ q และ r คือ 2 เวคเตอร์ที่ต้องการนำมาเปรียบเทียบ ค่าสหสัมพันธ์โคไซน์จะมีค่าอยู่ระหว่าง -1 ถึง 1 โดยมีความหมายดังนี้

ถ้าค่าเข้าใกล้ 1 หมายถึง ทั้ง 2 เวคเตอร์มีความสัมพันธ์กันมากไปในทิศทางเดียวกัน

ถ้าค่าเข้าใกล้ -1 หมายถึง ทั้ง 2 เวคเตอร์มีความสัมพันธ์กันมากไปในทิศทางตรงข้ามกัน

ถ้าค่าเข้าใกล้ 0 หมายถึง ทั้ง 2 เวคเตอร์ไม่มีความสัมพันธ์กัน

3.2 Convolutional Neural Network คือ การเรียนรู้เชิงลึกมีพื้นฐานมาจาก Neural Network ซึ่งเป็นสาขาหนึ่งของเทคโนโลยี Artificial Intelligence ที่มีแนวคิดมานานตั้งแต่ปี 1943 คือ การพัฒนาเครือข่ายของอัลกอริทึมให้ทำงานแบบเดียวกับเครือข่ายระบบประสาทของมนุษย์

Convolutional Neural Network (CNN) [12] หรือโครงข่ายประสาทเทียมแบบสังวัตนาการจะจำลองการมองเห็นของมนุษย์ที่มองพื้นที่เป็นที่ย่อย ๆ และนำกลุ่มของพื้นที่ย่อย ๆ มาผสานกัน เพื่อดูว่าสิ่งที่เห็นอยู่เป็นอะไร โดยการใช้ Matrix ชนิดพิเศษที่เรียกว่า Convolution Matrix ซึ่งทำหน้าที่สกัดเอาส่วนต่าง ๆ ของภาพออกมา เช่น เส้นขอบของวัตถุต่าง ๆ เพื่อให้สามารถเรียนรู้ลักษณะของภาพได้อย่างมีประสิทธิภาพและแม่นยำ

3.3 TensorFlow [13] คือ ไลบรารีสำหรับสร้างแบบจำลองการเรียนรู้เชิงลึกซึ่งมีการนำไปประยุกต์เพื่อเพิ่มประสิทธิภาพกับผลิตภัณฑ์หลากหลายประเภท เช่น เครื่องมือค้นหา (Search Engine), การแปลภาษา (Translation), คำบรรยายภาพ (Image Captioning) และการแนะนำ (Recommendations)

TensorFlow ถูกพัฒนาโดย Google เพื่อให้นักวิจัยและนักพัฒนาทำงานกับแบบจำลอง AI ได้ เมื่อพัฒนาและปรับปรุงชักระยะหนึ่ง TensorFlow ก็ถูกปล่อยออกมาให้คนทั่วไปใช้งานได้โดยเปิดโอเพนซอร์สในปี 2015 และปล่อยตัวสมบูรณ์ออกมาในปี 2017 พร้อมลิขสิทธิ์แบบ Apache Open Source ให้คนทั่วไปสามารถใช้งาน ดัดแปลง และแจกจ่ายตัวที่ถูกดัดแปลงมาแล้วโดยที่ไม่มีค่าใช้จ่าย

TensorFlow เป็นที่นิยม เนื่องจากถูกสร้างมาเพื่อให้ทุกคนเข้าถึงได้ง่าย TensorFlow ได้รวมเอา API ที่แตกต่างกันเพื่อ

สร้างสถาปัตยกรรมแบบ Deep Learning อย่าง Convolutional Neural Network (CNN) และ Recurrent Neural Network (RNN) TensorFlow ยังมี Graph เป็นตัวคำนวณหลักเพื่อช่วยให้นักพัฒนาเห็นภาพโครงสร้าง Neural Network รวมทั้งสามารถทำงานร่วมกับ Tensorboad เครื่องมือที่ช่วยให้นักพัฒนาหาข้อผิดพลาดของโปรแกรมและสุดท้าย TensorFlow สามารถทำงานได้ทั้งบน CPU และ GPU

