ระบบตรวจจับผู้ขับขี่รถจักรยานยนต์ที่ไม่สวมหมวกกันน็อค

NON-HELMETED MOTORCYCLIST DETECTION SYSTEM

กาญจน์ณุมาศ เอี่ยมธรรมรักษ์

ณัฏฐกิตติ์ หวังเจริญ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

ระบบตรวจจับผู้ขับขี่รถจักรยานยนต์ที่ไม่สวมหมวกกันน็อค

นายกาญจน์ณุมาศ เอี่ยมธรรมรักษ์ 59010093
นายณัฏฐกิตติ์ หวังเจริญ 59010401
รศ.ดร.เกียรติกูล เจียรนัยธนะกิจ อาจารย์ที่ปรึกษา
ปีการศึกษา 2562

บทคัดย่อ

หมวกกันน็อคเป็นอุปกรณ์ที่ผู้ขับขี่รถจักรยานยนต์จำเป็นต้องสวมใส่ขณะขับขี่ เพื่อป้องกันศีรษะเมื่อเกิดอุบัติเหตุ แต่ความเป็นจริงนั้นผู้ขับขี่รถจักรยานยนต์มักสวมใส่เพราะกลัวตำรวจ มากกว่ารักความปลอดภัยของตนเอง แต่เจ้าหน้าที่ตำรวจนั้น ไม่สามารถตรวจตราได้ทุกท้องที่ตลอดเวลา ดังนั้นถ้ามีระบบอัตโนมัติที่สามารถช่วยเจ้าหน้าที่ได้ นอกจากจะลดภาระงานเจ้าหน้าที่แล้ว ยังฝึกวินัยให้ผู้ขับขี่ให้สวมหมวกตลอดเวลา เพราะแม้ไม่มีตำรวจก็โดนตรวจจับได้ ระบบตรวจจับผู้ขับขี่รถจักรยานยนต์โดยไม่ใส่หมวกกันน็อค จะใช้ภาพจากกล้องวงจรปิดที่เชื่อมต่อกับคอมพิวเตอร์ โดยใช้การประมวลผลภาพและการเรียนรู้เชิงลึก ในการตรวจจับภาพผู้ไม่สวมใส่หมวกกันน็อคพร้อมตัดย่อภาพ บันทึกเป็นหลักฐานการกระทำผิดของผู้ขับขี่รถจักรยานยนต์

NON-HELMETED MOTORCYCLIST

DETECTION SYSTEM

Mr. Karnnumart Iamthammarak 59010093

Mr. Nattakitt Wangcharoen 59010401

Assoc.Prof.Dr. Kietikul Jearanaitanakij Advisor

Academic Year 2019

ABSTRACT

Helmet is an equipment that motorcyclists need to wear while driving. Instead of being concerned about their own safety, it is interesting that many of Thai motorcyclists wear helmet because of being afraid of getting fined by the police. Due to a limited number of officers who cannot monitor all motorcyclists at the same time, it would be a good idea to have an automated system that can identify the motorcycle riders who do not wear helmets. This system not only reduces the officer's workload, but also enforces a good habit to those motorcyclists to wear a helmet while they are driving. Non-helmeted Motorcyclist Detection System feeds a color image from a camera which is connected through a personal computer to an image processing unit. Then the resulting image will be taken as an input of the deep learning system which in turn classifies whether the motorcyclist in an image is wearing a helmet. The full capture of the image will be reported as an evidence of a traffic violation.

กิตติกรรมประกาศ

โครงงานนี้จะไม่ประสบความสำเร็จไปได้ หากไม่ได้รับความร่วมมือและความช่วยเหลือจากบุคคลหลายฝ่าย ซึ่งให้การสนับสนุนในการจัดทำโครงงาน ทั้งการให้คำปรึกษาทั้งก่อนและระหว่างการจัดทำโครงงาน การให้ทรัพยากรในการดำเนินในการทำโครงงาน ตลอดจนคอยเสนอแนะแนวทางในการแก้ไขปัญหาระหว่างการจัดทำโครงงาน

ขอขอบคุณ รศ.ดร.เกียรติกูล เจียรนัยธนะกิจ อาจารย์ที่ปรึกษาโครงงานที่ให้คำปรึกษาและคำชี้นำในเรื่องต่างๆ ตั้งแต่การเสนอหัวข้อโครงงานตลอดจนจัดทำโครงงาน

ขอขอบคุณสำนักงานบริหารทรัพยากรกายภาพและสิ่งแวดล้อม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้ความอนุเคราะห์ในการจัดหาไฟล์วิดีโอ ซึ่งเป็นทรัพยากรที่สำคัญอย่างมากในการจัดทำโครงงานนี้

กาญจน์ณุมาศ เอี่ยมธรรมรักษ์

ณัฏฐกิตติ์ หวังเจริญ

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญภาพ VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาของปัญหา 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ประโยชน์ที่คาดว่าจะได้รับ 2

1.4 ขอบเขตของโครงงาน 2

1.5 ข้อจำกัดของโครงงาน 2

1.6 แผนการดำเนินงาน 3

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 4

2.1 ข้อกฎหมายที่เกี่ยวข้อง 4

2.1.1 พระราชบัญญัติ จราจรทางบก (ฉบับที่ 12) พ.ศ. 2562 4

2.1.1.1 มาตรา 142/1 4

2.1.2 พระราชบัญญัติจราจรทางบก พ.ศ.๒๕๒๒ 5

2.1.2.1 มาตรา ๑๒๒ 5

2.1.3 กฎกระทรวง 5

2.1.3.1 ฉบับที่ 14 (พ.ศ. 2535) 5

2.2 เทคโนโลยีด้านการตรวจจับและคัดแยกภาพ 7

2.2.1 การลบภาพนิ่งพื้นหลัง 7

2.2.1.1 OpenCV 7

สารบัญ (ต่อ)

หน้า

2.2.2 การตรวจจับภาพรถจักรยานยนต์ 8
2.2.2.1 TensorFlow 8
2.2.2.2 Keras 8
2.2.3 โมเดลที่ใช้ในการตรวจจับภาพรถจักรยานยนต์ 9
2.2.3.1 Convolutional neural networks (CNN) 9
2.2.3.1.1 การทำงานของโครงข่ายประสาทเทียม CNN 9
2.2.3.2 ResNet152V2 10
2.2.3.3 Inception-ResNet-V2 11
2.2.3.4 Inception-V3 13
2.3 เทคโนโลยีที่ช่วยในการประมวลผลเพื่อสร้างโมเดล 15
2.3.1 Google Colab 15
2.4 เทคโนโลยีที่ช่วยในการสร้างส่วนติดต่อผู้ใช้งาน 15
2.4.1 Electron Js 15
บทที่ 3 การออกแบบและพัฒนา 16
3.1 ความต้องการของระบบ 16
3.1.1 Input 16
3.1.2 Functional 16
3.2 ภาพรวมของระบบ 17
3.3 ขั้นตอนการทำงาน 19
3.3.1 การเตรียมข้อมูล 19
3.3.2 การแยกข้อมูล 19
3.3.3. การสร้างโมเดล 21
3.4 ส่วนติดต่อผู้ใช้งาน 22
บทที่ 4 การทดลองและผลการทดลอง 25
4.1 การสร้างโมเดล 25
4.1.1 ResNet152V2 25

สารบัญ (ต่อ)

หน้า

4.1.2 InceptionResnetV2 26
4.1.3 Inceptionv3 27
4.2 การทดสอบโมเดลโดยใช้ Test set จากข้อมูลที่ใช้สร้างโมเดล 28
4.2.1 ResNet152V2 28
4.2.1.1 รูปที่ผิดพลาดจาก test set ผู้ที่สวมหมวกกันน็อค 28
4.2.1.2 รูปที่ผิดพลาดจาก test set ผู้ที่ไม่สวมหมวกกันน็อค 29
4.2.2 InceptionResnetV2 29
4.2.2.1 รูปที่ผิดพลาดจาก test set ผู้ที่สวมหมวกกันน็อค 29
4.2.2.2 รูปที่ผิดพลาดจาก test set ผู้ที่ไม่สวมหมวกกันน็อค 30
4.2.3 Inceptionv3 30
4.2.3.1 รูปที่ผิดพลาดจาก test set ผู้ที่สวมหมวกกันน็อค 30
4.2.3.2 รูปที่ผิดพลาดจาก test set ผู้ที่ไม่สวมหมวกกันน็อค 31
4.3 สรุปผลที่ได้จากการทดสอบโมเดลโดยใช้ Test set จากข้อมูลที่ใช้สร้างโมเดล 31
4.3.1 ResNet152V2 31
4.3.2 InceptionResnetV2 31
4.3.3 Inceptionv3 32
4.4 การทดสอบโมเดลโดยใช้ไฟล์วิดีโอ 32
4.4.1 มุมบริเวณทางเข้า-ออกตึกนาโน 32
4.4.1.1 ความถูกต้องในการทำนาย 32
4.4.1.2 ความผิดพลาดในการทำนาย 32
4.4.1.3 ความผิดพลาดในการตรวจจับ 33
4.4.2 มุมทางม้าลายบริเวณถนนฉลองกรุง 1 33
4.4.2.1 ความถูกต้องในการทำนาย 33
4.4.2.2 ความผิดพลาดในการทำนาย 33
4.4.2.3 ความผิดพลาดในการตรวจจับ 33

สารบัญ (ต่อ)

หน้า

4.4.3 มุมบริเวณหลังอาคาร 12 ชั้น 33
4.4.3.1 ความถูกต้องในการทำนาย 33
4.4.3.2 ความผิดพลาดในการทำนาย 34
4.4.3.3 ความผิดพลาดในการตรวจจับ 34
4.5 สรุปการทดสอบโมเดลโดยใช้ไฟล์วิดีโอ 35
4.5.1 ความถูกต้องในการทำนาย 35
4.5.2 ความผิดพลาดในการทำนาย 36
4.5.3 ความผิดพลาดในการตรวจจับ 37
4.5.4 ตาราง Confusion Matrix ของโมเดล 38
4.5.4.1 ResNet152V2 38
4.5.4.2 InceptionResnetV2 38
4.5.4.3 Inceptionv3 38
บทที่ 5 บทสรุปและข้อเสนอแนะ 40
5.1 สรุปผล 40
5.2 ปัญหาและแนวทางแก้ไข 40
5.3 แนวทางการพัฒนาต่อ 41
บรรณานุกรม 42

สารบัญตาราง

ตาราง **หน้า**

1.1 แผนการดำเนินงาน 3

4.1 Confusion Matrix ของโมเดล ResNet152V2 ในการจำแนกคนสวมหมวกกับคนไม่สวมหมวก โดยใช้ Test set จากข้อมูลที่ใช้สร้างโมเดล 31

4.2 Confusion Matrix ของโมเดล InceptionResnetV2 ในการจำแนกคนสวมหมวกกับคนไม่สวมหมวก โดยใช้ Test set จากข้อมูลที่ใช้สร้างโมเดล 31

4.3 Confusion Matrix ของโมเดล Inceptionv3 ในการจำแนกคนสวมหมวกกับคนไม่สวมหมวก โดยใช้ Test set จากข้อมูลที่ใช้สร้างโมเดล 32

4.4 สรุปผลความแม่นยำและความผิดพลาดโดยใช้ Test set จากข้อมูลที่ใช้สร้างโมเดล 32

4.5 Confusion Matrix ของโมเดล ResNet152V2 ในการจำแนกคนสวมหมวกกับคนไม่สวมโดยใช้ไฟล์วิดีโอ 38

4.6 Confusion Matrix ของโมเดล InceptionResnetV2 ในการจำแนกคนสวมหมวกกับคนไม่สวมหมวก โดยใช้ไฟล์วิดีโอ 38

4.7 Confusion Matrix ของโมเดล Inceptionv3 ในการจำแนกคนสวมหมวกกับคนไม่สวมหมวก โดยใช้ไฟล์วิดีโอ 38

4.8 สรุปผลความแม่นยำโดยใช้ไฟล์วิดีโอ 39

สารบัญรูป

รูป **หน้า**

2.1 ตัวอย่างของหมวกหมวกนิรภัยทั้ง 3 แบบ 6
2.2 สัญลักษณ์ของ OpenCV 7
2.3 สัญลักษณ์ของ TensorFlow 8
2.4 สัญลักษณ์ของ Keras 8
2.5 สถาปัตยกรรมของ Convolutional neural networks (CNN) 9
2.6 การทำงานของ skip connections ของโมเดล Resnet 10
2.7 โมดูลทั้งหมดของโมเดล Resnet 10
2.8 ภายในโมดูล Convolutions ของโมเดล Resnet 10
2.9 ภายในโมดูล Identity ของโมเดล Resnet 10
2.10 ตัวอย่างการทำงานของโมดูล Inception 11
2.11 โมดูลทั้งหมดของโมเดล Inception-ResNet-V2 11
2.12 ภายในโมดูล Stem ของโมเดล Inception-ResNet-V2 11
2.13 ภายในโมดูล Inception-A ของโมเดล Inception-ResNet-V2 11
2.14 ภายในโมดูล Inception-B ของโมเดล Inception-ResNet-V2 12
2.15 ภายในโมดูล Inception-C ของโมเดล Inception-ResNet-V2 12
2.16 ภายในโมดูล Reduction-A และB ของโมเดล Inception-ResNet-V2 12
2.17 ตัวอย่างการทำงานของโมดูล Inception 13
2.18 โมดูลทั้งหมดของโมเดล Inception-v3 13
2.19 ภายในโมดูล stem ของโมเดล Inception-v3 13
2.20 ภายในโมดูล Inception-A , B และ C ของโมเดล Inception-v3 14
2.21 ภายในโมดูล Reduction-A และ B ของโมเดล Inception-v3 14
2.22 สัญลักษณ์ของ Google Colab 15
2.22 สัญลักษณ์ของ Electron Js 15
3.1 ภาพการทำงานโดยรวมของระบบ 17
3.2 ไฟล์นามสกุล n3r ที่เปิดโดยใช้โปรแกรม NX_Viewer 19
3.3 ตัวอย่างรูปที่ได้จากการตัดภาพในขั้นตอนการแยกข้อมูล 20

สารบัญรูป (ต่อ)

