บทความวิจัย

Computer Science Department
Faculty of Informatics, Mahasarakham University

แอปพลิเคชันจำแนกสายพันธุ์งูที่พบในไทยด้วย Deep Learning

Thai Snake Classification Application Using Deep Learning

ศุภกิตติ์ เฉิดฉาย, ปรเมศร์ บ่อแก้ว, พรทิวา ปะวะระ
สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม
j.pbpromammothz@gmail.com, boosh23.2544@gmail.com, pornntiwa.p@msu.ac.th

Computer Science Department
Faculty of Informatics, Mahasarakham University

บทคัดย่อ

โปรเจคนี้เป็นการสร้าง Mobile Application ในการจำแนกสายพันธุ์งู มีวัตถุประสงค์เพื่อจำแนกสายพันธุ์ของงูและบอกรายละเอียดเกี่ยวกับงูชนิดนั้นพร้อมวิธีการปฐมพยาบาลเบื้องต้น ทำให้รู้วิธีในการรับมือกับงูที่พบเจอ ในการจำแนกสายพันธุ์งูเรามีกระบวนการหลัก ๆ 2 อย่างคือ (1) การทำ Object Detection เพื่อหาตำแหน่งของงูในภาพก่อน เพราะสภาพแวดล้อมในภาพจะทำให้ Classification ทำได้ยาก (2) นำตำแหน่งงูที่ได้จากการทำ Object Detection มาใช้ทำ Classification เพื่อให้ได้ความแม่นยำเพิ่มขึ้น ในการแยกความต่างของงูแต่ละสายพันธุ์ดังนั้นโครงงานฉบับนี้จึงนำเสนอการทำ Mobile application เพื่อจำแนกสายพันธุ์งู และบอกรายละเอียดเกี่ยวกับงูที่พบเจอเจอเพื่อเป็นประโยชน์ให้แก่ผู้ใช้ โดยงานนี้จะอยู่ลักษณะของการแยกสายพันธุ์งูพร้อมบอกรายละเอียดของสายพันธุ์นั้นๆ โดยอาศัยการจำแนกภาพ (Classification) พร้อมบอกวิธีการปฐมพยาบาลเบื้องต้น

1.บทนำ

บนโลกใบนี้มีเหล่าสัตว์ที่เป็นอันตรายต่อมนุษย์อยู่หลายชนิด งูนั้นนับเป็นอันดับที่สามจากสิบอันดับสัตว์ที่อันตรายต่อมนุษย์มากที่สุดในโลกด้วยอัตราการเสียชีวิตที่เกิดจากงูเฉลี่ยราว 138,000 รายต่อปี[1] ทั้งนี้ยังสามารถพบได้ในทุกภูมิภาคทั่วประเทศไทย ซึ่งงูหลายสายพันธุ์ไม่ได้มีพิษที่ร้ายแรงต่อชีวิตมนุษย์แต่ด้วยลักษณะของงูส่วนใหญ่ที่มีความคล้ายกันทำให้ผู้คนที่ได้พบเกิดความตระหนกหรือหวาดกลัว ทั้งนี้ยังมีผลกระทบอื่นที่อาจเกิดขึ้นเพราะงูได้ เช่น หากมีงูเข้าไปในบ้านของคนก็จะเกิดการแตกตื่นเพราะไม่รู้วิธีรับมือหรือการท่องเที่ยวในป่าที่สามารถเจองูได้ทุกเวลา หากโดนกัดแล้วไม่รู้จักสายพันธุ์ของงูชนิดนั้นก็ยากที่ปฐมพยาบาลเป็นต้น

เนื่องจากงูทุกชนิดจะมีเอกลักษณ์ที่แตกต่างกันออกไป แต่ก็ยังมีลักษณะหลายอย่างที่อาจจะแยกไม่ได้ในทันทีด้วยสายตาของมนุษย์ ทำให้การสร้างโมเดลจำแนกสายพันธุ์ของงู (Snake Classification Model) ขึ้นมาเพื่อจำแนกสายพันธุ์ของงู ด้วยอัลกอริทึมโครงข่ายประสาทเทียม (Convolutional Neural Network : CNN) โดยมีการจำแนกเอกลักษณ์เบื้องต้นของงู (Multiclass Classification) เพื่อทำการประมวลในการจัดกลุ่มสายพันธุ์ จากเหตุผลที่กล่าวข้างต้น จึงเกิดแนวคิดประยุกต์ใช้ Multi-Class

