รายงานสหกิจศึกษาฉบับสมบูรณ์

การยืนยันตัวตนด้วยมือ

Hand Password

นายสุริยะ ประเสริฐปัญญา

สาขาวิชาวิศวกรรมสารสนเทศ

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงงานสหกิจศึกษา (ภาษาไทย) การยืนยันตัวตนด้วยมือ Hand Password

ชื่อ-สกุล นักศึกษา นายสุริยะ ประเสริฐปัญญา

คณะ วิศวกรรมศาสตร์ **ภาควิชา** วิศวกรรมคอมพิวเตอร์

ชื่อ-สกุล อาจารย์นิเทศ ผศ.ไพศาล สิทธิโยภาสกุล

ชื่อ-สกุล ผู้นิเทศงาน นายอนันตโชค ดวงสาโรจน์

สถานประกอบการ บริษัท กสิกร เทคโนโลยี กรุ๊ป เซเคร็ทเรียต จำกัด

บทคัดย่อ

ธนาคารกสิกรไทยมุ่งเน้นที่จะพัฒนาเทคโนโลยีและผลิตภัณฑ์ของตนเองอยู่เสมอมา สำหรับธนาคารแล้วระบบยืนยันตัวตนเป็นหนึ่งในสิ่งที่สำคัญที่สุด ปัจจุบันได้มีการนำข้อมูลทางด้าน Biometrics มาใช้ในการยืนยันตัวตนหลายชนิด ยกตัวอย่างเช่น ภาพถ่ายใบหน้า และลายนิ้วมือ

อีกรูปแบบหนึ่งของข้อมูลที่มีรายงานว่าสามารถใช้ทำการระบุตัวตนได้ดีคือภาพถ่ายของมือ และทางทีม Biometrics ของ KBTG ให้ความสนใจกับการค้นคว้านี้ ความสนใจในแนวคิดเบื้องต้นนั้นทำให้เกิดโปรเจกต์นี้ขึ้นมา ทั้งหมดของโปรเจกต์นี้มีไว้สำหรับการทดลอง และพัฒนาในอนาคต

คำสำคัญ: การยืนยันตัวตนด้วยมือ Hand Password

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Cooperative Title: Hand Recognition – Hand Password

Student intern name: Mr. Suriya Prasertpunya

Faculty: Engineering **Department:** Computer Engineering

Advisor name: Asst. Prof. Paisan Sithiyopasakul

Mentor name: Mr. Anantachok Duangsarot

Company: Kasikorn Technology Group Secretariat Co.,Ltd

ABSTRACT

Kasikorn Bank is always motivated by the aim of improving its technologies and products. As a bank, identity verification is one of the most important systems. At the moment, there are several biometric informations that have been used for this identification task e.g. facial images and finger prints etc.

Another biometrics information that was reported to be a good source for human identification is the hand photos, and the Biometrics team of KBTG (owned by Kasikorn Bank) is interested in the research. The interest brought in this study. This project is for experimental and the future development.

Keywords: Hand Recognition – Hand Password

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานฉบับนี้ ได้รับการสนับสนุน คำแนะนำ และคำปรึกษาจากหลาย ๆ ท่านในทีม Biometrics ของ KBTG โดยเฉพาะความรู้สำหรับการทำหัวข้องาน ที่เป็นประโยชน์ให้สามารถทำหัวข้อนี้ต่อไปได้ และผู้ที่เป็นคนนิเทศงานและรับผิดชอบการตรวจสอบการทำงานทั้งหมด คือ คุณอนันตโชค ดวงสาโรจน์ ตำแหน่ง Associate Data Scientist คอยให้ความรู้ในด้าน Machine Learning การจัดการข้อมูล ขั้นตอนการทำงานและแนะแนวการแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้น ต้องขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณอาจารย์ที่ปรึกษา สาขาวิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่คอยให้คำแนะนำในเรื่องการทำสหกิจศึกษาในครั้งนี้

และสุดท้ายนี้ขอขอบคุณรุ่นพี่ของวิศวกรรมสารสนเทศและเพื่อน ๆ ทุกคนทั้งต่างมหาลัยต่างคณะที่ให้คำปรึกษาในเรื่องการทำงาน การแก้ไขปัญหาต่าง ๆ ตั้งแต่เริ่มทำสหกิจศึกษาจนสำเร็จรายงานฉบับนี้ไปได้ด้วยดี

นาย สุริยะ ประเสริฐปัญญา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VII
สารบัญตาราง	X
บทที่ 1 บทนำ	**1**
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์ของการทำวิจัย	2
1.3 ขอบเขตของการวิจัย	2
1.4 วิธีการดำเนินการวิจัย	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง	**4**
2.1 ทฤษฎีเกี่ยวกับงานวิจัย	4
2.2 งานวิจัยที่เกี่ยวข้อง	18
บทที่ 3 วิธีการดำเนินงาน	**18**
3.1 กระบวนการทำความเข้าใจงานวิจัยแหล่งอ้างอิง	18
3.1.1 โมเดลที่ใช้ในงานวิจัย	18
3.1.2 Dataset ของทางงานวิจัย	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การเตรียมข้อมูลของงานวิจัย 19

3.1.4 กระบวนการ Training ของงานวิจัย 21

3.1.5 ผลลัพธ์ของ Biometric Identification Training 21

3.2 กระบวนการเตรียมข้อมูล 22

3.2.1 Dataset ของ 11K Hands 22

3.2.2 Dataset ของ IITD 23

3.2.3 Dataset ของ Hongkong University 23

3.2.4 ข้อมูลที่เก็บมาจาก website 24

3.2.5 ข้อมูลรูปภาพจากกล้องมือถือ 25

3.2.6 ข้อมูลที่จากการรวบรวมด้วยตัวเองจากการทำ website 26

3.2.6.1 ส่วนของ Front-End 26

3.2.6.2 ส่วนของ Back-End 26

3.2.6.3 การเปิดให้งาน website 28

3.3 ขั้นตอนการ Train CNN models 29

3.4 ผลลัพธ์ของการ Train CNN models 36

3.5 ขั้นตอนการ Train with Triplet Loss Function 42

3.6 ผลลัพธ์ของการ Train with Triplet Loss Function 45

3.7 ศึกษาวิธีการต่าง ๆ ที่จะเพิ่มประสิทธิภาพให้โมเดล 46

3.7.1 Autoencoder 46

3.7.2 Segmentation 47

3.7.3 Hand Keypoint Detection 48

3.8 การแสดงตัวอย่างของ filter ของโมเดล และ features ของรูปภาพ 50
ในขบวนการ CNN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 ผลการดำเนินงาน **55**

4.1 การทำ web application เพื่อใช้เก็บข้อมูลเพิ่มในการเรียนรู้ 55

4.2 Service Hand Detection 55

4.3 การทดลองโมเดลจากขั้นตอนการ Train ด้วย CNN 55

4.3.1 การเปรียบเทียบกับผลลัพธ์ของ 11K Hands 55

4.3.2 การเปรียบเทียบโมเดลด้วยการเปลี่ยนพารามิเตอร์ต่าง ๆ 56

4.3.3 การทดลองโมเดลด้วย dataset ต่าง ๆ 56

4.4 การทดลองโมเดลจากขั้นตอนการใช้ Triplet loss 56

บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ **58**

5.1 สรุปผลการดำเนินงาน 58

5.2 ปัญหาและอุปสรรค 58

5.3 ข้อเสนอแนะและแนวทางในอนาคต 59

เอกสารอ้างอิง **60**

ประวัติผู้เขียน **65**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ 2.1 ตัวอย่างการทำงานของ Deep Learning 5

ภาพที่ 2.2 อัลกอริทึม Gradient Descent 7

ภาพที่ 2.3 ตัวอย่างของ filter 7

ภาพที่ 2.4 ตัวอย่างการทำงานของ filter 7

ภาพที่ 2.5 ตัวอย่างการทำงานของ filter 8

ภาพที่ 2.6 ตัวอย่างการทำงานของ Convolution 8

ภาพที่ 2.7 การเลื่อนของ filter ด้วย stride เท่ากับ 1 9

ภาพที่ 2.8 การเลื่อนของ filter ด้วย stride เท่ากับ 2 9

ภาพที่ 2.9 การ padding ของรูปภาพ 10

ภาพที่ 2.10 แนวคิดการใช้ pooling 10

ภาพที่ 2.11 ตัวอย่างการใช้ pooling 11

ภาพที่ 2.12 หลักการทำงาน Flatten 12

ภาพที่ 2.13 Sigmoid Function 12

ภาพที่ 2.14 Tanh Function 12

ภาพที่ 2.15 ReLU Function 12

ภาพที่ 2.16 หลักการทำงาน Dropout 14

ภาพที่ 2.17 กราฟแสดงการอธิบายของแต่ละ fitting 16

ภาพที่ 2.18 ตัวอย่างกราฟการเรียนรู้ของโมเดลที่ overfitting 16

ภาพที่ 2.19 ตัวอย่างรูปต้นฉบับที่นำ augment 17

ภาพที่ 2.20 ตัวอย่างรูปหลัง augment 17

ภาพที่ 3.1 Architecture โมเดลของ 11K Hands 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3.2 การทำงานของ Guided filter 21

ภาพที่ 3.3 ตัวอย่างการทำงานของ Guided filter 21

ภาพที่ 3.4 ตัวอย่างภาพ detailed layer 22

ภาพที่ 3.5 Metadata ของ 11K Hands ไฟล์ csv ที่เก็บข้อมูลต่าง ๆ ของคน 189 คน 23

ภาพที่ 3.6 ตัวอย่าง Dataset ของ IITD 24

ภาพที่ 3.7 ตัวอย่าง Dataset ของ IITD 25

ภาพที่ 3.8 ตัวอย่างรูปภาพใน Dataset ที่มาจาก website ที่ไม่สามารถระบุตัวตนได้ 25

ภาพที่ 3.9 ตัวอย่างรูปภาพใน Dataset ที่มาจาก website ที่สามารถระบุตัวตนได้ 26

ภาพที่ 3.10 ตัวอย่างรูปภาพจากกล้องโทรศัพท์ที่มีสภาพแวดล้อมที่แตกต่างกัน 26

ภาพที่ 3.11 ตัวอย่างของ website ที่จัดทำ 27

ภาพที่ 3.12 ตัวอย่างการทำงานของอัลกอริทึม YOLO 28

ภาพที่ 3.13 หลักการ IOU 29

ภาพที่ 3.14 ผลลัพธ์ของการทดสอบด้วย mAP 29

ภาพที่ 3.15 ตัวอย่างการ import library 30

ภาพที่ 3.16 ตัวอย่างการใช้ ImageDataGenerator 32

ภาพที่ 3.17 ตัวอย่างการใช้ flow_from_directory ในการทำ generator 33

ภาพที่ 3.18 Sharpen kernel ขนาด 3x3 33

ภาพที่ 3.19 ตัวอย่างการใช้ sharpen kernel ขนาด 3x3 34

ภาพที่ 3.20 Generator ที่ใช้สำหรับการเรียนรู้ของโมเดล 34

ภาพที่ 3.21 Architecture ของโมเดล 35

ภาพที่ 3.22 ตัวอย่างภาพ detailed layer 36

ภาพที่ 3.23 Architecture ของโมเดลที่ใช้รูปภาพ detailed layer 36

ภาพที่ 3.24 กราฟผลลัพธ์ของโมเดลที่ทำการ train 37

ภาพที่ 3.25 ตัวอย่าง dataset ของ Mutah University 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3.26 ตัวอย่าง dataset ของ European Cities 1M Dataset 40

ภาพที่ 3.27 ตัวอย่าง dataset ที่ทำการ random background 40

ภาพที่ 3.28 ตัวอย่าง dataset ที่ทำหมุนให้มือมีลักษณะทิศทางเดียวกัน 41

ภาพที่ 3.29 หลักการทำงานของ Triplet Loss 42

ภาพที่ 3.30 สมการของ Triplet Loss 42

ภาพที่ 3.31 ฟังก์ชันการเรียกใช้งาน Triplet Loss 42

ภาพที่ 3.32 ฟังก์ชันสำหรับการเปลี่ยนคำตอบให้เป็นตัวเลข 43

ภาพที่ 3.33 Generator สำหรับ Triplet Loss 44

ภาพที่ 3.34 กราฟผลลัพธ์ของการ train ด้วย Triplet Loss 45

ภาพที่ 3.35 ตัวอย่างของ Autoencoder 46

ภาพที่ 3.36 ตัวอย่างของ Segmentation 47

ภาพที่ 3.37 ตัวอย่างของ Segmentation ด้วยรูปภาพในโปรแกรม 48

ภาพที่ 3.38 ตัวอย่างการทำ Label ของ Segmentation ด้วยตนเอง 48

ภาพที่ 3.39 ตัวอย่างการทำงานของ Affine Transformation 49

ภาพที่ 3.40 ตัวอย่างของ Hand Keypoint Detection 49

ภาพที่ 3.41 ตัวอย่างการทดสอบ Hand Keypoint Detection ด้วย dataset ของ 11K Hands 49

ภาพที่ 3.42 Architecture โมเดลด้วยคำสั่ง Model Summary 50

ภาพที่ 3.43 ตัวอย่างของ filter ที่ชั้นที่ 2 ของโมเดลตั้งต้น 51

ภาพที่ 3.44 ตัวอย่างของภาพที่ผ่าน filter ที่ชั้นที่ 2 ของโมเดลตั้งต้น 52

ภาพที่ 3.45 ตัวอย่างของภาพที่ผ่าน filter ที่ชั้นที่ 84 ของโมเดลตั้งต้น 53

ภาพที่ 3.46 ตัวอย่างของภาพที่ผ่าน filter ที่ชั้นที่ 145 ของโมเดลตั้งต้น 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ 3.1 ผลลัพธ์ของการทดสอบกับ datasets ของงานวิจัย 11K Hands 22

ตารางที่ 3.2 ผลลัพธ์การทดสอบโมเดลต่าง ๆ กับ IITD dataset 26

ตารางที่ 3.3 ผลลัพธ์การทดสอบด้วยโมเดลที่ได้เปรียบเทียบกับโมเดลของ 11K Hands 37

ตารางที่ 3.4 ผลลัพธ์การทดสอบด้วยโมเดลการเปลี่ยนพารามิเตอร์ต่าง ๆ 38

ตารางที่ 3.5 ตัวอย่างของการตรวจสอบของการแยกลักษณะ 39

ตารางที่ 3.6 ผลลัพธ์การทดสอบด้วย datasets อื่น ๆ 41

ตารางที่ 3.7 ผลลัพธ์ของการทดสอบโมเดล Triplet Loss 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

KASIKORN Business-Technology Group (บริษัท กสิกร บิซิเนส-เทคโนโลยี กรุ๊ป) เป็นบริษัทด้านเทคโนโลยี แนวหน้าของธนาคารในประเทศไทย โดยบริษัทมีหน้าที่ค้นคว้าเทคโนโลยีใหม่ ที่จะนำมาประยุกต์ใช้ให้เกิดประโยชน์กับธนาคารกสิกรไทยและพาร์ทเนอร์ ไม่ว่าจะเป็นในด้านต่าง ๆ เช่น ด้านการเงิน ด้านบุคลากร เป็นต้น นอกจากนี้ KBTG ยังมีเป้าหมายหลักในการเป็นบริษัททางด้านเทคโนโลยีที่ดีที่สุดในภูมิภาคเอเชียตะวันออกเฉียงใต้ ดังนั้นบริษัท จำเป็นต้องพัฒนาบุคลากรและสร้างวัฒนธรรมการทำงานให้มีความเหมาะสมต่อการคิดค้นเทคโนโลยีที่ก้าวหน้า และทันต่อการเปลี่ยนแปลงของโลกอยู่ตลอดเวลา

ในยุคปัจจุบันที่มีจำนวนข้อมูลมากขึ้นเรื่อย ๆ การวิเคราะห์ข้อมูลในแบบเดิมอาจจะทำให้ใช้เวลานานและไม่แม่นยำ จึงมีใช้เวลาในการวิเคราะห์ข้อมูลมากเป็นเวลานาน จึงมีศาสตร์ที่ชื่อว่า Machine Learning ที่ทำให้ระบบคอมพิวเตอร์ หรือเครื่องจักรต่าง ๆ สามารถเรียนรู้ได้ด้วยตนเองจากข้อมูลที่ได้รับ ซึ่งคอมพิวเตอร์สามารถวิเคราะห์หาจุดสำคัญและความสัมพันธ์ของข้อมูลได้รวดเร็วกว่ามนุษย์ จากนั้นเครื่องจะสามารถทำนายผลลัพธ์ของข้อมูลที่มีลักษณะคล้ายกับข้อมูลที่ที่เครื่องจักรเรียนรู้

หนึ่งในแขนงย่อยของ Machine Learning คือ Deep learning หรือการเรียนรู้ในเชิงลึก โดยมีหลักการที่เป็นการจำลองวิธีการทำงานของสมองมนุษย์ ทำให้เครื่องจักรสามารถทำงานได้เหมือนระบบประสาทของมนุษย์ เพื่อให้เครื่องจักรสามารถเรียนรู้ข้อมูลจำนวนมากและสามารถแยกแยะสิ่งที่เรียนรู้ไปได้อย่างแม่นยำมากขึ้น ในบางกรณีโมเดล Deep Learning สามารถทำการวิเคราะห์และทำนายผลได้ดีกว่ามนุษย์

บริษัท กสิกร บิซิเนส-เทคโนโลยี กรุ๊ป มีแผนก Machine learning ที่คอยจัดการเกี่ยวกับข้อมูลจำนวนมากภายในบริษัท ไม่ว่าข้อมูลการเงิน ข้อมูลลูกค้า เป็นต้น โดยผมเข้าร่วมโครงการสหกิจศึกษากับทีม Biometrics หนึ่งในทีมย่อยของทีม Machine Learning เพื่อหาแนวทางในการยืนยันตัวตนแบบใหม่ (Identification) ที่นอกจากการใช้ใบหน้าแล้ว ยังมีส่วนต่าง ๆ ของบุคคลที่คาดว่าจะสามารถนำมาประยุกต์ใช้ได้ในการยืนยันตัวตนได้ สิ่งที่จะนำมาทดลองนำมาประยุกต์ในครั้งนี้ก็คือ ลายมือ โดยปกติแล้วจะมีเครื่องสแกนลายนิ้วมือแต่ว่าต้องสแกนระยะประชิด ทาบไปกับอุปกรณ์ฮาร์ดแวร์ ดังนั้นทางทีม Biometrics คิดจะเปลี่ยนขั้นตอนจากการสแกนเป็นการถ่ายรูปแทน ซึ่งจะคล้ายกับการถ่ายรูปใบหน้าเพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยืนยันตัวตน ข้อดีคือวิธีการนี้ใช้แค่กล้องถ่ายรูปทั่วไปก็สามารถได้ข้อมูลรูปถ่ายที่ต้องการ แต่การนำรูปมือมา ใช้ยืนยันตัวตนก็ถือว่าเป็นสิ่งที่ท้าทายใหม่ของทีม

1.2 วัตถุประสงค์ของการทำวิจัย

1.2.1 ค้นคว้าวิธีการใหม่ของการยืนยันตัวตน

1.2.2 ทำการศึกษาข้อมูลการยืนยันตัวตนด้วยฝ่ามือ

1.2.3 หาแนวทางในการพัฒนาในอนาคต

1.3 ขอบเขตของการวิจัย

สามารถแบ่งขอบเขตการวิจัยได้ดังนี้

1.3.1 ขอบเขตด้านแหล่งอ้างอิงโดยอ้างอิงหลักมาจาก 11K Hands

1.3.2 ขอบเขตด้านแหล่งข้อมูลรูปภาพมือ

1.3.2.1 dataset ของ 11K Hands

1.3.2.1 dataset ของ IITD

1.3.2.2 dataset ของ Mutah University

1.4 วิธีการดำเนินการวิจัย

การวางแผนการทำงานในแต่ละขั้นตอนเป็นไปตามขอบเขตการวิจัยซึ่งแบ่งออกเป็นแต่ละส่วนตามลักษณะการทำงาน

1.4.1 ศึกษาเกี่ยวกับ Machine learning

1.4.1.1 ศึกษาเกี่ยวกับ Deep learning

1.4.1.2 ศึกษาเกี่ยวกับ Convolution Neural Network

1.4.2 การจัดเตรียมข้อมูลเพื่อนำใช้กับ Deep learning

1.4.2.1 หาแหล่งข้อมูล Dataset ที่เปิดให้สามารถนำไปใช้ได้

1.4.2.2 ทำการเพิ่มข้อมูลด้วยตัวเอง

1.4.3 ทำการสร้างโมเดล Deep learning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.4 นำผลลัพธ์ไปทำนายผลกับข้อมูลที่ไว้ทดสอบ

1.4.5 การปรับโมเดลและข้อมูล เพื่อเพิ่มความแม่นยำของโมเดล

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าได้รับจากบริษัท กสิกร บิซิเนส-เทคโนโลยี กรุ๊ป ภายในระยะเวลาสหกิจศึกษา เป็นจำนวน 4 เดือน คือ

1.5.1 ประโยชน์ต่อบริษัท

1.5.1.1 ได้รับ Research จากผู้ดำเนินงาน

1.5.1.2 มีผู้ดำเนินงานได้ช่วยงานที่บริษัทไม่มีเวลาดำเนินงาน

1.5.2 ประโยชน์ต่อผู้ดำเนินงาน

1.5.2.1 ได้รับความรู้ด้าน Machine Learning สามารถนำไปต่อยอดความรู้ในภายภาคหน้า

1.5.2.2 ได้รับทักษะด้านการเขียนโปรแกรมและการใช้ระบบปฏิบัติการ Ubuntu ระหว่างดำเนินงาน

1.5.2.3 เรียนรู้การทำงานของบริษัท ว่ามีการแบ่งหน้าที่กันอย่างไร

1.5.2.4 กระตุ้นให้ตัวเองใฝ่หาความรู้ให้มากขึ้น

1.5.2.5 ทำความเข้าใจในข้อผิดพลาดของตัวเองและปรับข้อผิดพลาดของตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีเกี่ยวกับงานวิจัย

AI : Artificial Intelligence หรือปัญญาประดิษฐ์ คือเครื่องจักรที่สามารถเรียนรู้ สิ่งต่าง ๆ โดยการเรียนรู้ของเครื่องจักรก็จะเปรียบเทียบกับสมองมนุษย์ ซึ่งสมองของเครื่องจักร คือ Machine Learning

Machine Learning ก็คือการเรียนรู้ของเครื่องจักร ที่เรียนรู้ด้วยตัวเองโดยใช้ข้อมูลที่ได้รับมา โดยไม่ต้องให้ผู้สอนเขียนโปรแกรมให้ว่าเรียนรู้อย่างไร การเรียนรู้จะเริ่มต้นจากการอาศัยข้อมูลที่ได้รับมาเพื่อหาความสัมพันธ์ของข้อมูลกับคำตอบที่ได้รับ เพื่อทำนายความเป็นไปได้ของคำตอบในอนาคต (Prediction) ว่า ถ้าได้รับข้อมูลเข้ามา ควรจะมีคำตอบเป็นอย่างไร

ประเภทของการเรียนรู้ของเครื่องจักร แบ่งได้ออกเป็น

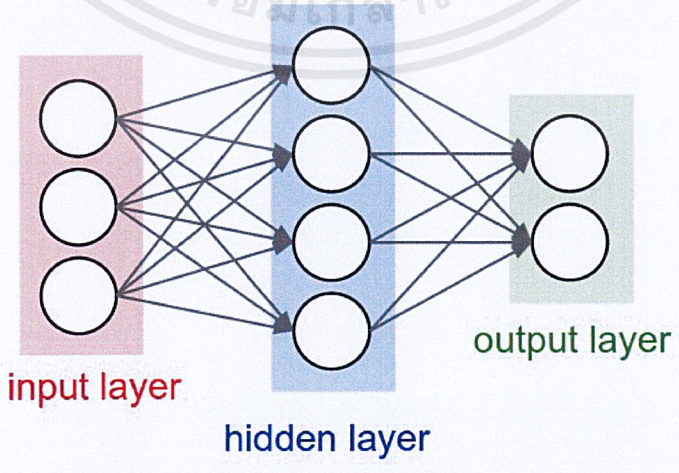
1) Surpervised learning การเรียนรู้แบบมีผู้สอน โดยข้อมูลที่เครื่องจักรจะเรียนรู้นั้น ผู้สอนจะทำการสอนให้รู้จักในช่วงแรก แล้วให้เครื่องจักรไปเรียนรู้และหาความสัมพันธ์ของข้อมูลของข้อมูลที่รับไปกับผลลัพธ์ได้
2) Unsupervised learning การเรียนรู้แบบไม่มีผู้สอน โดยจะให้ข้อมูลกับเครื่องจักรไปหาความสัมพันธ์ของข้อมูลเอง คำตอบที่ได้มาจะเป็นแค่การแบ่งกลุ่มข้อมูลเท่านั้น
3) Reinforcement learning การเรียนรู้แบบเสริมกำลัง เกี่ยวข้องกับการดำเนินการในสภาพแวดล้อมที่เปลี่ยนแปลงตลอดเวลา มักถูกใช้ในการเรียนรู้ของการเล่นเกมส์กับฝ่ายตรงข้ามที่เป็นมนุษย์
4) Semi Supervised Learning (การเรียนรู้แบบกึ่งมีผู้สอน) จะเป็นการสอนแบบไม่สมบูรณ์ ผู้สอนจะทำให้คำตอบไปแค่ครึ่งเดียวของข้อมูลที่ป้อนไป
5) Transduction (ทรานดักชั่น) เป็นกรณีพิเศษของ Semi Supervised Learning คือจะเป็นหาคำตอบให้กับข้อมูลที่ไม่มีคำตอบ ไม่ได้ทำนายคำตอบในอนาคต จึงไม่สามารถนไปใช้กับข้อมูลชุดอื่น ๆ ได้โดยตรง
6) Learning to learn, meta-learning (การเรียน หรือ วิธีการเรียน) เป็นการเรียนรู้วิธีการเรียนรู้ของตัวเอง โดยใช้การเรียนรู้จากประสบการณ์ที่ผ่านมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำตอบของ machine learning นั้นสามารถแบ่งออกได้เป็นหลายประเภทดังนี้