รูป **หน้า**

3.4 ตัวอย่างรูปผู้ที่สวมหมวกกันน็อค 20

3.5 ตัวอย่างรูปผู้ที่ไม่สวมหมวกกันน็อค 20

3.6 ตัวอย่างรูปที่เสีย 21

3.7 หน้าหลักของส่วนติดต่อผู้ใช้งาน 22

3.8 ตัวอย่างการทำงานเมื่อทำการเลือกไฟล์วิดีโอ 22

3.9 ตัวอย่างการทำงานเมื่อทำการเลือกที่อยู่ในการจัดเก็บ 23

3.10 ตัวอย่างการทำงานเมื่อทำการเลือกตำแหน่งที่ต้องการเน้นแล้ว 23

3.11 ตัวอย่างการทำงานเมื่อเริ่มการทำนาย 24

3.12 ตัวอย่างภาพที่ได้จากการทำนาย 24

4.1 กราฟแสดงผลลัพธ์ที่ได้ของโมเดล ResNet152V2 ที่ Epoch = 10 25

4.2 กราฟแสดงผลลัพธ์ที่ได้ของโมเดล ResNet152V2 ที่ Epoch = 100 26

4.3 กราฟแสดงผลลัพธ์ที่ได้ของโมเดล InceptionResnetV2 ที่ Epoch = 10 26

4.4 กราฟแสดงผลลัพธ์ที่ได้ของโมเดล InceptionResnetV2 ที่ Epoch = 100 27

4.5 กราฟแสดงผลลัพธ์ที่ได้ของโมเดล Inceptionv3 ที่ Epoch = 10 27

4.6 กราฟแสดงผลลัพธ์ที่ได้ของโมเดล Inceptionv3 ที่ Epoch = 100 28

4.7 ตัวอย่างรูปที่ผิดพลาดจาก test set ผู้ที่สวมหมวกกันน็อค 28

4.8 ตัวอย่างรูปที่ผิดพลาดจาก test set ผู้ที่ไม่สวมหมวกกันน็อค 29

4.9 ตัวอย่างรูปที่ผิดพลาดจาก test set ผู้ที่สวมหมวกกันน็อค 29

4.10 ตัวอย่างรูปที่ผิดพลาดจาก test set ผู้ที่ไม่สวมหมวกกันน็อค 30

4.11 ตัวอย่างรูปที่ผิดพลาดจาก test set ผู้ที่สวมหมวกกันน็อค 30

4.12 ตัวอย่างรูปที่ผิดพลาดจาก test set ผู้ที่ไม่สวมหมวกกันน็อค 31

4.13 กราฟเปรียบเทียบจำนวนรูปที่ทำนายถูกต้องแยกตามมุม 35

4.14 กราฟเปรียบเทียบจำนวนรูปที่ทำนายผิดแยกตามมุม 36

4.15 กราฟเปรียบเทียบจำนวนรูปที่ตรวจจับวัตถุผิดแยกตามมุม 37

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

ในปัจจุบันผู้ขับขี่รถจักรยานยนต์ในประเทศไทยส่วนมากไม่คำนึงถึงประโยชน์ของหมวกกันน็อคในการป้องกันศีรษะและลดความเสียหายของศีรษะเมื่อเกิดอุบัติเหตุ แต่จะสวมเพราะป้องกันเจ้าหน้าที่ตำรวจจับและต้องจ่ายเงินค่าปรับ มีการตั้งเพจใน Facebook เพื่อประชาสัมพันธ์ถึงการตั้งด่านของตำรวจในพื้นที่นั้นๆ เพื่อจะเตือนผู้ขับขี่รถจักรยานยนต์ที่สัญจรในพื้นที่นั้นสวมหมวกเพื่อหลีกเลี่ยงการถูกปรับ

สาเหตุที่ผู้ขับขี่รถจักรยานยนต์ส่วนใหญ่ไม่สวมหมวกกันน็อคนั้นส่วนมากเกิดจากการที่ผู้ขับขี่คิดว่าไม่จำเป็นต้องสวมเนื่องจากขับไปในพื้นที่บริเวณใกล้เคียง บางส่วนคิดว่าไม่ได้ออกไปในบริเวณที่มีรถเยอะหรือบริเวณถนนเส้นหลักที่มีรถสัญจรเยอะจึงคิดว่าไม่จำเป็นที่ต้องสวมหมวกกันน็อค มีผู้ขับอีกบางส่วนที่ให้เหตุผลว่าผู้ที่นั่งมาด้วยไม่สวมหมวกกันน็อคตนเองจึงไม่สวมด้วยเพราะมีโอกาสที่จะเกิดอุบัติเหตุน้อย

ในปัจจุบันอุบัติเหตุบนท้องถนนนั้นส่วนมากเกิดกับรถจักรยานยนต์มากที่สุด และเกิดการสูญเสียมากที่สุด ซึ่งเกิดจากการที่ผู้ขับขี่ไม่สวมหมวกกันน็อค ดังนั้นผู้จัดทำจึงสนใจที่จะทำระบบตรวจจับผู้ขับขี่รถจักรยานยนต์ที่ไม่สวมหมวกกันน็อคเพื่อทำให้ผู้ขับขี่สวมหมวกกันน็อคมากขึ้นโดยไม่ต้องคอยสวมหมวกขณะที่เจ้าหน้าที่ตำรวจตั้งด่านตรวจจับโดยระบบตรวจจับนี้จะทำการบันทึกภาพผู้ขับขี่รถจักรยานยนต์ที่ไม่สวมหมวกกันน็อคไว้เป็นหลักฐานในการทำผิด

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อทำให้ผู้ขับขี่รถจักรยานยนต์สวมหมวกกันน็อคมากขึ้น
2) ศึกษาการประมวลผลข้อมูลภาพและวิดีโอผ่านกระบวนการ deep learning
3) ศึกษาและพัฒนาแอพพลิเคชั่นเพื่อตรวจจับผู้กระทำความผิดทางท้องถนน

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1) ผู้ขับขี่รถจักรยานยนต์สวมหมวกกันน็อคมากขึ้น
2) เพื่อลดการสูญเสียที่เกิดจากรถจักรยานยนต์
3) สามารถนำระบบไปพัฒนาต่อได้
4) ตรวจจับผู้กระทำความผิดทางท้องถนนได้
5) ลดภาระงานของเจ้าหน้าที่ตำรวจ

1.4 ขอบเขตของโครงงาน

ระบบตรวจจับนี้สามารถทำงานได้ในมุมกล้องมุมเดียวเท่านั้น เนื่องจากแต่ละมุมภาพต้องทำการปรับแก้ค่าในพารามิเตอร์ให้มีค่าต่างกัน

ระบบนี้เรียนรู้จากข้อมูลภาพในเวลากลางวันเท่านั้น อาจใช้ไม่ได้ในกรณีที่ฝนตก หรือภาพเวลากลางคืน

ระบบนี้ไม่สามารถบันทึกภาพทะเบียนรถที่ชัดเจนได้เนื่องจากไม่ได้มีการติดตั้งอุปกรณ์ในพื้นที่และใช้ภาพจากกล้องวงจรปิดของสถาบันในการประมวลผลเท่านั้น

1.5 ข้อจำกัดของโครงงาน

มุมภาพจากกล้องวงจรปิดที่และเงาที่เกิดจากแสงแดดที่ต่างกันอาจส่งผลให้ความแม่นยำของระบบตรวจจับมีความแม่นยำน้อยลง ซึ่งหากต้องการความแม่นยำมากขึ้นต้องทำการปรับค่าพารามิเตอร์ให้มีค่าเฉพาะของมุมภาพแต่ละมุม และในช่วงเวลากลางคืนหรือบริเวณที่มีแสงน้อยไม่สามารถทำการตรวจจับได้เนื่องจากรายละเอียดของภาพที่ได้มีความแตกต่างกันมากเกินไป รวมไปถึงสภาพแสงเงาของเมฆในบริเวณนั้น เนื่องจากระบบตรวจจับจากความแตกต่างของพิกเซล เพื่อหาวัตถุที่เคลื่อนไหว หากพาหนะเคลื่อนผ่านกล้องในเวลาที่แสงเปลี่ยน อาจทำให้ไม่สามารถตรวจจับได้

1.6 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	เดือน									
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1.ศึกษาเทคโนโลยีที่ใช้ในการพัฒนา										
1.1 ศึกษาวิธีการทำงานของเทคโนโลยีที่ใช้										
1.2 ติดตั้งเทคโนโลยีที่ใช้										
2.รวบรวมข้อมูล										
2.1 รวบรวมไฟล์กล้องวงจรปิดและทำการแปลงไฟล์เป็นวิดีโอ										
2.2 ใช้ image processing ในการตัดภาพจากวิดีโอ										
2.3 คัดแยกภาพ										
3.Training model										
3.1 Train model โดยใช้ Transfer learning										
3.2 ตรวจสอบและวัดผลที่ได้										
3.3 ปรับปรุงแก้ไขการตัดภาพให้ดีขึ้น ลด error										
3.4 ทดสอบกับไฟล์กล้องวงจรปิด และวัดผล										
4. Implement										
4.1 Implement										
4.2 ทดสอบใช้และวัดผล										

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ข้อกฎหมายที่เกี่ยวข้อง

2.1.1 พระราชบัญญัติ จราจรทางบก (ฉบับที่ 12) พ.ศ. 2562

2.1.1.1 มาตรา 142/1

เพื่อประโยชน์ในการควบคุมความประพฤติของผู้ได้รับใบอนุญาตขับขี่ให้เกิดความปลอดภัยในการจราจร ในกรณีที่ผู้ขับขี่ผู้ใดกระทำความผิดตามพระราชบัญญัตินี้หรือตามกฎหมายอื่นอันเกี่ยวกับรถหรือการใช้ทาง ให้สำนักงานตำรวจแห่งชาติจัดให้มีระบบการบันทึกคะแนนความประพฤติในการขับรถของผู้ได้รับใบอนุญาตขับขี่

ระบบการบันทึกคะแนนความประพฤติในการขับรถตามวรรคหนึ่ง อย่างน้อยต้องประกอบด้วยการกำหนดคะแนน การตัดคะแนน และการคืนคะแนน โดยวิธีดำเนินการดังกล่าวให้เป็นไปตามหลักเกณฑ์วิธีการ และเงื่อนไขที่ผู้บัญชาการตำรวจแห่งชาติและอธิบดีกรมการขนส่งทางบกร่วมกันกำหนดโดยประกาศในราชกิจจานุเบกษา ทั้งนี้ ในการกำหนดคะแนนความประพฤติในการขับรถให้คำนึงถึงประเภทของใบอนุญาตขับขี่ และเหตุแห่งการกระทำความผิดด้วย

ในกรณีที่ผู้ได้รับใบอนุญาตขับขี่ผู้ใดถูกตัดคะแนนความประพฤติในการขับรถจนหมดคะแนนตามที่กำหนดไว้ ให้ผู้บัญชาการตำรวจแห่งชาติหรือผู้ซึ่งได้รับมอบหมายจากผู้บัญชาการตำรวจแห่งชาติในแต่ละท้องที่สั่งพักใช้ใบอนุญาตขับขี่ของผู้ได้รับใบอนุญาตขับขี่ผู้นั้นคราวละเก้าสิบวัน ทั้งนี้ ตามระเบียบที่ผู้บัญชาการตำรวจแห่งชาติกำหนด

2.1.2 พระราชบัญญัติจราจรทางบก พ.ศ.๒๕๒๒

2.1.2.1 มาตรา ๑๒๒

ผู้ขับขี่รถจักรยานยนต์และคนโดยสารรถจักรยานยนต์ ต้องสวมหมวกที่จัดทำขึ้นโดยเฉพาะ เพื่อป้องกันอันตรายในขณะขับขี่ และโดยสารรถจักรยานยนต์ ห้ามมิให้ผู้ขับขี่รถจักรยานยนต์ตามวรรคหนึ่งขับขี่รถจักรยานยนต์ในขณะที่คนโดยสารรถจักรยานยนต์ มิได้ สวมหมวกที่จัดทำขึ้นโดยเฉพาะเพื่อป้องกันอันตราย

ลักษณะและวิธีการใช้หมวกเพื่อป้องกันอันตรายตามวรรคหนึ่งให้เป็นไปตามที่กำหนดในกฎกระทรวง ความในวรรคหนึ่ง มิให้ใช้บังคับแก่ภิกษุ สามเณร นักพรต นักบวช หรือผู้นับถือลัทธิศาสนาอื่นที่ใช้ผ้าหรือสิ่งอื่นโพกศีรษะตามประเพณีนิยมนั้น หรือบุคคลใดที่กำหนดในกฎกระทรวง

2.1.3 กฎกระทรวง

2.1.3.1 ฉบับที่ 14 (พ.ศ. 2535)

อาศัยอำนาจตามความใน มาตรา 5 แห่ง พระราชบัญญัติ จราจรทางบก พ.ศ. 2522 และ มาตรา 122 วรรคสาม แห่ง พระราชบัญญัติ จราจรทางบก พ.ศ. 2522 ซึ่งแก้ไขเพิ่มเติมโดย พระราชบัญญัติ จราจรทางบก (ฉบับที่ 2) พ.ศ. 2522 รัฐมนตรีว่าการ กระทรวงมหาดไทยออกกฎกระทรวงไว้ ดังต่อไปนี้

ข้อ 1 ในกฎกระทรวงนี้

"หมวกนิรภัย" หมายความว่า หมวกที่จัดทำขึ้นโดยเฉพาะเพื่อป้องกันอันตรายในขณะขับขี่และโดยสารรถจักรยานยนต์

"หมวกนิรภัยแบบปิดเต็มหน้า" หมายความว่า หมวกนิรภัยที่เปลือกหมวก เป็นรูปทรงกลมปิดด้านข้าง ด้านหลัง ขากรรไกร และคาง ในกรณีที่มีบังลม บังลม ต้องทำจากวัสดุโปร่งใสและไม่มีสี

"หมวกนิรภัยแบบเต็มใบ" หมายความว่า หมวกนิรภัยที่เปลือกหมวกเป็น รูปทรงกลม ปิดด้านข้างและดานหลังเสมอแนวขากรรไกรและต้นคอด้านหลัง ด้านหน้าเปิด เหนือคิ้ว

"หมวกนิรภัยแบบครึ่งใบ" หมายความว่า หมวกนิรภัยที่เปลือกหมวกเป็นรูปครึ่งทรงกลมปิดด้านข้างและด้านหลังเสมอระดับหู ในกรณีที่มีบังลม บังลม ต้องทำจาก วัสดุโปร่งใสและไม่มีสี

ข้อ 2 หมวกนิรภัยให้ใช้ได้ 3 แบบ คือ หมวกนิรภัยแบบปิดเต็มหน้า หมวกนิรภัยแบบเต็มใบและหมวกนิรภัยแบบครึ่งใบ

ในกรณีที่ได้มีการกำหนดมาตรฐานผลิตภัณฑ์อุตสาหกรรมสำหรับหมวกนิรภัยแบบใดไว้ตามกฎหมายว่าด้วยมาตรฐานผลิตภัณฑ์อุตสาหกรรมแล้ว หมวกนิรภัยที่จะใช้ต้องเป็นไปตามมาตรฐานผลิตภัณฑ์อุตสาหกรรม

ข้อ 3 ในขณะขับขี่หรือโดยสารรถจักรยานยนต์ ผู้ขับขี่และคนโดยสารต้องสวมหมวกนิรภัยโดยจะต้องรัดคางด้วยสายรัดคางหรือเข็มขัดรัดคางให้แน่นพอที่จะป้องกันมิให้ หมวกนิรภัยหลุดจากศีรษะได้หากเกิดอุบัติเหตุ

ก) ข) ค)