3.4 TensorFlow Lite [14] (TFLite) คือ Tools ที่ช่วยให้นักพัฒนาสามารถรันแบบจำลองของ TensorFlow ทำ Inference บนโทรศัพท์มือถือ Android หรือ iOS อุปกรณ์ Edge, IoT Device, Raspberry Pi, Jetson Nano, Arduino และ Microcontroller ได้เนื่องจากแบบจำลองของ TFLite มีความซับซ้อนน้อยกว่าแบบจำลองของ Tensorflow ปกติ ทำให้แบบจำลองของ TFLite มีขนาดเล็กลง ทำงานได้เร็วขึ้น แต่อย่างไรก็ตามต้องยอมรับกับความแม่นยำของแบบจำลองที่ลดลง

3.5 MobileFaceNets [7] เป็นผลงานที่ยอดเยี่ยมของนักวิจัยที่ Watchdata Inc. ในกรุงปักกิ่ง ประเทศจีน พวกเขานำเสนอโมเดล CNN ที่มีประสิทธิภาพสูงซึ่งออกแบบมาโดยเฉพาะสำหรับการตรวจสอบใบหน้าแบบเรียลไทม์ที่มีความแม่นยำสูงบนอุปกรณ์มือถือ โมเดลมีความเร็วที่น่าประทับใจด้วยความแม่นยำที่สูงมากด้วยขนาดของโมเดลเพียง 4.0 MB ความแม่นยำที่ได้รับนั้นใกล้เคียงกับโมเดลอื่น ๆ ที่ซับซ้อนกว่ามาก เช่น โมเดล FaceNet

จากรูปที่ 1 โครงสร้างของ MobileFaceNets จะใช้ภาพที่มีความละเอียด 112 × 112 พิกเซล เป็นอินพุตและเริ่มสกัดคุณสมบัติของใบหน้าจากภาพดังกล่าว ในขณะที่คุณสมบัติที่ได้รับเป็นเอาท์พุตมีขนาด 512 มิติ โมเดลมีทั้งหมด 20 เลเยอร์

Input	Operator	t	c	n	s
$112^2 \times 3$	conv3x3	-	64	1	2
$56^2 \times 64$	depthwise conv3x3	-	64	1	1
$56^2 \times 64$	bottleneck	2	64	5	2
$28^2 \times 64$	bottleneck	4	128	1	2
$14^2 \times 128$	bottleneck	2	128	6	1
$14^2 \times 128$	bottleneck	4	128	1	2
$7^2 \times 128$	bottleneck	2	128	2	1
$7^2 \times 128$	conv1x1	-	512	1	1
$7^2 \times 512$	linear GDConv7x7	-	512	1	1
$1^2 \times 512$	linear conv1x1	-	128	1	1

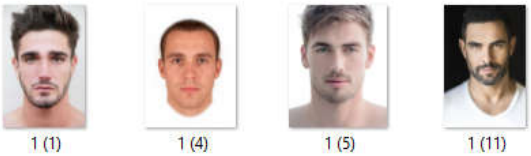
รูปที่ 1. แสดงโครงสร้างของ MobileFaceNets [7]

3.6 Google ML Kit [15] คือ SDK สำหรับ Machine Learning บน Android และ IOS ซึ่งประกอบไปด้วยสองส่วนหลัก ๆ คือ Base APIs และ Custom model ซึ่ง ML Kit นั้นสามารถใช้งานได้ทั้งแบบ Offline (On-device) และ Online (Google Cloud AI) ซึ่งแบบ Offline นั้นสามารถใช้ได้ฟรี แต่มีขีดจำกัดความสามารถอยู่บางส่วน Google ML Kit มาพร้อมกับ API พื้นฐานสำหรับงาน ML ทั่ว ๆไปหลายตัว แต่ในงานวิจัยนี้เลือกใช้เฉพาะการตรวจจับใบหน้า (Face detection) [16] ซึ่ง API ตรวจจับใบหน้าของ ML Kit จะช่วยให้เราสามารถตรวจจับใบหน้าในรูปภาพ ระบุลักษณะใบหน้าหลัก และทำความเข้าใจลักษณะของใบหน้าที่ตรวจพบได้