1) Classification การแบ่งประเภทข้อมูล ข้อมูลที่ได้รับจะมีความแตกต่างกัน ผู้สอนจะต้องสร้างโมเดลที่สามารถแยกความแตกต่างนั้นได้จากข้อมูลที่ไม่เคยเห็นมาก่อน โดยปกติจะใช้วิธีการเรียนรู้แบบมีผู้สอน เช่น ใส่ข้อมูลรูปสุนัขกับแมว ข้อมูลเซตนี้ก็จะมีจำนวน 2 classes คือ สุนัข กับ แมว แล้วให้เครื่องจักรเรียนรู้การแบ่งประเภทข้อมูลให้ได้คำตอบตาม class นี้
2) Regression การวิเคราะห์การถดถอย จะเป็นการวิเคราะห์ถึงความสัมพันธ์ของข้อมูล ซึ่งความสัมพันธ์จะเป็นลักษณะของตัวเลขที่ต่อเนื่อง เช่น ปริมาณฝนที่วัดได้ในกรุงเทพมหานครฯ ทุกวันเป็นเวลา 1 ปี
3) Clustering การแบ่งกลุ่มข้อมูล เป็นคำตอบที่ถูกแบ่งออกเป็นกลุ่ม ๆ ตามความสัมพันธ์ของคำตอบ โดยดูจากข้อมูลที่มีลักษณะเหมือนกัน หรือ คล้ายกันมาจัดไว้ในกลุ่มเดียวกัน โดยใช้วิธีการวัดระยะห่างของเวกเตอร์ข้อมูลขาเข้า ซึ่งเป็นการเรียนรู้แบบไม่มีผู้สอน
4) Density Estimation การประเมินความหนาแน่น

Deep Learning ก็เป็นส่วนหนึ่งของ Machine Learning เป็นอัลกอริทึมการเรียนรู้แบบเชิงลึก โดยใช้หลักการของ Artificial Neural Networks คือการพยายามเลียนแบบการทำงานของสมองมนุษย์ สมองมนุษย์มีระบบประสาทที่มีเซลล์ประสาทหลาย ๆ เซลล์เชื่อมต่อกัน และส่งสัญญาณไปยังเซลล์ประสาทที่ติดกัน ซึ่ง Neural Networks ก็เป็นเครื่องมือหนึ่งที่คิดมาเพื่อเลียนแบบการทำงานนั้น เพื่อจำลองความสัมพันธ์ของข้อมูลที่ไม่ใช่ความสัมพันธ์เชิงเส้น (Non-linear relation) Neural Networks จะมีอยู่สามส่วน ส่วนแรกคือส่วนรับข้อมูล หรือ Input เข้าไปจำนวนมาก ส่วนที่สอง คือส่วนที่ระบบการทำงานนำข้อมูลไปประมวลผลโดยเป็นการเชื่อมต่อกันแบบขนาน ส่วนที่สามคือ ส่วนผลลัพธ์ หรือ Output ที่ได้ออกมาจากการประมวลผล



ภาพที่ 2.2 ตัวอย่างการทำงานของ Deep Learning [3]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Deep learning จะคล้าย ANN แต่ระบบประมวลผล จะมีจำนวนหลายเลเยอร์ซ้อนกัน ทำให้มีโครงสร้างที่ลึกยิ่งขึ้น จึงได้ชื่อว่า Deep Learning โดยในแต่ละชั้นจะมีการประมวลผลข้อมูล input ให้เป็นรูปธรรมที่เข้าใจง่ายมากขึ้นและส่งข้อมูลที่ประมวลผลแล้วให้กับเลเยอร์ชั้นถัดไป โดยที่แต่ละชั้นจะมีค่าถ่วงน้ำหนัก (weight) และค่าอื่น ๆ ที่สำคัญในขบวนการสามารถเรียนรู้ว่า features ใดควรอยู่ระดับใดจึงจะเหมาะสมที่สุด

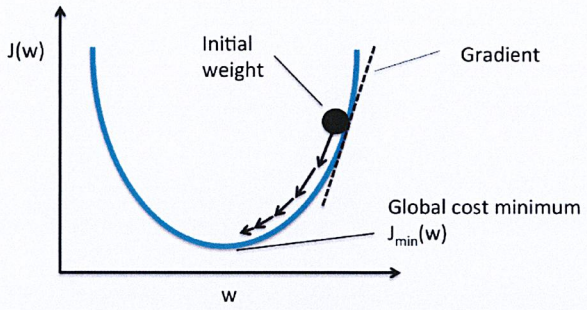
Deep learning ในการเชื่อมต่อแต่ละชั้นของ neural networks จะมีค่าถ่วงน้ำหนัก (weight) เป็นการกำหนดความสำคัญข้อมูลที่แต่ละ node รับมาจากชั้นก่อนหน้า และจะมี Activation Function ที่เป็นฟังก์ชันที่รับผลรวมการประมวลผล ของข้อมูลที่รับมาภายในแต่ละ node แล้วนำมาคำนวณหาว่าค่าผลลัพธ์ที่ได้จะเป็นเท่าไหร่ เพื่อส่งไปยัง node ถัดไป

ในการทำงานของ neural networks นั้นจะเป็นการใช้ผลลัพธ์ที่ได้ออกมาจากชั้นหนึ่งไปเป็นข้อมูลส่งไปยังชั้นต่อไป จากชั้น input layer ไปยัง output layer ข้อมูลถูกส่งไปข้างหน้าเสมอ ไม่ถูกส่งย้อนกลับมา เรียกการทำงานนี้ว่า Feed Forward

การเรียนรู้ของ nerual networks จากขบวนการ feed forward จะมีโอกาสจะเกิดค่าความผิดพลาด (error/loss/cossfunction) โดยนำค่าที่ได้จากการทำ feed forward มาเทียบกับค่า ground truth ที่เป็นผลลัพธ์ที่ควรจะเป็น Neural Networks ก็จะทำการเรียนรู้จากจุดนี้ โดยเรียนรู้จากค่าความผิดพลาดที่คำนวณไป เพื่อปรับ weights ที่จะทำให้ค่า loss น้อยลงเรื่อย ๆ จนถึงจุดที่ค่า loss น้อยที่สุด

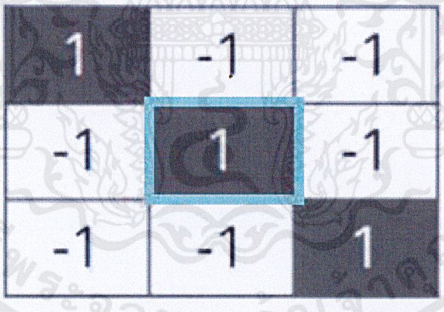
การปรับค่าถ่วงน้ำหนัก weight ใช้อัลกอริทึม Back Propagation สำหรับการคำนวณการปรับ gradients โดยเป็นการใช้สมการทางคณิตศาสตร์ในการปรับค่าถ่วงน้ำหนัก โดยดูจากค่าความผิดพลาดที่เกิดขึ้นในแต่ละ node ของชั้น โดยจะเริ่มทำจากชั้น output ย้อนกลับไปยังชั้น input การปรับค่าถ่วงน้ำหนัก (weight) หรือการ Optimization แต่ละ node ใน neural networks ใช้อัลกอริทึมเรียกว่า Gradient Descent ใช้ค้นหาค่าต่ำสุดของฟังก์ชัน โดยจะทยอยปรับค่าถ่วงน้ำหนัก weight โดยดูจากการหาอนุพันธ์ค่าความผิดพลาด จะทำให้เห็นสัดส่วนการไล่ระดับของค่าความผิดพลาด เมื่อสิ้นสุดการเรียนรู้ของ neural networks จะได้รับค่า weight ที่ปรับค่าให้เหมาะสมที่ทำให้เกิดค่าความผิดพลาดน้อยที่สุดจากการเรียนรู้นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

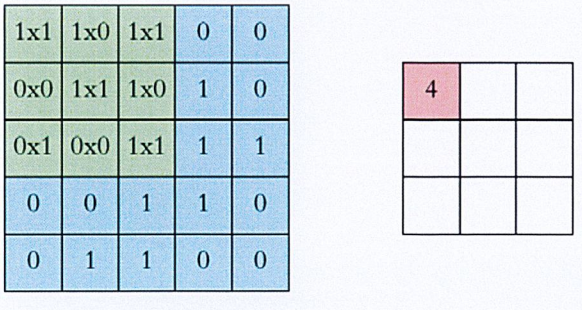


ภาพที่ 2.2 อัลกอริทึม Gradient Descent [8]

CNN (Convolutional Neural Networks) เป็นอัลกอริทึมที่เป็นนิยมมากที่สุดของ Neural Networks สำหรับการวิเคราะห์ข้อมูลภาพถ่าย หลักการทำงานของ CNN นั้นใช้หลักการของ Deep Learning โดยใช้เลเยอร์หลาย ๆ ชั้นในการหา features ของรูปภาพ เมื่อข้อมูลผ่านเลเยอร์หลายชั้น จะได้ลักษณะสำคัญของรูปภาพที่เป็น highler level features แต่ละชั้นจะมีตัวกรองที่เรียกว่า filter หรือ kernal อยู่ในแต่ละ node จะมีขนาดแตกต่างกันออกไปตามความเหมาะสมของ model input layer จะเชื่อมต่อกับ filter ที่แตกต่างกัน ดังนั้นจะสามารถดึง Features ที่แตกต่างกันกันของข้อมูลภาพที่รับมา filter จะเป็นตารางสองมิติ แล้วนำไปเทียบกับข้อมูลรูปภาพ pixel ต่อ pixel



ภาพที่ 2.3 ตัวอย่างของ filter [9]



ภาพที่ 2.4 ตัวอย่างการทำงานของ filter [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

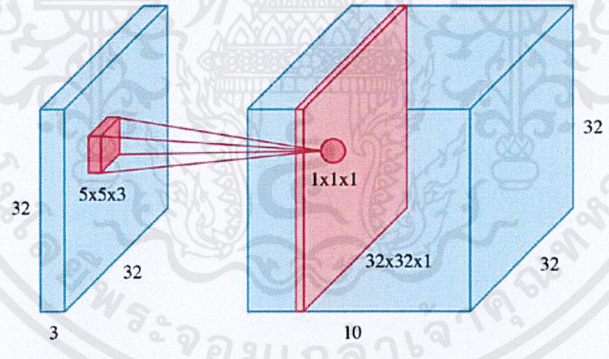
โดยแต่ละ filter จะเลื่อนไปทุก pixel ของข้อมูลภาพ แล้วทำการคูณเมทริกซ์ แล้วหาผลรวมจนครบทุก pixel ผลลัพธ์ที่ได้ออกมาจะเรียกว่า Feature map

1	1	1	0	0
0	1	1	1	0
0	0	1x1	1x0	1x1
0	0	1x0	1x1	0x0
0	1	1x1	0x0	0x1

4	3	4
2	4	3
2	3	4

ภาพที่ 2.5 ตัวอย่างการทำงานของ filter [9]

ซึ่งในปกติแล้วการทำ Convolutions จะถูกทำใน 3 มิติ คือภาพที่มีความสูง ความกว้าง และความลึก ความลึกจะมีความสอดคล้องกับช่องของสี (RGB)



ภาพที่ 2.6 ตัวอย่างการทำงานของ Convolution [9]

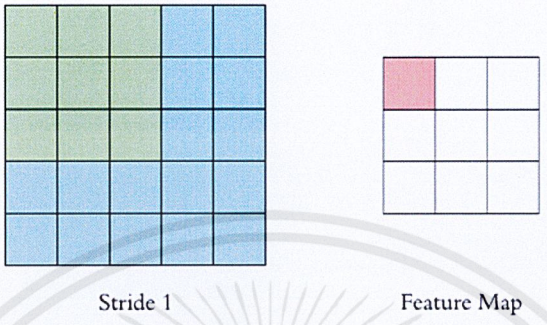
โดยภาพที่ 2.6 เป็นการที่ข้อมูลรูปภาพมีขนาด 32x32x3 แล้ว filter มีขนาด 5x5x3 แล้วก็จะทำขบวนการ convolution ทำผลรวมของการคูณเมทริกซ์ใน 3D โดย Feature map ที่ได้ จะมีขนาดเป็น 32x32x1 (ในกรณีมีการกำหนด padding เพื่อให้ได้ภาพขนาดเดิม)

ซึ่งในตัวอย่างจะเป็นการใช้ filter ที่แตกต่างกันสิบตัว จะได้ Feature map ขนาด 32x32x1 ทั้งหมด 10 feature map วางซ้อนกันตามมิติความลึก จะให้ผลลัพธ์สุดท้ายของชั้น convolution นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

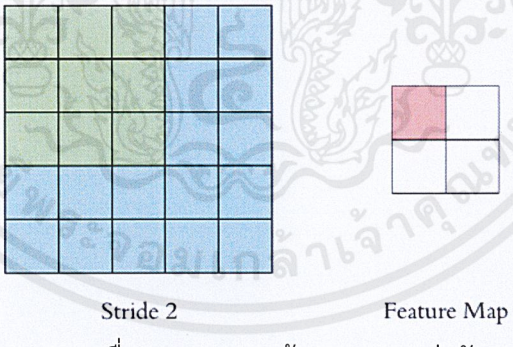
ใน Convolutions Neural Networks มีตัวแปรต่าง ๆ ที่สามารถเปลี่ยนได้ ที่จะส่งผลต่อผลลัพธ์ ที่อาจมีทั้งผลข้างเคียงในบางส่วน

Stride เป็นตัวแปรที่กำหนดว่า จะเลื่อน filter ซ้อนทับกับข้อมูลรูปภาพ ด้วยก้าวที่ยาวเท่าไหร่ โดยค่าเริ่มต้นคือ 1 ตัวอย่างในภาพที่ 2.7



ภาพที่ 2.7 การเลื่อนของ filter ด้วย stride เท่ากับ 1 [9]

ถ้าต้องการให้การทับซ้อนมีน้อยลง สามารถปรับจำนวนช่องที่จะเลื่อน filter ได้ ยิ่งช่องที่เลื่อนต่อครั้งใหญ่เท่าไหร่ features map ก็จะเล็กลงไปด้วย เหมือนดังภาพที่ 2.7 ที่ข้อมูลรูปภาพมีขนาดเป็น 5x5 และ Filter ที่มีขนาด 3x3 เลื่อนขยับด้วย Stride = 1 และภาพที่ 2.8 ข้อมูลรูปภาพมีขนาดเป็น 5x5 และ Filter มีขนาด 3x3 เลื่อนด้วย stride = 2

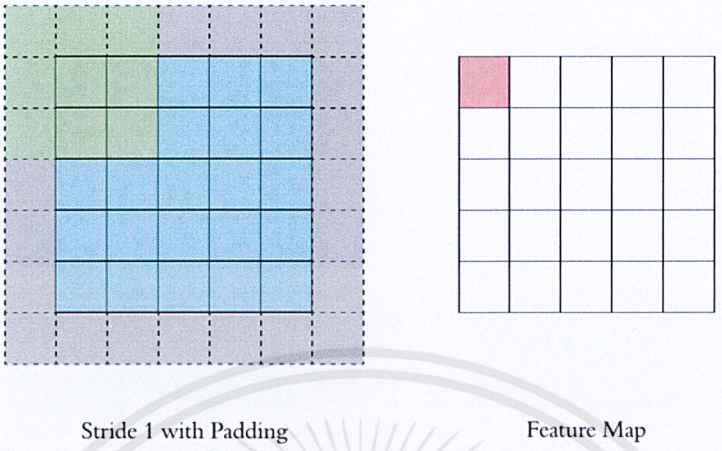


ภาพที่ 2.8 การเลื่อนของ filter ด้วย stride เท่ากับ 2 [9]

Padding เป็นการแก้ปัญหาของการที่บางข้อมูลที่ถูก filter ทับซ้อนหายไป เนื่องจากอยู่บนขอบภาพ ทำให้ filter ไม่สามารถ convolution ได้ วิธีที่ง่ายและมีประสิทธิภาพคือ Padding ในการแก้ปัญหานี้ โดยเติมข้อมูล 0 เพิ่มจากขอบออกไป เพื่อขยายขนาดออกไป ทำให้ filter สามารถ convolution ข้อมูลที่อยู่บนขอบได้ Feature map ที่ได้ออกมาก็จะมีขนาดเท่ากับ input ได้ ทำให้สามารถเก็บข้อมูลที่อาจเป็นส่วนสำคัญของข้อมูลรูปภาพไว้ได้ ยกตัวอย่างการทำ padding ดังภาพที่ 2.7 ข้อมูลรูปภาพมีขนาด 5x5 Filter มีขนาด 3x3 เลื่อนด้วย Stride =1 ปกติแล้ว feature map ที่ได้ออกมา

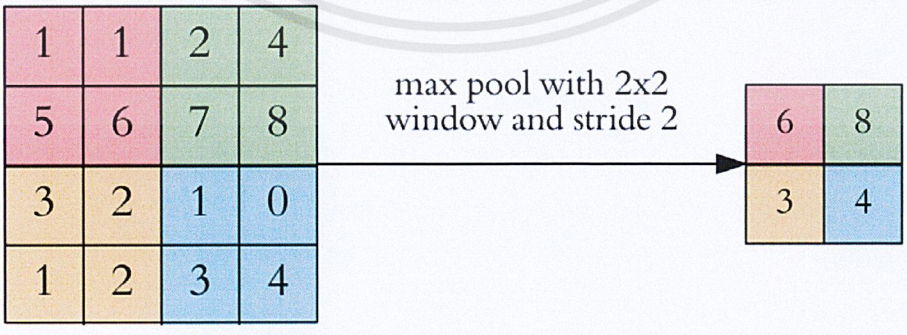
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ 3x3 แต่ถ้า padding ข้อมูลรูปภาพ จะมีขนาดเป็น 7x7 ทำให้สามารถ convolution pixel ที่อยู่บริเวณขอบได้ดังภาพที่ 2.9 Feature map ที่ได้จึงเป็น 5x5 ขนาดเท่าเดิมกับข้อมูลรูปภาพ



ภาพที่ 2.9 การ padding ของรูปภาพ [9]

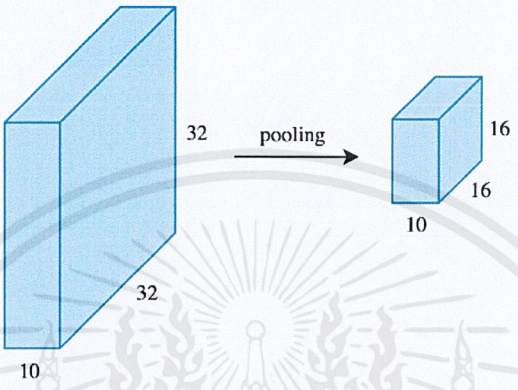
Pooling คือการรวม แนวคิดหลักก็คือการ down-sampling เพื่อลดความซับซ้อนของข้อมูลสำหรับชั้นถัดไป ถ้าเป็นรูปภาพ เป็นการลดความละเอียดของภาพลง พารามิเตอร์ก็จะลดลง ช่วยลดเวลาในการเรียนรู้และการover-fitting ที่มากเกินไป Max-Pooling เป็นหนึ่งใน Pooling ที่ใช้กันมากที่สุด หลักการทำงานของ Max-Pooling คือ จะมีกรอบสี่เหลี่ยมที่เหมือน filter ทาบไปยังบนข้อมูลรูปภาพ แล้วเปลี่ยนจากการคูณเมทริกซ์ เป็นการหาค่าที่มากที่สุดของในกรอบนั้นออกมาเป็นผลลัพธ์ ขนาดของกรอบที่ใช้กันมากที่สุดใน Max-Pooling คือขนาด 2x2 ตัวอย่างภาพที่ 10 เป็น Max-Pooling ขนาด 2x2 มี stride เท่ากับ 2



ภาพที่ 2.10 แนวคิดการใช้ pooling [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของการใช้ pooling นั้นดังภาพที่ 2.11 ข้อมูลรูปภาพมีขนาด 32x32x10 ใช้ Max-Pooling ขนาด 2x2 จะได้ feature map ขนาด 16x16x10 เห็นได้ว่าความสูง ความกว้างของ feature map หายไปครึ่งหนึ่ง แต่ความลึกจะไม่เปลี่ยนแปลง ทำให้สามารถลดค่าถ่วงน้ำหนักไปได้ ¼ ของข้อมูล เป็นเรื่องที่ดีสำหรับการทำ CNN



ภาพที่ 2.11 ตัวอย่างการใช้ pooling [9]

Hyperparameters ตัวแปรที่สำคัญที่ส่งผลต่อการตัดสินใจและผู้เขียนโมเดลจะต้องปรับค่าเอง CNN จะไม่ปรับค่าให้อัตโนมัติ ในตอนนี้จะพิจารณาจากชั้น convolution ก่อนโดยไม่สนใจชั้น pooling มีสี่ตัวแปรหลัก ๆ ด้วยกันคือ

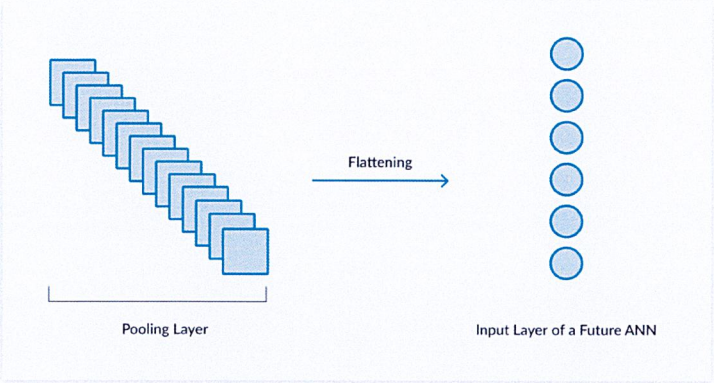
1) Filter size ส่วนใหญ่จะมีขนาดที่ 3x3 5x5 และ 7x7 ขึ้นอยู่กับการใช้งาน

2). Filter count เป็นตัวแปรที่ส่งผลต่อการตัดสินใจมากที่สุด โดยจะมีค่าเป็นสองยกกำลัง x หรือในระหว่าง 32 ถึง 1024 ซึ่งการใช้ filter จำนวนมากก็ทำให้การเรียนรู้มีประสิทธิภาพมากขึ้น แต่ถ้ามากเกินไปก็ทำให้เกิด overfitting

3.) Stride การก้าวของ filter ค่าเริ่มต้นคือ 1

4.) Padding โดยปกติมักจะใช้เพื่อรักษาข้อมูลไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

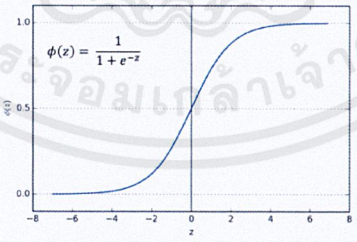


ภาพที่ 2.12 หลักการทำงาน Flatten [40]

หลังจากชั้นของขบวนการ Convolution แล้วจะเพิ่ม Fully connected ต้องทำ Flatten ของข้อมูลที่ออกมาจากขบวนการ convolution เพื่อแปลงขนาดมิติของข้อมูลจากชั้น Convolution ข้อมูลที่ออกมาเป็น 3 มิติ แล้วไปต่อด้วย Fully connected ที่เป็น 1 มิติได้ ซึ่งสามารถปรับหรือตัดผลลัพธ์ให้ได้ตามที่ต้องการ หรือก็คือ Activation Function

Activation Function แบ่งได้สองแบบ คือ linear function และ non-linear function แต่ว่า linear function เป็นแค่ฟังก์ชันธรรมดาที่ไม่มีความซับซ้อน จึงไม่นิยมนำมาใช้ Non-Linear function ที่นำมาใช้มีดังนี้

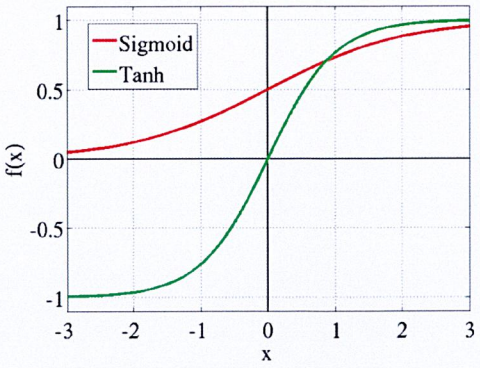
1) Sigmoid Function เป็นฟังก์ชัน S Curve ค่า output ที่ได้ออกมามีค่าเป็น 0 ถึง 1 และมีอนุพันธ์ที่ไม่เป็นลบ จึงเหมาะสมกับผลลัพธ์ค่าความน่าจะเป็น (Probability)