รูปที่ 2.1 ตัวอย่างของหมวกหมวกนิรภัยทั้ง 3 แบบ

ก) หมวกนิรภัยแบบปิดเต็มหน้า

ข) หมวกนิรภัยแบบเต็มใบ

ค) หมวกนิรภัยแบบครึ่งใบ

2.2 เทคโนโลยีด้านการตรวจจับและคัดแยกภาพ

2.2.1 การลบภาพนิ่งพื้นหลัง

2.2.1.1 OpenCV

OpenCV คือ Library ที่อยู่ในรูปแบบของฟังก์ชันในการเขียนโปรแกรม โดยจุดประสงค์ในการแสดงผลด้วยคอมพิวเตอร์ให้อยู่ในรูปแบบเรียลไทม์ (Real Time) OpenCV ได้ถูกจัดทำและพัฒนาขึ้นโดยบริษัท Intel โดย OpenCV สามารถใช้งานได้ฟรีโดยมีเจ้าของลิขสิทธิ์คือ BSD แบบ Open Source นอกจากนี้ OpenCV ยังสามารถรองรับ Framework ในการทำ Deep Learning เช่น TensorFlow, Torch, PyTorch ด้วย นอกจากนี้ OpenCV ถูกพัฒนาขึ้นมาจากภาษา C++ และสามารถรองรับภาษา Python, Java และ MATLAB ได้

OpenCV นิยมนำมาใช้ในการพัฒนาโปรแกรมที่เกี่ยวข้องกับทางด้าน Image Processing และ Computer Vision โดยสามารถทำงานได้ทั้งในระบบปฏิบัติการ Windows และ Linux

รูปที่ 2.2 สัญลักษณ์ของ OpenCV

2.2.2 การตรวจจับภาพรถจักรยานยนต์

2.2.2.1 TensorFlow

TensorFlow คือ Library ที่ใช้พัฒนา Machine Learning จัดทำโดยบริษัท Google โดย TensorFlow นั้นใช้ภาษา python ในการใช้งาน สามารถทำงานได้ทั้งบน CPU และ GPUs ได้หลายตัวและสามารถทำงานในโทรศัพท์มือถือได้ โดยมีระบบปฏิบัติการที่รองรับได้แก่ Linux, macOS, Windows และ Android

รูปที่ 2.3 สัญลักษณ์ของ TensorFlow

2.2.2.2 Keras

Keras เป็น API โครงข่ายประสาทเทียม (neural network) ระดับสูงที่ใช้งานโดยภาษา Python เวอร์ชัน 2.7-3.6 และสามารถทำงานบน TensorFlow, CNTK หรือ Theano โดย Keras ได้พัฒนาโดยมีจุดประสงค์คือการเพิ่มความรวดเร็วในการเทรนหรือการคำนวณเพื่อทำนายคำตอบ โดยการใช้ Keras จะช่วยสามารถสร้าง Model ได้รวดเร็วและไม่จำเป็นต้องสร้างเองทั้งหมดจึงทำให้เกิดความสะดวกสบายในการพัฒนาเป็นอย่างมาก และ Keras ยังรองรับทั้งเครือข่าย Convolutional และรวมทั้งสามารถรวมโครงข่ายประสาทเทียมได้

รูปที่ 2.4 สัญลักษณ์ของ Keras

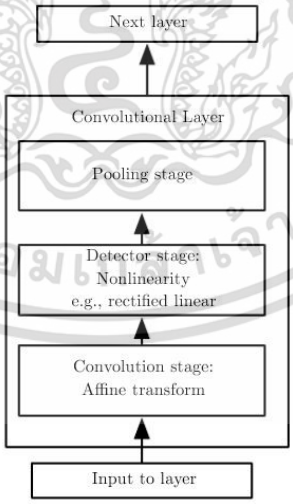
2.2.3 โมเดลที่ใช้ในการตรวจจับภาพรถจักรยานยนต์

2.2.3.1 Convolutional neural networks (CNN)

เป็นโครงข่ายประสาทเทียมที่จำลองการมองเห็นของมนุษย์โดยมองพื้นที่จากชิ้นใหญ่เป็นชิ้นย่อย และนำกลุ่มของชิ้นย่อยมาวิเคราะห์ โดยการมองเป็นชิ้นย่อยจะมีการแยกคุณลักษณะ (feature) ของชิ้นนั้น เช่น ลายเส้น และการตัดกันของสี ซึ่งการที่รู้ว่าพื้นที่ตรงนี้เป็นเส้นตรงหรือสีตัดกัน เพราะเกิดจากการนำเอาชิ้นย่อยที่อยู่ติดกันมาประกอบกัน

2.2.3.1.1 การทำงานของโครงข่ายประสาทเทียม CNN

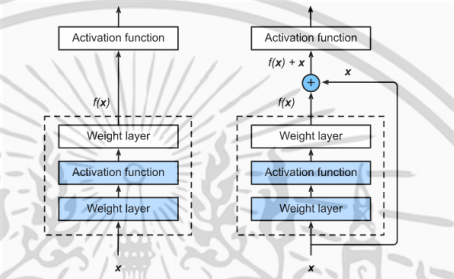
1. Convolution stage จะสร้าง Sliding window (Filter) มาสแกนรูปที่เป็นข้อมูลเพื่อแยกองค์ประกอบของรูป(feature) ออกมา เช่น สี รูปทรง ขอบ เป็นต้น
2. Detector stage รับ output จากข้อแรกมาแปลงให้อยู่ในรูป nonlinear โดยใช้ activation เพื่อสะดวกในการคำนวณผลและเพิ่มประสิทธิภาพของผลลัพธ์
3. Pooling stage ทำการ resize ข้อมูลให้มีขนาดที่เล็กลงโดยที่รายละเอียดของรูปที่เป็นข้อมูลนั้นยังครบถ้วน ซึ่งช่วยในเรื่องเพิ่มความไวในการคำนวณ และแก้ปัญหา overfitting
4. ใน layer ถัดไปอาจจะทำซ้ำเดิมก็ได้ แต่ส่วนที่จะมาเป็น Final layer นั้นต้องเป็น fully connected NN (Feed-forward NN) เท่านั้น



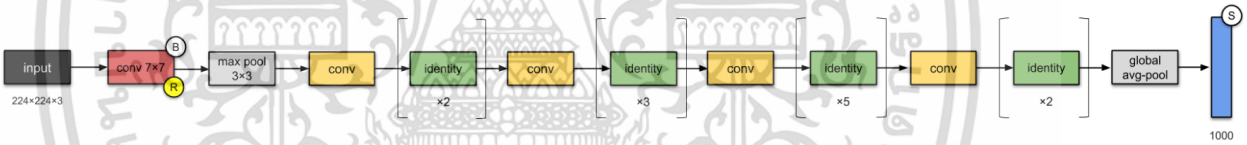
รูปที่ 2.5 สถาปัตยกรรมของ Convolutional neural networks (CNN)

2.2.3.2 ResNet152V2

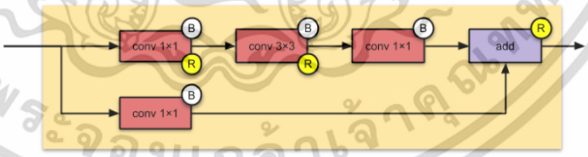
เป็นโมเดลที่พัฒนาจาก CNN ส่วนมากจะมีแต่การเพิ่มจำนวนของ layer ในการออกแบบเพื่อเพิ่มประสิทธิภาพให้ดีขึ้น แต่การเพิ่มความลึกนั้นอาจทำให้เกิด vanishing gradient และใช้ระยะเวลานานมากขึ้น โมเดล ResNet มีวิธีการที่เรียกว่า skip connections เพื่อข้ามขั้นตอนของ layer บาง layer ที่ output มีค่าเหมือนกับ input หรือ layer ที่ไม่ต้องการได้ โดยเมื่อไม่ต้องการ layer แล้วจะทำการปรับค่า weight ที่ F(x) ให้เป็น 0 จึงทำให้ค่าที่ผ่านการคำนวณที่ block นั้นไม่มีการเปลี่ยนแปลง โดยมีพารามิเตอร์จำนวน 60 ล้านพารามิเตอร์



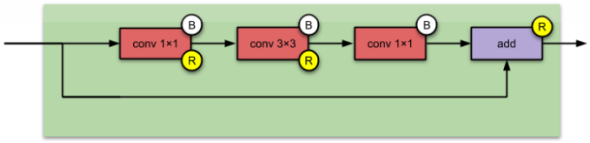
รูปที่ 2.6 การทำงานของ skip connections ของโมเดล Resnet



รูปที่ 2.7 โมดูลทั้งหมดของโมเดล Resnet



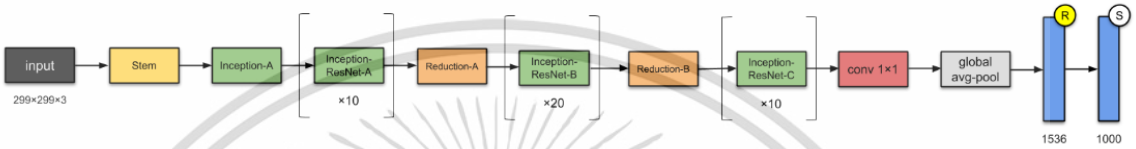
รูปที่ 2.8 ภายในโมดูล Convolutions ของโมเดล Resnet



รูปที่ 2.9 ภายในโมดูล Identity ของโมเดล Resnet

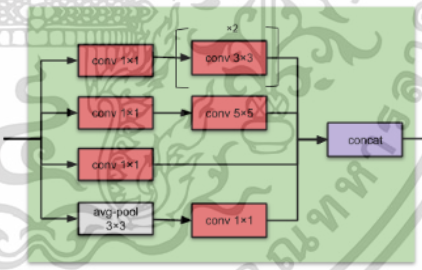
2.2.3.3 Inception-ResNet-V2

โมเดล Inception-ResNet-V2 พัฒนาจาก Inception-v3 และ Inception-v4 ข้อแตกต่างที่สำคัญคือโมดูล Stem และการเปลี่ยนแปลงเล็กน้อยในโมดูล Inception โดยจัดทำขึ้นเนื่องจากโมเดล inception นั้นเมื่อต้องการประสิทธิภาพในการทำงานมากจำนวนชั้นก็มากขึ้นตามไปด้วยจนทำให้เกิดปัญหาคอขวดขึ้น ซึ่งได้นำ skip connections ของโมเดล Resnet มาใช้งานร่วมกันในการแก้ไขไม่ให้เกิดคอขวดขึ้น

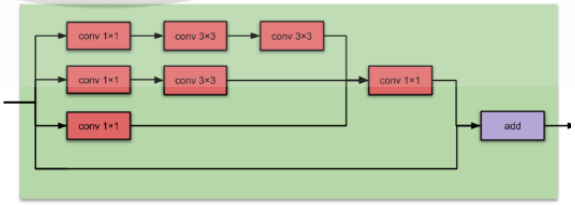


รูปที่ 2.10 โมดูลทั้งหมดของโมเดล Inception-ResNet-V2

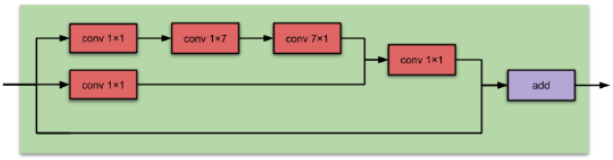
รูปที่ 2.11 ภายในโมดูล Stem ของโมเดล Inception-ResNet-V2



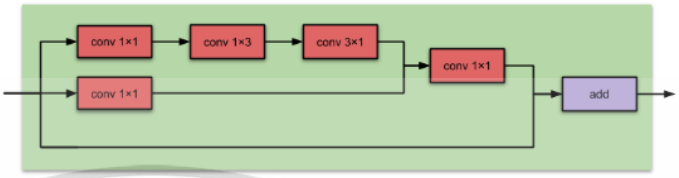
รูปที่ 2.12 ภายในโมดูล Inception-A ของโมเดล Inception-ResNet-V2



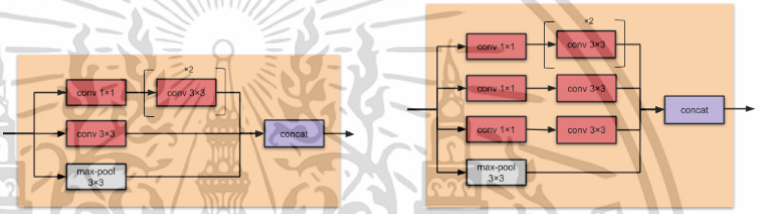
รูปที่ 2.13 ภายในโมดูล Inception-A ของโมเดล Inception-ResNet-V2



รูปที่ 2.14 ภายในโมดูล Inception-B ของโมเดล Inception-ResNet-V2



รูปที่ 2.15 ภายในโมดูล Inception-C ของโมเดล Inception-ResNet-V2



ก) ข)

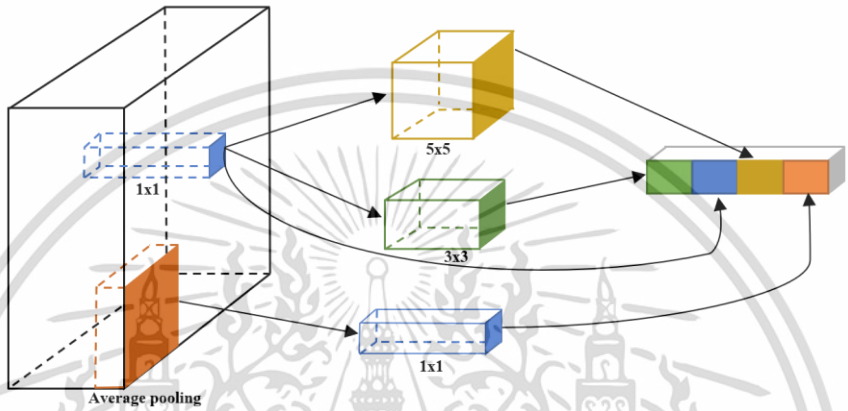
รูปที่ 2.16 ภายในโมดูล Reduction-A และB ของโมเดล Inception-ResNet-V2

ก) ภายในโมดูล Reduction-A

ข) ภายในโมดูล Reduction-B

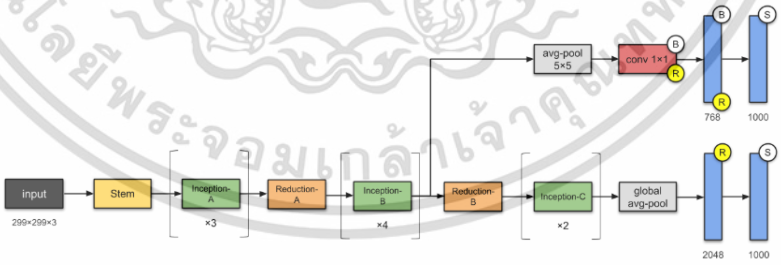
2.2.3.4 Inception-v3

Inception เวอร์ชันแรกนั้นมีจำนวน layer อยู่ 22-layer และมีพารามิเตอร์จำนวน 5 ล้านพารามิเตอร์ การออกแบบสถาปัตยกรรมของโมดูล Inception นั้นเป็นงานวิจัยเกี่ยวกับโครงสร้างที่กระจัดกระจาย โดยได้จากการนำเอาข้อมูลมาผ่านการทำ convolutions หลากหลายรูปแบบโดยทำงานแบบ parallel แล้วนำค่าที่ได้เหล่านั้นมารวมเข้าด้วยกัน

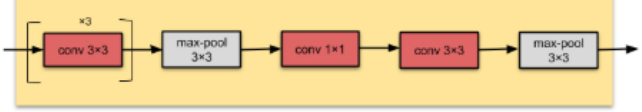


รูปที่ 2.17 ตัวอย่างการทำงานของโมดูล Inception

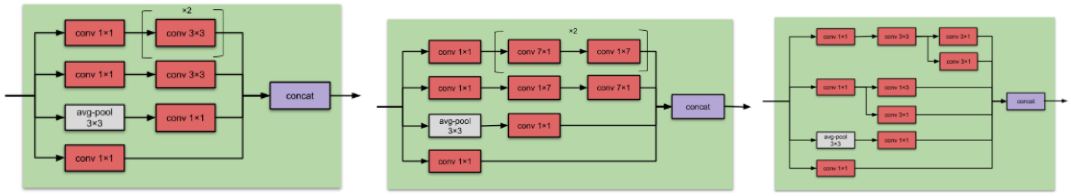
Inception-v3 นั้นถูกพัฒนาขึ้นมาจาก Inception-v1 มีพารามิเตอร์จำนวน 24 พารามิเตอร์ Inception-v3 เป็นเครือข่ายที่การปรับแต่งเพื่อเพิ่มประสิทธิภาพให้ฟังก์ชั่น ลดความผิดพลาดและเพิ่ม batch normalization ใน layer เสริมในเครือข่าย



รูปที่ 2.18 โมดูลทั้งหมดของโมเดล Inception-v3



รูปที่ 2.19 ภายในโมดูล stem ของโมเดล Inception-v3



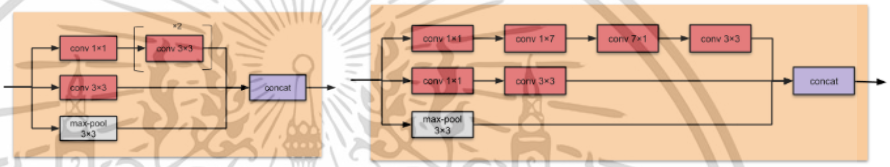
ก) ข) ค)

รูปที่ 2.20 ภายในโมดูล Inception-A , B และ C ของโมเดล Inception-v3.