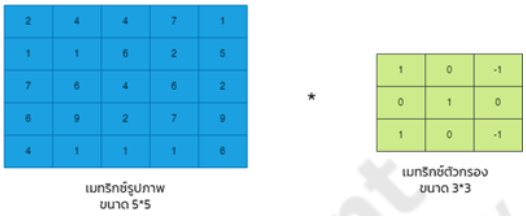
Classification ในการสร้างโมเดลจำแนกสายพันธุ์งูพบในไทยให้มีประสิทธิภาพในการจำแนกสายพันธุ์งูและเพื่อให้ข้อมูลเกี่ยวกับงูสายพันธุ์นั้นๆ

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

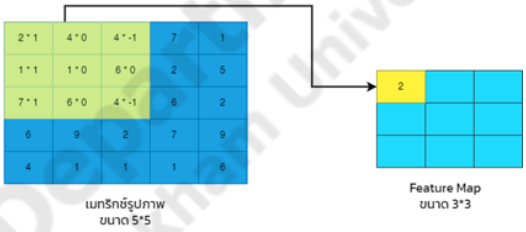
2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 Convolutional Neural Network[2](CNN) เป็นอัลกอริทึม Deep Learning ที่รับอินพุตรูปภาพ กำหนดความสำคัญให้กับวัตถุต่าง ๆ ในภาพ โดยที่ CNN จะจำลองการ มองเห็นของมนุษย์ที่มองพื้นที่แบ่งออกเป็นที่ย่อย ๆ ซึ่งจะมีการแยกคุณลักษณะของพื้นที่ย่อยนั้นเพื่อดู ว่าสิ่งที่เห็นอยู่เป็นอะไรและสามารถแยกแยะความแตกต่าง โดย CNN ประกอบด้วยเลเยอร์หลัก 3 ประเภท คือ Convolutional Layer, Pooling Layer, และ fully connected โดยแต่ละ layer จะสร้างจากโหนดที่ตรวจสอบข้อมูลอินพุตและสร้างเอาต์พุตและจำมีวิธีการคำนวณที่แตกต่างกัน

2.1.1.1 Convolutional Layer[3] เป็นเลเยอร์ที่จะดึงเอาคุณลักษณะของรูปภาพเก็บไว้และเป็นเลเยอร์แรกสุดของ CNN เสมอ ประกอบด้วยเมทริกซ์ตัวกรอง(Filters)ซึ่งกำหนดค่าได้ เช่น ต้องการ 11 หาเส้นตรง ข้อมูลใน filters จะเป็นแบบหนึ่ง ถ้าจะหาทรงกลม ข้อมูลใน filters ก็จะเป็นอีกแบบหนึ่ง การคำนวณของ Convolutional layer นั้นทำโดยการนำพิกเซลของรูปภาพมาคูณในระดับสมาชิก(Elementwise) กับ filters แล้วนำผลลัพธ์สุดท้ายจากการบวกค่าทั้งหมดมาเก็บไว้ในผังคุณลักษณะ (Feature map)



ภาพประกอบที่ 1 การคำนวณของ Convolutional Layer



ภาพประกอบที่ 2 Feature map

2.1.1.2 Pooling Layer[4] คือชั้นที่ทำหน้าที่ในการลดจำนวนของพารามิเตอร์ในกรณีที่ภาพมีขนาดใหญ่เกินไป การทำ Pooling หรือที่เรียกว่า Sub-sampling หรือ Down-sampling เป็นกระบวนการลดมิติของ Feature Map ลงแต่ยังคงข้อมูลสำคัญเอาไว้