ภาพที่ 2.13 Sigmoid Function [12]

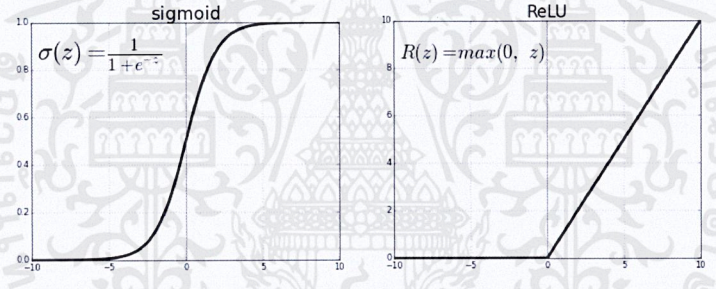
2) Tanh Function เป็นฟังก์ชันลักษณะคล้ายกับ Sigmoid ค่า output ที่ได้มีค่า -1 ถึง 1 ถ้า input ใกล้ 0 แล้วหาอนุพันธ์จะได้ความชันที่ชันกว่า sigmoid ทำให้ gradient ได้มากกว่า เหมาะสมกับการนำไปใช้ Classification ที่มี 2 class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.14 Tanh Function [12]

3) ReLU เป็นฟังก์ชันที่จะแปลงค่าที่ต่ำกว่า 0 เป็นผลลัพธ์ที่เท่ากับ 0 Input เป็นบวก Slope จึงเท่ากับ 1 ตลอด ทำให้ไม่ให้เกิด Vanishing Gradient หรือการที่ Gradient มีขนาดเล็กลงเรื่อย ๆ จนเท่ากับ 0



ภาพที่ 2.15 ReLU Function [12]

4) Softmax เป็นฟังก์ชันที่รับ input แล้ว normalize ออกมาเป็นความน่าจะเป็น ที่ผลรวมเท่ากับ 1 โดยมักนำไปไว้ layer สุดท้ายของ Neural Network เพื่อให้ output ออกมาเป็นความน่าจะเป็น เหมาะกับใช้ในงาน single class Classification

ดังนั้นที่นิยมในปัจจุบันก็คือ Rectified Linear Unit (ReLU) เพราะมีหลายเหตุผลคือ

1) ReLU มีคำจำกัดความที่ง่ายทั้ง ฟังก์ชันและ Gradient

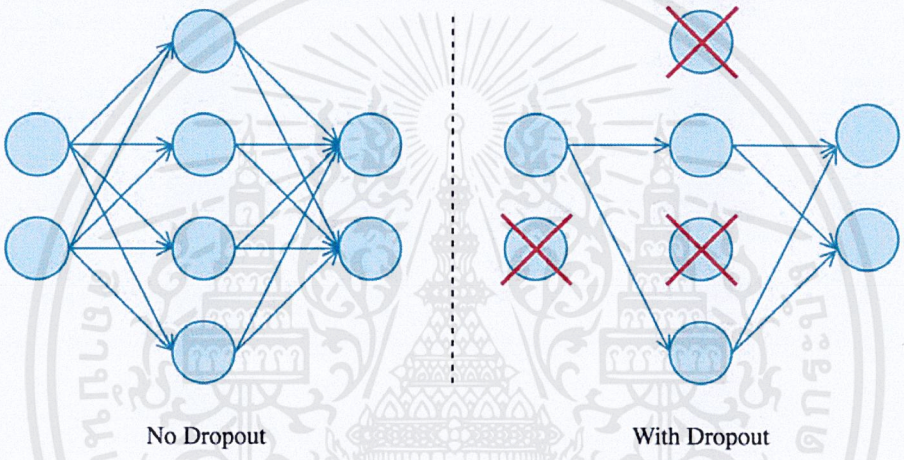
2) ฟังก์ชันอิ่มตัวบางฟังก์ชัน ทำให้เกิดปัญหาในการทำ Back propagation โดยเกิดขึ้นเมื่อ Gradient ของฟังก์ชันเหล่านั้นเข้าใกล้ศูนย์เกือบทุกที่ แต่อยู่ตรงกลาง ทำให้เกิด Vanishing Gradient

3) การหาอนุพันธ์ของ ReLU ไม่ยาก มีค่าเป็น 0 กับ 1 ขึ้นอยู่กับ input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การใช้ ReLU ทำให้การเรียนรู้ของ Deep Nerual Network เร็วมากขึ้น

มีเทคนิคที่นิยมนำมาใช้ใน Deep Neural networks เพื่อแก้ปัญหา Overfitting ก็คือ Dropout คือในขณะที่โมเดลกำลังเรียนรู้ในแต่ละรอบ node บางตัวจะถูกสั่งไม่ให้ทำงาน เพื่อทำให้ข้อมูลไม่ถูกส่งมาใน node นี้ โดยทั่วไปที่ตัวเลขของ Dropout ที่ใช้กันคือ 50 เปอร์เซ็นต์ ซึ่งทำให้ model ทำงานได้ดีขึ้น เหตุผลที่ใช้ก็เพื่อป้องกัน model ไปเจาะจงกับข้อมูลจำนวนที่น้อยเกินไป และทำให้ทุก node สามารถทำงานได้อิสระ นอกจากนี้ Dropout ปกติจะทำในขั้นตอนของการเรียนรู้ ขั้นตอนของการทดสอบการเรียนรู้ไม่นิยมใส่กัน



ภาพที่ 2.16 หลักการทำงาน Dropout [40]

ตัวอย่างการทำงานของ Dropout สมมติว่างานในบริษัท ซึ่งมีพนักงานคนหนึ่งทำส่วนหน้าที่ของการเขียนโปรแกรมอยู่คนเดียวในบริษัท โดยคุณบอกบริษัทไว้ว่าจะทำงานทุกวัน แต่เมื่อเกิดพนักงานคนนั้นเกิดขี้เกียจขึ้น โยนเหรียญหัวก้อยว่าจะไปทำงานหรือไม่ เพื่อนร่วมงานในบริษัทก็จะเกิดปัญหาขึ้น เมื่อพนักงานคนนั้นไม่มาทำงาน งานก็จะไม่เดินต่อ เพื่อนร่วมงานต้องปรับตัว เพราะไม่มั่นใจว่าวันไหนพนักงานคนนั้นจะไม่มาทำงาน จึงต้องเรียนรู้การเขียนโปรแกรมและแบ่งปันความรู้ที่เรียนเขียนโปรแกรมให้คนอื่น ๆ ด้วย เพื่อช่วยกันในการทำงานนี้ ซึ่งส่งผลดีต่อบริษัท ทำให้บริษัทมีความยืดหยุ่นขึ้น พนักงานทุกคนมีทักษะและคุณภาพในตัวเอง มีเทคนิคที่คล้ายกับ Dropout กำลังนิยมในช่วงนี้ก็คือ Batch Normalization

Batch Normalization เป็นการปรับ shift และ scale ของ Activation ที่อยู่ภายใน Hidden layer โดยการ normalizes output ด้วยการลบด้วยค่าเฉลี่ยแล้วหารด้วยค่าเบี่ยงเบนมาตรฐานของ ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละชั้นในโมเดลสามารถเรียนรู้ได้ด้วยตัวเอง เป็นอิสระจากกันมากขึ้น ข้อดีคือ ช่วยทำให้ Gradient ไหลได้ดีขึ้นและทำให้กำหนดค่า learning rate ได้มากกว่าเดิม

หลังจากการสร้างโมเดลแล้ว สิ่งที่ต้องคำนึงถึงต่อมาคือ loss function ซึ่งเป็นตัวชี้วัดโมเดล machine learning ว่าทำงานได้ดีหรือไม่ เป้าหมายของการทำโมเดลนั้น ต้องทำให้ loss ให้มีค่าต่ำที่สุด โดย loss นั้นคิดจากการทำนายผลลัพธ์ของโมเดล แล้วนำมาเปรียบเทียบกับคำตอบว่ามีความแตกต่างกันเท่าใด แต่มีวิธีการคิดที่แตกต่างกันไประหว่าง loss function อื่น ๆ ซึ่งในงาน classification นั้น ส่วนใหญ่เลือกใช้ loss function คือ cross entropy loss เป็น loss function ที่ใช้คู่กับ softmax activation

Cross Entropy Loss นั้นเป็นค่าแสดงตัวเลขความไม่แน่นอนของคำตอบ หรือเรียกว่าความน่าจะเป็น ยกตัวอย่างเช่น การทดสอบโมเดลโดยใช้รูปภาพสุนัข ซึ่งโมเดลได้เรียนรู้คลาสทั้งหมดสองคลาส คือ สุนัขและแมว โมเดลจะทำนายออกมาเป็น ความน่าจะเป็นที่รูปภาพจะเป็นสุนัข 80 เปอร์เซ็นต์ และความน่าจะเป็นที่รูปภาพจะเป็นแมว 20 เปอร์เซ็นต์

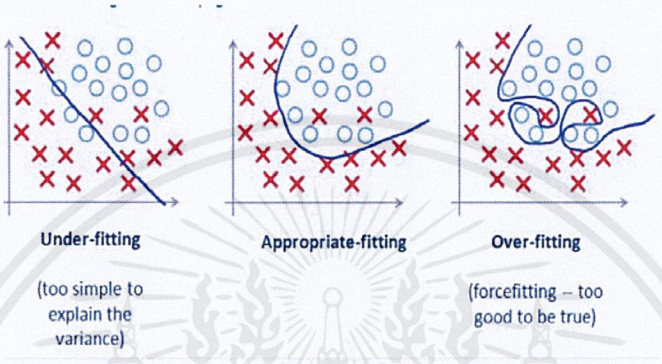
ต่อไปเป็นการกำหนดวิธีการ optimization ให้กับโมเดล ซึ่งในการเรียนรู้ในแต่ละรอบของโมเดลนั้น ต้องมีการปรับค่าถ่วงน้ำหนักและค่าอื่น ๆ เพื่อลดค่า loss ให้มากที่สุด ในการหาค่าต่ำสุดของ gradient นั้นมีสมการปรับค่าถ่วงน้ำหนักมากมาย ซึ่งในแต่ละสมการจะมีค่าตัวแปรหลัก ๆ คือ learning rate เป็นอัตราการปรับค่า เนื่องจากส่วนใหญ่ learning rate นั้นเป็นค่าที่ถูกนำไปคูณกับค่าถ่วงน้ำหนักในสมการ หาก learning rate มีค่าน้อยนั้น การเปลี่ยนแปลงของค่าถ่วงน้ำหนักก็จะเปลี่ยนแปลงน้อย loss จะเปลี่ยนแปลงน้อยเช่นกัน ถ้า learning rate นั้นมีค่ามาก การเปลี่ยนแปลงของค่าถ่วงน้ำหนักก็จะมากขึ้น loss เปลี่ยนแปลงมากขึ้น ถ้าพูดถึง optimizer ที่นิยมใช้กันมากที่สุดก็คือ Adam (Adaptive Moment Estimation) เป็น optimizer ที่จะปรับ learning rate ในแต่ละรอบการเรียนรู้ได้ และสามารถ decaying learning rate ได้ หรือเป็นการปรับเปลี่ยนค่า learning rate ตามจำนวนรอบ เนื่องจากการเรียนรู้ช่วงแรก ๆ นั้น หากใช้ learning rate ที่มีค่าน้อย จะทำให้การเรียนรู้เป็นไปได้ช้ากว่าจะถึงไปจุดต่ำสุด ดังนั้นจึงแก้ปัญหาโดยการเพิ่ม learning rate ในช่วงแรกหลังจากนั้นค่อย ๆ ลดลง เพื่อเพิ่มความเร็วของการเรียนรู้

แต่ไม่ได้แสดงว่า Adam นั้นเป็น optimizer ที่ดีที่สุด เพราะมีการออก optimizer ใหม่อยู่เรื่อย ๆ และยังมี optimizer ตัวอื่น ๆ ที่อาจเหมาะกับแต่ละโมเดลมากกว่า เช่น SGD RMSProp เป็นต้น

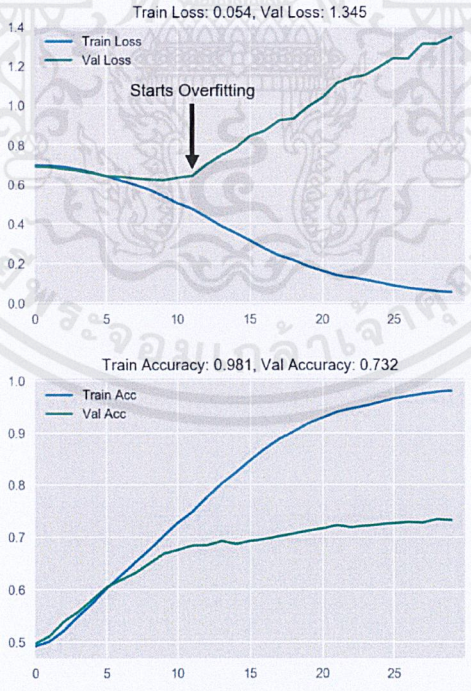
เมื่อมาพูดถึงประสิทธิภาพของโมเดลแล้ว จะดูกราฟ loss กับ ความแม่นยำ (Accuracy) จากภาพที่ 2.17 แสดงให้เห็นค่าความผิดพลาดของการเรียนรู้ (Training loss) ค่าความผิดพลาดการทดสอบ (Validation loss) ความแม่นยำของการเรียนรู้ (Training accuracy) และความแม่นยำของการทดสอบ (Validation accuracy) เห็นได้ว่าจากกราฟแสดงค่าความผิดพลาด จากจำนวน epoch ที่ 10 เป็นต้นไป ซึ่งหมายความว่าโมเดลกำลังเรียนรู้ข้อมูลได้แล้ว แต่ว่าเรียนรู้ในมุมมองที่ผิด โมเดลกำลังจำข้อมูลว่าต้องมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะยังไง ถึงจะได้ผลลัพธ์แบบนี้ แต่เมื่อไปเจอข้อมูลที่ไม่ใช่ข้อมูล training หรือข้อมูลที่ไม่ได้มีลักษณะเหมือนกับที่เรียนรู้ไป จะทำให้ผลของการทดสอบของโมเดลแย่ลง เรียกว่า Overfitting เปรียบเทียบได้เช่น มีข้อสอบชุดหนึ่ง นักเรียนได้ทำการสอบหลาย ๆ รอบ นักเรียนได้จำโจทย์ข้อสอบไปว่า ข้อนี้ต้องตอบอะไร เมื่อไปเจอโจทย์ใหม่ที่ไม่ได้จำมาก็จะไม่รู้ว่าต้องตอบอะไร เป็นต้น แต่มีอีกกรณีหนึ่ง โมเดลที่ไม่สามารถเรียนรู้อะไรได้จากข้อมูลเลย ไม่สามารถวิเคราะห์ลักษณะสำคัญของข้อมูลได้ ทำให้ไม่สามารถทำนายได้ไม่ดีมากนัก เรียกว่า Underfitting



ภาพที่ 2.17 กราฟแสดงการอธิบายของแต่ละ fitting [17]



ภาพที่ 2.18 ตัวอย่างกราฟการเรียนรู้ของโมเดลที่ overfitting [11]

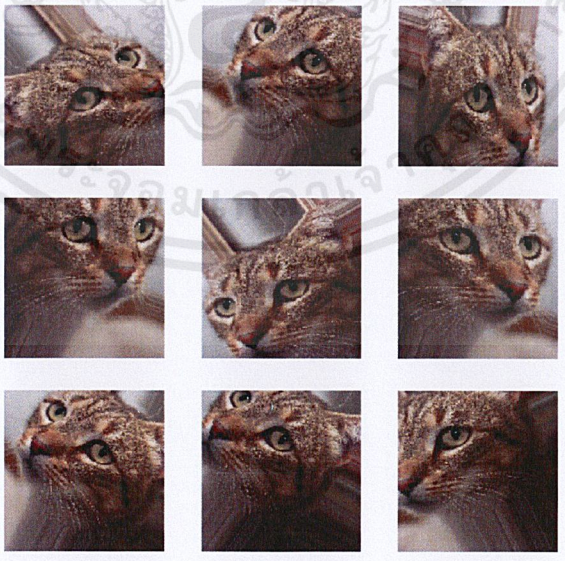
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีวิธีแก้ปัญหาได้หลากหลายวิธีด้วยกันในการทำให้ประสิทธิภาพของโมเดลดีขึ้นจากเดิม โดยวิธีในการแก้ปัญหา Overfitting นั้นแก้ได้หลายวิธี วิธีแรกก็คือ เทคนิคการ Dropout ที่ได้กล่าวไปแล้ว โดยเป็นการเพิ่มความยากของการเรียนรู้ ด้วยการปิดข้อมูลบางส่วนในชั้นของ node วิธีที่สอง คือการทำ Data Augmentation เป็นต้น

การทำ Data Augmentation เป็นการเพิ่มข้อมูลที่ใช้สำหรับการเรียนรู้ (Training) โดยใช้ข้อมูลที่มีอยู่แล้วมาสร้างเป็นข้อมูลใหม่ที่มีความแตกต่าง โดยไม่ทำให้ความหมาย หรือลักษณะของข้อมูลนั้น ๆ เสียหรือเพี้ยนไปมากเกินไป การทำ Data Augmentation เพื่อให้โมเดลเจอข้อมูลที่มีความหลากหลายมากขึ้น เช่น การหมุนขนาดของรูป การปรับขนาดของรูป การปรับแสงสีของรูป ความคมชัดของรูป เป็นต้น

ภาพที่ 2.19 ตัวอย่างรูปต้นฉบับที่นำ augment [11]

โดยจะนำรูปแมวที่เป็นต้นฉบับนั้นมาทำการหมุน แต่ว่าความหมายภาพก็ยังคงเดิมที่ว่า รูปก็ยังคงเป็นแมว ไม่ใช่อย่างอื่น



ภาพที่ 2.20 ตัวอย่างรูปหลัง augment [11]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่ใช้อ้างอิงคือ 11K Hands [21] ในงานนี้มีรูปภาพของที่พร้อมกับข้อมูลเพื่อใช้สำหรับ gender recognition และ biometric identification โดยใช้ข้อมูลเหล่านี้มาทำการ train ในขบวนการ CNN สำหรับการทำ gender recognition ส่วนโมเดลที่ได้เรียนรู้แล้วนำนำมาทำการแยก feature เพื่อป้อนเข้าไปใน support vector machine (SVM) เป็นอัลกอริทึมที่ใช้แก้ปัญหาการจำแนกข้อมูล สอนให้ระบบเรียนรู้ โดยเน้นไปการแยกแยะกลุ่มข้อมูลได้ดีที่สุด สำหรับการทำ biometric identification

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

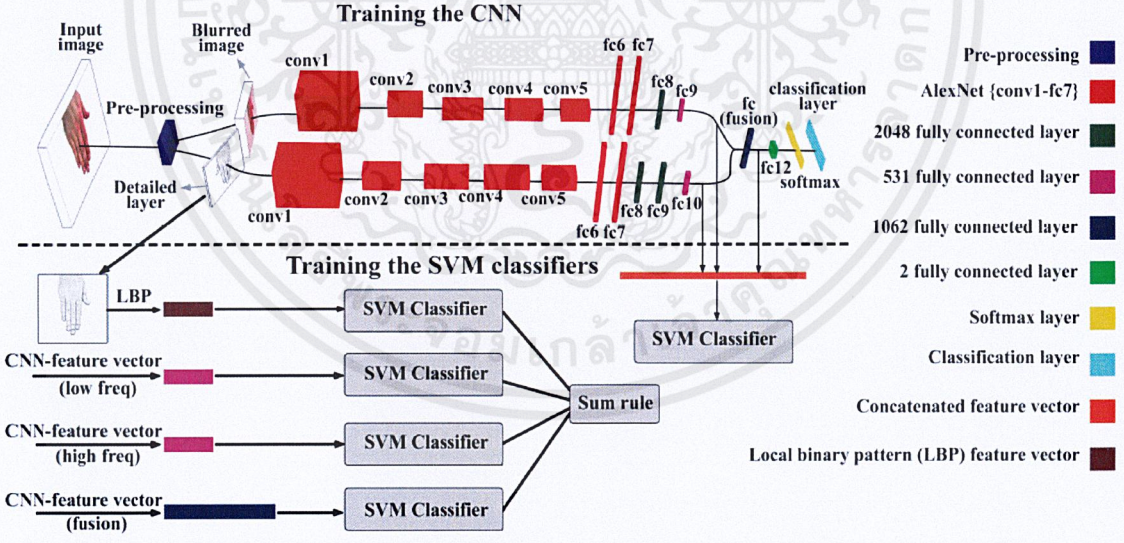
บทที่ 3

วิธีการดำเนินงาน

โปรเจกต์นี้เป็นการค้นคว้าวิธีการยืนยันตัวตนแบบใหม่ของทางทีม Biometrics ของทาง KBTG โดยได้แหล่งอ้างอิงจากงานวิจัยชื่อว่า 11K Hands ที่ได้ใช้ทำการทดสอบการใช้ Machine Learning ในการทำนายผลลัพธ์ของงานวิจัยการยืนยันตัวตนด้วยมือ โดยทำความเข้าใจแล้วนำมาดัดแปลงเพื่อใช้ในงานจริง

3.1 กระบวนการการทำความเข้าใจงานวิจัยแหล่งอ้างอิง

ในขั้นตอนแรกนั้นต้องทำความเข้าใจงานวิจัยว่ามีแนวคิดอย่างไร มีจุดประสงค์อะไร โดยงานวิจัยได้ระบุไว้ว่า มีการศึกษาหลายชิ้นที่แสดงให้เห็นว่าสามารถดึงคุณสมบัติที่โดดเด่นจากมือ ที่สามารถบ่งบอกถึงเพศได้ โดยเป็นข้อมูลในชุดที่รู้จักกันในชื่อว่า soft biometrics ในเร็ว ๆ นี้ได้มีการใช้ convolutional neural networks ในการจัดการคุณสมบัติของมือพร้อมกับ Dataset ของทางงานวิจัยเพื่อนำมาใช้กับโมเดลที่ได้สร้างขึ้นมา มีสองวัตถุประสงค์คือเพื่อการแยกเพศ กับการยืนยันตัวตน



ภาพที่ 3.1 Architecture โมเดลของ 11K Hands [21]

3.1.1 โมเดลที่ใช้ในงานวิจัย

เนื่องจากการแยกเพศกับการยืนยันตัวตนจากรูปภาพเป็นปัญหาการจำแนกที่มีความท้าทาย จึงจำเป็นต้องใช้ Deep learning ในการแก้ปัญหา ทาง 11K Hands ได้เลือกที่ใช้เป็น two-stream CNN ซึ่งเป็นการใช้ข้อมูลขาเข้าที่แตกต่างกันสองแบบที่มาจากรูปมือ ผ่านเข้าไปในกระบวนการ Convolutional

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Neural networks โดย feature บางส่วนจะถูกนำไปใช้ต่อในการเรียนรู้ของ SVM เพื่อแก้ปัญหาของการขาดข้อมูลในการเรียนรู้ของการยืนยันตัวตน (Biometric Identification)

3.1.2 Dataset ของทางงานวิจัย

ข้อมูลของงานวิจัยที่ทางนั้นได้รวบรวมมาคือจำนวนรูปมือจำนวน 11,076 รูป โดยมีขนาดแต่ละรูปที่ 1600x1200 pixels ได้มาจากรูปมือของคนจำนวน 190 คน โดยมีอายุเฉลี่ยอยู่ระหว่าง 18–75 ปี โดยแต่ละคนจะมีรูปฝ่ามือกับหลังมือทั้งซ้ายและขวาอยู่บนพื้นหลังสีขาวเหมือนกันทุกคน ระยะห่างจากกล้องเท่ากัน และมีข้อมูลที่บันทึกประกอบไปด้วย

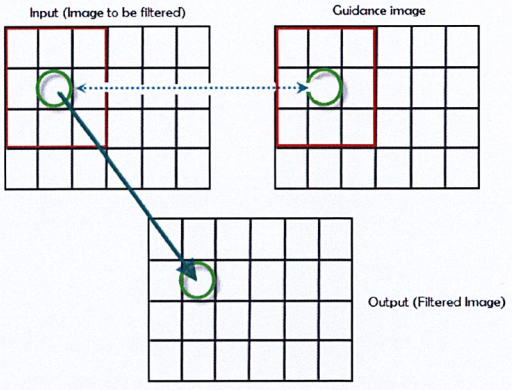
1) เลขไอดีของแต่ละคน
2) เพศ
3) อายุ
4) สีผิว
5) ข้อมูลเพิ่มเติมของรูป เช่น เครื่องประดับ สีน้ำยาทาเล็บ ความผิดปกติของมือ