ก) ภายในโมดูล Inception-A

ข) ภายในโมดูล Inception-B

ค) ภายในโมดูล Inception-C



ก) **ข)**

รูปที่ 2.21 ภายในโมดูล Reduction-A และ B ของโมเดล Inception-v3

2.3 เทคโนโลยีที่ช่วยในการประมวลผลเพื่อสร้างโมเดล

2.3.1 Google Colab

Google Colab ชื่อเต็มคือ Google Colaboratory คือ Jupyter notebook ที่ทำงานอยู่บน cloud โดยที่สามารถใช้งานได้เพียงมีบัญชีที่สามารถเข้าใช้ Google Drive ก็จะสามารถใช้งานได้ โดยสามารถจัดเก็บ Code ไว้บน Google Drive ของตนเองได้ ไม่ต่างกับการเขียนโปรแกรมบนเครื่อง notebook และยังสามารถที่จะแชร์ให้กับเพื่อนได้และสามารถที่จะทำงานในเวลาเดียวกันได้ ซึ่งภาษาหลักที่ใช้เขียนจะเป็นภาษา python โดยรองรับ python version 2 และ Python version 3.x ซึ่งสามารถเลือกได้ว่าจะใช้งานเวอร์ชันใด และมี GPU, TPU ให้ใช้โดยไม่เสียค่าใช้จ่ายภายใต้เงื่อนไขที่กำหนด

นอกจากนี้ Google Colab ยังสามารถเชื่อมต่อกับ GitHub หรือ Export ออกมาเป็นไฟล์ python หรือแชร์ให้เพื่อนใช้งานต่อได้ โดย python library ที่ใช้งานกันบ่อยบน Google Colab นั้นจะมีการบันทึกให้มีอยู่ในตัวโดยอัตโนมัติ จึงสามารถเรียกใช้งานได้ทันที หรือจะติดตั้งเสริมเพิ่มเติมก็ได้

รูปที่ 2.22 สัญลักษณ์ของ Google Colab

2.4 เทคโนโลยีที่ช่วยในการสร้างส่วนติดต่อผู้ใช้งาน

2.4.1 Electron Js

Electron เป็น Open source library ที่ถูกพัฒนาโดย GitHub เอาไว้ใช้สำหรับการพัฒนาเดสก์ท็อป ครอสแพลตฟอร์มแอปพลิเคชัน ด้วยภาษา HTML, CSS, และ JavaScript การทำงานของ Electron เกิดขึ้นจากการผสมผสานระหว่าง Chromium และ Node.js เข้าด้วยกัน และตัวแอปสามารถนำไปใช้ได้กับทั้งระบบปฏิบัติการ Mac, Windows, และ Linux

รูปที่ 2.23 สัญลักษณ์ของ Electron Js

บทที่ 3

การออกแบบและพัฒนา

3.1 ความต้องการของระบบ

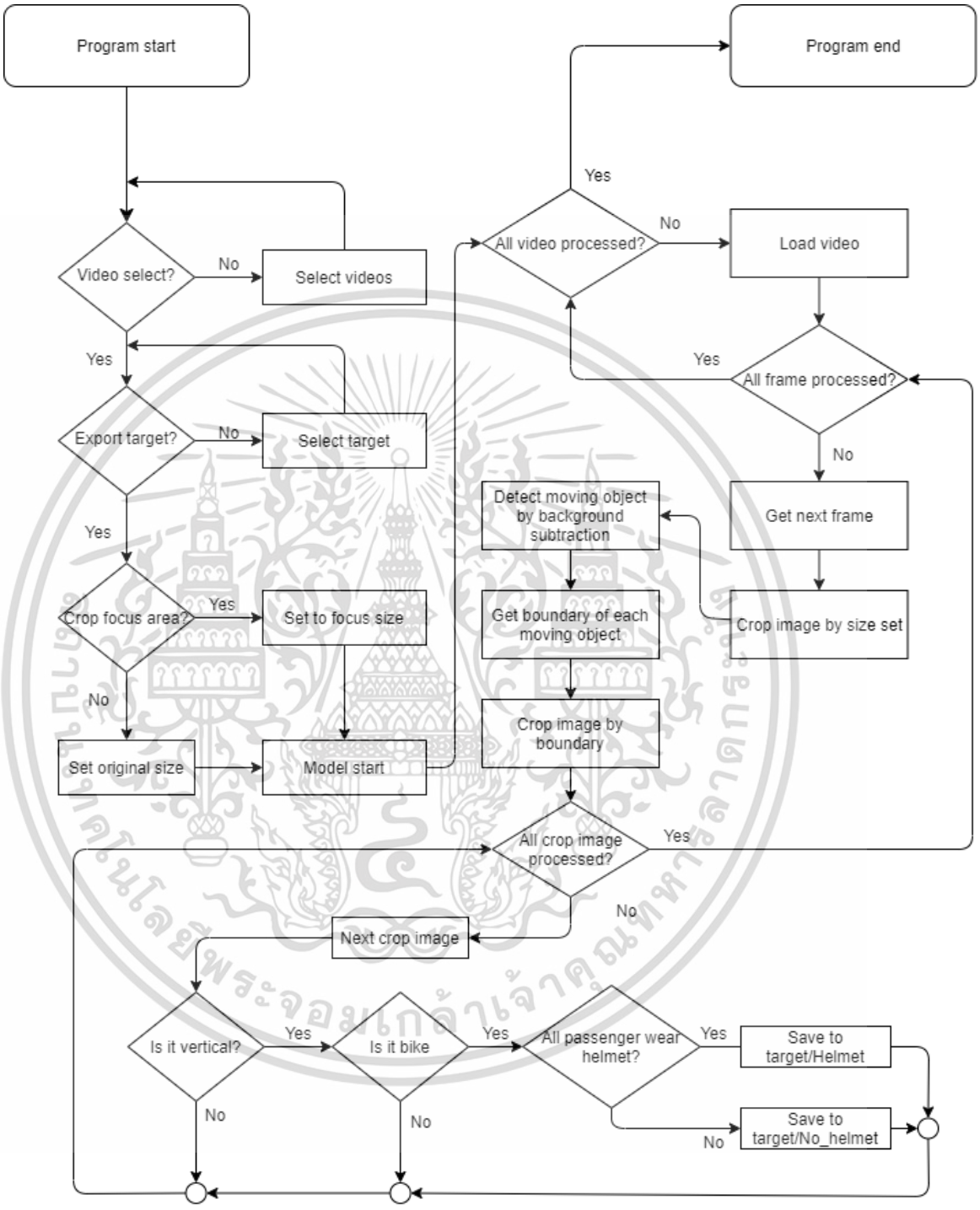
3.1.1 Input

- ข้อมูลที่นำเข้าต้องเป็นภาพเคลื่อนไหวเท่านั้น
- ต้องเป็นภาพสี
- ภาพรถจักรยานยนต์ที่จับได้ต้องมีขนาดใหญ่กว่า 299x299 พิกเซล

3.1.2 Functional

- สามารถลบภาพนิ่งพื้นหลังได้
- สามารถตีกรอบภาพวัตถุที่เคลื่อนไหวได้
- สามารถแยกแยะรูปว่าเป็นรถจักรยานยนต์ได้
- สามารถแยกแยะผู้สวมหมวกกันน็อคกับผู้ไม่สวมหมวกกันน็อคได้
- สามารถจับภาพผู้สวมหมวกกันน็อคกับผู้ไม่สวมหมวกกันน็อค จากภาพเคลื่อนไหวและบันทึกภาพเก็บไว้ได้

3.2 ภาพรวมของระบบ



รูปที่ 3.1 ภาพการทำงานโดยรวมของระบบ

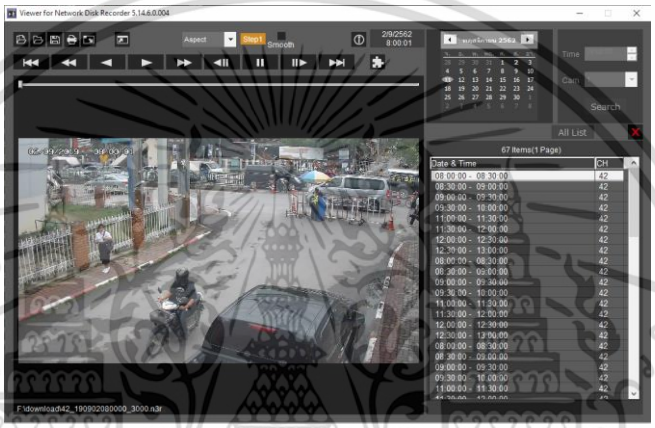
จากรูปที่ 3.1 ในขั้นตอนแรกของการทำงานผู้ใช้ต้องทำการเลือกไฟล์วิดีโอที่ต้องการใช้ในการทำนายซึ่งสามารถเลือกหลายไฟล์ได้ในครั้งเดียว หลังจากที่เลือกไฟล์วิดีโอที่ต้องการเรียบร้อยตัวระบบจะเก็บข้อมูลความละเอียดของไฟล์วิดีโอนั้นไว้และนำภาพตัวอย่างมาแสดง จากนั้นทำการเลือกที่อยู่สำหรับเก็บภาพที่ได้จากการทำนาย หลังจากทำการเลือกแล้วถ้าต้องการเลือกจุดหรือมุมที่ต้องการเน้นในการทำนายโดยเฉพาะให้ทำการกำหนดตำแหน่งโดยการลากกรอบสี่เหลี่ยมบนรูปตัวอย่าง(ถ้าไม่กำหนดจะตั้งค่าให้เป็นขนาดเต็มของไฟล์วิดีโอนั้น) จากนั้นจะเข้าสู่กระบวนการการทำงานในส่วนของการทำนายโดยเริ่มจากการโหลดไฟล์วิดีโอและทำการใช้ background subtraction ในการตรวจหาเฟรมของวิดีโอที่มีการเปลี่ยนแปลง ซึ่งมีหน้าที่ในการจับภาพวัตถุที่เคลื่อนไหวโดยใช้เฟรมก่อนหน้านี้เทียบกับเฟรมปัจจุบัน จากนั้นทำการตีกรอบวัตถุที่เคลื่อนไหวจะทำให้ได้พิกัดของวัตถุที่เคลื่อนไหวจำนวน 4 พิกัด จากนั้นใช้โมเดล InceptionResNetV2 เพื่อทำนายหารูปที่เป็นรถจักรยานยนต์ โดยในตัวโมเดลของ InceptionResNetV2 มี Label วัตถุ 1000 ชนิด โดยถ้ารถจักรยานยนต์อยู่ภายใน 5 อันดับแรกของการทำนายจะตัดสินว่าวัตถุนั้นเป็นรถจักรยานยนต์ และจะทำการพลิกภาพเพื่อตรวจสอบอีกครั้ง จากนั้นรูปที่ไม่ใช่รถจักรยานยนต์จะทำการข้ามไป ส่วนรูปที่เป็นรถจักรยานยนต์จะเข้าสู่กระบวนเพื่อทำการแยกแยะระหว่างผู้สวมหมวกกันน็อคกับผู้ที่ไม่สวมหมวกกันน็อคโดยใช้โมเดลที่พัฒนาขึ้นมาจากโมเดล ResNet152V2, Inceptionresnetv2 และ Inceptionv3 ผ่านกระบวนการ Transfer Learning โดยกำหนด Label ใหม่ให้มีจำนวน 2 อย่างคือ ถูกกฎหมายหรือไม่สามารถระบุได้ (legal) และผิดกฎหมาย (Illegal) หลังจากที่ทำการทำนายแล้วจะทำการบันทึกรูปในที่อยู่ที่ได้ทำการเลือกไว้โดยจะแยกไฟล์ไว้ 2 ไฟล์คือไฟล์ที่ทำนายว่าสวมหมวก และไฟล์ที่ทำนายว่าไม่สวมหมวก

3.3 ขั้นตอนการทำงาน

3.3.1 การเตรียมข้อมูล

ข้อมูลที่นำมานั้นเป็นไฟล์นามสกุล n3r ซึ่งเป็นไฟล์ของกล้องวงจรปิดของบริษัท Panasonic โดยได้ความอนุเคราะห์มาจากสำนักงานบริหารทรัพยากรกายภาพและสิ่งแวดล้อม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

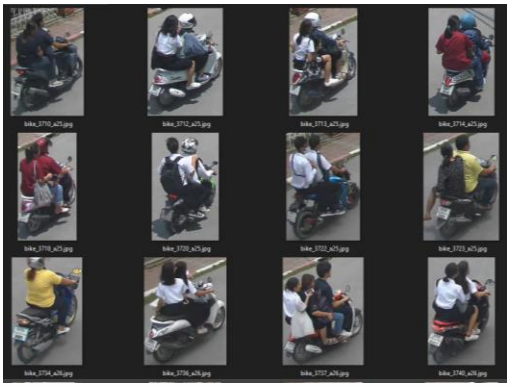
จากนั้นทำการแปลงไฟล์นามสกุล n3r เป็นไฟล์นามสกุล mp4 เนื่องจากไฟล์ n3r ไม่สามารถเปิดเพื่ออ่านเฟรมของวิดีโอได้ หลังจากได้ไฟล์ mp4 แล้วต่อไปจะทำการตัดภาพในขั้นตอนต่อไป