3.7 จีพีเอส (GPS) [17] ย่อมาจากคำว่า global positioning system หรือ ระบบตรวจสอบตำแหน่งของวัตถุ จีพีเอสจะบอกพิกัดตำแหน่งบนโลกโดยมีการติดต่อระหว่างอุปกรณ์รับสัญญาณจีพีเอสบนพื้นผิวโลกและดาวเทียมที่โคจรอยู่รอบโลก ซึ่งตำแหน่งดังกล่าวได้มาจากการคำนวณพิกัดของดาวเทียมระบุพิกัดที่ลอยอยู่ในอวกาศ จำนวน 24 ดวง ทั่วโลกการที่ระบบจีพีเอสจะทำงานได้นั้น ต้องประกอบไปด้วย 3 ส่วนหลัก ๆ คือ 1) สถานีฐาน 2) ดาวเทียมจีพีเอส และ 3) เครื่องรับสัญญาณจีพีเอส งานวิจัยนี้ใช้จีพีเอสเพื่อติดตามตำแหน่งของโทรศัพท์มือถือเพื่อให้ทราบว่าผู้ใช้งานอยู่พิกัดที่ต้องการหรือไม่

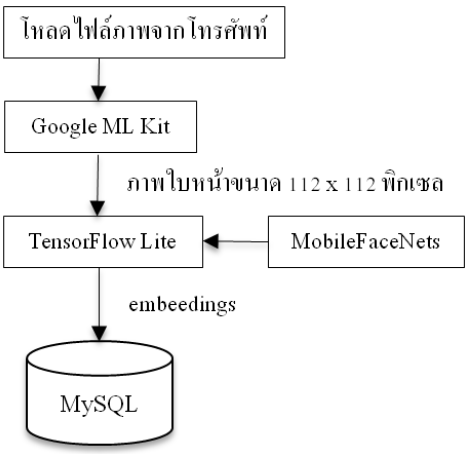
4. วิธีการวิจัย

4.1 ภาพใบหน้าที่นำมาใช้ในการทดลอง ได้นำมาจากเว็บไซต์ https://www.kaggle.com/datasets/ashwingupta3012/human-faces มีภาพใบหน้าทั้งหมด 7,219 ภาพ ซึ่งบางภาพซ้ำกัน ผู้วิจัยจึงคัดเลือกภาพที่ไม่ซ้ำกันและเป็นภาพหน้าตรง เป็นภาพผู้หญิง จำนวน 111 ภาพ และเป็นภาพผู้ชายจำนวน 399 ภาพ รวมทั้งสิ้น 510 ภาพ ทุกภาพจะถูกปรับให้ตั้งฉากเพื่อให้การประมวลผลไม่มี bias แล้วคัดลอกไปเก็บในโทรศัพท์



รูปที่ 2. ตัวอย่างภาพใบหน้าที่ใช้ทดลอง

4.2 พัฒนาแอนดรอยด์แอปพลิเคชัน โดยเรียกใช้งาน Google ML Kit, TensorFlow Lite และ MobileFaceNets เพื่อประมวลผลหาค่า embeedings (feature vector) ของแต่ละภาพจากข้อที่ 4.1 ซึ่งค่า embeedings ที่ได้จาก MobileFaceNets จะเป็นชนิดอาร์เรย์ 1 มิติ ขนาด 192 สมาชิก แต่ละสมาชิกเป็นชนิด float จากนั้นนำชื่อไฟล์ภาพและค่า embeedings ที่ได้ไปเก็บลงในฐานข้อมูล MySQL ขั้นตอนการทำงานดังรูปที่ 3



รูปที่ 3. แสดงขั้นตอนการประมวลผลหาค่า embeedings

รูปที่ 3 แสดงขั้นตอนการประมวลผลหาค่า embeedings จำนวน 1 ภาพ ซึ่งจะต้องประมวลผลจำนวน 510 รอบตามจำนวนภาพที่ใช้ทดลอง จะได้ฐานข้อมูลตามตัวอย่างในรูปที่ 4

filename	COL2	COL3	COL4	COL5	COL6
1 (1).jpg	-0.0182230980	0.0021961373	-0.0049122646	0.0061100530	-0.0229468230
1 (1005).jpg	0.0039141290	0.0003929171	-0.0130407360	0.0030874915	0.0128907060
1 (103).jpg	0.0010564298	-0.0080726845	-0.0118220460	-0.0021692850	-0.0020287314
1 (105).jpg	0.0013289162	0.0002461397	-0.0005984420	-0.0023782216	0.0012135789
1 (1053).jpg	-0.0121675220	-0.0011689715	-0.0002184557	-0.0007582008	0.0025738797
1 (1062).jpg	0.0043364903	-0.0035046497	-0.0052110700	-0.0083811890	0.0030967258