3.1.3 การเตรียมข้อมูลของงานวิจัย

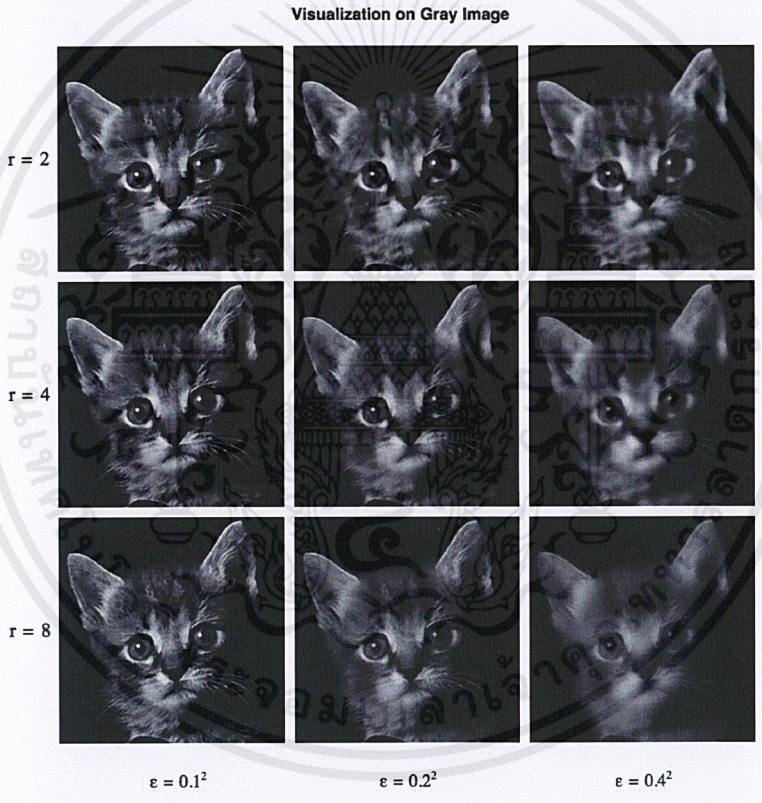
ข้อมูลก่อนจะนำไปใช้เป็นข้อมูลขาเข้าโมเดล จะเห็นได้ว่ามีสองข้อมูลขาเข้าที่ต้องใส่เข้าไปในโมเดล โดยสองรูปนั้นต้องเป็นรูปเดียวกัน แต่มีลักษณะที่แตกต่างกัน โดยรูปแรกจะเป็นรูปมีความละเอียดต่ำ เช่น blurred หรือ smoothed โดยใช้ filters ที่มีชื่อว่า Guided filter เพื่อทำให้รูปที่ได้เป็นรูปที่ smoothed จากรูปต้นฉบับ

Guided Filter เป็นเทคนิคที่ทำให้ขอบภาพเรียบขึ้น โดยใช้เนื้อหาของภาพที่สองที่เรียกว่า guidance image ไว้ใช้เป็น filter โดยจะเป็นภาพที่แตกต่างกับภาพที่หนึ่งเล็กน้อย หรือมาก ๆ โดยจะนำภาพ Guidance Image มาเป็นตัวชี้นำเหมือน filter อื่น ๆ โดยสมการเบื้องหลังของ Guided Filter มี parameters ให้ปรับหลัก ๆ อยู่สองตัว ค่าแรกคือ radius เป็นขนาดหน้าต่าง กับ ε *(epsilon)* เป็นค่าเล็กน้อยสำหรับให้สมการมีความเสถียร โดยในงานวิจัยนี้ได้ใช้ radius เท่ากับ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.2 การทำงานของ Guided filter [19]

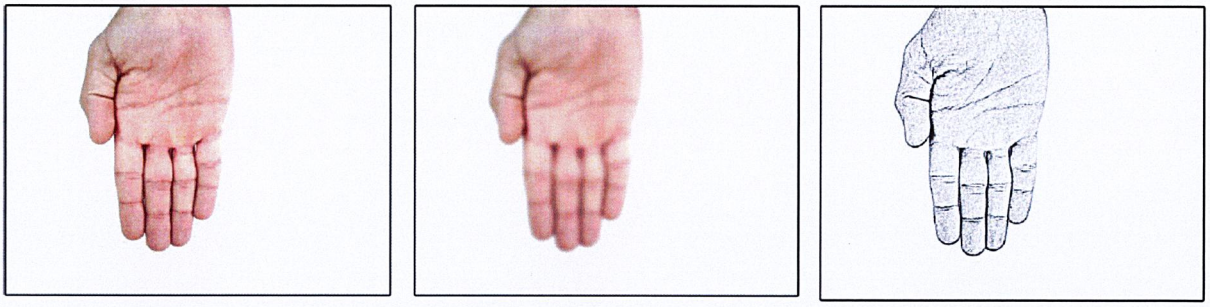


ภาพที่ 3.3 ตัวอย่างการทำงานของ Guided filter [20]

หลังจากได้ภาพ smoothed image มา จะนำภาพนี้มาดัดแปลงโดย pixel-wise division เพื่อให้ได้ภาพที่มี detailed layer ตามสมการ

$$Detailed\ layer = \frac{original\ image}{smoothed\ image + \varepsilon}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a) Input hand image (b) Smoothed version (c) Detailed layer

ภาพที่ 3.4 ตัวอย่างภาพ detailed layer [21]

3.1.4 กระบวนการ Training ของงานวิจัย

การเรียนรู้ของ convolution neural networks โดยใช้โมเดลที่กล่าวไว้ในหัวข้อ 3.1.1 ผลลัพธ์ที่โดยใช้ค่าถ่วงน้ำหนัก (weights) จากการ pre-trained AlexNet เป็นค่าเริ่มต้นของตัวแปรการเรียนรู้ หลังจากนั้น เมื่อผลลัพธ์ออกมาจาก two-stream CNN จะมาทำการ convergence กัน โดยในขบวนการ pre-trained กำหนด learning rate λ เท่ากับ 10^{-4} และหลังออกจาก pre-trained กำหนด learning rate λ เท่ากับ $2x10^{-3}$

ส่วนของการ train SVM นั้นก็เอาผลลัพธ์จาก 4 ชั้นมาทำ SVM classification แล้วนำมาหาผลรวม โดย 4 ชั้นนั้นได้แก่ detailed layer, CNN-feature vector (low frequency), CNN-feature vector (high frequency) และ CNN-feature vector (fusion)

3.1.5 ผลลัพธ์ของ Biometric Identification Training

SVM classifiers ทั้ง 4 ตัวมีประสิทธิภาพมากกว่าการใช้ single SVM classifier ที่ใช้ combined CNN-feature vector โดยตารางที่ 3.1 เป็นผลลัพธ์จากการทดสอบกับ datasets ของพวกเขาเอง โดยใช้จำนวนคนที่แตกต่างกันออกไป แล้วแบ่งออกเป็นหน้ามือกับหลังมือ ส่วนตารางที่ 3.2 นั้นได้ลองนำมาใช้กับ IITD dataset ซึ่งเป็นข้อมูลที่มีจุดประสงค์คล้ายกัน โดยผลลัพธ์จากการทดสอบจากต้นฉบับ นำมาเทียบกับโมเดลของ 11K Hands ได้แสดงว่าตัวเองทำผลลัพธ์ได้ความแม่นยำที่ดี แสดงว่าโมเดลไม่ได้จำกัดอยู่ที่ dataset ของ 11K Hands เพียงอย่างเดียว สามารถนำไปใช้กับ dataset ที่แตกต่างกันได้

Method	(80-P)	(100-P)	(120-P)	(80-D)	(100-D)	(120-D)
CNN-features/SVM	**0.948**	0.929	0.933	**0.964**	**0.960**	**0.962**
CNN-features+LBP/SVM	**0.960**	**0.953**	**0.956**	**0.962**	**0.967**	**0.97**

ตารางที่ 3.1 ผลลัพธ์ของการทดสอบกับ datasets ของงานวิจัย 11K Hands โดย (N-S) ในตาราง หมายถึง ความแม่นยำโดยเฉลี่ยที่ใช้ ด้านของมือ (S) และจำนวนคนที่นำมาใช้เป็นจำนวนทดลอง (N) [21]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Method	Features	$S = 230$	$S = 137$	$S = 100$
Kumar and Shekhar [20]	Gabor	**0.95**	-	-
Kumar and Shekhar [20]	Radon transform	0.928	-	-
Charfi *et al.* [6]	SIFT	0.94	-	-
Bera *et al.* [5]	finger contour profile	**0.952**	**0.978**	-
Charfi *et al.* [7]	standard SIFT	0.803	-	0.852
Charfi *et al.* [7]	SIFT+SR	-	-	**0.962**
Ours	CNN-features	0.9	0.912	0.891
Ours	CNN-features+LBP	**0.948**	**0.964**	**0.94**

ตารางที่ 3.2 ผลลัพธ์การทดสอบโมเดลต่าง ๆ กับ IITD dataset โดยมีจำนวนคน (S) 230 คน, 137 คน และ 100 คน ตามตาราง [21]

3.2 กระบวนการเตรียมข้อมูล

การยืนยันตัวตนด้วยมือนั้น ข้อมูลที่ใช้สำหรับโครงงานนี้ก็ต้องเป็น datasets ที่มีรูปภาพมือ พร้อมกับการระบุตัวตนของเจ้าของมือไม่ว่าจะวิธีการใด ๆ เช่น การตั้งชื่อ การทำไฟล์ csv เป็นต้น เพื่อให้สามารถรู้ว่าเป็นภาพของคนใด จึงจะสามารถบอกคำตอบให้กับโมเดลได้ในการเรียนรู้ โดยได้รวบรวมข้อมูลจากแหล่งต่าง ๆ ดังนี้

3.2.1 Datasets ของ 11K Hands

จากข้อ 3.1.2 นั้น datasets ของ 11K Hands มีจำนวนรูปทั้งหมด 11,076 รูป โดยมี metadata ที่เป็นไฟล์ csv และไฟล์ json ซึ่งเป็นข้อมูลที่บันทึกชื่อไฟล์แต่ละรูปทั้งหมด และสามารถระบุตัวตนของแต่ละ ID ได้ ซึ่งแต่ละ ID จะมีรูปภาพมือทั้งด้าน palmar และ dolsar

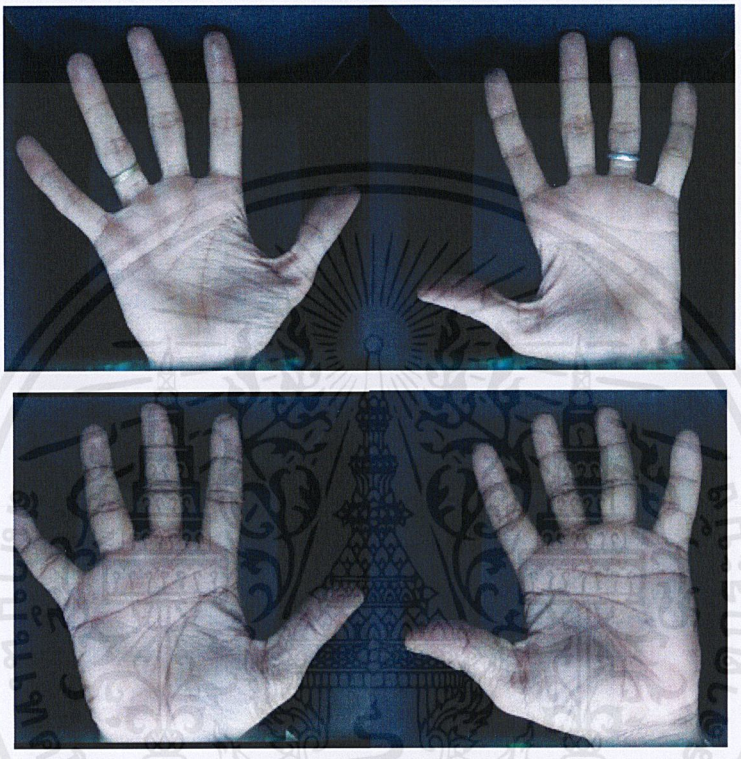
	A	B	C	D	E	F	G	H	I
1	id	age	gender	skinColor	accessories	nailPolish	aspectOfHand	imageName	irregularities
2	0	27	male	fair	0	0	dorsal right	Hand_0000002.jpg	0
3	0	27	male	fair	0	0	dorsal right	Hand_0000003.jpg	0
4	0	27	male	fair	0	0	dorsal right	Hand_0000004.jpg	0
5	0	27	male	fair	0	0	dorsal right	Hand_0000005.jpg	0
6	0	27	male	fair	0	0	dorsal right	Hand_0000006.jpg	0
7	0	27	male	fair	0	0	dorsal right	Hand_0000007.jpg	0
8	0	27	male	fair	0	0	dorsal right	Hand_0000008.jpg	0
9	0	27	male	fair	0	0	dorsal right	Hand_0000009.jpg	0
10	0	27	male	fair	0	0	dorsal right	Hand_0000010.jpg	0
11	0	27	male	fair	0	0	dorsal right	Hand_0000011.jpg	0
12	0	27	male	fair	0	0	dorsal right	Hand_0000012.jpg	0
13	0	27	male	fair	0	0	dorsal right	Hand_0000013.jpg	0
14	0	27	male	fair	0	0	dorsal right	Hand_0000014.jpg	0
15	0	27	male	fair	0	0	dorsal right	Hand_0000015.jpg	0
16	0	27	male	fair	0	0	dorsal right	Hand_0000016.jpg	0
17	0	27	male	fair	0	0	dorsal right	Hand_0000020.jpg	0
18	0	27	male	fair	1	0	dorsal left	Hand_0000021.jpg	0
19	0	27	male	fair	1	0	dorsal left	Hand_0000022.jpg	0
20	0	27	male	fair	1	0	dorsal left	Hand_0000023.jpg	0

ภาพที่ 3.5 Metadata ของ 11K Hands ไฟล์ csv ที่เก็บข้อมูลต่าง ๆ ของคน 189 คน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Dataset ของ IITD

ภาพมือทั้งหมดถูกรวบรวมในสภาพแวดล้อมในร่มและใช้หลอดไฟฟลูออเรสเซนต์แบบกลมรอบเลนส์กล้อง มีจำนวนคนทั้งหมด 235 เป็นช่วงคนอายุ 12-57 ปี ซึ่งแต่ละคนจะมีภาพคนละ 7 ภาพ ทั้งซ้ายและขวา ซึ่งแต่ละรูปมีบอกตัวเลขระบุ ID ขนาดรูปแต่ละรูปมีขนาด 800x600 pixels แต่ว่าภาพทั้งหมดเป็นภาพด้าน palmar side เท่านั้น

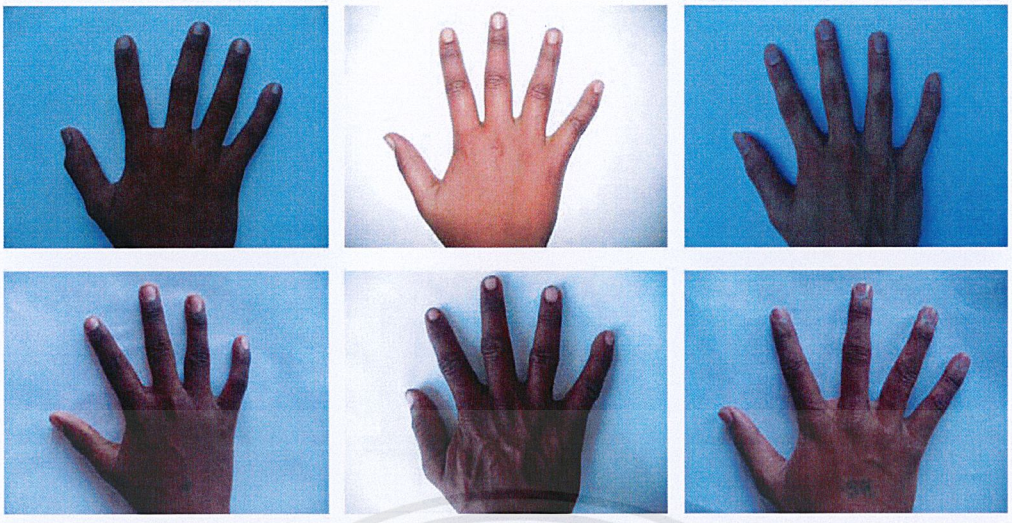


ภาพที่ 3.6 ตัวอย่าง Dataset ของ IITD [21]

3.2.3 Dataset ของ Hongkong University

เป็น dataset ของรูป dorsal hand ที่รวบรวมมาจากอาสาสมัครผู้ชายและผู้หญิง มีรูปหลังมืออยู่ทั้งหมด 2,505 รูปที่มาจากมือขวาของคนที่แตกต่างกัน 501 คน และมีรูปเพิ่มเติมอีกจำนวน 211 คน แต่ว่ารูปไม่มีความชัดเจนที่พอใช้ได้ จึงไม่ได้นำมาใช้ในการทดลอง

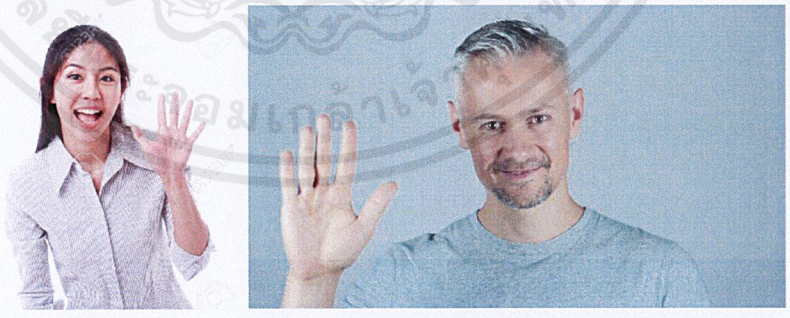
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.7 ตัวอย่าง Dataset ของ IITD [22]

3.2.4 ข้อมูลที่เก็บมาจาก website

ในการหาข้อมูลที่สามารถเข้าถึงคนได้ง่ายในยุคนี้ คือ ข้อมูลจาก website โดยสามารถหาสิ่งที่ต้องการได้จากการป้อน keyword ที่ต้องการไป ซึ่งสิ่งที่ต้องการคือรูปภาพมือ ซึ่งการหาว่ารูปภาพมือระบุตัวตนของบุคคลนั้นไม่สามารถทราบได้ เพราะว่ามีจำนวนข้อมูลที่น้อยเกินไปจึงเป็นข้อมูลที่นำไปใช้ในการเรียนรู้ของโมเดลการยืนยันตัวตนไม่ได้ แต่มีข้อมูลรูปภาพบางส่วนที่สามารถระบุตัวตนได้ จากการค้นหารูปภาพจากบุคคล บุคคลหนึ่งที่มีรูปจำนวนมาก จะสามารถบันทึกข้อมูลได้ว่าเป็นรูปภาพมือของบุคคลใด



ภาพที่ 3.8 ตัวอย่าง รูปภาพใน Dataset ที่มาจาก website ที่ไม่สามารถระบุตัวตนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3.9 ตัวอย่าง รูปภาพใน Dataset ที่มาจาก website ที่สามารถระบุตัวตนได้

3.2.5 ข้อมูลรูปภาพจากกล้องมือถือ

รูปภาพจากการถ่ายด้วยกล้องมือถือ โดยเป็นภาพถ่ายของบุคคลใกล้ตัวจึงสามารถระบุตัวตนได้ คนหนึ่งคนมีรูปอย่างน้อย 5 รูป โดยมีพื้นหลังหรือสภาพแวดล้อมที่แตกต่างกันออกไป ตำแหน่งของมือที่แตกต่างกัน แต่ละคนจะมีรูปภาพทั้งหน้ามือและหลังมือ ด้านซ้ายและด้านขวา ไม่มีการกำหนดลักษณะการกางนิ้ว เป็นรูปภาพที่เป็นอิสระ

ภาพที่ 3.10 ตัวอย่างรูปภาพจากกล้องโทรศัพท์ที่มีสภาพแวดล้อมที่แตกต่างกัน

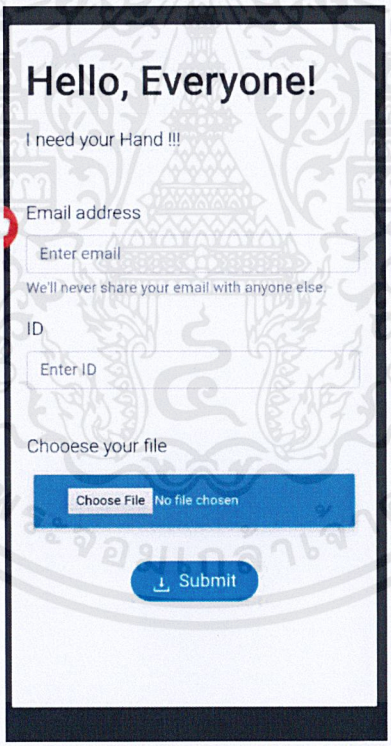
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 ข้อมูลที่จากการรวบรวมด้วยตัวเองจากการทำ website

เนื่องจากการยืนยันตัวตนต้องการรูปภาพมือ พร้อมด้วยข้อมูลที่สามารถระบุตัวตนได้ ผู้จัดทำจึงได้สร้าง web application เพื่อเก็บข้อมูลดังกล่าว เนื่องจากสามารถเข้าถึงบุคคลอื่น ๆ ได้ง่าย ไม่ว่าจะทางโทรศัพท์ คอมพิวเตอร์ ก็สามารถเข้าถึงได้ โดย web application ที่จัดทำขึ้นนั้นแบ่งออกเป็นสองส่วนดังนี้

3.2.6.1 ส่วนของ Front-End

ใน Front-End ได้เลือกใช้ Platform ที่ชื่อว่า React Native ซึ่งเป็น Open source จากทีมผู้พัฒนา Facebook มี library ให้เลือกใช้มากมายและไม่มีค่าใช้จ่ายในการใช้ จึงเป็นที่นิยมในปัจจุบัน โดยสิ่งที่ต้องการคือข้อมูลที่สามารถระบุตัวตนได้และรูปภาพ จึงทำ web application ออกมาในรูปแบบฟอร์มการเก็บข้อมูล โดยให้ผู้ใช้จำเป็นต้องกรอก Email address, ID (ในที่นี้เป็นเลขประจำตัวนักเรียน หรือ เลขประจำตัวของพนักงาน) และใส่รูปภาพ แล้วนำข้อมูลมาแปลงเป็นรูปข้อมูลฟอร์ม JSON เพื่อติดต่อกับ Back-End โดยใช้ Rest API



ภาพที่ 3.11 ตัวอย่างของ website ที่จัดทำ

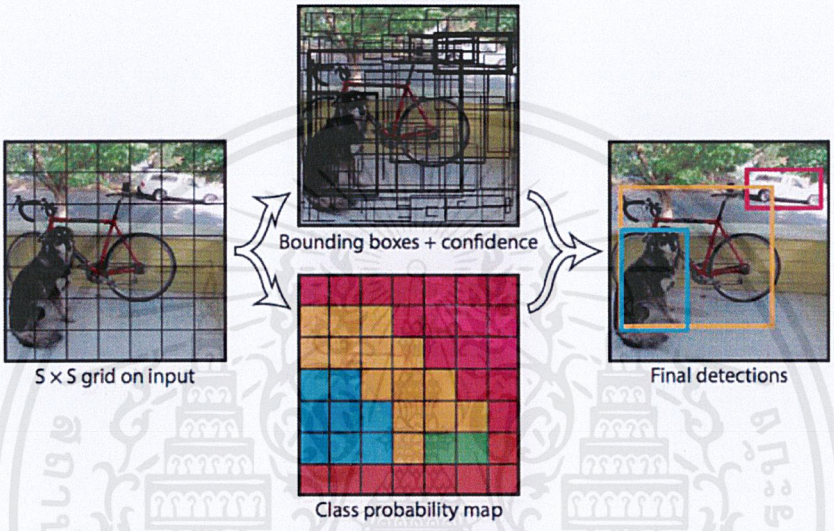
3.2.6.2 ส่วนของ Back-End

ใช้ส่วนของการทำงานเบื้องหลังนั้น ได้เลือกใช้ framework คือ Flask เป็น framework สำหรับพัฒนาระบบ web application ที่ใช้ภาษา python โดยการทำงานของ Back-End นั้นจะเป็น service ที่รับข้อมูลที่ส่งมาจาก Front-End โดยใช้ Rest API นำข้อมูล JSON มาแปลงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วนำมาเก็บไว้ในเครื่องของ Host ที่เปิดเอาไว้และบันทึกข้อมูลต่าง ๆ ลงใน Database ข้อมูลรูปภาพที่ได้รับมานั้นจะถูกเก็บออกเป็นสองแบบ แบบแรกคือรูปภาพปกติที่ผู้ใช้อัพโหลดมา แบบที่สองคือรูปที่ผ่าน Hand Detection จะได้รูปที่ตัดเฉพาะมือออกมาจากแต่ละรูป

Hand Detection นั้นเป็น service ที่ไว้ทำการทำนายว่ารูปที่ใส่ลงไปแต่ละรูปนั้นมีมืออยู่ส่วนไหนของรูป โดย Hand Detection นั้นใช้อัลกอริทึมที่มีชื่อว่า Yolo (You only look one) เป็นการทำนาย bounding box และความน่าจะเป็นของคลาสต่าง ๆ ของโมเดลไปพร้อมกัน