รูปที่ 3.2 ไฟล์นามสกุล n3r ที่เปิดโดยใช้โปรแกรม NX_Viewer

3.3.2 การแยกข้อมูล

หลังจากได้ไฟล์เป็นนามสกุล mp4 แล้ว ขั้นตอนต่อไปเป็นการตัดภาพรถจักรยานยนต์ในไฟล์ mp4 ที่ทำการแปลงมา โดยใช้โมเดล InceptionResNetV2 ผ่าน Library ของ Keras ในการทำนายหารถจักรยานยนต์ และใช้ Library ของ opencv ช่วยในการทำ background subtraction คือจับภาพเฟรมที่มีการเปลี่ยนแปลงแล้วทำการตีกรอบรูปนั้น จะได้พิกัดของกรอบจากนั้นเพื่อนำไปผ่านการทำนายของโมเดล InceptionResNetV2 ว่าเป็นรถจักรยานยนต์หรือไม่ ถ้าเป็นรถจักรยานยนต์จะทำการตัดรูปและบันทึกเป็นไฟล์นามสกุล jpg

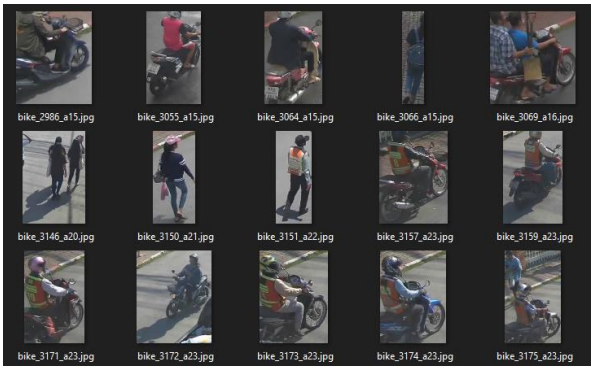


รูปที่ 3.3 ตัวอย่างรูปที่ได้จากการตัดภาพในขั้นตอนการแยกข้อมูล

หลังจากตัดภาพรถจักรยานยนต์ออกมาได้แล้วจะทำการแยกระหว่างรูปที่ผู้สวมหมวกกันน็อคกับผู้ที่ไม่สวมหมวกกันน็อค โดยได้ รูปผู้ที่สวมหมวกกันน็อค จำนวน 5,802 รูป รูปผู้ที่ไม่สวมหมวกกันน็อค จำนวน 4,289 รูป และรูปเสียจำนวน 2,880 รูป

รูปที่ 3.4 ตัวอย่างรูปผู้ที่สวมหมวกกันน็อค

รูปที่ 3.5 ตัวอย่างรูปผู้ที่ไม่สวมหมวกกันน็อค



รูปที่ 3.6 ตัวอย่างรูปที่เสีย

3.3.3. การสร้างโมเดล

นำเข้าโมเดลที่มีให้ใน Library ของ Keras โดยนำ layer สุดท้ายที่ใช้ทำนาย Class ของ ImageNet ออกไป จากนั้นสร้าง Layer ของ Dense (Fully connected) 2 node เพื่อใช้ทำนาย Class เพื่อแยกระหว่างผู้สวมหมวกกันน็อคกับผู้ไม่สวมหมวกกันน็อค ต่อมาจะทำการปรับค่า model ให้บางส่วนเป็น trainable ในส่วนของ Train model บน Google Colab กับ Dataset ที่ทำการแยกไว้ โดยได้ทำการแบ่งข้อมูลสำหรับการสร้างโมเดลไว้ ในขั้นตอนการแยกข้อมูล ซึ่งได้ทำการสุ่มหยิบรูปใน Dataset ทั้งหมดเป็นจำนวน 6,000 รูป เป็นข้อมูลสำหรับการเทรนโมเดล ซึ่งมีรายละเอียดดังนี้ใน

1.รูปของผู้ที่สวมหมวกกันน็อคจำนวน 2,050 รูป

2.รูปภาพที่เสียจำนวน 950 รูป

3.รูปของผู้ที่ไม่สวมหมวกกันน็อค จำนวน 3,400 รูป

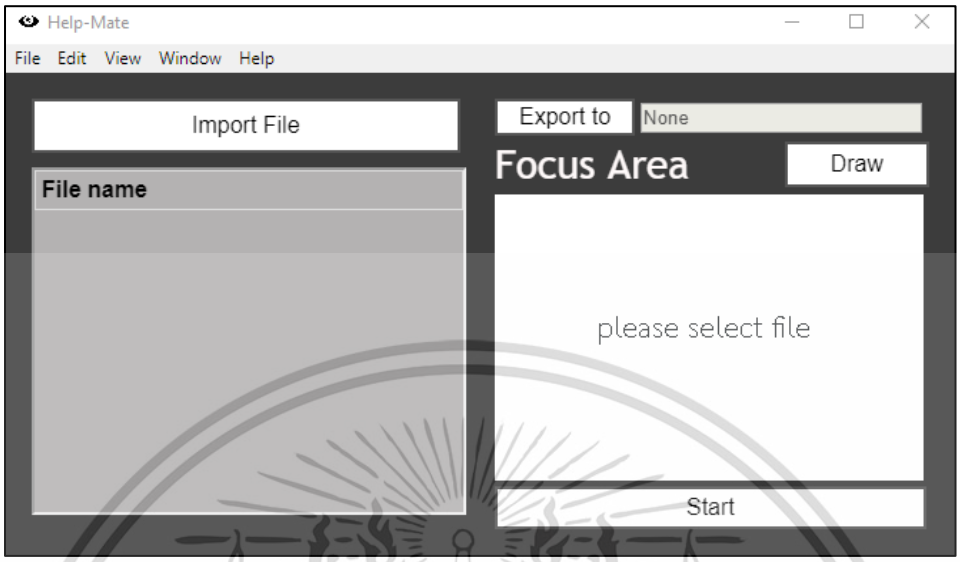
โดยรูปภาพที่เสียจะนำไปรวมกับรูปของผู้ที่สวมหมวกกันน็อค ซึ่งรวมกันเป็น 3,400 รูป เท่ากับจำนวนรูปของผู้ที่ไม่สวมหมวกกันน็อค นอกจากนั้นยังมีรูปที่ใช้ในการทดสอบแยกไว้อีกจำนวน 2,400 รูป โดยแบ่งเป็น 2 ส่วนคือ

1.รูปที่ใช้ในการทดสอบผู้ที่สวมหมวกกันน็อคจำนวน 1,200 รูป

2.รูปที่ใช้ในการทดสอบผู้ที่ไม่สวมหมวกกันน็อคจำนวน 1,200 รูป

จากนั้นทำการศึกษาผลลัพธ์ที่ได้จากการเทรนโมเดล จากจำนวนของ epoch และ จำนวนของ Layer ที่กำหนดให้ใช้ในการเทรนได้ ทำการบันทึกโมเดลที่ดีที่สุดมาใช้ โดยโมเดลทั้งหมดที่ใช้ในการเทรนได้แก่ ResNet152V2, Inceptionresnetv2 และ Inceptionv3

3.4 ส่วนติดต่อผู้ใช้งาน



รูปที่ 3.7 หน้าหลักของส่วนติดต่อผู้ใช้งาน

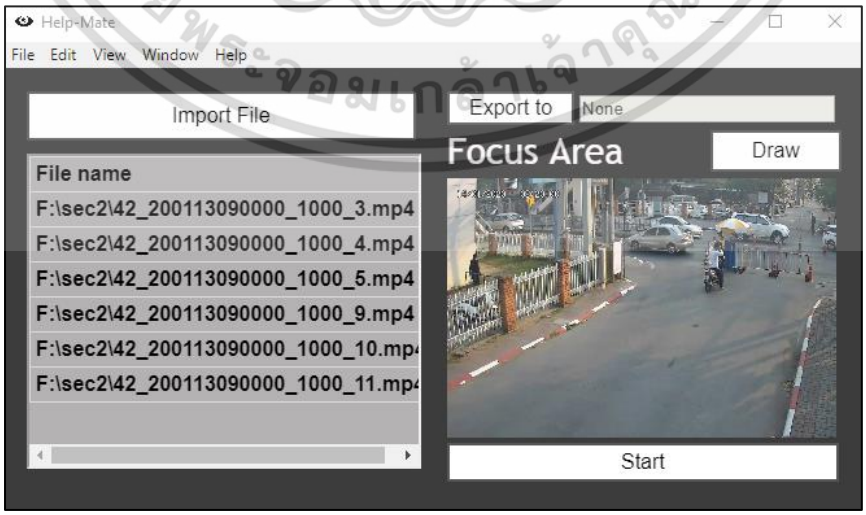
หน้าหลักของส่วนติดต่อผู้ใช้งานประกอบไปด้วย

1.ปุ่ม Import file ใช้ในการเลือกไฟล์วิดีโอที่ต้องการใช้เพื่อแยกผู้สวมหมวกกับผู้ไม่สวมหมวก

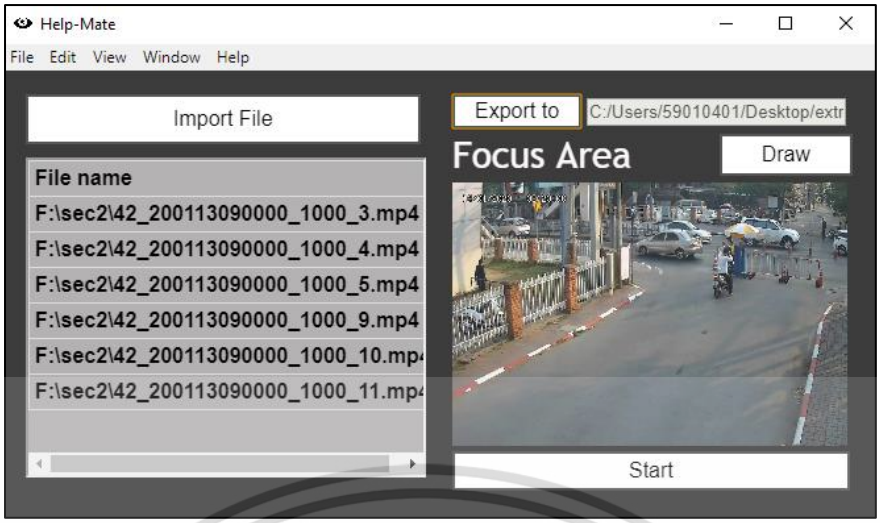
2.ปุ่ม Export to ใช้ในการเลือกที่อยู่ในการจัดเก็บภาพที่ได้จากการทำนาย

3.ปุ่ม Draw ใช้เพื่อกำหนดจุดที่ต้องการทำนาย ซึ่งถ้าหากไม่เลือกจะทำการทำนายโดยใช้ขนาดเต็มของไฟล์วิดีโอนั้น

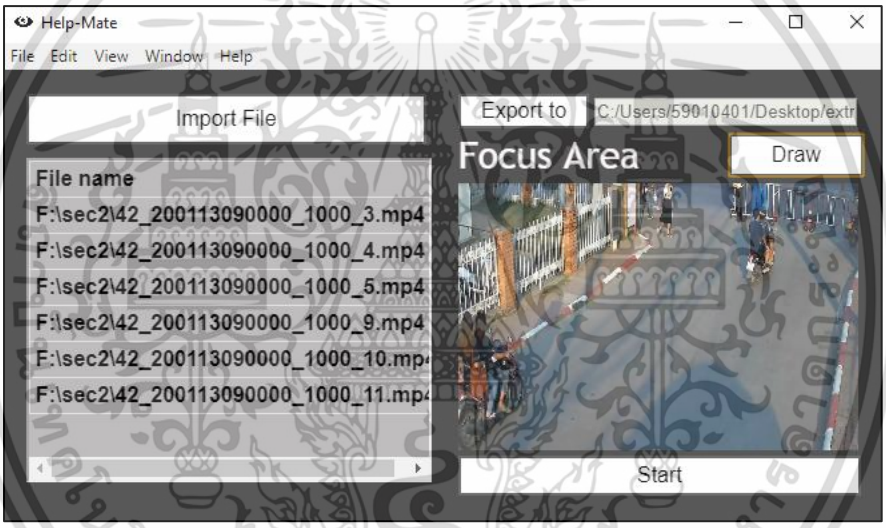
4.ปุ่ม Start ใช้ในการเริ่มการทำงานของโปรแกรม



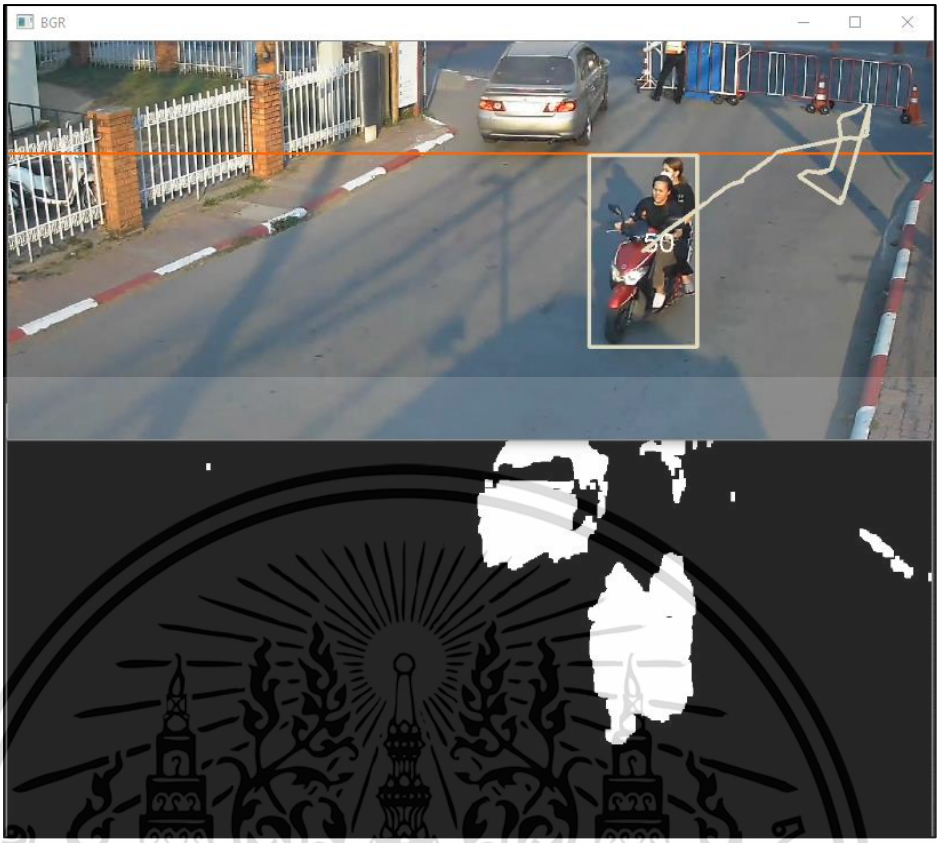
รูปที่ 3.8 ตัวอย่างการทำงานเมื่อทำการเลือกไฟล์วิดีโอ