รูปที่ 4. ตัวอย่างข้อมูลที่เก็บในฐานข้อมูล

รูปที่ 4 คอลัมน์ filename หมายถึงชื่อไฟล์ภาพ COL2 COL3 หมายถึง สมาชิกของ embeedings ตัวที่ 1 และ 2 เป็นต้น จำนวนแถวเท่ากับจำนวนภาพ คือ 510 แถว

4.3 นำข้อมูลจากฐานข้อมูลตามข้อที่ 4.2 มาคำนวณหาระยะห่างโดยใช้ฟังก์ชันตามข้อที่ 3.1 ได้แก่ ฟังก์ชันยูคลิเดียน ฟังก์ชันแมนฮัตตันและฟังก์ชันสหสัมพันธ์โคไซน์ ตัวอย่างการประมวลผล เช่น ภาพที่ 1 เปรียบเทียบกับทุกภาพ ภาพที่ 2 เปรียบเทียบกับทุกภาพ ไปเรื่อย ๆ จนครบ 510 ภาพ ฉะนั้นจำนวนครั้งในการประมวลผลต่อฟังก์ชัน เท่ากับ 510x510= 260,100 ครั้ง ทำการเก็บข้อมูลระยะเวลาที่ใช้ในการประมวลผลของแต่ละฟังก์ชันเพื่อสรุปผลการทดลอง ตัวอย่างข้อมูลที่ได้แสดงดังรูปที่ 5

filename	compare	distance	distance_man	distance_cosine
1 (1).jpg	1 (1005).jpg	1.379	11.147	0.049
1 (1).jpg	1 (103).jpg	1.063	8.274	0.435
1 (1).jpg	1 (105).jpg	1.327	10.406	0.120
1 (1).jpg	1 (1053).jpg	1.231	9.799	0.243

รูปที่ 5. ตัวอย่างผลการคำนวณระยะห่าง

รูปที่ 5 ข้อมูลแถวแรก หมายถึง ไฟล์ภาพ 1 (1).jpg กับ ไฟล์ 1 (1005).jpg ระยะห่างจากฟังก์ชันยูคลิเดียน เท่ากับ 1.379 ระยะห่างจากฟังก์ชันแมนฮัตตัน เท่ากับ 11.147 และระยะห่างจากฟังก์ชันสหสัมพันธ์โคไซน์ เท่ากับ 0.049 เป็นต้น

ข้อมูลตามรูปที่ 5 แต่ละฟังก์ชันจะมีจำนวน 260,100 แถว (510x510) นำข้อมูลดังกล่าวไปวิเคราะห์หาค่า threshold หรือระยะห่างที่ให้ผลลัพธ์ถูกต้อง 100% ของแต่ละฟังก์ชัน วิเคราะห์โดยนักวิจัยเอง ผลการวิเคราะห์ดังตารางที่ 1

ตารางที่ 1. แสดงผลการวิเคราะห์ค่า threshold ที่เหมาะสม

ฟังก์ชัน	threshold
ฟังก์ชันยูคลิเดียน	<= 0.752
ฟังก์ชันแมนฮัตตัน	<= 5.796
ฟังก์ชันสหสัมพันธ์โคไซน์	>= 0.717

จากข้อมูลในตารางที่ 1 สรุปว่า ภาพ ก จะเหมือนกันกับภาพ ข ถ้าระยะห่าง <= 0.752 (ฟังก์ชันยูคลิเดียน) หรือ <= 5.796 (ฟังก์ชันแมนฮัตตัน) หรือ >= 0.717 (ฟังก์ชันสหสัมพันธ์โคไซน์) ซึ่งค่า threshold ที่ได้มาผู้วิจัยก็จะนำไปใส่ในแอปพลิเคชันต่อไป

ตารางที่ 2. เปรียบความเร็วของแต่ละฟังก์ชัน

ฟังก์ชัน	ความเร็ว (วินาที)
ฟังก์ชันยูคลิเดียน	12.880
ฟังก์ชันแมนฮัตตัน	10.261
ฟังก์ชันสหสัมพันธ์โคไซน์	23.398