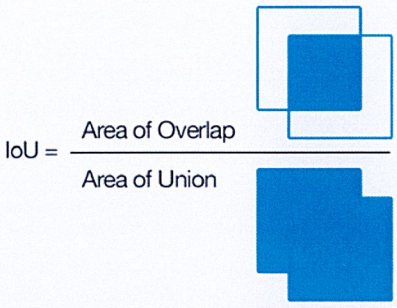


ภาพที่ 3.12 ตัวอย่างการทำงานของอัลกอริทึม YOLO [36]

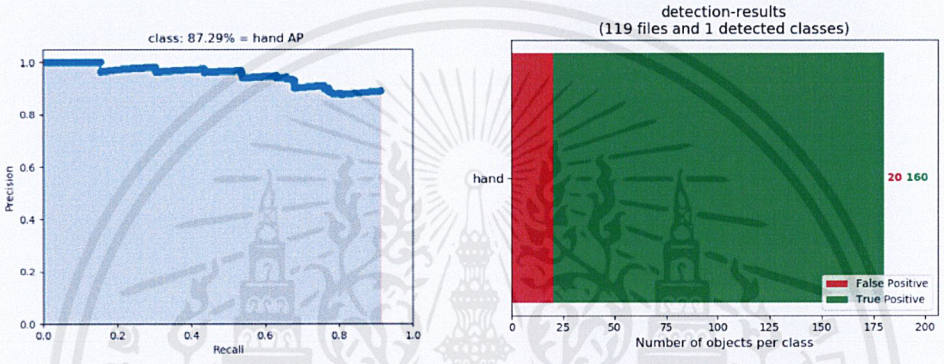
ซึ่งโมเดลต้องเรียนรู้คลาสก่อนจึงจะสามารถทำนายคลาสนั้นออกมาตามที่ต้องการได้ ผู้จัดทำจึงนำ Datasets ของ 11K Hands จำนวน 1,158 รูป รูปภาพจาก website และรูปภาพจากกล้องโทรศัพท์ เป็นรูปภาพทั้งหมด 2,939 รูป นำรูปภาพทั้งหมดไป train ด้วยโมเดลของ Yolo แล้วนำโมเดลไปทดสอบด้วยการทำ mAP (Mean Average Precision) ใน github [24] โดยนำข้อมูลอีกชุดที่ไม่ได้อยู่ใน training set มาทำการทดสอบ

mAP เป็นเกณฑ์ที่กำหนดไว้ในการแข่งขัน PASCAL VOC 2012 ซึ่งจะคำนวณ AP (Average Precision) ของแต่ละคลาส โดยใช้หลักการ IoU (Intersection over Union) เอาพื้นที่ที่กรอบ intersect กันออกมาหารด้วยพื้นที่กรอบทั้งหมด ถ้าค่า Iou เกิน 0.5 แสดงว่ากรอบที่ทำนายสามารถเจอคลาสนั้นได้ แล้วนำค่า AP ของทุกคลาสมาหาค่าเฉลี่ยจึงสามารถหาค่า mAP ออกมาได้ ซึ่งคลาสที่นำไป train มีเพียงคลาสเดียว คือ มือ ทำการทดสอบด้วยรูปภาพมือจำนวน 119 รูป ทุกรูปมีจำนวนมือตั้งแต่ 1 เป็นต้นไป แล้วได้ค่า mAP ออกมาอยู่ที่ 87.29 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.13 ภาพแสดงหลักการ IOU [41]



ภาพที่ 3.14 ผลลัพธ์ของการทดสอบด้วย mAP

3.2.6.3 การเปิดใช้งาน website

การเปิด website ให้คนทั่วไป สามารถเข้ามาใช้งานได้ ได้เลือกใช้เครื่องมือที่มีชื่อว่า ngrok โดยทำให้สามารถ ให้คนทั่วไปเข้า website ที่รันอยู่บนเครื่อง localhost ได้ โดยทั้ง Front-End และ Back-End รันอยู่บนเครื่อง localhost ไม่ได้ใช้เว็บโฮสติ้งและไม่ได้ทำการจดโดเมน เครื่องมือ ngrok นี้จะทำการสร้าง url โดยสุ่มขึ้นมาเพื่อให้คนทั่วไปใช้ url ในการเข้าถึง website สาเหตุที่เลือกใช้ เพราะว่า ngrok เหมาะสำหรับการ demo ทำให้สามารถทดลอง website โดยไม่ต้องเสียค่าใช้จ่าย ไม่จำเป็นต้องไปเสียค่าใช้จ่ายกับการเช่า cloud แต่ก็มีข้อเสียคือ มีความปลอดภัยไม่แน่นหนา ถ้าจัดการเรื่อง permission ไม่ดีพอ จะทำให้ผู้ไม่ประสงค์ดีสามารถเข้ามาจัดการไฟล์ในเครื่อง localhost ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ขั้นตอนการ Train CNN models

การยืนยันตัวตนนั้น ต้องหาลักษณะสำคัญของมือ โดยไม่บอกว่าเครื่องจักรว่าให้ดูลักษณะอย่างไร จึงจำเป็นต้องใช้วิธีการ Deep Learning ผู้จัดทำได้เลือกใช้ Deep Learning Library ที่ชื่อว่า Keras เป็น Library ในภาษา Python ที่มี Back-End ทั้ง Tensorflow และ Theano ซึ่งเป็น Deep learning Library ที่มีประสิทธิภาพทั้งคู่ ทำให้ Keras นั้นเป็นที่นิยมและมีประสิทธิภาพที่สูงในการรันโมเดลเช่นกัน โดย Keras นั้นมี library ให้ใช้จำนวนมากสำหรับหลายกระบวนการเช่น ในการสร้างโมเดล data pre-processing เป็นต้น เพียง import library ที่ต้องการใช้

```
from keras import backend as K
from keras.layers import Dropout, BatchNormalization, Conv2D, Input, PReLU, MaxPool2D, Flatten, Dense,
from keras.models import Model,load_model,Sequential
from keras import optimizers
```

ภาพที่ 3.15 ตัวอย่างการ import library

การทำ Data Pre-processing นั้นใช้ Hand Detection ในการทำให้รูปมีขนาดเล็กลง ทำให้กระบวนการเรียนรู้เร็วขึ้น โดย Datasets ที่ใช้ คือ 11K Hands กับ IITD dataset นำมารวมกัน โดยผู้จัดทำได้นำเฉพาะภาพหน้ามือ (Palmar side) มาทำการ train โดยทำการบันทึกเป็นรูปภาพใหม่หลังจากได้ผ่าน Hand Detection รูปภาพที่ได้ออกมาจะมีขนาดเป็นสี่เหลี่ยมจัตุรัส หรือก็คือ มีความกว้างและความสูงเท่ากัน โดยในการเรียนรู้นั้น ต้องแบ่งข้อมูลออกเป็นสามส่วนด้วยกัน คือ training set, validation set และ test set ส่วนแรก training set คือเป็นส่วนที่ให้โมเดลได้เรียนรู้ แล้วทำการปรับค่าถ่วงน้ำหนักโดยดูจากค่าความผิดพลาดการทดสอบ (Validation loss) จากการทดสอบข้อมูลด้วย validation set ขณะเรียนรู้ในแต่ละรอบ (epoch) เพื่อต้องทำให้ค่าผิดพลาดการเรียนรู้ (Training loss) ลดลงให้ได้มากที่สุด โดยการแบ่งข้อมูลนั้น ได้นำ test set ที่เหมือนกันกับของ 11K Hands ซึ่งเป็นรูปมือของ 120 คน เป็นรูป palmar side ทั้งหมด แล้วนำรูปภาพที่เหลือหลังจากตัด test set ออก มาแบ่งเป็นส่วนที่มีรูปภาพไม่เท่ากัน แบ่งรูปภาพทั้งหมดของคนหนึ่งคนเป็น training set ประมาณ 70 เปอร์เซ็นต์ และ validation set 30 เปอร์เซ็นต์ แต่ทว่าคนบางคนมีจำนวนรูปภาพทั้งหมดเป็นเลขที่หาร 70 เปอร์เซ็นต์ไม่ลงตัว จึงนำส่วนที่หารไม่ลงตัวมารวมอยู่ใน validation ซึ่งทำให้อัตราส่วนของการแบ่งเป็นประมาณ 67 และ 32 เปอร์เซ็นต์ตามลำดับ

เมื่อได้ข้อมูลออกมาเป็นส่วน ๆ แล้ว นำมาทำ Data Augmentation เพื่อเพิ่มความหลากหลายของรูป เนื่องจากบางที dataset ที่มีนั้นไม่ได้มีจำนวนมากและหลากหลายเพียงพอ Data Augmentation เป็นประโยชน์ในการแก้ปัญหานี้ เป็นกระบวนการดัดแปลงข้อมูลด้วยหลากหลายวิธีและเพิ่มลงในชุดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในที่นี้ได้เลือกใช้ Image Preprocessing ซึ่งเป็นของ Keras โดยมี generator ที่ดึงข้อมูลจากโฟลเดอร์ได้ และฟังก์ชันที่สามารถทำ augmentation ได้ที่มีชื่อว่า ImageDataGenerator

Generator คือ ฟังก์ชันปกติทั่วไป แต่มีตัวสำคัญคือ yield ซึ่งทำหน้าที่คล้าย return ของฟังก์ชันจะคืนค่าบางอย่างกลับคืนไป เมื่อ generator โดนเรียกมันจะทำงาน จะเริ่มต้นทำงานและคืนค่ากลับไป แต่จะจดจำไว้ว่า ทำงานถึงส่วนไหนแล้ว เมื่อโดนเรียกทำงานอีกครั้งหนึ่ง จะเริ่มทำงานต่อจากค่าที่จดจำไว้ เหตุผลที่ต้องใช้งาน generator เนื่องจากการทำงานกับชุดรูปภาพซึ่งมีขนาดใหญ่ การที่จะโหลดชุดข้อมูลที่มีขนาดใหญ่นั้นมาเก็บไว้ในตัวแปร python คอมพิวเตอร์จะต้องทำงานหนักขึ้น ต้องใช้เมมโมรี่ (Ram) จำนวนมาก เลยไม่ใช่ทางเลือกที่ดี generator จึงถูกนำมาใช้เพื่อดึงชุดข้อมูลออกมาทีละชุด ที่มีขนาดเล็กลงตามที่กำหนด ยกตัวอย่างเช่น มีชุดรูปภาพทั้งหมด 50,000 รูป ใช้ generator ในการดึงรูปภาพออกมาทีละชุด (batch) ชุดละ 50 รูปแรก เมื่อเรียกใช้ generator ครั้งถัดไป จะดึงชุด 50 รูปต่อจากชุดแรก

ImageDataGenerator สามารถ augmentation ให้กับข้อมูลได้หลากหลายวิธี วิธีที่ใช้คือ

1) rescale โดยปกติรูปภาพมีขนาด pixel แต่ละ pixel อยู่ที่ 255 แต่จะทำให้โมเดลเรียนรู้ได้ช้าลง จึงนำมา normalize ให้แต่ละ pixel หารด้วย 255 ทำให้ค่าอยู่ในระหว่าง 0 ถึง 1 ถึงรูปภาพจะมีขนาด pixel ลดลง แต่รูปจะยังถูกมองคล้ายเดิมอยู่ ไม่มีลักษณะเปลี่ยนแปลง
2) horizontal_flip กำหนดให้สุ่มทำการหมุนแบบ horizontal เพราะว่าต้องการไม่ให้โมเดลเรียนรู้สับสนระหว่างรูปภาพของคนเดียวกันที่เป็นมือคนละฝั่ง (ซ้ายและขวา) ต้องการให้โมเดลเรียนรู้ลักษณะสำคัญของมือแต่ละคน แทนที่เป็นการจำรูปภาพ
3) rotation_flip กำหนดให้สุ่มหมุนรูปตามองศาที่กำหนดให้ ต้องการเพิ่มความหลากหลายให้กับข้อมูล ทำให้ทิศทางมืออยู่แตกต่างกัน ซึ่งอาจเป็นทั้งข้อดีและข้อเสียได้เช่นกัน ข้อดีก็คือโมเดลอาจเรียนรู้ลักษณะสำคัญได้ดีขึ้น เพราะว่าข้อมูลมีความแตกต่างกัน ข้อเสียคือการที่เพิ่มความหลากหลาย โมเดลจะเรียนรู้ได้ยากขึ้น ค่าที่กำหนดคือ 180 องศา รูปจะสุ่มหมุนระหว่าง 0 ถึง 180 องศา
4) brightness_range กำหนดให้สุ่มความสว่างของรูป ค่าที่กำหนดคือ 0.6 ถึง 1 รูปที่ได้จะเป็นรูปที่มีความสว่างปกติ ไปถึงงง รูปที่มีความว่างน้อยลงถึง 0.6 เท่า
5) fill_mode เป็นการกำหนดให้พื้นที่ว่างที่เกินขอบเขตรูปมาจากการ augmentation ต่าง ๆ เช่น การหมุนของรูปที่จะทำให้มีพื้นที่ว่างขึ้นมา กำหนดว่าให้พื้นที่เป็นอะไร โดยค่าที่กำหนดคือ constant พื้นที่ว่างจะเป็นสีดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
train_datagen = ImageDataGenerator(rescale=1./255,
                                   horizontal_flip=True,
                                    rotation_range=180,
                                    brightness_range=[0.6,1],
                                    fill_mode="constant"
                                   )
```

ภาพที่ 3.16 ตัวอย่างการใช้ ImageDataGenerator

เมื่อสร้าง train_datagen ซึ่งเป็น Prepossing จากการใช้ ImageDataGenerator แล้ว ทำการสร้าง train_generator ทำหน้าที่เป็น generator โดยใส่พารามิเตอร์ต่าง ๆ โดยใช้คำสั่ง flow_from_directory จะเป็นการสร้าง generator โดยนำข้อมูลมาจากโฟลเดอร์ โดยกำหนดค่าพารามิเตอร์ต่อไปนี้

1) พารามิเตอร์แรกที่ต้องใส่คือ directory เป็นการบอกที่อยู่ของโฟลเดอร์ข้อมูลที่ต้องการ ภายในโฟลเดอร์ต้องมีโฟลเดอร์ย่อยที่มีรูปอยู่ข้างใน โดยแบ่งตามคลาสที่ต้องการจะแบ่ง ซึ่งในที่นี้โฟลเดอร์ย่อยเป็นไอดีของคนแต่ละคน มีรูปมือ palmar side อยู่ข้างใน
2) พารามิเตอร์ที่สองคือ target size ขนาดของรูปภาพที่ดึงจากโฟลเดอร์ต้องการให้มีขนาดเท่าใด ค่าที่กำหนดคือ 224x224
3) พารามิเตอร์ที่สามคือ batch size กำหนดขนาดของชุดข้อมูล จะดึงข้อมูลมาทีละกี่รูปในโฟลเดอร์ ค่าที่กำหนดคือ 20
4) พารามิเตอร์ที่สี่คือ class mode กำหนดประเภทของ array คำตอบที่ต้องการจะ return โดยค่าที่กำหนดคือ categorical ทำให้คำตอบเป็น 2D one-hot เป็นรูปแบบข้อมูลที่เหมาะสมกับอัลกอริทึม ML เพื่อให้การทำงานดีขึ้น ยกตัวอย่างเช่น มีคลาสทั้งหมดคือ A B C ถ้าคำตอบของรูปแรกคือคลาส ฤ คำตอบแบบ 2D one-hot ของคำตอบของรูปแรกคือ 1, 0, 0 ตามลำดับ ถ้าคำตอบของรูปที่สองคือคลาส B คำตอบแบบ 2D one-hot ก็คือ 0, 1, 0 ตามลำดับ
5) shuffle ค่าที่กำหนดเพื่อให้ชุดข้อมูลไม่มีการสุ่มหยิบเกิดขึ้น จะหยิบตามลำดับข้อมูลที่อ่านได้
6) seed ค่าที่กำหนดเผื่อในกรณีสำหรับการ shuffle และการสุ่มค่าของการเปลี่ยนแปลงรูปภาพนั้น ให้มีผลลัพธ์เหมือนเดิมเสมอตามค่าที่กำหนด seed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
train_generator = train_datagen.flow_from_directory(
    directory=train_dir,
    target_size=(224,224),
    batch_size=50,
    class_mode='categorical',
#     color_mode='grayscale'
    shuffle=False,
     seed=1
    )
```

```
Found 5080 images belonging to 419 classes.
```

ภาพที่ 3.17 ตัวอย่างการใช้ flow_from_directory ในการทำ generator

เมื่อได้ train_generator แล้ว ในส่วนของการ validation ทำคล้ายกันเพียงแต่การ preprocessing นั้นกำหนดเพียงแค่ค่า rescale หารด้วย 255 เท่านั้น หลังจากนั้นจะนำ generator นั้นนำมาสร้าง generator อีกฟังก์ชันหนึ่ง เพราะว่าโมเดลที่ใช้จะเป็นโมเดลที่ต้องการ 2 ข้อมูลขาเข้า แต่ตอนนี้ generator ส่งมาเพียงค่าเดียว ดังนั้นจึงสร้าง train และ valid generator เพิ่มมาอีก generator หนึ่ง แต่ทำการเปลี่ยนพารามิเตอร์ อันดับแรกคือ directory รูปภาพที่นำมาจะไม่ใช่รูปภาพแบบเดิม แต่เป็นรูปภาพที่นำไป preprocessing โดยใช้ filter kernel ซึ่งเป็น matrix ขนาด 3x3 ไปคูณแต่ละ pixel ด้วยขนาดที่สอดคล้องกับ kernel แล้วนำมาหาผลรวม จะทำให้รูปภาพมีขอบที่ชัดขึ้น (sharpen) ซึ่งจะทำให้สามารถเห็นความแตกต่างชัดเจนระหว่างสิ่งของได้ ต่อไปเพิ่มพารามิเตอร์เข้าไปใน generator คือ color_mode เนื่องจากรูปภาพที่ได้จาก preprocessing นั้นเป็นรูปภาพขาวดำ และ color channel เพียงช่องเดียว จึงต้องกำหนดให้ color_mode เป็น grayscale

0	-1	0
-1	5	-1
0	-1	0

ภาพที่ 3.18 Sharpen kernel ขนาด 3x3 [42]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3.19 ตัวอย่างการใช้ sharpen kernel ขนาด 3x3 [42]

เมื่อได้ generator ทั้งสอง input เรียบร้อยแล้ว นำมาใช้เป็นส่วนหนึ่ง generator ที่ใช้สำหรับการ training โดยเขียนฟังก์ชัน generator ให้รับค่าจาก generator ข้อมูลทั้งสองตัว แล้ว yield ข้อมูลออกไปในรูปแบบสองรูปภาพหนึ่งคำตอบ เนื่องจากทั้งสองรูปภาพเป็นคลาสเดียวกัน

```
def generate_data_generator():
    while True:
        X1i = train_generator.next()
        X2i = train_generator2.next()
        yield [X2i[0],X1i[0]], X1i[1]

def generate_val_generator():
    while True:
        Y1i = val_generator.next()
        Y2i = val_generator2.next()
        yield [Y2i[0] ,Y1i[0]], Y1i[1]
```

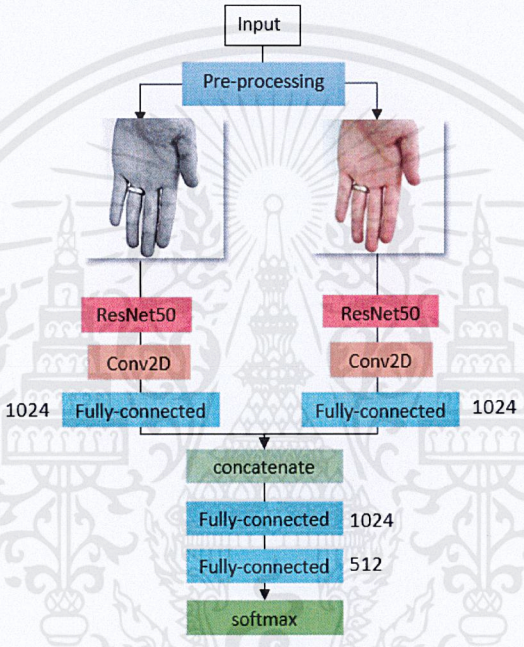
ภาพที่ 3.20 Generator ที่ใช้สำหรับการเรียนรู้ของโมเดล

การจดจำรูปภาพเพื่อนำไปใช้การยืนยันตัวตน โดยใช้ CNN (Convolutional neural network) ในการวิเคราะห์รูปภาพ เพื่อหาความสัมพันธ์ของ feature ของรูปภาพ ทดสอบสร้างโมเดลโดยอ้างอิงจากโมเดลของงานวิจัย 11K Hands โมเดลที่สร้างเป็น two-stream CNN รับข้อมูลขาเข้า 2 ตัว ในกระบวนการ CNN นั้นได้ใช้ Transfered Learning หรือ Pre-trained เป็นการนำความรู้หรือค่าถ่วงน้ำหนักจากโมเดลอื่นที่เรียนรู้มาแล้ว ที่มีจุดประสงค์คล้ายคลึงกัน ก็คือทำ Image Classification ผู้จัดทำได้เลือกโมเดลจาก ResNet50 ซึ่งมีความลึกของโมเดลมากกว่า VGG16 และ VGG19 กำหนด include_top เป็น False ทำให้นำเอา fully-connected layer ของ ResNet50 ออก แล้วนำมาต่อด้วย fully-connected layer ที่สร้างขึ้น ทั้ง two-stream แล้วนำมา concatenate กันต่อด้วย fully-connected layer อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทุก fully-connected layer ใช้ Relu เป็น activation function แล้วชั้นสุดท้ายใช้ softmax เป็น activation function มีคลาสทั้งหมด 419 คลาส หรือ 419 คน ใช้ optimizer เป็น adam , learning rate เท่ากับ 10^{-5} และ losss function ใช้เป็น categorical crossentropy

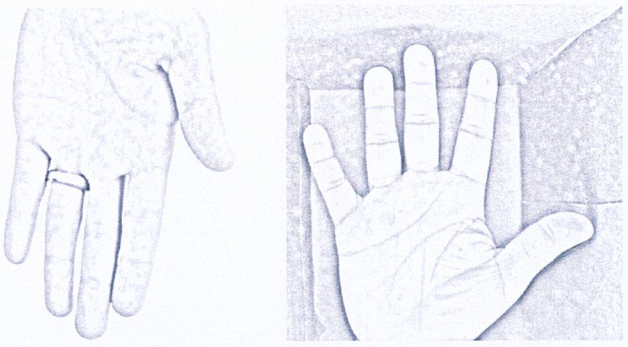
ทำการเทรนโดนใช้ fit_generator ใส่ training set, validation set จาก generator ที่สร้างขึ้นเอง ต่อมากำหนด step_per_epoch, validation_step จำนวนครั้งเรียก generator ของ training set, validation set ตามลำดับ คือ 254, 121 และจำนวนรอบที่ใช้ในการเรียนรู้ 300 epoch ซึ่งได้ทำการปรับจูนโมเดลโดยการเพิ่มจำนวนชั้น layer จำนวน filter ค่า learning rate เป็นต้น จนได้ผลลัพธ์ที่พึงพอใจ



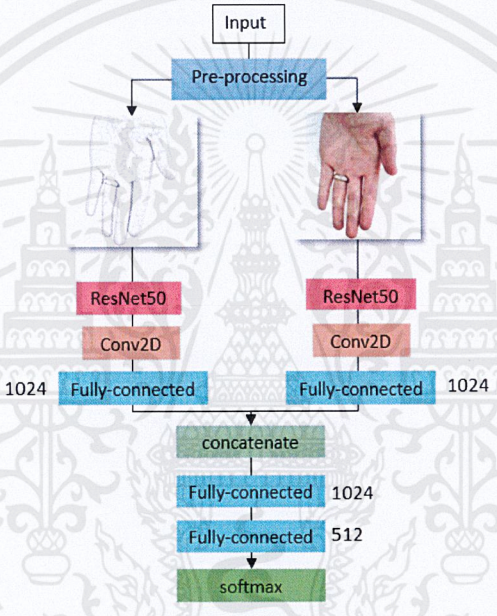
ภาพที่ 3.21 Architecture ของโมเดล

หลังจากได้โมเดลตั้งต้น จึงได้ทำการทดลองเปลี่ยนพารามิเตอร์ โดยใช้หลักการเดียวกับงานวิจัย 11K Hands คือรูป detailed layer โดยนำรูปต้นฉบับมาหารด้วยรูปที่ smoothed บวกด้วยค่าเล็กน้อย รูป smoothed นั้นใช้ guided filter โดยใช้รูปต้นฉบับนำด้วยรูปต้นฉบับกำหนดค่า radius เท่ากับ 16 และ ε เท่ากับ 0.01 แล้วนำรูป detailed layer ที่ได้นั้นมาใช้แทนรูปภาพ grayscale เพื่อดูผลลัพธ์ของความแตกต่างระหว่างข้อมูลขาเข้า ซึ่งในการ train กำหนด step_per_epoch, validation_step เป็น 495, 256 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.22 ตัวอย่างภาพ detailed layer



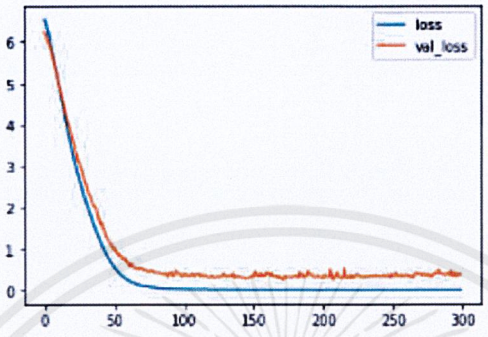
ภาพที่ 3.23 Architecture ของโมเดลที่ใช้รูปภาพ detailed layer