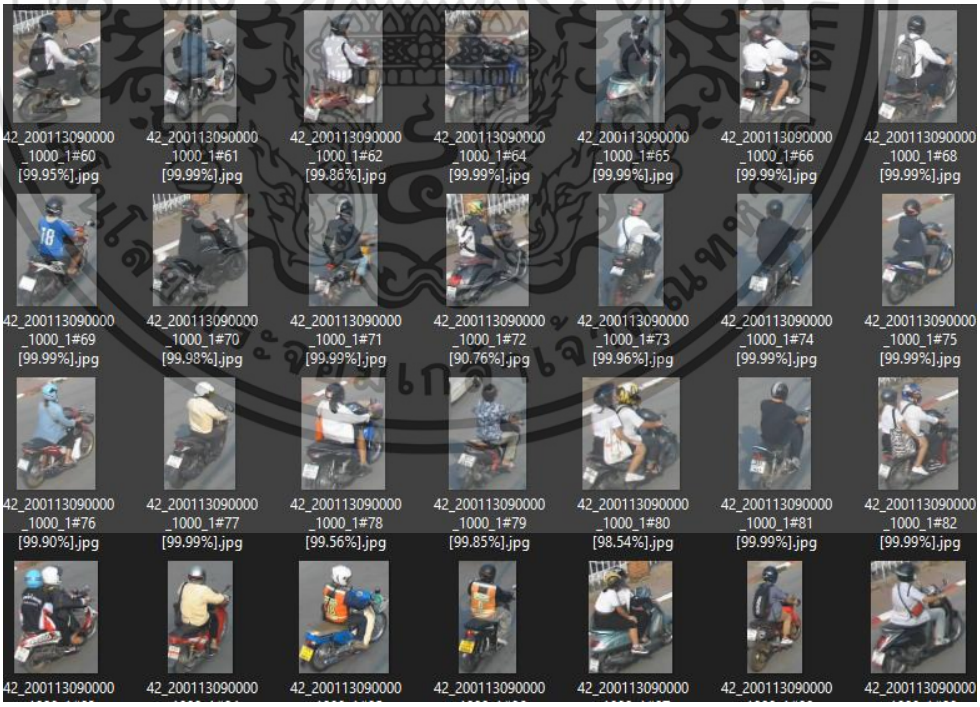
รูปที่ 3.9 ตัวอย่างการทำงานเมื่อทำการเลือกที่อยู่ในการจัดเก็บ



รูปที่ 3.10 ตัวอย่างการทำงานเมื่อทำการเลือกตำแหน่งที่ต้องการเน้นแล้ว



รูปที่ 3.11 ตัวอย่างการทำงานเมื่อเริ่มการทำนาย



รูปที่ 3.12 ตัวอย่างภาพที่ได้จากการทำนาย

บทที่ 4

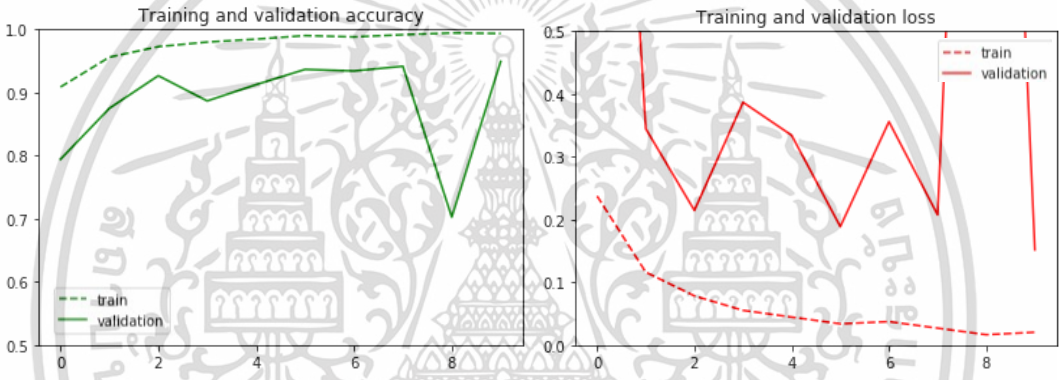
การทดลองและผลการทดลอง

4.1 การสร้างโมเดล

ในการสร้างโมเดลของโครงงานนี้ได้ใช้ Epoch = 10 รอบเนื่องจากโมเดลทั้ง 3 ที่ใช้เริ่มมีการ Overfitting เกิดขึ้นในช่วงหลังของ Epoch ที่ 10 และจำนวน Layer ของโมเดลที่ไม่ต้องการนำมาเทรนโมเดลคิดเป็นร้อยละ 50 ของจำนวน Layer ที่มีอยู่ในโมเดลนั้นได้ผลลัพธ์ดังนี้

4.1.1 ResNet152V2

ได้ความแม่นยำอยู่ที่ร้อยละ 94 และค่าความผิดพลาดอยู่ที่ร้อยละ 16

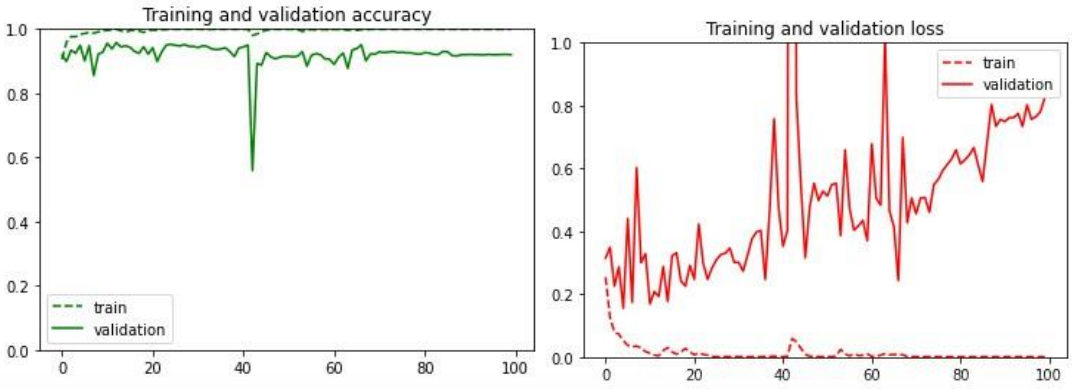


ก) ข)

รูปที่ 4.1 กราฟแสดงผลลัพธ์ที่ได้ของโมเดล ResNet152V2 ที่ Epoch = 10

ก) กราฟแสดงความแม่นยำในการสร้างโมเดลของ ResNet152V2

ข) กราฟแสดงความผิดพลาดในการสร้างโมเดลของ ResNet152V2



ก) ข)

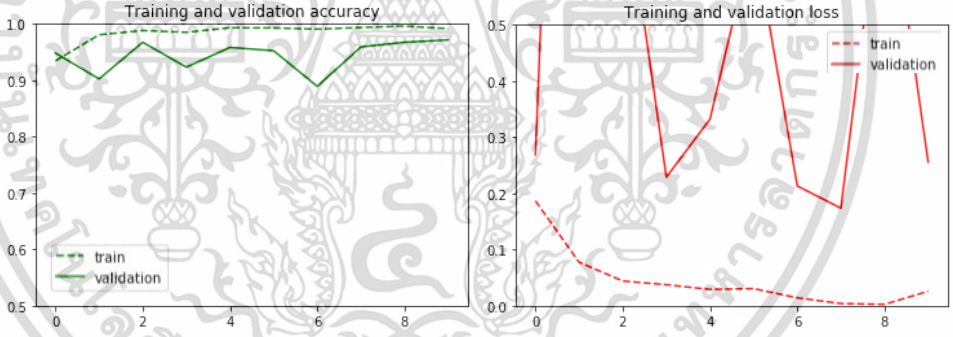
รูปที่ 4.2 กราฟแสดงผลลัพธ์ที่ได้ของโมเดล ResNet152V2 ที่ Epoch = 100

ก) กราฟแสดงความแม่นยำในการสร้างโมเดลของ ResNet152V2

ข) กราฟแสดงความผิดพลาดในการสร้างโมเดลของ ResNet152V2

4.1.2 InceptionResnetV2

ได้ความแม่นยำอยู่ที่ร้อยละ 97 และค่าความผิดพลาดอยู่ที่ร้อยละ 10

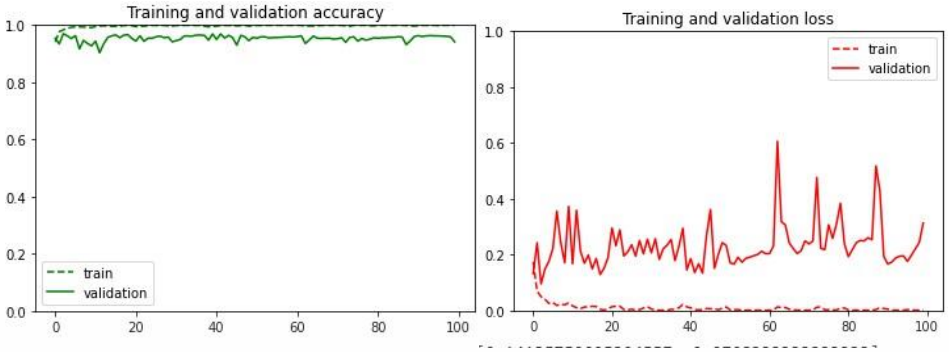


ก) ข)

รูปที่ 4.3 กราฟแสดงผลลัพธ์ที่ได้ของโมเดล InceptionResnetV2 ที่ Epoch = 10

ก) กราฟแสดงความแม่นยำในการสร้างโมเดลของ InceptionResnetV2

ข) กราฟแสดงความผิดพลาดในการสร้างโมเดลของ InceptionResnetV2



ก) ข)

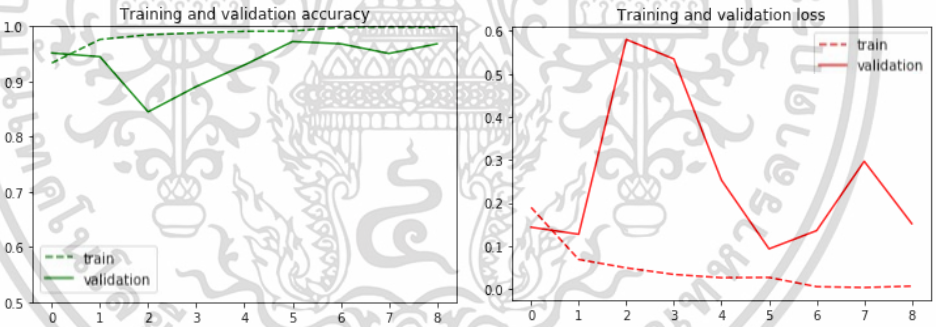
รูปที่ 4.4 กราฟแสดงผลลัพธ์ที่ได้ของโมเดล InceptionResnetV2 ที่ Epoch = 100

ก) กราฟแสดงความแม่นยำในการสร้างโมเดลของ InceptionResnetV2

ข) กราฟแสดงความผิดพลาดในการสร้างโมเดลของ InceptionResnetV2

4.1.3 Inceptionv3

ได้ความแม่นยำอยู่ที่ร้อยละ 96 และค่าความผิดพลาดอยู่ที่ร้อยละ 18

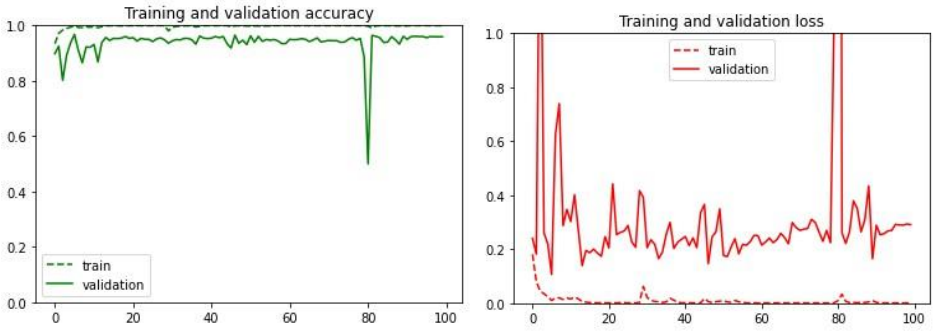


ก) ข)

รูปที่ 4.5 กราฟแสดงผลลัพธ์ที่ได้ของโมเดล Inceptionv3 ที่ Epoch = 10

ก) กราฟแสดงความแม่นยำในการสร้างโมเดลของ Inceptionv3

ข) กราฟแสดงความผิดพลาดในการสร้างโมเดลของ Inceptionv3



ก) ข)

รูปที่ 4.6 กราฟแสดงผลลัพธ์ที่ได้ของโมเดล Inceptionv3 ที่ Epoch = 100

ก) กราฟแสดงความแม่นยำในการสร้างโมเดลของ Inceptionv3

ข) กราฟแสดงความผิดพลาดในการสร้างโมเดลของ Inceptionv3

จากรูปที่ 4.2 , 4.4 และ 4.6 จะสังเกตว่ากราฟเกิดการ spike ขึ้นซึ่งเกิดขึ้นจากขั้นตอนการกำหนดค่าในการเทรนโมเดลค่าของ batch size ให้มีขนาดเท่ากับ 30 ซึ่งเป็นขนาดที่ใหญ่

4.2 การทดสอบโมเดลโดยใช้ Test set จากข้อมูลที่ใช้สร้างโมเดล

ผลจากการนำโมเดลที่สร้างขึ้นมาทดสอบกับ test set โดยแยกเป็นผู้ที่สวมหมวกกันน็อคกับผู่ที่ไม่สวมหมวกกันน็อค ได้ผลลัพธ์ดังนี้

4.2.1 ResNet152V2

ทดสอบกับ test set ของผู้ที่สวมหมวกกันน็อคจำนวน 1,200 ผิดพลาด 67 รูป

ทดสอบกับ test set ของผู้ที่ไม่สวมหมวกกันน็อคจำนวน 1,200 ผิดพลาด 58 รูป

4.2.1.1 รูปที่ผิดพลาดจาก test set ผู้ที่สวมหมวกกันน็อค

รูปที่ได้จากการทดสอบนั้นส่วนมากจะนับรวมคนทุกคนที่อยู่ในรูปว่าเป็นผู้ขับขี่รถจักรยานยนต์ด้วย และผมที่เลยออกมาจากหมวกกันน็อคก็มีผลต่อการทำนายด้วย

รูปที่ 4.7 ตัวอย่างรูปที่ผิดพลาดจาก test set ผู้ที่สวมหมวกกันน็อค

4.2.1.2 รูปที่ผิดพลาดจาก test set ผู้ที่ไม่สวมหมวกกันน็อค

รูปที่ได้จากการทดสอบทำให้เห็นว่าสีบริเวณศีรษะมีผลต่อการทำนาย ไม่ว่าจะเป็นสีของผมหรือสีของหมวกที่ไม่ใช่หมวกกันน็อค