จากข้อมูลในตารางที่ 2 สรุปว่า ฟังก์ชันแมนฮัตตัน มีความเร็วมากที่สุด ฉะนั้นงานวิจัยนี้จะเลือกใช้ฟังก์ชันแมนฮัตตัน โดยกำหนดค่า threshold <= 5.796

4.4 พัฒนาแอนดรอยด์แอปพลิเคชัน โดยเรียกใช้งาน Google ML Kit, TensorFlow Lite และ MobileFaceNets เพื่อใช้สำหรับการเช็คชื่อเข้าร่วมกิจกรรม โดยแอปพลิเคชัน ดังกล่าวมีความสามารถดังนี้

1) การลงทะเบียนใบหน้า

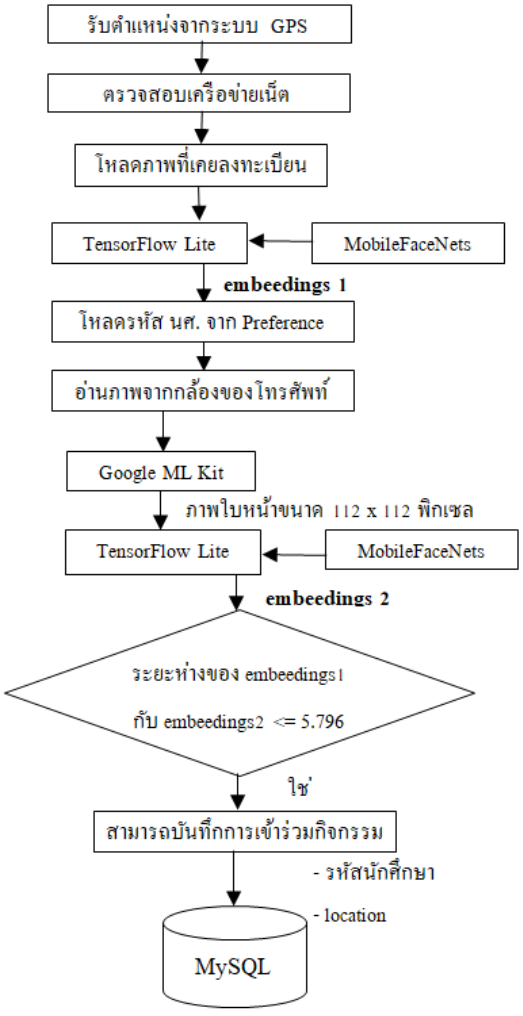
รูปที่ 6. หน้าจอหลักของแอปพลิเคชัน

จากรูปที่ 6 หมายเลข 1 ใช้สำหรับสลับกล้องหน้า-หลัง หมายเลข 2 สำหรับลงทะเบียนใบหน้าและหมายเลข 3 ใช้สำหรับบันทึกข้อมูลการเข้าร่วมกิจกรรม เมื่อผู้ใช้แตะหมายเลข 2 ผลลัพธ์ดังรูปที่ 7

รูปที่ 7. ผลลัพธ์เมื่อแตะหมายเลข 2

จากรูปที่ 7 ให้กรอกรหัสนักศึกษา ปรับใบหน้าให้ตรงแล้วแตะปุ่ม OK แอปพลิเคชันจะทำการบันทึกใบหน้าไปเป็นไฟล์ภาพขนาด 112 x 112 พิกเซล เก็บในโทรศัพท์มือถือพร้อมกับบันทึกรหัสนักศึกษาในตัวแปรชนิด Preference

2) การเช็คชื่อเข้าร่วมกิจกรรม ขั้นตอนการทำงานของโปรแกรมดังรูปที่ 8



รูปที่ 8. แสดงขั้นตอนการทำงานของแอปพลิเคชัน

4.5 นำแอปพลิเคชันไปทดสอบกับนักศึกษามหาวิทยาลัยราชภัฏร้อยเอ็ด จำนวน 30 คน วิธีการทดสอบก็คือ ให้นักศึกษาแต่ละคนใช้โทรศัพท์มือถือของตัวเองลงทะเบียนภาพใบหน้าของตัวเองจากนั้นนำโทรศัพท์มือถือไปถ่ายภาพใบหน้าของผู้ร่วมทดสอบคนอื่นแล้วสรุปผลว่าแอปพลิเคชันให้คำตอบอย่างไร รายชื่อนักศึกษาที่มาร่วมทดสอบแอปพลิเคชันมีดังตารางที่ 3 ซึ่งนักศึกษาทุกคนยินยอมให้เผยแพร่ในบทความวิจัย