พารามิเตอร์ต่อมาที่ทำการทดลองคือ เลเยอร์ชั้นที่สุดท้ายที่เป็นชั้น classification เนื่องจากโมเดลตั้งต้นนั้นเป็นการ classify มือทั้งซ้ายและขวาของแต่ละบุคคลเป็นคลาสเดียวกัน พบว่ามือทั้งสองข้างควรเป็นคนละคลาส เพราะว่ารายละเอียดของมือทั้งสองนั้นมีความแตกต่างกันอยู่ ถ้าให้โมเดลเรียนรู้อาจทำให้มีความสับสนในการจำแนก features ของมือได้ผิดพลาด จึงแก้คลาสจากเดิมที่มือซ้ายและมือขวาเป็นคลาสเดียวกันกลายเป็นคนละคลาส และแก้จำนวนคลาสของชั้น softmax ในเลเยอร์ชั้นสุดท้ายใหม่ เมื่อทำการแบ่งคลาสใหม่ จะได้จำนวนคลาสทั้งหมด 835 คลาส ซึ่งโฟลเดอร์ที่ใช้ในการแบ่งคลาสของ ImageDataGenerator ต้องเปลี่ยนด้วยเช่นกัน และพารามิเตอร์ใน generator ที่จำเป็นต้องเปลี่ยนเมื่อใช้คลาสแบบนี้คือ horizontal_flip เนื่องจากจะทำให้รูปภาพมีคำตอบที่ผิดเพี้ยนไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นนำพารามิเตอร์เหล่านี้มา train ด้วยโมเดลตั้งต้น แต่เปลี่ยนพารามิเตอร์ใน generator เล็กน้อยคือ นำการสุ่มหมุนออกไป แต่ยังคงให้สุ่มแสงของรูปภาพอยู่ เนื่องจากคาดว่าให้โมเดลมีความซับซ้อนลดลง เรียนรู้ได้ง่ายขึ้น โดยกำหนดให้ learning rate เท่ากับ 10^{-5} และจำนวนรอบเป็น 300

3.4 ผลลัพธ์ของการ Train CNN models



ภาพที่ 3.24 กราฟผลลัพธ์ของโมเดลที่ทำการ train

โมเดลตั้งต้นขณะกำลังเรียนรู้ loss ทั้ง training loss และ validation loss ลดลงเรื่อย ๆ คู่กันไป แล้วค่อยคงที่เมื่อผ่านไปประมาณ 100 epoch เมื่อโมเดลทำการเรียนรู้เสร็จสิ้น นำโมเดลที่ได้ทดลองกับชุด test set ที่แบ่งเอาไว้ จำนวน 120 คน เป็นจำนวน 480 ภาพ โดยใช้ evaluate ในการทำนายคำตอบจากชุดข้อมูลที่ใส่ลงไป ได้ผลลัพธ์ดังตารางที่ 3.3

Method	Test Accuracy (%)
Our model	94.1
Model 11K Hands CNN/SVM	93.3
Model 11K Hands CNN+LBP/SVM	95.6

ตารางที่ 3.3 ผลลัพธ์การทดสอบด้วยโมเดลที่ได้เปรียบเทียบกับโมเดลของ 11K Hands

ต่อมาได้ทำการปรับโมเดลโดยเปลี่ยนพารามิเตอร์ของรูปภาพขาวดำและจำนวนคลาสในชั้น softmax ซึ่งมี architecture โมเดลคล้ายกัน จึงนำผลลัพธ์มาเปรียบเทียบกัน แบ่งออกได้เป็น 4 โมเดล คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.) โมเดลที่ใช้รูปภาพขาวดำจากการใช้ sharpen kernel และการแบ่งคลาสแบบที่นับมือซ้ายและมือขวาเป็นคลาสเดียวกัน มีจำนวนคลาสทั้งหมด 419 คลาส

2.) โมเดลที่ใช้รูปภาพขาวดำจาก detailed layer และการแบ่งคลาสแบบที่นับมือซ้ายและมือขวาเป็นคลาสเดียวกัน มีจำนวนคลาสทั้งหมด 419 คลาส

3.) โมเดลที่ใช้รูปภาพขาวดำจากการใช้ sharpen kernel และการแบ่งคลาสแบบที่นับมือซ้ายและมือขวาเป็นคนละคลาสกัน มีจำนวนคลาสทั้งหมด 835 คลาส

4.) โมเดลที่ใช้รูปภาพขาวดำจาก detailed layer และการแบ่งคลาสแบบที่นับมือซ้ายและมือขวาเป็นคนละคลาสกันมีจำนวนคลาสทั้งหมด 835 คลาส

เนื่องจากโมเดลที่มีจำนวนคลาสที่แตกต่างกันนั้นสามารถนำมาเปรียบเทียบกันได้ เพราะว่าได้ทำการทดลองนำโมเดลที่แบ่งคลาสแบบที่นับมือซ้ายและมือขวาเป็นคนละคลาส มาทำการเปลี่ยนคำตอบของแต่ละรูปในข้อมูล test set ที่ได้นำมาทดลองกับโมเดล โดยสามารถเปรียบเทียบมือซ้ายและมือขวาของคนเดียวกันเป็นคลาสเดียวกันได้ จากนั้นทำการทดลองอีกครั้ง แล้วทำให้เห็นได้ว่า ผลลัพธ์มีค่าเท่ากัน

Method	Test Accuracy (%)
419 class + sharpen kernel	92.50
419 class + detailed layer	91.87
835 class + sharpen kernel	91.25
835 class + detailed layer	90.20

ตารางที่ 3.4 ผลลัพธ์การทดสอบด้วยโมเดลการเปลี่ยนพารามิเตอร์ต่าง ๆ

นำโมเดลที่ได้มาทดลองกับ datasets อื่น ๆ เพื่อทดสอบว่าเมื่อเจอข้อมูลของคนอื่นที่ไม่เรียนรู้มาก่อน จะสามารถแยกความต่างออกจากกันได้หรือไม่ ทำการใช้ embedding features ในการแยกความต่างของลักษณะของข้อมูล โดยตัดชั้น Output layer ของโมเดลออกไป หรือก็คือชั้น softmax ที่ไว้ classify ว่าเป็นมือของบุคคลใด ชั้นก่อนหน้าจะเป็นชั้น embedding features ที่แยกคุณลักษณะของมือไว้ก่อนจะนำมา classify แล้วนำมาคำนวณหาระยะทางของแต่ละรูปที่ output ออกมาเป็น embedding features โดยคนหนึ่งคนต้องมีรูปสองรูป นำมา predict หา embedding features ของแต่ละรูป แล้วนำข้อมูลทั้งหมดมาเทียบกัน หากระยะทางของ embedding features ค่าความต่างของ embedding ควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะน้อยที่สุด เมื่อสองรูปที่ถูกเปรียบเทียบเป็นมือของคนเดียวกัน ถ้าค่าความต่างของคนเดียวกันไม่ได้น้อยที่สุด แสดงว่าไม่สามารถแยกคุณลักษณะได้

	A - 1	B - 1	C – 1	D – 1	E - 1
A - 2	0.0092	1.548	1.425	1.368	1.566
B - 2	1.554	0.0068	1.687	1.778	1.425
C - 2	1.014	1.213	1.548	1.678	1.444
D - 2	1.876	1.223	1.246	0.0876	1.546
E - 2	1.101	1.021	1.432	1.324	0.0976

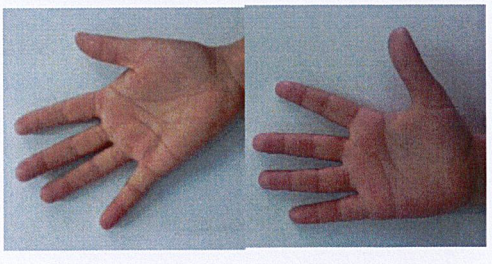
ตารางที่ 3.5 ตัวอย่างของการตรวจสอบของการแยกลักษณะ

ตารางที่ 3.5 เป็นการใช้ระยะทางของ embedding features เปรียบเทียบกัน โดยมี ID คนคือ A, B, C, D, E มีรูปคนละสองรูป หากมีระยะทางน้อยที่สุดเทียบกับรูปของคนอื่น จะแสดงว่าแยกได้ถูกต้อง (สีเขียว) หากรูปของคนเดียวกันไม่ได้ระยะทางน้อยที่สุด คือแยกไม่ได้ (สีแดง)

เมื่อนำจำนวนที่โมเดลทำนายถูกต้องมาทำเป็นเปอร์เซ็นต์เพื่อให้รู้ประสิทธิภาพของโมเดลต่อ dataset นั้น ๆ

โดยมี Dataset ที่นำมาทดสอบทั้งหมด 3 dataset

1.) ข้อมูลของ Mutah University โดยนำข้อมูลรูปภาพมาทั้งหมด 50 คน เป็นจำนวนรูปทั้งหมด 100 รูป โดยเป็นรูปภาพจากกล้องโทรศัพท์มือถือ โดยรูปภาพจะมีลักษณะมือระยะห่างต่าง ๆ มีพื้นหลังส่วนใหญ่เป็นสีพื้นสีเดียวเช่นสีขาว และภาพของคนหนึ่งคนมีลักษณะที่แตกต่างกันน้อยมาก โอกาสทำนายถูกต้องจึงสูง



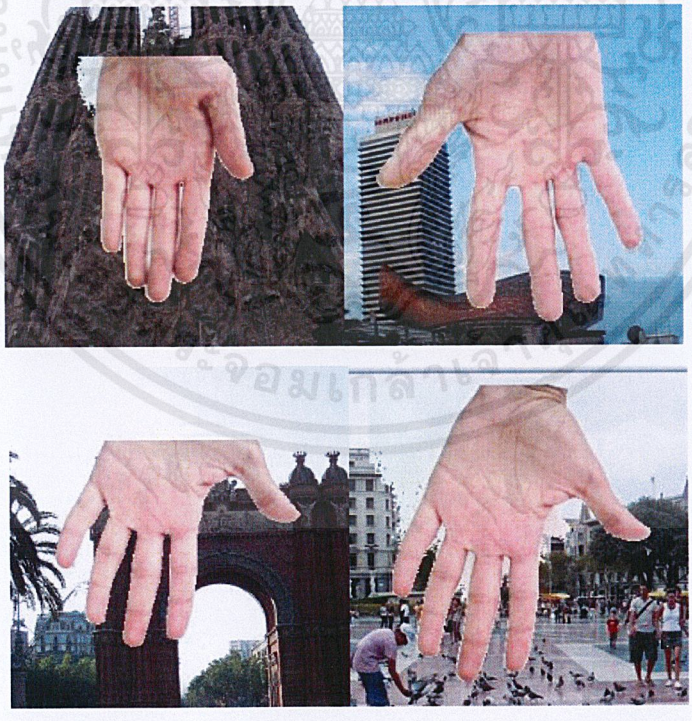
ภาพที่ 3.25 ตัวอย่าง dataset ของ Mutah University [29]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) ข้อมูลจากกล้องมือถือ จากข้อ 3.2.5 นำข้อมูลรูปภาพมาทั้งหมด 18 คน เป็นจำนวนรูปทั้งหมด 36 รูป พื้นหลังของรูปทั้งหมดมีความแตกต่างกัน รูปภาพแต่ละรูปมีรายละเอียดสูง

3.) ข้อมูลจากการนำรูปภาพ 11K Hands ทำการตัดพื้นหลังสีขาวออก โดยใช้วิธีการพื้นฐานคือนำ pixel ที่เป็น 235-255 ออกแล้วทำให้เป็นภาพนามสกุล .png ต่อมานำรูปภาพเหล่านั้นมาทำการสุ่มวางบนรูปภาพพื้นหลังจาก European Cities 1M Dataset [30] เป็นรูปภาพของเมืองในยุโรป แล้วใช้ Hand Detection ตัดภาพมาเฉพาะส่วนที่มีมืออยู่เท่านั้น

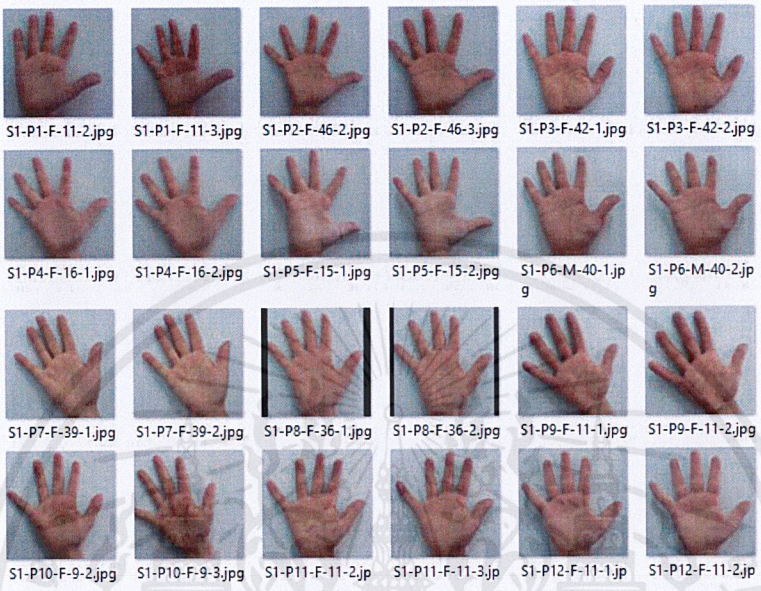
ภาพที่ 3.26 ตัวอย่าง dataset ของ European Cities 1M Dataset [30]



ภาพที่ 3.27 ตัวอย่าง dataset ที่ทำการ random background

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 ผลลัพธ์จากการทดสอบด้วย Dataset ต่าง ๆ โดย Dataset ของ Mutah University และจากกล้องมือถือ มีอย่างละสอง Dataset ซึ่งอันแรกคือรูปภาพปกติที่มาจากแหล่งข้อมูล อันที่สองคือรูปภาพที่ทำการหมุนให้ลักษณะมืออยู่ในทิศทางเดียวกันของทุกรูปใน dataset นั้น ๆ จากปกติรูปภาพมือบางรูปจะถูกถ่ายคนละมุมกัน



ภาพที่ 3.28 ตัวอย่าง dataset ที่ทำหมุนให้มือมีลักษณะทิศทางเดียวกัน

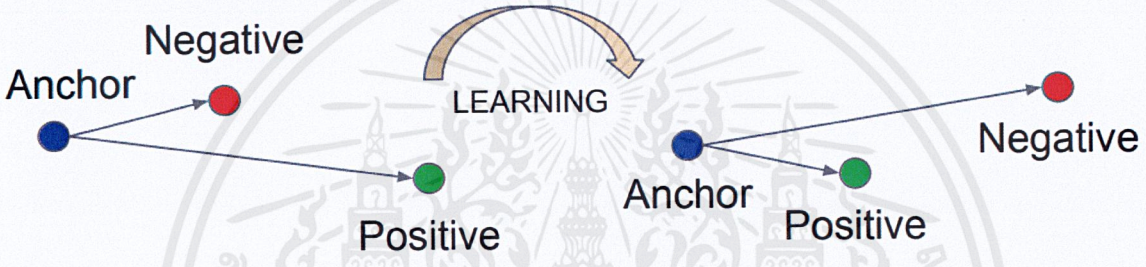
Dataset	Test Accuracy (%)
Mutah University	86
Mutah University with Rotation	82
Dataset from camera	50
Dataset from camera with Rotation	38
Dataset 11K Hands with Random Background	8.125

ตารางที่ 3.6 ผลลัพธ์การทดสอบด้วย datasets อื่น ๆ

3.5 ขั้นตอนการ Train with Triplet Loss Function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากได้ทำการทดสอบด้วย embedding features จึงได้ลองนำ Triplet Loss มาทำการทดสอบด้วย เนื่องจากการยืนยันตัวตนด้วยใบหน้านั้น มีบางงานวิจัยได้ใช้ FaceNet กับ Triplet Loss ในการทำ Face Recognition จึงลองนำมาใช้กับการยืนยันตัวตนด้วยมือ โดย Triplet Loss นั้น เป็นการทำให้ embedding ของสิ่งที่เป็นคำตอบเดียวกันอยู่ใกล้กันมากกว่าคำตอบอื่น ประกอบไปด้วยสามส่วนคือ 1). Anchor Example เป็นตัวอย่างตั้งต้นที่ใช้เปรียบเทียบกับอีกสองส่วน 2). Positive Example ตัวอย่างของรูปภาพที่เป็นคำตอบเดียวกับ Anchor 3.) Negative Example ตัวอย่างของรูปภาพที่เป็นคำตอบต่างกับ Anchor จุดประสงค์ของ Triplet Loss นั้นเพื่อ Optimize ระยะห่างของทั้งสามส่วนโดยมีเป้าหมายหลัก ๆ คือ 1). ลดระยะห่างของ Anchor และ Positive ให้มีระยะห่างที่ใกล้กันมากขึ้น 2). เพิ่มระยะห่างของ Anchor และ Positive ให้มีระยะห่างไกลจากกันมากขึ้น



ภาพที่ 3.29 หลักการทำงานของ Triplet Loss

ซึ่งมีสมการของ triplet ดังนี้ โดย a คือ Anchor, p คือ Positive, n คือ Negative

$$\mathcal{L} = max(d(a,p) - d(a,n) + margin, 0)$$

ภาพที่ 3.30 สมการของ Triplet Loss

```
def triplet_loss(y_true, y_pred):
    y_true = Lambda(lambda  x:tf.squeeze(x,1))(y_true)
    loss = Lambda(lambda  x: metric_learning.triplet_semihard_loss(x[0],x[1],margin=0.3))([y_true,y_pred])
    return loss
```

ภาพที่ 3.31 ฟังก์ชันการเรียกใช้งาน Triplet Loss

ขั้นตอนการสร้างโมเดล ก่อนอื่นสร้างฟังก์ชัน Triplet โดยจากภาพที่ 3.31 เป็นการเรียกใช้งาน Triplet Loss โดย import มาจาก metric_learning ของ tensorflow และการนำข้อมูลเข้าโมเดลที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งาน Triplet ดังที่กล่าวไป ในชุดข้อมูลหนึ่งในการเรียนรู้ต้องมีรูปมือของคนหนึ่งคนอย่างน้อยสองรูปอยู่ในชุดข้อมูลเดียวกัน ถ้าหากมีรูปภาพมือคน ๆ หนึ่งที่มีเพียงแค่หนึ่งรูปในชุดข้อมูล จะทำให้ Triplet ไม่สามารถหาสามส่วนครบได้ ค่า Loss จะลดลงเป็น NaN ดังนั้นจึงต้องสร้าง generator ขึ้นมาใหม่ โดยในชุดข้อมูลหนึ่งต้องมีรูปภาพมือคนหนึ่งอย่างน้อยสองรูป

```
def gen_new_df(csv_file, label):
    data_ = pd.read_csv(csv_file)  ####อ่านไฟล
    label_list = data_[label].to_list()  ####ทำแถว label ให้เป็น list
    label_list = np.array(label_list)
    label_encoder = LabelEncoder()    #### แปลง label เป็นตัวเลข
    integer_encoded = label_encoder.fit_transform(label_list)  #### แปลง label เป็นตัวเลข
    new_df = data_.copy()
    new_df[label] = list(integer_encoded)
    new_df['id'] = pd.Series(integer_encoded, index=new_df.index)
    return new_df
```

ภาพที่ 3.32 ฟังก์ชันสำหรับการเปลี่ยนคำตอบให้เป็นตัวเลข

โดยภาพที่ 3.32 เป็นฟังก์ชันเพื่ออ่านคำตอบจาก ID ให้เป็น ตัวเลขเพื่อนำไปใช้ใน generator โดยจะรับข้อมูลไฟล์ csv และชื่อคอลัมน์ที่ต้องการอ่านคำตอบ แล้วใช้ pandas เป็น library ที่ใช้อ่านไฟล์ csv ตัวหนึ่งใน python จะได้ข้อมูลออกมาในรูปแบบตาราง (Dataframe) ต่อมาต้องแปลงเป็น array เพื่อนำเข้า LabelEncoder ซึ่งจะแปลงข้อมูลใน array ให้อยู่ในรูปแบบตัวเลข เมื่อได้ข้อมูลออกมานำไปทำเป็น dataframe ใหม่อีกครั้งหนึ่ง ที่มีข้อมูลแบบไฟล์ csv ที่อ่านมาในตัวแรกแต่เปลี่ยนข้อมูลในแถวคอลัมน์เป็นข้อมูลที่แปลงให้อยู่ในรูปแบบตัวเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```python
def generator(image_dir, csv_data, image_list_txt, batch_size, label='id', train=True):
    data_ = gen_new_df(csv_data, label)
    i = 0
    with open(image_list_txt) as f:
        image_name_list = f.readlines()

    while True:
        batch_x = list()  # use a dict for multiple inputs
        batch_y = list()
        for b in range(batch_size):
            if i == len(image_name_list):
                i = 0
            image_file_path = image_name_list[i]
            data_cor_image = data_[data_['imageName'] == image_file_path[:-1]]
            label_ = np.array(data_cor_image[label].tolist()[0])
            image = Image.open(os.path.join(image_dir, image_file_path[:-1]))
            image = np.array(image)

            if train:
                seq =iaa.Sequential([ iaa.Add(value=(-25, 45)),iaa.Affine(rotate=(-50, 50),mode="constant")])

                image = seq.augment_image(image)
                image.astype(float)
                image = image / 255.0
                batch_x.append(image)
                batch_y.append(label_)
            else:
                image.astype(float)
                image = image / 255.0
                batch_x.append(image)
                batch_y.append(label_)