รูปที่ 4.8 ตัวอย่างรูปที่ผิดพลาดจาก test set ผู้ที่ไม่สวมหมวกกันน็อค

4.2.2 InceptionResnetV2

ทดสอบกับ test set ของผู้ที่สวมหมวกกันน็อคจำนวน 1,200 ผิดพลาด 40 รูป

ทดสอบกับ test set ของผู้ที่ไม่สวมหมวกกันน็อคจำนวน 1,200 ผิดพลาด 27 รูป

4.2.2.1 รูปที่ผิดพลาดจาก test set ผู้ที่สวมหมวกกันน็อค

รูปที่ได้จากการทดสอบนั้นส่วนมากจะนับรวมคนทุกคนที่อยู่ในรูปว่าเป็นผู้ขับขี่รถจักรยานยนต์ด้วย และผมที่เลยออกมาจากหมวกกันน็อคก็มีผลต่อการทำนายด้วย

รูปที่ 4.9 ตัวอย่างรูปที่ผิดพลาดจาก test set ผู้ที่สวมหมวกกันน็อค

4.2.2.2 รูปที่ผิดพลาดจาก test set ผู้ที่ไม่สวมหมวกกันน็อค

รูปที่ได้จากการทดสอบทำให้เห็นว่าสีบริเวณศีรษะมีผลต่อการทำนาย ไม่ว่าจะเป็นสีของผมหรือสีของหมวกที่ไม่ใช่หมวกกันน็อค และรูปร่างของทรงผมก็มีผลต่อการทำนายด้วย

รูปที่ 4.10 ตัวอย่างรูปที่ผิดพลาดจาก test set ผู้ที่ไม่สวมหมวกกันน็อค

4.2.3 Inceptionv3

ทดสอบกับ test set ของผู้ที่สวมหมวกกันน็อคจำนวน 1,200 ผิดพลาด 21 รูป

ทดสอบกับ test set ของผู้ที่ไม่สวมหมวกกันน็อคจำนวน 1,200 ผิดพลาด 35 รูป

4.2.3.1 รูปที่ผิดพลาดจาก test set ผู้ที่สวมหมวกกันน็อค

รูปที่ได้จากการทดสอบนั้นส่วนมากจะนับรวมคนทุกคนที่อยู่ในรูปว่าเป็นผู้ขับขี่รถจักรยานยนต์ด้วย และสีของหมวกกันน็อคที่ดูคล้ายกับพื้นหลังก็มีส่วนในการทำให้การทำนายมีความผิดพลาด

รูปที่ 4.11 ตัวอย่างรูปที่ผิดพลาดจาก test set ผู้ไม่สวมหมวกกันน็อค

4.2.3.2 รูปที่ผิดพลาดจาก test set ผู้ที่ไม่สวมหมวกกันน็อค

รูปที่ได้จากการทดสอบทำให้เห็นว่าสีบริเวณศีรษะมีผลต่อการทำนาย ไม่ว่าจะเป็นสีของผมหรือสีของหมวกที่ไม่ใช่หมวกกันน็อค

รูปที่ 4.12 ตัวอย่างรูปที่ผิดพลาดจาก test set ผู้ที่ไม่สวมหมวกกันน็อค

4.3 สรุปผลที่ได้จากการทดสอบโมเดลโดยใช้ Test set จากข้อมูลที่ใช้สร้างโมเดล

4.3.1 ResNet152V2

ตาราง 4.1 Confusion Matrix ของโมเดล ResNet152V2 ในการจำแนกคนสวมหมวกกับคนไม่สวมหมวก โดยใช้ Test set จากข้อมูลที่ใช้สร้างโมเดล

Prediction \ Label	Legal	Illegal
Legal	94.42%	5.58%
Illegal	4.83%	95.17%

4.3.2 InceptionResnetV2

ตาราง 4.2 Confusion Matrix ของโมเดล InceptionResnetV2 ในการจำแนกคนสวมหมวกกับคนไม่สวมหมวก โดยใช้ Test set จากข้อมูลที่ใช้สร้างโมเดล

Prediction \ Label	Legal	Illegal
Legal	96.67%	3.33%
Illegal	2.25%	97.75%

4.3.3 Inceptionv3

ตาราง 4.3 Confusion Matrix ของโมเดล Inceptionv3 ในการจำแนกคนสวมหมวกกับคนไม่สวมหมวก โดยใช้ Test set จากข้อมูลที่ใช้สร้างโมเดล

Prediction \ Label	Legal	Illegal
Legal	98.25%	1.75%
Illegal	2.92%	97.08%

ตาราง 4.4 สรุปผลความแม่นยำและความผิดพลาด โดยใช้ Test set จากข้อมูลที่ใช้สร้างโมเดล

หัวข้อ \ โมเดล	ResNet152V2	InceptionResnetV2	Inceptionv3
ความแม่นยำ	94.79%	97.21%	97.67%
ความผิดพลาด	5.21%	2.79%	2.33%

4.4 การทดสอบโมเดลโดยใช้ไฟล์วิดีโอ

ผลจากการนำโมเดลที่สร้างขึ้นมาทดสอบกับไฟล์วิดีโอ โดยไฟล์วิดีโอของกล้องวงจรปิดมีมุมกล้องที่แตกต่างกันจำนวน 3 มุม ซึ่งได้ผลลัพธ์ดังนี้

4.4.1 มุมบริเวณทางเข้า-ออกตึกนาโน

- จำนวนรูปจากการใช้โมเดล ResNet152V2 จำนวน 2,161 รูป
- จำนวนรูปจากการใช้โมเดล InceptionResnetV2 จำนวน 2,246 รูป
- จำนวนรูปจากการใช้โมเดล Inceptionv3 จำนวน 2,152 รูป

4.4.1.1 ความถูกต้องในการทำนาย

- โมเดล ResNet152V2 จำนวน 1,771 รูป
- โมเดล InceptionResnetV2 จำนวน 1,956 รูป
- โมเดล Inceptionv3 จำนวน 1,947 รูป

4.4.1.2 ความผิดพลาดในการทำนาย

- โมเดล ResNet152V2 จำนวน 260 รูป
- โมเดล InceptionResnetV2 จำนวน 130 รูป
- โมเดล Inceptionv3 จำนวน 113 รูป

4.4.1.3 ความผิดพลาดในการตรวจจับ

- โมเดล ResNet152V2 จำนวน 130 รูป
- โมเดล InceptionResnetV2 จำนวน 160 รูป
- โมเดล Inceptionv3 จำนวน 92 รูป

4.4.2 มุมทางม้าลายบริเวณถนนฉลองกรุง 1

- จำนวนรูปจากการใช้โมเดล ResNet152V2 จำนวน 15,542 รูป
- จำนวนรูปจากการใช้โมเดล InceptionResnetV2 จำนวน 15,668 รูป
- จำนวนรูปจากการใช้โมเดล Inceptionv3 จำนวน 15,663 รูป

4.4.2.1 ความถูกต้องในการทำนาย

- โมเดล ResNet152V2 จำนวน 11,641รูป
- โมเดล InceptionResnetV2 จำนวน 13,163รูป
- โมเดล Inceptionv3 จำนวน 13,934รูป

4.4.2.2 ความผิดพลาดในการทำนาย

- โมเดล ResNet152V2 จำนวน 2,571รูป
- โมเดล InceptionResnetV2 จำนวน 1,364รูป
- โมเดล Inceptionv3 จำนวน 681รูป

4.4.2.3 ความผิดพลาดในการตรวจจับ

- โมเดล ResNet152V2 จำนวน 1,330รูป
- โมเดล InceptionResnetV2 จำนวน 1,141รูป
- โมเดล Inceptionv3 จำนวน 1,048รูป

4.4.3 มุมบริเวณหลังอาคาร 12 ชั้น

- จำนวนรูปจากการใช้โมเดล ResNet152V2 จำนวน 13,935 รูป
- จำนวนรูปจากการใช้โมเดล InceptionResnetV2 จำนวน 13,964 รูป
- จำนวนรูปจากการใช้โมเดล Inceptionv3 จำนวน 13,987 รูป

4.4.3.1 ความถูกต้องในการทำนาย

- โมเดล ResNet152V2 จำนวน 8,242รูป
- โมเดล InceptionResnetV2 จำนวน 10,430รูป
- โมเดล Inceptionv3 จำนวน 12,707รูป

4.4.3.2 ความผิดพลาดในการทำนาย

- โมเดล ResNet152V2 จำนวน 3,358รูป
- โมเดล InceptionResnetV2 จำนวน 1,287รูป
- โมเดล Inceptionv3 จำนวน 801 รูป

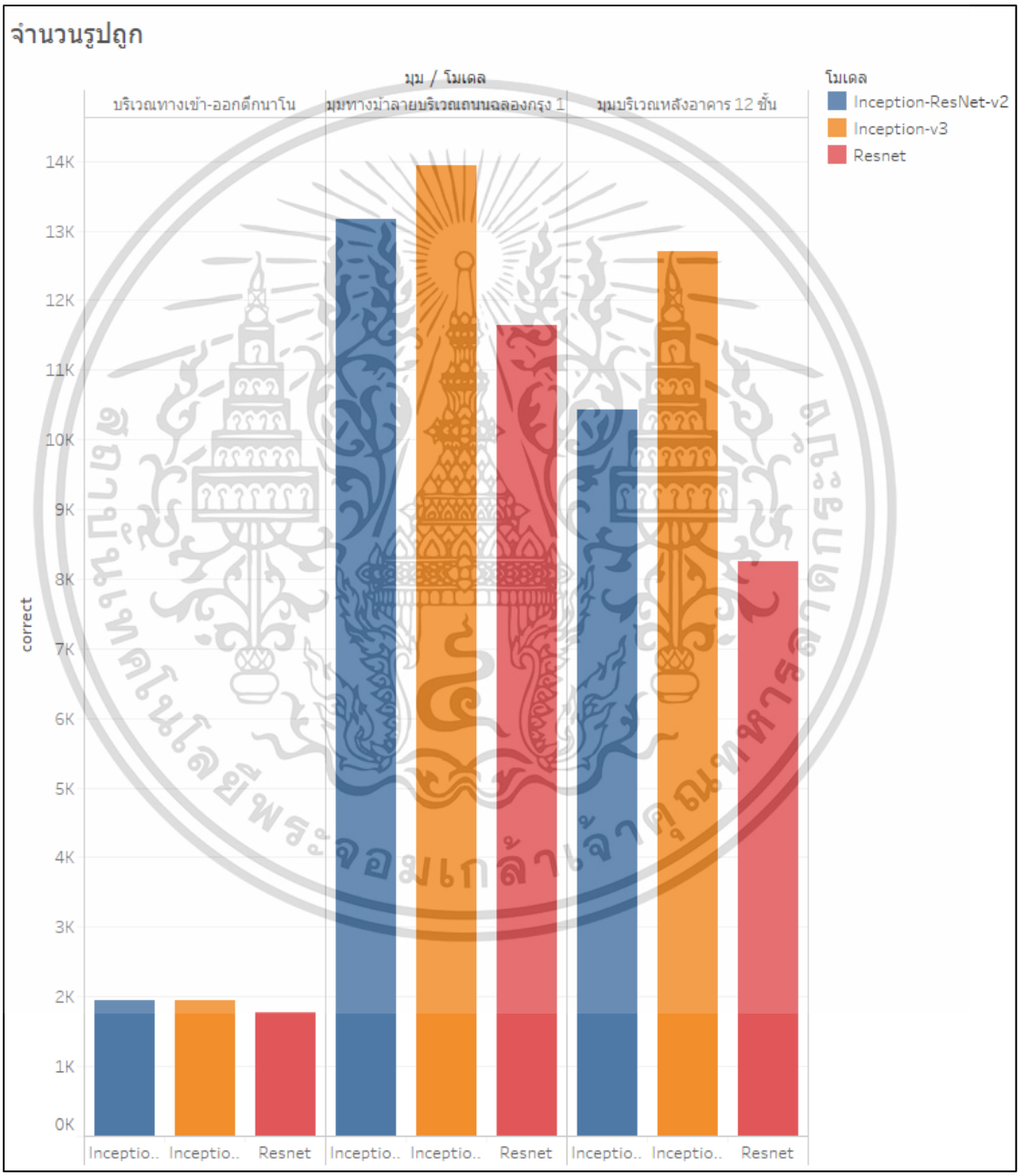
4.4.3.3 ความผิดพลาดในการตรวจจับ

- โมเดล ResNet152V2 จำนวน 2,335รูป
- โมเดล InceptionResnetV2 จำนวน 2,247รูป
- โมเดล Inceptionv3 จำนวน 479รูป

4.5 สรุปการทดสอบโมเดลโดยใช้ไฟล์วิดีโอ

4.5.1 ความถูกต้องในการทำนาย

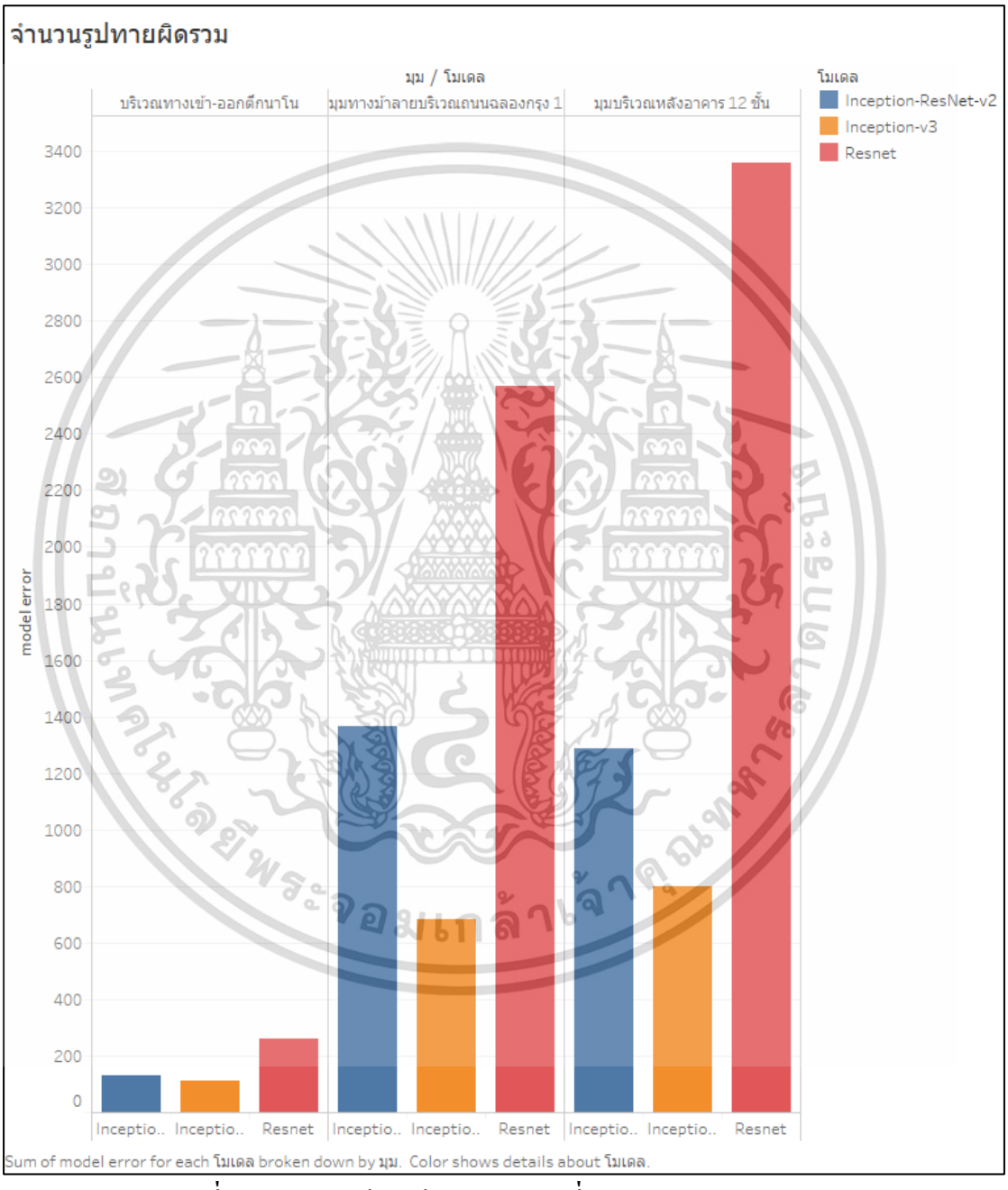
จากการทดสอบด้วยไฟล์วิดีโอพบว่าโมเดลที่มีความแม่นยำที่สุดคือ Inceptionv3 ซึ่งจะสังเกตได้ว่ามุมบริเวณทางเข้า-ออกตึกนาโน มีความถูกต้องใกล้เคียงกันเนื่องจากเป็นมุมที่ใช้สร้างโมเดล โดยผลจะเห็นได้ชัดเจนขึ้นถ้านำไปใช้กับมุมอื่นที่ไม่เคยใช้