ตารางที่ 3. รายชื่อนักศึกษาที่เข้าร่วมทดสอบแอปพลิเคชัน

ที่	ชื่อ-นามสกุล	สาขาวิชา
1	นายเทวา มาศวรรณา	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
2	นายชาญวิทย์ วรบุตร	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
3	นายไกรวิชญ์ แสงจันทร์ศรี	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
4	นายทศพร มาศวรรณา	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
5	น.ส.วัชราภรณ์ น้ำคำ	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
6	นายธเนศ เหง่าง่า	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
7	นายอรรถวัต พรหมคุณ	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
8	น.ส.ณัฐพร สีแสงหน่อม	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
9	น.ส.นิพาดา ธงสันเทียะ	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
10	น.ส.ปนัดดา ศรีบู้อ่ำ	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
11	นายสุวิโรจน์ ทวีศักดิ์	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
12	นายธีระศักดิ์ ยอดดี	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
13	น.ส.อนุชศรา พานาดี	การประถมศึกษา
14	น.ส.วิภาวี หล่าบุญทัน	การประถมศึกษา
15	น.ส.อมรรัตน์ จอดนอก	เคมี
16	นายธนชัย ไกยเกษ	วิทยาศาสตร์การกีฬา
17	น.ส.ละอองดาว เหลากลม	เคมี
18	นายเชษฐา ศรีขัดเค้า	วิทยาศาสตร์การกีฬา
19	นายวรภาส ละลอกน้ำ	คอมพิวเตอร์ธุรกิจ
20	น.ส.อุมาพร จันทร	การประถมศึกษา
21	นายพิชิตชัย ทับทิมไสย์	วิทยาการคอมพิวเตอร์
22	น.ส.ชลธิชา ไชยมาตย์	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
23	น.ส.ดุสิตา คาอามาตย์	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
24	นายอโรชา สมบัติศรี	วิทยาการคอมพิวเตอร์
25	นายกรกฤษ ศรีวะลา	วิทยาการคอมพิวเตอร์
26	นายสุพลชัย หงษ์ทอง	คอมพิวเตอร์และเทคโนโลยีสารสนเทศ
27	นายชยณัฐ หงส์ทอง	วิทยาศาสตร์การกีฬา
28	น.ส.ภัทราภรณ์ สุนสันเขตต์	ภาษาอังกฤษธุรกิจ
29	น.ส.รัตนาภรณ์ จำปาหอม	วิทยาการคอมพิวเตอร์
30	นายภัคพล ธรรมมนตรี	วิทยาการคอมพิวเตอร์

ผลการทดลองดังตารางที่ 4

ตารางที่ 4. ผลการทดสอบแอปพลิเคชัน

ที่	ชื่อ-นามสกุล	ถ่ายตนเอง	ถ่ายคนอื่น
1	นายเทวา มาศวรรณา	✓	✕
2	นายชาญวิทย์ วรบุตร	✓	✕
3	นายไกรวิชญ์ แสงจันทร์ศรี	✓	✕
4	นายทศพร มาศวรรณา	✓	✕
5	น.ส.วัชราภรณ์ น้ำคำ	✓	✕
6	นายธเนศ เหง่าง่า	✓	✕
7	นายอรรถวัต พรหมคุณ	✓	✕
8	น.ส.ณัฐพร สีแสงหน่อม	✓	✕
9	น.ส.นิพาดา ธงสันเทียะ	✓	✕
10	น.ส.ปนัดดา ศรีบู้อ่ำ	✓	✕