            i += 1

        batch_x = np.array(batch_x)
        batch_y = np.array(batch_y)
        yield batch_x, batch_y
```

ภาพที่ 3.33 Generator สำหรับ Triplet Loss

โดยภาพที่ 3.33 เป็น generator ที่ต้องรับข้อมูล โฟลเดอร์ของรูปภาพ ไฟล์ csv ไฟล์ txt และขนาดของชุดข้อมูล (batch size) การทำงานแรกของ generator นี้จะไปนำ dataframe จากฟังก์ชันของภาพที่ 3.32 ที่เป็นคำตอบ label แล้วอ่านไฟล์ txt ซึ่งภายในไฟล์จะมีชื่อไฟล์ของรูปภาพที่ได้ทำการแบ่ง training set, validation set และ test set เหมือนกับขั้นตอนการ train CNN แต่มีส่วนแตกต่างกันคือจะแบ่งอัตราส่วน training set และ validation set เป็น 80, 20 ตามลำดับ ตัดรูปภาพสุดท้ายออกถ้าหากจำนวนรูปภาพของคนเป็นเลขคี่และชื่อรูปภาพของคนหนึ่งคนจะอยู่ติดกันสองชื่อแล้วต่อไปเป็นของคนต่อไป ซึ่งรูปภาพของคนทุกคนมีจำนวนรูปภาพไม่เท่ากัน จึงทำให้การรันชื่อของรูปภาพนั้นในช่วงหลังทำให้จำนวนคนไม่มีความคงที่ ต่อมาหลังจากอ่านไฟล์ txt จะเข้าลูบ While ที่จบด้วย yield ซึ่งเป็นการทำ generator ให้สามารถเรียกใช้ต่อจากเดิมได้ ซึ่งข้างในนั้น เป็นลูปอ่านชื่อไฟล์รูปภาพในไฟล์ txt ตามจำนวน batch size ที่กำหนดไว้ แล้วนำชื่อไฟล์รูปภาพมาเทียบกับ dataframe จากฟังก์ชันเพื่อหา label ของรูปภาพนี้และอ่านรูปภาพในโฟลเดอร์ที่กำหนดไว้ โดยถ้าเป็นขั้นตอนของการ train รูปภาพจะถูก augment โดยใช้ library ที่ชื่อว่า ImgAug จะทำแปลงรูปภาพคือ สุ่มเพิ่มแสง และ สุ่มหมุนรูป หลังจากนั้นนำรูปภาพใส่ลงไปใน list ที่สร้างไว้และคำตอบ label ใส่ลงใน list อีกอัน แต่ถ้าเป็นขั้นตอนการ

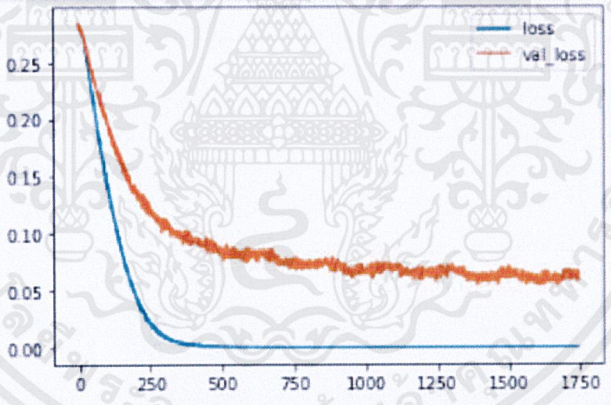
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

validation รูปภาพจะไม่ถูก augment หลังจากเสร็จสิ้นจำนวนลูป จะแปลง list ทั้งสองเป็น array แล้ว yield กลับไป

หลังจากนั้นสร้างโมเดลโดยใช้โครงสร้างเพียงนำ Pre-trained CNN ของ ResNet50 ต่อด้วย fully-connected layer และส่วนชั้นสุดท้ายเป็นการใส่ L2 Normalized ซึ่งการใส่ L2 Normalized นั้นคือการทำให้สามารถจัดกลุ่มของ embedding features จากชั้นก่อนหน้าได้ โดยจะจัดกลุ่มของ embedding features ที่อยู่ใกล้กันเป็นกลุ่มเดียวกันและจัดกลุ่มให้โดยการวัดระยะทางว่าอยู่ใกล้กับกลุ่มไหนมากที่สุด จากนั้นทำการ compile model โดยกำหนดให้ loss function คือ triplet ฟังก์ชันที่สร้างขึ้นมา opitimizer เป็น adam และ learning rate คือ 10^{-6}

ทำการเทรนโดนใช้ fit_generator ใส่ train data, validation data จาก generator ที่สร้างขึ้นเอง ต่อมากำหนด step_per_epoch, validation_step เป็นจำนวนครั้งเรียก generator ของ training set, validation set ตามลำดับ คือ 93, 29 และกำหนดจำนวนรอบที่ใช้ในการเรียนรู้เป็นจำนวน 1744 epoch

3.6 ผลลัพธ์ของการ Train with Triplet Loss Function



ภาพที่ 3.34 กราฟผลลัพธ์ของการ train ด้วย Triplet Loss

ได้โมเดลหลังจากการ train เรียบร้อยแล้วนำมาทดสอบด้วยวิธีการเดิมเช่นเดียวกับ การทดสอบ CNN โดยใช้การวัดระยะทางจาก embedding features โดยนำมาทดสอบกับ dataset เดิมที่ใช้ทดสอบไป ยกเว้น dataset รูปภาพที่มีการสุ่มพื้นหลัง แต่เพิ่มรายละเอียดการทดสอบโดยเพิ่ม augmentation เช่นเดียวกับที่ทำในขั้นตอน train

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

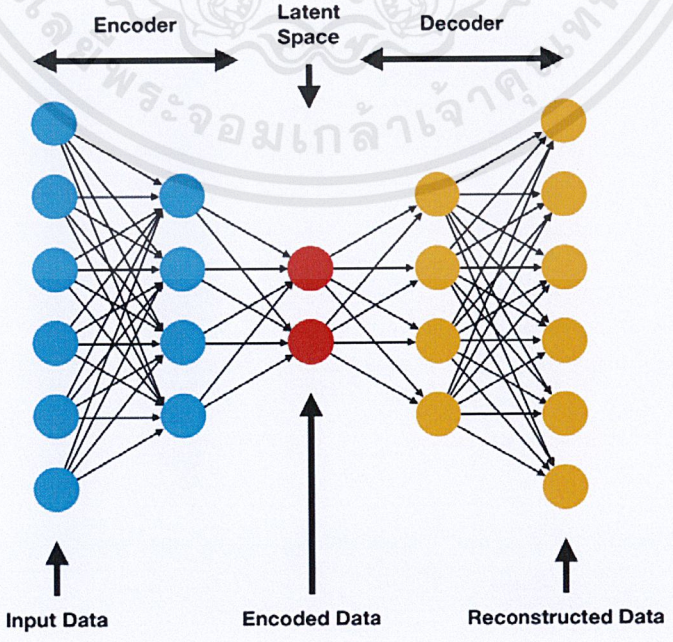
Dataset	Test Accuracy (%)
Mutah University without augmentation	78
Mutah University augment with rotation	78
Mutah University augment with rotation and brightness	40
Data from camera without augmentation	33.33
Data from camera augment with rotation	38.88
Data from camera augment with rotation and brightness	27.77

ตารางที่ 3.7 ผลลัพธ์ของการทดสอบโมเดล Triplet Loss

3.7 ศึกษาวิธีการต่าง ๆ ที่จะเพิ่มประสิทธิภาพให้โมเดล

มีวิธีการต่าง ๆ มากมายที่สามารถทดลองนำมาใช้กับโมเดลได้ เพื่อทำให้โมเดลมีความแม่นยำในการหาลักษณะของมือดีขึ้น แต่ภายในเวลาจำกัดจึงยังไม่สามารถนำมาดัดแปลงใช้กับโมเดลได้

3.7.1 Autoencoder



ภาพที่ 3.35 ตัวอย่างของ Autoencoder [33]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

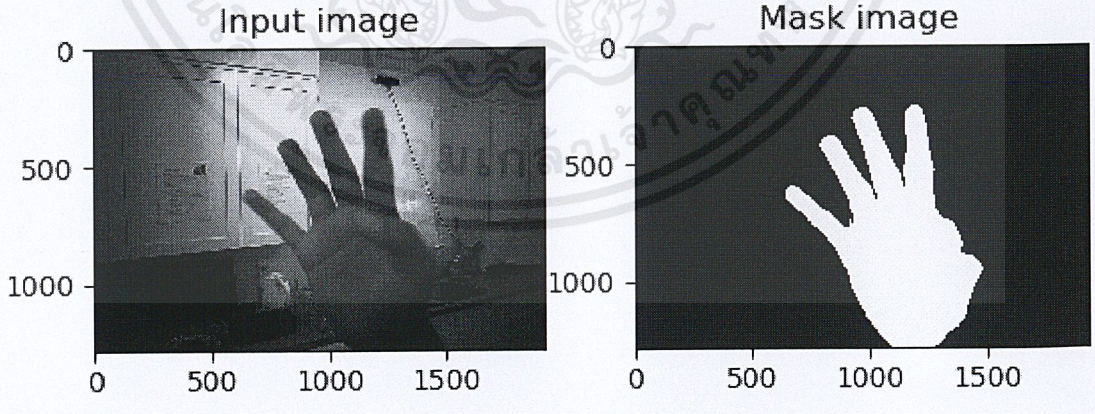
เนื่องจากได้ใช้ pre-trained จาก ResNet50 ซึ่งมีความเป็นไปได้ว่า ResNet50 ที่ผ่านการเรียนรู้มาก่อนหน้านั้นอาจไม่ได้เรียนรู้คลาสที่ต้องการเรียนรู้ได้มากพอ คือ คลาสมือ จึงได้ทดลองโมเดลที่สามารถเรียนรู้จากข้อมูลจำกัด เพื่อนำมาเป็น pre-trained ได้ คือ Autoencoder เป็น unsupervised learning ที่ไม่ต้องเตรียมคำตอบให้กับโมเดล เนื่องจาก Autoencoder เป็น network ที่รับข้อมูล Input เข้ามาแล้วส่ง Output ออกมาเป็นข้อมูลเดิม สองส่วนคือ Encoder และ Decoder

ส่วน Encoder นั้นจะรับข้อมูลมาทำการเข้ารหัสข้อมูล (encoder) ให้มีขนาดที่กำหนดไว้ในครึ่งแรกของโมเดล และส่วน Decoder ครึ่งหลังของโมเดลจะทำหน้าที่สร้างข้อมูลให้เหมือนเดิม โดยทำการถอดรหัสข้อมูล (decode) โดยอิงข้อมูลจากข้อมูลที่ถูกเข้ารหัส ให้เหมือนกับข้อมูลขาเข้า

ทดลองโดยการสร้างโมเดลขึ้นมา ทำเหมือนขั้นตอน train โมเดล แต่ในส่วน generator นั้นค่าที่ส่งไปให้โมเดลเรียนรู้เป็นข้อมูลกับคำตอบคือ รูปภาพเดียวกัน ทั้งในส่วน train และ validation

3.7.2 Segmentation

ข้อมูลรูปภาพมือมีพื้นหลังที่แตกต่างกัน เป็นส่วนที่โมเดลไม่จำเป็นต้องเรียนรู้ก็ได้ เนื่องจากไม่ใช่คลาสหนึ่งในโมเดล จึงได้หา Hand Segmentation Model ที่ผ่านการ train มาแล้วจาก github [34] โดยในตัวอย่างในภาพที่ 3.36 เป็นตัวอย่างของโมเดลได้ว่าสามารถแบ่งมือได้ชัดเจนระหว่างมือกับพื้นหลัง แต่เมื่อนำมาทดลองใช้กับรูปตัวอย่างที่ไว้ทดลองของโปรแกรม ภาพที่ 3.37 เห็นได้ว่ายังมีพื้นหลังบางส่วนที่ยังไม่สามารถแยกออกได้ โมเดลจำเป็นต้องพัฒนาหรือแก้ไขเพิ่ม

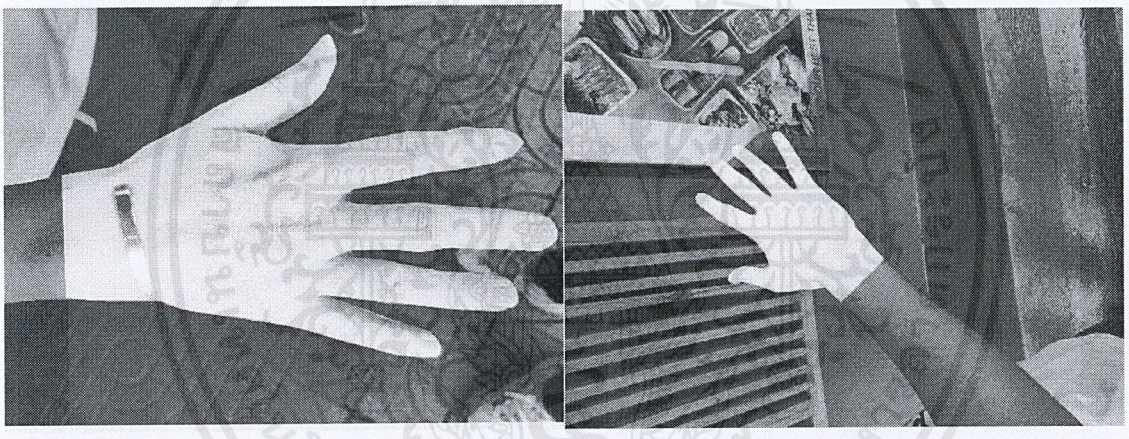


ภาพที่ 3.36 ตัวอย่างของ Segmentation [34]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3.37 ตัวอย่างของ Segmentation ด้วยรูปภาพในโปรแกรม [34]

จึงได้ทำการทดลอง label รูปภาพด้วยตนเอง จากการลากจุดให้ครอบคลุมมือที่ต้องการ แต่สามารถทำไปได้เพียงจำนวนน้อย จึงยังไม่สามารถนำมา train ในการทำ segmentation ได้ เป็นงานที่ต้องใช้เวลาในการทำค่อนข้างมาก

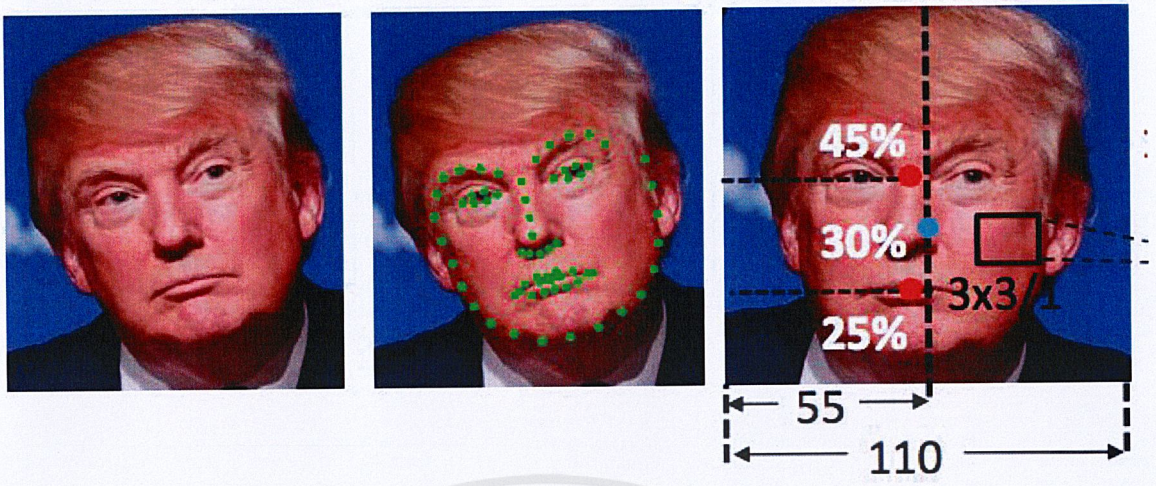


ภาพที่ 3.38 ตัวอย่างการทำ Label ของ Segmentation ด้วยตนเอง

3.7.3 Hand Keypoint Detection

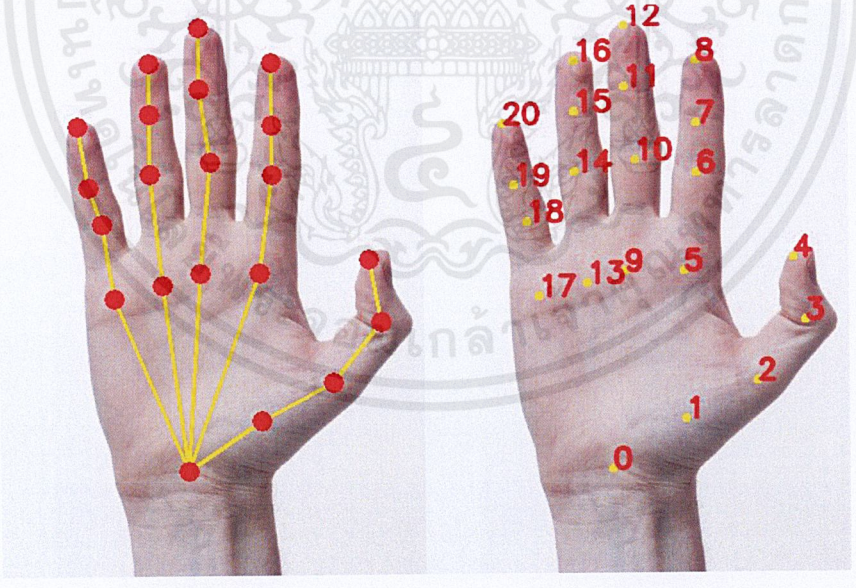
เนื่องจากข้อมูลรูปภาพมืออาจมีการถ่ายจากมุมต่าง ๆ ทำให้ทิศทางมือนั้นแตกต่างกัน ซึ่งอาจทำให้โมเดลเรียนรู้ได้ช้าลง โดยใช้หลักการเดียวกับการทำ face recognition ที่ต้องทำ affine transformation หรือทำให้ใบหน้าให้มีองค์ประกอบอยู่ในตำแหน่งที่คล้ายเดิม เช่น ตา จมูก ปาก แล้วนำเข้าโมเดล ทำให้โมเดลนั้นสามารถแยกและเรียนรู้ได้ดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



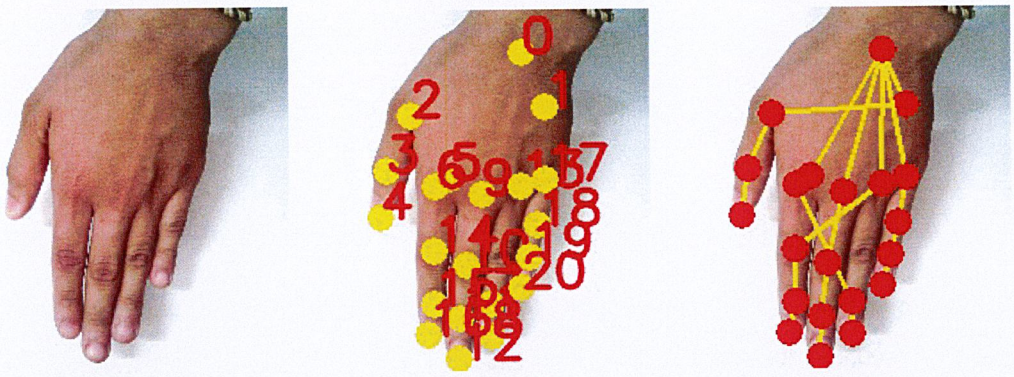
ภาพที่ 3.39 ตัวอย่างการทำงานของ Affine Transformation [37]

จึงได้หา Hand Keypoint Detection ใน github [36] ในภาพที่ 3.40 เป็นตัวอย่างของโมเดลที่ได้เรียนรู้มาแล้ว ซึ่งจะหาจุด keypoint เป็นจำนวนทั้งหมด 21 จุด และอีกจุดหนึ่งคือพื้นหลัง จึงได้นำโปรแกรมมาทดสอบกับรูปภาพของ 11K Hands ดังภาพ 3.41 ที่ แต่เห็นได้ว่าโปรแกรมยังไม่สามารถหา keypoint ได้อย่างแม่นยำ มีการสลับระหว่าง keypoint ไปมาและบางรูปไม่สามารถหา keypoint ได้ครบ



ภาพที่ 3.40 ตัวอย่างของ Hand Keypoint Detection [36]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.41 ตัวอย่างการทดสอบ Hand Keypoint Detection ด้วย dataset ของ 11K Hands

3.8 การแสดงตัวอย่างของ filter ของโมเดล และ features ของรูปภาพในขบวนการ CNN

เป็นการตรวจสอบการทำงานของ CNN ว่าเข้าใจคุณลักษณะของรูปภาพที่ต้องการให้เรียนรู้ได้หรือไม่ อ้างอิงวิธีการแสดงจาก [39] โดยได้เลือกโมเดล softmax activation เป็นโมเดลตั้งต้นในการแสดงตัวอย่าง ขั้นแรกทำ model summary ออกมาเพื่อแสดงชั้นต่าง ๆ ของโมเดล ดังภาพที่ 3.42 โมเดลมีจำนวนชั้นเป็นจำนวนมาก

Layer (type)	Output Shape	Param #	Connected to
input_6 (InputLayer)	(None, 224, 224, 3)	0	
conv1_pad (ZeroPadding2D)	(None, 230, 230, 3)	0	input_6[0][0]
conv1 (Conv2D)	(None, 112, 112, 64)	9472	conv1_pad[0][0]
bn_conv1 (BatchNormalization)	(None, 112, 112, 64)	256	conv1[0][0]
activation_148 (Activation)	(None, 112, 112, 64)	0	bn_conv1[0][0]
pool1_pad (ZeroPadding2D)	(None, 114, 114, 64)	0	activation_148[0][0]
max_pooling2d_4 (MaxPooling2D)	(None, 56, 56, 64)	0	pool1_pad[0][0]
res2a_branch2a (Conv2D)	(None, 56, 56, 64)	4160	max_pooling2d_4[0][0]
. . .			
batch_normalization_14 (BatchNo	(None, 512)	2048	dense_9[0][0]
dense_10 (Dense)	(None, 419)	214947	batch_normalization_14[0][0]

Total params: 78,354,601
Trainable params: 78,237,097
Non-trainable params: 117,504

ภาพที่ 3.42 Architecture โมเดลด้วยคำสั่ง Model Summary

การแสดงให้เห็นภาพของ filter โมเดลตั้งต้นนั้นจะทำการดึงค่าถ่วงน้ำหนักจากโมเดล โดยเลือกชั้นที่ต้องการให้แสดงภาพตัวอย่างของ filter ชั้นนั้น โดยได้เลือกชั้นที่ 2 เป็นชั้น conv2D ชั้นแรกของ

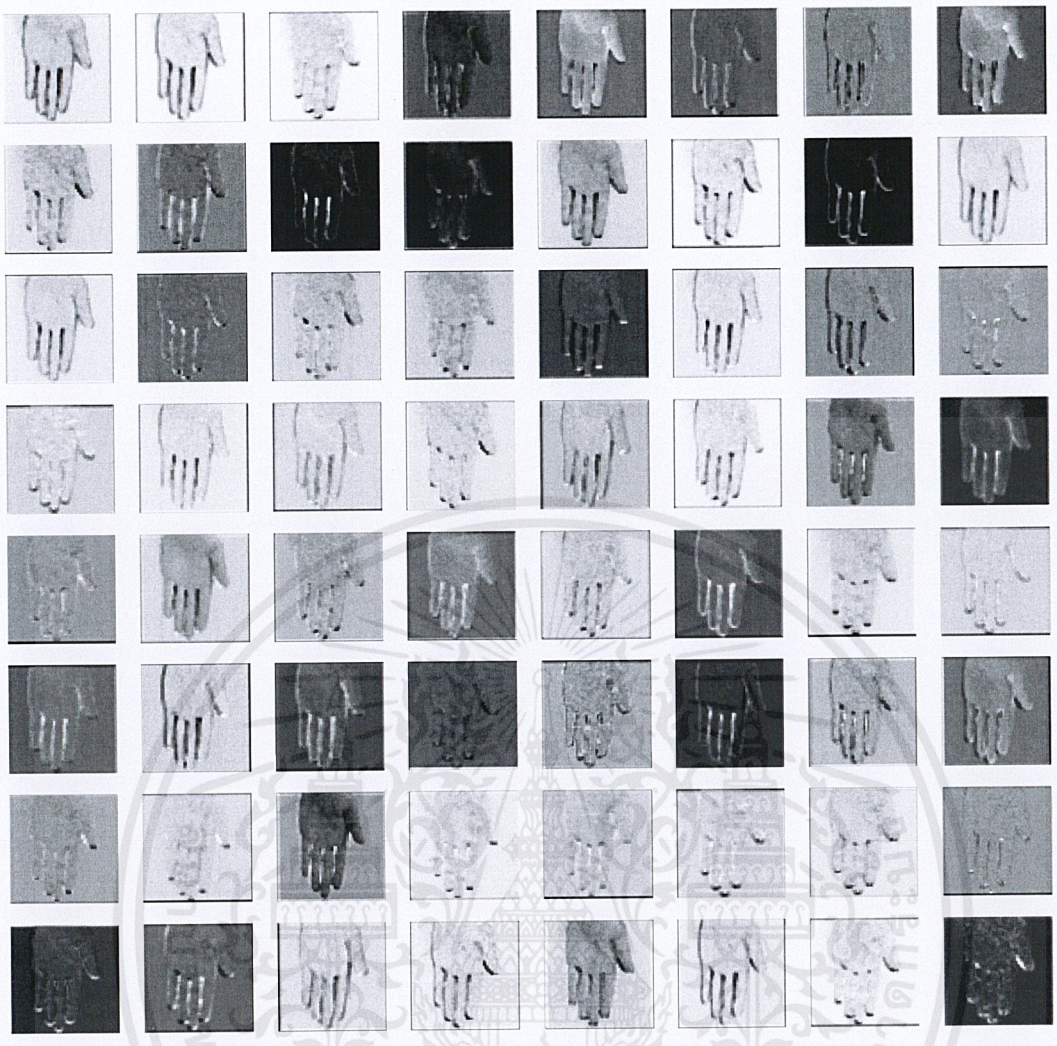
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ResNet50 แล้วนำมาแสดงโดยใช้ matplotlib ที่เป็น library ที่สามารถสร้างกราฟเส้นขึ้นมาและยังสามารถแสดงรูปภาพได้เช่นกัน ชั้นแรกของ ResNet50 นั้นมี filter ทั้งหมด 64 filter นำมาแสดงเพียง 10 filter ซึ่งในแต่ละแถวนั้นเป็น filter ของแต่ละ channel ทั้งหมด 3 ช่อง

ภาพที่ 3.43 ตัวอย่างของ filter ที่ชั้นที่ 2 ของโมเดลตั้งต้น

ในส่วนของการแสดงภาพตัวอย่างของ feature Maps ของโมเดล ต้องทำการสร้างโมเดลขึ้นมาใหม่โดยการตัดโมเดลตั้งต้นจากชั้นแรกจนถึงชั้นที่ต้องการให้แสดงภาพตัวอย่าง แล้วนำรูปภาพมาทำนาย (predict) สุดท้ายที่ได้ออกมาจะเป็น features รูปภาพที่ออกมาจากชั้น Convolution ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.44 ตัวอย่างของภาพที่ผ่าน filter ที่ชั้นที่ 2 ของโมเดลตั้งต้น

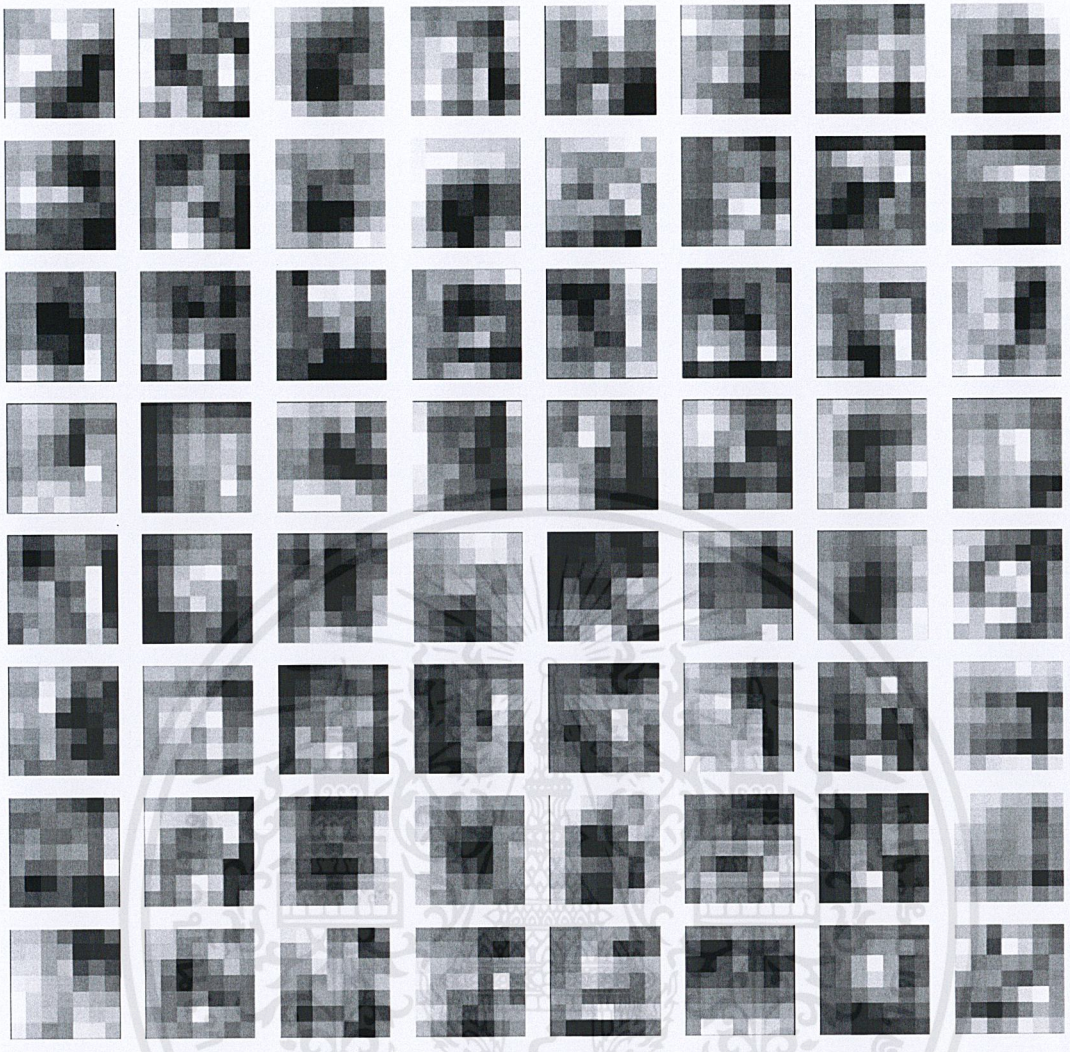
จากภาพที่ 3.44 เป็นโมเดลที่ตัดจนถึงชั้น Conv2D แรกของโมเดล ซึ่งเป็น CNN ของ ResNet50 หรือชั้นที่ 2 ของโมเดลตั้งต้น ซึ่งรูปภาพที่ได้จากชั้นที่ 2 นั้นจะถูกเข้ารหัสให้เหลือขนาด 112x112 และผ่าน filter ทั้งหมด 64 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3.45 ตัวอย่างของภาพที่ผ่าน filter ที่ชั้นที่ 84 ของโมเดลตั้งต้น

จากภาพที่ 3.45 เป็นโมเดลที่ตัดจนถึงชั้น 84 ของโมเดลตั้งต้น เป็นชั้น Conv2D ของ Resnet50 ซึ่งรูปภาพที่ได้จากชั้นที่ 84 นั้นจะถูกเข้ารหัสให้เหลือขนาด 14x14 และผ่าน filter ทั้งหมด 256 ตัว แต่โชว์ตัวอย่างเพียง 64 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.46 ตัวอย่างของภาพที่ผ่าน filter ที่ชั้นที่ 145 ของโมเดลตั้งต้น

จากภาพที่ 3.46 เป็นโมเดลที่ตัดจนถึงชั้น 145 ของโมเดลตั้งต้น เป็นชั้น Conv2D ของ Resnet50 ซึ่งรูปภาพที่ได้จากชั้นที่ 145 นั้นจะถูกเข้ารหัสให้เหลือขนาด 7x7 และผ่าน filter ทั้งหมด 256 ตัว แต่โชว์ตัวอย่างเพียง 64 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการดำเนินงาน

ตลอดระยะเวลาที่ได้ศึกษาการยืนยันตัวตนเพื่อทดสอบการยืนยันตัวตนด้วยมือจนสิ้นสุดโครงการสหกิจศึกษานั้น ทำให้เห็นผลลัพธ์ของการทดลองด้วยโมเดลที่ได้ train ขณะอยู่ในช่วงโครงการสหกิจศึกกษา เพื่อเป็นแหล่งอ้างอิงในการนำไปปรับปรุงในการทำโมเดลครั้งต่อ ๆ ไป

ผลการดำเนินงานทั้งหมดนั้นสามารถแบ่งออกเป็นสามส่วน

4.1 การทำ web application เพื่อใช้เก็บข้อมูลเพิ่มในการเรียนรู้

Web application จะเก็บข้อมูลต่าง ๆ ของผู้ใช้และเก็บรูปภาพที่จะนำมาเพิ่มจำนวนข้อมูลให้กับโมเดลการยืนยันตัวตน ข้อมูลรูปภาพที่ผู้ใช้ส่งมาในการใช้ web application จะผ่านกระบวนการ Hand Detection ทำให้แยกรูปภาพเฉพาะมือออกมาจากรูปภาพที่ผู้ใช้ส่งเข้ามา และทำการยืนยันโดยผู้ใช้อีกทีหนึ่งเพื่อไม่ให้มีข้อมูลที่ผิดพลาดถูกจัดเก็บลงไปในฐานข้อมูล

4.2 Service Hand Detection

เป็น service ที่ใช้อัลกอริทึมของ yolo (you only look one) สามารถตรวจจับรูปภาพมือได้จากการ train โมเดลด้วยรูปภาพมือต่าง ๆ แล้วทดลองด้วย mAP ที่เป็นเกณฑ์ในการวัดความถูกต้องของ object detection ได้ค่าออกมาที่ 87.29 เปอร์เซ็นต์

โดย service Hand Detection นี้สามารถนำไปใช้กับการยืนยันตัวตนได้ โดยนำไปเป็น preprocessing สำหรับรูปภาพในขั้นตอนการต่าง ๆ เช่น รูปภาพที่นำไปใช้ train โมเดล เป็นต้น service นี้ทำให้ข้อมูลมีขนาดเล็กลง จากการตัด pixel ที่ไม่ต้องการออกไป ทำให้โมเดลสามารถเรียนรู้ได้รวดเร็วขึ้นและเรียนรู้สิ่งที่ต้องการได้ชัดเจนมากขึ้น

4.3 การทดลองโมเดลจากขั้นตอนการ Train ด้วย CNN

ผลลัพธ์ที่ได้ออกมาจากการทดลองนั้น มีโมเดลที่แตกต่างกันออกไป ในการเปรียบเทียบนั้นสามารถแบ่งออกเป็น 3 ส่วนได้แก่

4.3.1 การเปรียบเทียบกับผลลัพธ์ของ 11K Hands

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดลองโมเดลตั้งต้นเป็นโมเดลที่ได้ทำการปรับจูนโมเดลเปลี่ยนพารามิเตอร์ต่าง ๆ โดยใช้รูปภาพปกติกับรูปภาพที่ใช้ sharpnen kernel เป็นข้อมูลขาเข้า ซึ่งมี architecture โมเดลคล้ายกับโมเดลของ 11K Hands แต่มีส่วนที่อาจแตกต่างกันไปบ้าง จากตารางที่ 3.3 เห็นได้ว่าโมเดลตั้งต้นที่ได้ออกมานั้นนำมาทดลองข้อมูล test set อันเดียวกับของ 11K Hands มีผลลัพธ์ test accuracy ที่ใกล้เคียงกับโมเดลของ 11K Hands

4.3.2 การเปรียบเทียบโมเดลด้วยการเปลี่ยนพารามิเตอร์ต่าง ๆ