รูปที่ 4.13 กราฟเปรียบเทียบจำนวนรูปที่ทำนายถูกต้องแยกตามมุม

4.5.2 ความผิดพลาดในการทำนาย

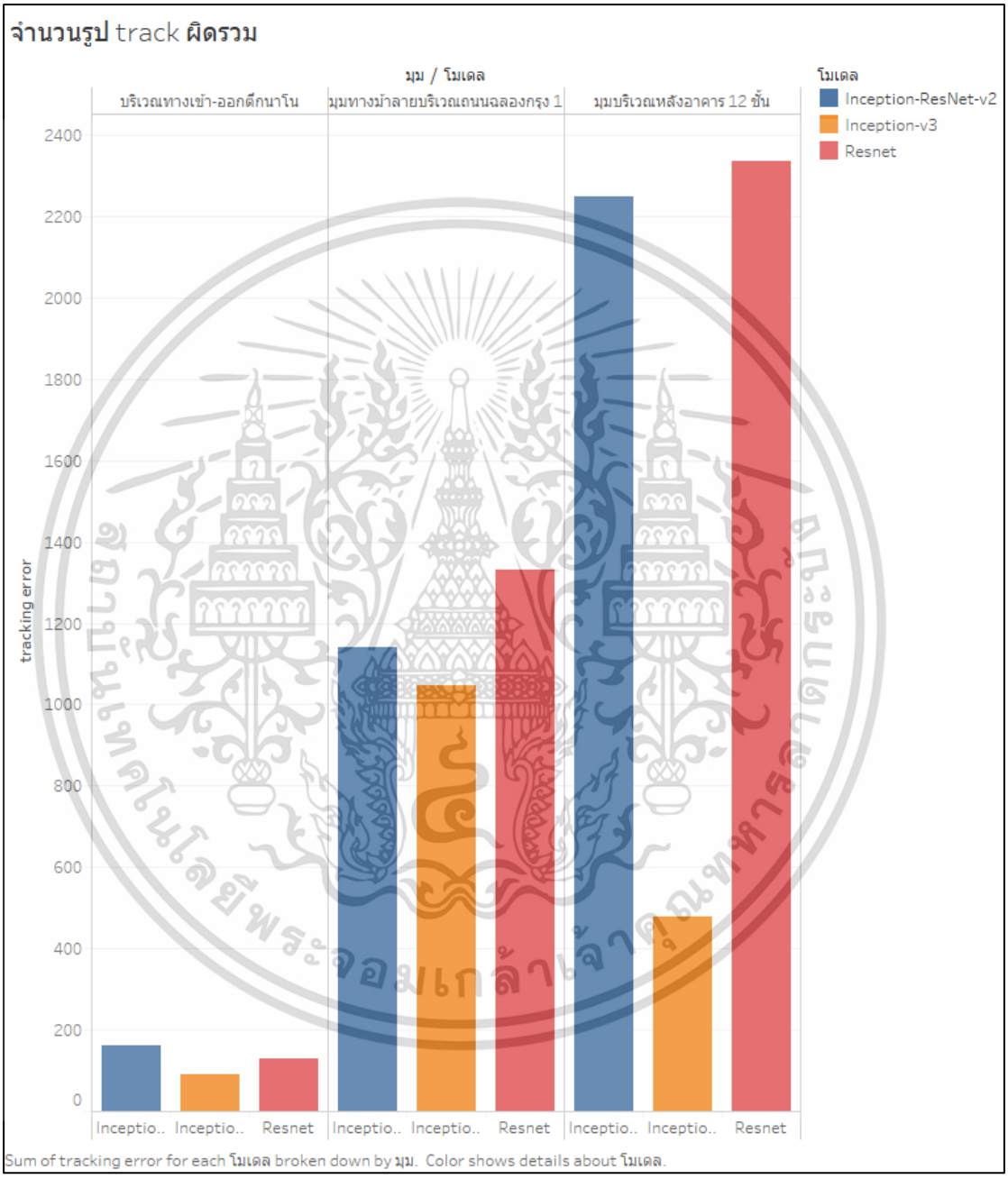
จากการทดสอบด้วยไฟล์วิดีโอพบว่าโมเดลที่มีความผิดพลาดมากที่สุดคือ ResNet152V2 ซึ่งจะสังเกตได้ว่ามุมบริเวณทางเข้า-ออกตึกนาโน มีความผิดพลาดน้อยเนื่องจากเป็นมุมที่ใช้สร้างโมเดล โดยจะเห็นความผิดพลาดมากขึ้นถ้านำไปใช้กับมุมอื่นที่ไม่เคยใช้



รูปที่ 4.14 กราฟเปรียบเทียบจำนวนรูปที่ทำนายผิดแยกตามมุม

4.5.3 ความผิดพลาดในการตรวจจับ

จากการทดสอบด้วยไฟล์วิดีโอพบว่าโมเดลที่มีอัตราการตรวจจับวัตถุผิดพลาดมากที่สุดคือ ResNet152V2 ซึ่งเป็นผลมาจากแสงและเงาจากสภาพแวดล้อมภายในวิดีโอนั้น



รูปที่ 4.15 กราฟเปรียบเทียบจำนวนรูปที่ตรวจจับวัตถุผิดแยกตามมุม

4.5.4 ตาราง Confusion Matrix ของโมเดล

4.5.4.1 ResNet152V2

ตาราง 4.5 Confusion Matrix ของโมเดล ResNet152V2 ในการจำแนกคนสวมหมวกกับคนไม่สวมหมวก โดยใช้ไฟล์วิดีโอ

Prediction \ Label	Legal	Illegal
Legal	49%	21%
Illegal	2%	28%

4.5.4.2 InceptionResnetV2

ตาราง 4.6 Confusion Matrix ของโมเดล InceptionResnetV2 ในการจำแนกคนสวมหมวกกับคนไม่สวมหมวก โดยใช้ไฟล์วิดีโอ

Prediction \ Label	Legal	Illegal
Legal	47%	7%
Illegal	3%	43%

4.5.4.3 Inceptionv3

ตาราง 4.7 Confusion Matrix ของโมเดล Inceptionv3 ในการจำแนกคนสวมหมวกกับคนไม่สวมหมวก โดยใช้ไฟล์วิดีโอ

Prediction \ Label	Legal	Illegal
Legal	53%	21%
Illegal	2%	28%

ตาราง 4.8 สรุปผลความแม่นยำโดยใช้ไฟล์วิดีโอ

โมเดล	ชนิดการทำนาย	จำนวนรูปที่ถูก	จำนวนรูปที่ผิด	ความแม่นยำ
ResNet152V2	สวมหมวก + ตรวจจับผิด	17,185	5,751	74.93%
	ไม่สวมหมวก	8,142	560	93.56%
InceptionResnetV2	สวมหมวก + ตรวจจับผิด	14,279	1,904	88.23%
	ไม่สวมหมวก	12,355	3,340	78.72%
Inceptionv3	สวมหมวก + ตรวจจับผิด	17,471	1,103	94.06%
	ไม่สวมหมวก	12,683	545	95.88%

ในตารางสรุปผลความแม่นยำโดยใช้ไฟล์วิดีโอ จะใช้จำนวนรูปที่ถูกรวมกับจำนวนรูปที่ตรวจจับผิด เพราะรูปที่ตรวจจับผิดได้นำไปใช้ในการสร้างโมเดลซึ่งใช้ใน Class ที่ระบุว่าสวมหมวก ดังนั้นรูปที่ตรวจจับผิดจึงนับว่าถูกต้องถ้ารูปที่ตรวจจับผิดถูกทำนายออกมาว่าสวมหมวก ส่วนถ้ารูปที่ตรวจจับผิดถูกทำนายออกมาว่าไม่สวมหมวกจะถือว่าผิด

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปผล

การดำเนินงานทั้งหมดใน 2 ภาคเรียนที่ผ่านมานี้ได้แก่ การศึกษาเทคโนโลยีที่ใช้ในการพัฒนาระบบตรวจจับผู้ขับขี่รถจักรยานยนต์ที่ไม่สวมหมวกกันน็อค การเก็บข้อมูล การสร้างชุดข้อมูลทั้งเพื่อสร้างโมเดลและทดสอบโมเดล ซึ่งสร้างโดยใช้ข้อมูลที่แปลงจากไฟล์ n3r ที่เป็นกล้องวงจรปิดเป็นไฟล์ mp4 จากนั้นใช้เทคนิค background subtraction ในการตรวจจับวัตุเคลื่อนไหวและใช้โมเดล InceptionResnetV2 ในการทำนายหารถจักรยานยนต์และทำการตัดภาพมาใช้เป็น Dataset จากนั้นทำการสร้างโมเดลผ่านกระบวนการ Transfer Learning โดยเปรียบเทียบความแตกต่างระหว่างโมเดล ResNet152V2, Inceptionresnetv2 และ Inceptionv3 โดยแยกเป็นผู้ทำถูกกฎหมายกับผู้ทำผิดกฎหมาย ทำการวัดผลความแม่นยำและความผิดพลาดที่ได้จากข้อมูล Test set ที่เป็นรูปภาพซึ่งรูปที่นำมาทดสอบเป็ยรูปมุมกล้องเดียวกันทั้งหมด หลังจากที่สร้างโมเดลทั้ง 3 เสร็จแล้ว ขั้นตอนต่อไปที่ได้ทำคือการสร้างส่วนติดต่อผู้ใช้งานขึ้นซึ่งจะทำให้สะดวกในการทดสอบและดูได้ง่ายขึ้น เมื่อสร้างส่วนติดต่อผู้ใช้งานแล้ว ต่อไปได้นำไฟล์วิดีโอจากกล้องวงจรปิดของ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งมีความแตกต่างกัน 3 มุมมาทดสอบกับโมเดลทั้ง 3 ผล ปรากฏว่าโมเดลที่พัฒนาจาก Inceptionv3 มีความแม่นยำมากที่สุดซึ่งมีความแม่นยำในการทำนายผู้สวมหมวกกันน็อค 94.06% และความแม่นยำในการทำนายผู้ไม่สวมหมวกกันน็อค 95.88%

5.2 ปัญหาและแนวทางแก้ไข

1) การจัดทำ Dataset จากการใช้โมเดล InceptionResnetV2 ในการทำนายหารถจักรยานยนต์มีความผิดพลาดในการทำนาย จึงทำการตั้งเงื่อนไขคือถ้าใน 5 อันดับแรกของการทำนายเป็นรถจักรยานยนต์ให้ตีความว่าวัตถุนั้นเป็นรถจักรยานยนต์

2) จากข้อที่ 1 การทำนายมีการทำนายผิดพลาด ซึ่งตัดภาพวัตถุอื่นที่ไม่ใช่รถจักรยานยนต์ติดมาด้วย จึงทำการนำรูปนั้นไปใส่ใน Dataset เพื่อใช้ในการสร้างโมเดลด้วย เพราะเมื่อตรวจพบวัตถุนั้นจะได้เห็นว่าไม่ทำผิดกฎหมาย

3)ในการทดสอบผ่านไฟล์วิดีโอจากมุมที่ต่างกันพบว่ามีความผิดพลาดเยอะกว่ามุมที่ใช้สร้างโมเดลอย่างเห็นได้ชัดจึงควรนำมุมที่ต้องการทำนายมาใช้ในการสร้างโมเดลด้วย

5.3 แนวทางการพัฒนาต่อ

หากต้องการความแม่นยำที่มากขึ้นควรนำภาพที่ได้จากการทดสอบผ่านไฟล์วิดีโอกล้องวงจรปิดทุกมุมที่ต้องการนำไปใช้ทำนายมาเป็น Data set ในการสร้างและพัฒนาโมเดลต่อไปซึ่งสามารถทำได้สะดวกยิ่งขึ้นเนื่องจากได้มีการพัฒนาในส่วนของการเก็บ Data set ไว้แล้ว

บรรณานุกรม

Sumit Saha 2018. **A Comprehensive Guide to Convolutional Neural Networks** [online]. Available https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Open Source Computer Vision 2017. **How to Use Background Subtraction Methods** [online]. https://docs.opencv.org/3.4/d1/dc5/tutorial_background_subtraction.html

Adrian Rosebrock 2018. **Simple object tracking with OpenCV** [online]. https://www.pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv

Open Source Computer Vision 2017. **Contours: Getting Started** [online]. https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html

Open Source Computer Vision 2017. **Morphological Transformations** [online]. https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html

Adrian Rosebrock 2017. **Image classification with Keras and deep learning** [online]. https://www.pyimagesearch.com/2017/12/11/image-classification-with-keras-and-deep-learning/

Aditya Ananthram 2018. **Deep Learning For Beginners Using Transfer Learning In Keras** [online]. https://towardsdatascience.com/keras-transfer-learning-for-beginners-6c9b8b7143e

Bharath Raj 2018. **A Simple Guide to the Versions of the Inception Network** [online]. https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202

google colab 2019. **Colab** [online]. https://colab.research.google.com

google colab 2019. **Colab.External data: Local Files, Drive, Sheets, and Cloud Storage** [online]. https://colab.research.google.com/notebooks/io.ipynb#scrollTo=BaCkyg5CV5jF

Keras 2018. **Keras The Python Deep Learning library** [online]. https://keras.io

Keras 2018. **Keras.Applications** [online]. https://keras.io/applications

ราชกิจจานุเบกษา 2019. **พระราชบัญญัติ จราจรทางบก (ฉบับที่ 12) พ.ศ. 2562** [online]. http://www.ratchakitcha.soc.go.th/DATA/PDF/2562/A/067/T_0067.PDF

พระราชบัญญัติ 1979. **พระราชบัญญัติ จราจรทางบก พ.ศ. 2522** [online]. http://kmcenter.rid.go.th/kmc17/data/Knowledge%2017/Legal/law_traffic2522.pdf