ที่	ชื่อ-นามสกุล	ถ่ายตนเอง	ถ่ายคนอื่น
11	นายสุวิโรจน์ ทวีศักดิ์	✓	×
12	นายธีระศักดิ์ ยอดดี	✓	×
13	น.ส.อนุชศรา พานาดี	✓	×
14	น.ส.วิภาวี หล่าบุญทัน	✓	×
15	น.ส.อมรรัตน์ จอดนอก	✓	×
16	นายธนชัย ไกยเกษ	✓	×
17	น.ส.ละอองดาว เหลากลม	✓	×
18	นายเชษฐา ศรีขัดเค้า	✓	×
19	นายวรภาส ละลอกน้ำ	✓	×
20	น.ส.อุมาพร จันทร	✓	×
21	นายพิชิตชัย ทับทิมไสย์	✓	×
22	น.ส.ชลธิชา ไชยมาตย์	✓	×
23	น.ส.ดุสิตา คาอามาตย์	✓	×
24	นายอโรชา สมบัติศรี	✓	×
25	นายกรกฤษ ศรีวะลา	✓	×
26	นายสุพลชัย หงษ์ทอง	✓	×
27	นายชยณัฐ หงส์ทอง	✓	×
28	น.ส.ภัทราภรณ์ สุนสันเขตต์	✓	×
29	น.ส.รัตนาภรณ์ จำปาหอม	✓	×
30	นายภัคพล ธรรมมนตรี	✓	×

หมายเหตุ ✓ หมายถึง แอปฯ สรุปใบหน้าที่ถ่ายตรงกับใบหน้าที่ลงทะเบียน × หมายถึง แอปฯ สรุปว่าไม่ตรงกัน

5. สรุปผลการทดลองและการอภิปราย

บทความนี้นำเสนอการพัฒนาแอนดรอยด์แอปพลิเคชันสำหรับจดจำและตรวจสอบใบหน้าโดยใช้โมบายเฟซเน็ต กรณีศึกษา การเช็คชื่อเข้าร่วมกิจกรรมมหาวิทยาลัยราชภัฏร้อยเอ็ด ซึ่งมีวัตถุประสงค์ 4 ข้อ ขอสรุปผลการทดลองตามวัตถุประสงค์ดังนี้

1) การพัฒนาแอนดรอยด์แอปพลิเคชันสำหรับจดจำและตรวจสอบใบหน้าโดยใช้โมบายเฟซเน็ต กรณีศึกษาการเช็คชื่อเข้าร่วมกิจกรรมมหาวิทยาลัยราชภัฏร้อยเอ็ด สามารถนำไปใช้งานได้จริง การใช้งานแอปพลิเคชัน ดังรูปที่ 6 และรูปที่ 7

2) จากข้อมูลตารางที่ 2 สรุปว่า ฟังก์ชันแมนฮัตตัน มีความเร็วมากที่สุด รองลงมาได้แก่ ฟังก์ชันยูคลิเดียนและฟังก์ชันสหสัมพันธ์โคไซน์ ตามลำดับ ฉะนั้นงานวิจัยนี้จึงเลือกใช้งานฟังก์ชันแมนฮัตตันเพื่อประสิทธิภาพที่ดีที่สุดของแอปพลิเคชัน

3) จากข้อมูลในฐานข้อมูลตามรูปที่ 5 นำไปวิเคราะห์เพื่อหาค่า threshold หรือระยะห่างที่ให้ผลลัพธ์ถูกต้อง 100% ของแต่ละฟังก์ชัน พบว่า ภาพ ก จะเหมือนกันกับภาพ ข ถ้าระยะห่าง <= 0.752 (ยูคลิเดียน) หรือ <= 5.796 (แมนฮัตตัน) หรือ >= 0.717 (สหสัมพันธ์โคไซน์) การวิเคราะห์จะวิเคราะห์โดยใช้ผู้วิจัยเอง

4) ผลการหาประสิทธิภาพของแอปพลิเคชัน ตามตารางที่ 4 พบว่า แอปพลิเคชัน ประมวลผลถูกต้อง 100%

6. ข้อเสนอแนะ

1) ในปัจจุบันระบบปฏิบัติการที่มีการใช้งานในโทรศัพท์มือถือนอกจากระบบปฏิบัติการแอนดรอยด์แล้ว ยังมีระบบปฏิบัติการ iOS ของค่าย apple อีกหนึ่งระบบปฏิบัติการที่มีผู้ใช้งานจำนวนหนึ่งในสัดส่วนที่น้อยกว่าระบบปฏิบัติการแอนดรอยด์ ฉะนั้นแนวทางในการพัฒนาแอปพลิเคชันที่มีวัตถุประสงค์เดียวกันบนระบบปฏิบัติการ iOS จึงเป็นแนวทางที่น่าสนใจอีกหนึ่งประเด็นงานวิจัย

เอกสารอ้างอิง

[1] R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra and J. M. Z. Maningo, "Object detection using convolutional neural networks," TENCON 2018 – 2018 IEEE Region 10 Conference, pp. 2023-2027, 2018.