ผลลัพธ์โมเดลที่ได้เปลี่ยนพารามิเตอร์ของรูปภาพขาวดำและจำนวนคลาสในชั้น softmax ซึ่งมี architecture โมเดลคล้ายกัน ตามตารางที่ 3.4 แสดงให้ถึงว่า โมเดลที่ใช้รูปภาพขาวดำและจำนวนคลาสที่นับว่ามือซ้ายและมือขวาเป็นคลาสเดียวกัน นั้นมีค่า test accuracy สูงที่สุด แต่อาจเป็นเพราะว่าในขั้นตอนการปรับจูนโมเดลในโมเดลนี้มากกว่าโมเดลอื่น ทำให้โมเดลอาจเรียนรู้กับข้อมูลชุดนี้มากกว่า

4.3.3 การทดลองโมเดลด้วย dataset ต่าง ๆ

จากตารางที่ 3.6 แสดงให้เห็นถึงการนำไปใช้จริงของโมเดล ว่าสามารถแยกลักษณะมือของคนที่ไม่เคยเรียนรู้ข้อมูลมาก่อนได้ดีหรือไม่ โดยสรุปผลได้ดังนี้

1.) โมเดลสามารถแยกลักษณะของมือได้ จากการทดลองด้วย Mutah University dataset ซึ่งพื้นหลังเป็นสีขาว เหมือนกับข้อมูลของการใช้ train

2.) จากค่าความแม่นยำของ dataset ที่ได้ทำการหมุนรูปภาพให้มีลักษณะมืออยู่ในทิศทางเดียวกัน ลดลงเมื่อเทียบกับค่าความแม่นยำของ dataset ปกติที่ไม่ได้ทำการหมุนรูปภาพ แสดงให้เห็นว่าโมเดลนั้นยังไม่สามารถแยกลักษณะของมือได้ดีพอ เป็นการจำ pixel ของข้อมูลแล้วนำมาเปรียบเทียบกัน

3.) โมเดลเมื่อเจอข้อมูลรูปภาพมือที่มีพื้นหลัง จะมีความแม่นยำลดลง จากการทดลองด้วย Dataset from camera เนื่องจากข้อมูล training set นั้นล้วนเป็นรูปภาพที่ไม่ได้มีพื้นหลังแตกต่างกัน

4.) จาก dataset รูปภาพ 11K Hands ที่ถูกใส่สุ่มพื้นหลังเข้าไป แสดงให้เห็นว่า ถึงแม้เป็นข้อมูลมือที่ได้ทำการเรียนรู้แล้ว โมเดลยังไม่สามารถแยกได้ เมื่อมีภาพพื้นหลังเพิ่มเข้าไป

4.4 การทดลองโมเดลจากขั้นตอนการใช้ Triplet loss

จากตารางที่ 3.7 ผลลัพธ์จากการทดลองด้วย dataset ต่าง ๆ นั้นมีผลลัพธ์ใกล้เคียงกับการทดลองโมเดล softmax activation ค่า test accuracy โมเดลของ triplet loss นั้น มีค่าลดลงเมื่อเทียบกับโมเดล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ใช้ softmax activation function ซึ่งโมเดลของ triplet loss นั้น มีจำนวนชั้นของโมเดลไม่เท่ากับโมเดล softmax activation และมี architecture แตกต่างกัน อาจเป็นสาเหตุที่ทำให้ค่าความแม่นยำน้อยกว่า และการทดลองโมเดล triplet loss ได้เพิ่มการ augment ของรูปภาพเข้าไป ผลที่ได้ออกมามีค่า test accuracy มีค่าลดลงมากขึ้นตามการเพิ่ม augment ของรูปภาพ ซึ่งเป็นเพราะว่าโมเดลนั้นไม่สามารถแยกความแตกต่างของรูปภาพมือทั้งสองของคน ๆ หนึ่ง ที่มีรูปภาพแตกต่างกันได้ดีพอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

ตลอดระยะเวลาการดำเนินงานการศึกษาการยืนยันตัวตนด้วยมือจนถึงการทดลองโมเดลต่าง ๆ จนกระทั่งสิ้นสุดโครงการสหกิจศึกษา สามารถสรุปผลการดำเนินงาน ปัญหาและอุปสรรค และข้อเสนอแนะแนวทางในอนาคตได้ดังต่อไปนี้

5.1 สรุปผลการดำเนินงาน

จากผลลัพธ์การทดลองทั้งหมด โมเดลที่มีความแม่นยำที่สุดในการทดลองนั้นเป็นโมเดลที่ใช้ softmax activation function และรูปภาพขาวดำที่ใช้ sharpen kernel แต่ในการนำไปใช้จริงในการยืนยันตัวตนนั้น ยังไม่สามารถทำได้ เนื่องจากค่าความแม่นยำไม่เพียงพอต่อการนำไปใช้ รูปภาพมือที่จะนำมาใช้กับโมเดลถูกจำกัดให้เป็นรูปภาพที่มีพื้นหลังสีขาว ถ้าเจอรูปภาพที่มีพื้นหลังต่างออกไป ทำให้ความแม่นยำลดลง เพราะว่าข้อมูลที่ใช้ train นั้นมีความหลากหลายไม่เพียงพอ จำนวนข้อมูลที่มีไม่มากพอต่อการเรียนรู้และต้องทำการปรับจูนโมเดลให้มากกว่านี้ เพื่อเพิ่มความแม่นยำของโมเดลให้มากขึ้นกว่า ณ ตอนนี้

ซึ่งการที่รูปภาพมือถูกกำจัดให้มีพื้นหลังสีขาวนั้น อาจสามารถแก้ปัญหาได้โดยการทำ image preprocessing ก่อนนำเข้าโมเดลคือ segmentation หรือการแยกรูปภาพมือออกมาจากพื้นหลังแล้วนำมา train โมเดลใหม่ จะทำให้โมเดลนั้นเรียนรู้ลักษณะของมือได้มากขึ้นจนโมเดลไม่สนใจพื้นหลัง และเมื่อนำไปใช้รูปภาพที่ถูกนำเข้าโมเดลเพื่อทำการทดลองโมเดล ไม่ว่ารูปภาพมีพื้นหลังหรือไม่ ก็สามารถนำเข้าโมเดลได้เพราะมีการ segment รูปภาพมือก่อน

5.2 ปัญหาและอุปสรรค

5.2.1 ข้อมูลรูปภาพที่ใช้ในการเรียนรู้ของ machine learning นั้น มีจำนวนไม่มากพอต่อการเรียนรู้เพื่อนำไปใช้ในการยืนยันตัวตน

5.2.2 การรวบรวมข้อมูลด้วยตนเองนั้น ยังไม่มีประสิทธิภาพมากพอที่จะเพิ่มจำนวนข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 พบข้อผิดพลาดระหว่างการเขียนฟังก์ชันเรียกข้อมูลรูปภาพ ซึ่งทำให้ระยะเวลาในการ train โมเดลลดลง จำเป็นต้องทำการแก้ไขฟังก์ชันในการเรียกข้อมูลรูปภาพใหม่

5.2.4 ขาดความรู้ความเข้าใจในด้าน Machine Learning มีประสบการณ์ไม่มากพอในการทำ Deep Learning

5.3 ข้อเสนอแนะและแนวทางในอนาคต

5.3.1 ควรมีการเก็บรวบรวมข้อมูลในงานวิจัยที่ต้องการค้นคว้าให้เพียงพอต่อความต้องการของงานวิจัยแต่ละอย่าง

5.3.2 การยืนยันตัวตนด้วยมือสามารถนำไปใช้กับสิ่งต่าง ๆ ได้มากมาย ถ้าสามารถทำให้สำเร็จได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Natthawat Phongchit. “Machine learning”. [Online].
แหล่งที่มา https://medium.com/convolab/machine-learning-basics-2b38700cb10b
(21 พฤศจิกายน 2562)

[2] Ajay Kapoor. “Deep learning vs. Machine Learning”. [Online].
แหล่งที่มา https://hackernoon.com/deep-learning-vs-machine-learning-a-simple-explanation-47405b3eef08 (22 พฤศจิกายน 2562)

[3] Vithan Minaphinant. “Deep Learning คืออะไร”. [Online].
แหล่งที่มา https://blog.finnomena.com/ deep-learning-คืออะไร-อาชีพไหนจะตกงานบ้าง-499c250784a1 (23 พฤศจิกายน 2562)

[4] Jure Zupan. “Introduction to Artificial Neural Network (ANN) Methods”. [Online].
แหล่งที่มา
https://www.researchgate.net/publication/251626579_Introduction_to_Artificial_Neural_Network_ANN_Methods_What_They_Are_and_How_to_Use_Them
(24 พฤศจิกายน 2562)

[5] Ian Goodfellow,Yoshua Bengio,Aaron Courville. “Deep Learning”. [Online].
แหล่งที่มา http://faculty.neu.edu.cn/yury/AAI/Textbook/DeepLearningBook.pdf
(25 พฤศจิกายน 2562)

[6] Nessessence. “Deep Learning คืออะไร”. [Online].
แหล่งที่มา https://www.thaiprogrammer.org/2018/12/deep-learning-คืออะไร/
(25 พฤศจิกายน 2562)

[7] Keng Surapong. “Activation Function”. [Online].
แหล่งที่มา https://www.bualabs.com/archives/1261/what-is-activation-function-what-is-sigmoid-function-activation-function-ep-1/amp_(25 พฤศจิกายน 2562)

[8] Pakpoom Thaweesitthichat. “Deep Learning ทำงานอย่างไร”. [Online].
แหล่งที่มา https://medium.com/@pakpoom.thawee/ deep-learning-มันทำงานอย่างไรกันนะ-8eef57561d19 (25 พฤศจิกายน 2562)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[9] Natthawat Phongchit. “Convolutional Neural Network (CNN) คืออะไร.” [Online]. แหล่งที่มา https://blog.datawow.io/มาลองดูวิธีการคิดของ-cnn-กัน-e3f5d73eebaa (26 พฤศจิกายน 2562)

[10] JUKKRIT KRAINET. “เรียนรู้และทำความเข้าใจ Convolutional Neural Network”. [Online]. แหล่งที่มา https://www.glurgeek.com/education/เรียนรู้และทำความเข้าใ-2/ (26 พฤศจิกายน 2562)

[11] Arden Dertat. “Applied Deep Learning - Part 4: Convolutional Neural Networks.” [Online]. แหล่งที่มา https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2 (26 พฤศจิกายน 2562)

[12] Mr.P L. “Optimization & Activation Function”. [Online]. แหล่งที่มา https://medium.com/mmp-li/deep-learning-แบบฉบับสามัญชน-ep-2-optimization-activation-function-เรียนกันสบายๆสไตส์ชิลๆ-9feb5a87e3b2 (27 พฤศจิกายน 2562)

[13] Keng Surapong. “Tanh Function”. [Online]. แหล่งที่มา https://www.bualabs.com/archives/1324/what-is-tanh-activation-function-compare-different-sigmoid-function-artificial-neural-network-activation-function-ep-2/ (27 พฤศจิกายน 2562)

[14] Keng Surapong. “RelU Function”. [Online]. แหล่งที่มา https://www.bualabs.com/archives/1355/what-is-relu-function-why-popular-deep-learning-training-deep-neural-network-activation-function-ep-3/ (28 พฤศจิกายน 2562)

[15] Keng Surapong. “BatchNorm คืออะไร”. [Online]. แหล่งที่มา https://www.bualabs.com/archives/2617/what-is-batchnorm-teach-batch-normalization-train-machine-learning-model-deep-convolutional-neural-network-convnet-ep-5/ (28 พฤศจิกายน 2562)

[16] Sanparith Marukatat. “สนุกกับ Neural Network”. [Online]. แหล่งที่มา https://medium.com/@sanparithmarukatat/สนุกกับ-neural-network-657fa293c4d1 (30 พฤศจิกายน 2562)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[17] Anup Bhande. “Underfitting and overfitting”. [Online].
แหล่งที่มา https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76 (30 พฤศจิกายน 2562)

[18] Engkarat Techapanurak. “Overfitting”. [Online].
แหล่งที่มา https://medium.com/@dopplerz/overfitting-จำคำตอบของข้อสอบอย่างเดียว-ไม่สามารถเข้าใจความรู้ที่แท้จริงได้-ปัญหาที่-ai-5bf27c266462 (31 พฤศจิกายน 2562)

[19] Mathworks. “Guided Image Filter”. [Online].
แหล่งที่มา https://www.mathworks.com/help/images/what-is-guided-image-filtering.html
(1 ธันวาคม 2562)

[20] Lisabug. “Guided Image Filter”. [Online].
แหล่งที่มา https://github.com/lisabug/guided-filter (1 ธันวาคม 2562)

[21] Mahmoud Afifi “11K Hands”. [Online].
แหล่งที่มา https://sites.google.com/view/11Khands (2 ธันวาคม 2562)

[22] Ajay Kumar. “HongKong University Database”. [Online].
แหล่งที่มา http://www4.comp.polyu.edu.hk/~csajaykr/knuckleV2.htm (2 ธันวาคม 2562)

[23] Ajay Kumar. “IITD Dataset”. [Online].
แหล่งที่มา https://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Palm.htm___
(2 ธันวาคม 2562)

[24] João Cartucho. “Mean Average Precision”. [Online].
แหล่งที่มา https://github.com/Cartucho/mAP (2 ธันวาคม 2562)

[25] Krucamper. “Ngrok”. [Online].
แหล่งที่มา https://medium.com/@Krucamper/บวกสกิล-developer-ด้วย-ngrok-674ea172ccaa
(3 ธันวาคม 2562)

[26] Big Data Analytics Thailand. “Keras คืออะไร”. [Online].
แหล่งที่มา https://www.facebook.com/truebigdata/posts/858304600987843/
(3 ธันวาคม 2562)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[27] Software Intel. “Image Data Preprocessing and Augmentation”. [Online].
แหล่งที่มา https://software.intel.com/en-us/articles/hands-on-ai-part-14-image-data-preprocessing-and-augmentation (4 ธันวาคม 2562)

[28] Vasudev. “What is One Hot Encoding”. [Online].
แหล่งที่มา https://hackernoon.com/what-is-one-hot-encoding-why-and-when-do-you-have-to-use-it-e3c6186d008f one hot (4 ธันวาคม 2562)

[29] Mutah University. “Mutah University Dataset”. [Online].
แหล่งที่มา https://xwww.mutah.edu.jo/biometrix/hand-images-databases.html
(4 ธันวาคม 2562)

[30] National Technical University of Athens. “European cities 1M dataset”. [Online].
แหล่งที่มา http://image.ntua.gr/iva/datasets/ec1m/ (4 ธันวาคม 2562)

[31] Poom Wettayakorn. “Facenet และ Triplet Loss”. [Online].
แหล่งที่มา https://medium.com/@poom.wettayakorn/ facenet-และ-triplet-loss-ในการจดจำใบหน้า-a523b71be9e1 (5 ธันวาคม 2562)

[32] Jason Brownlee. “One-shot Learning”. [Online].
แหล่งที่มา https://machinelearningmastery.com/one-shot-learning-with-siamese-networks-contrastive-and-triplet-loss-for-face-recognition/ (5 ธันวาคม 2562)

[33] Steven Flores. “Autoencoder”. [Online].
แหล่งที่มา https://www.compthree.com/blog/autoencoder/ (6 ธันวาคม 2562)

[34] Wudonghao. “Hand-Segmentation”. [Online].
แหล่งที่มา https://github.com/wudonghao/Hand-Segmentation (7 ธันวาคม 2562)

[35] Erezposner. “Hand Keypoint Detector”. [Online].
แหล่งที่มา https://github.com/erezposner/HandKeyPointDetector (8 ธันวาคม 2562)

[36] Sik-Ho Tsang. “Yolo”. [Online].
แหล่งที่มา https://towardsdatascience.com/yolov1-you-only-look-once-object-detection-e1f3ffec8a89 (9 ธันวาคม 2562)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[37] YellowPillow. “Affine transform”. [Online].
แหล่งที่มา https://stackoverflow.com/questions/43042275/how-would-i-achieve-this-in-opencv-with-an-affine-transform (10 ธันวาคม 2562)

[38] Kasidis Satangmongkol. “การวัดผล Machine Learning”. [Online].
แหล่งที่มา https://datarockie.com/2019/03/30/top-ten-machine-learning-metrics/
(10 ธันวาคม 2562)

[39] Jason Brownlee. “Visualize Filters and Feature Maps in CNN”. [Online].
แหล่งที่มา https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/ (11 ธันวาคม 2562)

[40] Missinglink. “Using the keras Flatten Operation”. [Online].
แหล่งที่มา https://missinglink.ai/guides/keras/using-keras-flatten-operation-cnn-models-code-examples/ (12 ธันวาคม 2562)

[41] Adrian Rosebrock. “Intersection over Union”. [Online].
แหล่งที่มา https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/ (12 ธันวาคม 2562)

[42] Victor Powell. “Image Kernel”. [Online].
แหล่งที่มา http://setosa.io/ev/image-kernels/ (12 ธันวาคม 2562)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล นายสุริยะ ประเสริฐปัญญา

ชื่อเล่น โอ

วัน/เดือน/ปีเกิด 5 กุมภาพันธ์ 2541

อีเมล์ suriya.pkb@gmail.com

วุฒิการศึกษา

กำลังศึกษาปริญญาตรี

สาขาวิชา วิศวกรรมสารสนเทศ

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

รหัสนักศึกษา 59011457

ที่อยู่

บ้านเลขที่ 1 ซอยช่างอากาศอุทิศ 10 แยก 1 แขวงดอนเมือง
เขตดอนเมือง จังหวัด กรุงเทพฯ 10210

ผลงาน

Mobile Application สำหรับผู้ความจำสั้น หรือ อัลไซเมอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้