[2] ภูมิภักดิ์ พรมรังกา, สหรัฐ แหวนทอง และ ชูพันธุ์ รัตนโภคา, "โมบายแอปพลิเคชันสำหรับจำแนกสายพันธุ์นกด้วยวิธีการเรียนรู้เชิงลึก," Journal of Information Science and Technology (JIST) Volume 12, NO 1, pp. 37-46, JAN – JUN 2022.

[3] นัศพ์ชาณัณ ชินปัญช์ธนะ, "แบบจำลองการเรียนรู้ท่าทางมนุษย์จากการเคลื่อนไหวด้วยเทคนิคโครงข่ายประสาทเทียมแบบสังวัตนาการและแบบหน่วยความจำระยะสั้นแบบยาว," Journal of Information Science and Technology (JIST) Volume 12, NO 1, pp. 27-36, JAN – JUN 2022.

[4] วัชรชัย คงศิริวัฒนา, พลกฤต หวังศิริกำโชค และวายุภักดิ์ ชันติโก, "ระบบตรวจสอบการเข้าชั้นเรียนและประเมินความสนใจผ่านลักษณะอารมณ์ทางใบหน้าด้วยกล้องเว็บแคม," วารสารวิทยาศาสตร์ลาดกระบัง ปีที่ 30, ฉบับที่ 2, หน้า 42-57, กรกฎาคม-ธันวาคม 2564.

[5] Salinda Hettiarachchi, " Analysis of different face detection and recognition models for Android, " M.S. thesis, Mid Sweden University, Faculty of Science, Sweden, 2021.

[6] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, pp. 815–823, 2015.

[7] Sheng Chen, Yang Liu, Xiang Gao and Zhen Han, "MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices," Apr, 2018. [Online], Available: https://arxiv.org/ftp/arxiv/papers/1804/1804.07573.pdf. [Accessed Aug. 1,2022].

[8] A. Rosebrock, "OpenCV Face Recognition," Sep, 2018. [Online], Available: https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition. [Accessed Aug.1,2022].

[9] E. Uri, "Converting Sandberg's FaceNet pre-trained model to TensorFlow Lite (using an "unorthodox way")," Jun 19, 2020. [Online], Available: https://medium.com/@ estebanuri/converting-sandbergs-facenet-pre-trained-model-to-tensorflow-lite-using-an-unorthodox-way- 7ee3a6ed02a3. [Accessed Aug. 1,2022].

[10] ชนาธิป หมั่นเพียรสุขและสุพจน์ เฮงพระพรหม, "ประสิทธิภาพของฟังก์ชันความเหมือนต่อขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคสำหรับการจำแนกประเภทข้อมูล," การประชุมวิชาการระดับชาติ ครั้งที่ 8 มหาวิทยาลัยราชภัฏนครปฐม, นครปฐม, หน้า 473-479, 2559.

[11] Lillian Lee, "Measures of distributional similarity," Proceedings of the 3 7 th annual meeting of the Association for Computational Linguistics on Computational Linguistics, pp. 25-32, 1999.

[12] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Under standing of a convolutional neural network," In International Conference on Engineering and Technology (ICET), pp. 1-6, 2017.

[13] "TensorFlow," TensorFlow, 2015. [Online]. Available:https://www.tensorflow.org. [Accessed Aug. 1,2022].

[14] "TensorFlowLite," TensorFlow, 2018. [Online]. Available: https://www.tensorflow.org/lite/guide. [Accessed Aug. 1,2022].

[15] "ML Kit คืออะไร," www.mindphp.com, 2562. [Online]. Available: https://www.mindphp.com/คู่มือ/73-คืออะไร/7110-what-is-ml-kit.html. [Accessed Aug. 1,2022].

[16] "Face detection," developers.google.com, 2022. [Online]. Available: https://developers.google.com/ml-kit/vision/face-detection. [Accessed Aug. 1,2022].

[17] ปุญญรัตน์ ปุญญา, "การพัฒนา MSU–GPS สำหรับการเดินทางในมหาวิทยาลัยมหาสารคาม," ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาสื่อนฤมิต,มหาวิทยาลัยมหาสารคาม, 2557.