ภาพประกอบที่ 3 ตัวอย่างการทำ Pooling

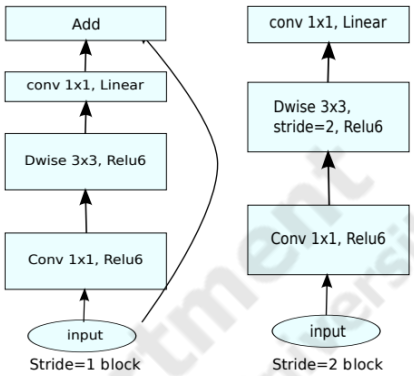
2.1.1.3 Fully Connected Layer[5] เป็นชั้นที่เชื่อมโยงระหว่าง Feature Map และ Output แบบสมบูรณ์ ดังนั้นใน layer นี้จะเป็น

Computer Science Department
Faculty of Informatics, Mahasarakham University

การรวบรวม Feature Map ทั้งหมดที่ถูกสร้างมาแปลงเป็นเวกเตอร์ด้วยการทำ Flatten จากนั้น neuron แต่ละตัวใช้การแปลงเชิงเส้นกับเวกเตอร์อินพุตด้วยการคูณกับ weight matrix แล้ว นำไปบวกกับ bias ซึ่งจะได้มาเป็นค่าของแต่ละโหนด จากนั้นจะใช้งาน Activation function เช่น softmax, sigmoid เพื่อจำแนกเอาต์พุตออกเป็นคลาสต่าง ๆ

2.1.2 Multi-Class Classification[6] การจัดหมวดหมู่แบบหลายคลาส(Multi-Class Classification) เป็นงานการจัดหมวดหมู่ Machine Learning ที่ประกอบด้วยคลาสหรือเอาต์พุตมากกว่า 2 คลาส ตัวอย่างเช่น ถ้าจำแนกว่าบุคคลคนหนึ่งว่าเป็นนักบาสเก็ตบอลหรือนักซูโม่หรือนักมวย ถ้านำมาแบ่งเป็นคลาสจะได้ 3 คลาส จากกีฬาทั้ง 3 ประเภท และในการ Classification จะได้คำตอบออกมาเพียงคลาสเดียว เช่น ถ้าส่งข้อมูลนาย A เข้าไป Classification จะได้คำตอบเป็นนักบาสเก็ตบอล

2.1.3 MobileNetV2[7] สถาปัตยกรรมเครือข่ายประสาทเทียมที่ซับซ้อนซึ่งสามารถทำงานได้ดีบนอุปกรณ์มือถือ มันขึ้นอยู่กับโครงสร้าง inverted residual structure ที่การเชื่อมต่อ residual ระหว่าง bottleneck layers เลเยอร์การขยายตัวระดับกลางใช้ lightweight depthwise convolutions เพื่อกรองคุณสมบัติเป็นแหล่งที่มาของ non-linearity โดยรวมแล้วสถาปัตยกรรมของ MobileNetV2 มีเลเยอร์ convolution เริ่มต้นอย่างสมบูรณ์พร้อม filter 32 ตัวขนาด (3x3) ตามด้วยชั้นคอขวดที่เหลือ 17 ชั้น



ภาพประกอบที่ 4 บล็อกของ bottleneck ใน MobileNetV2

ที่มา:https://paperswithcode.com/method/mobilenetv2

2.1.4 Transfer Learning[8] เป็นการนำค่าน้ำหนัก (weight) ของโมเดลที่มีการฝึกฝนไว้แล้วด้วยชุดข้อมูลจำนวนมากอย่าง imagenet ซึ่งเป็นที่นิยมในการใช้ทำ Transfer learning

2.1.5 Bacth Normolization[9] เป็นเทคนิคการทำ Data Normalization กับข้อมูลอินพุตทั้งหมดของทุก layer ใน Minibatch ทั้งหมด ก่อนจะส่งออกไปเป็น Input ของ layer ถัดไป ช่วยให้การ train นั้นรวดเร็วมากยิ่งขึ้น ใช้ Learning rate ได้สูงกว่า หรืออาจจะทำให้โมเดลทำงานได้ดีขึ้น

2.1.6 Global Average Pooling[10] เป็นการ pooling ที่ถูกออกแบบมาเพื่อแทนที่ flatten layer โดยจะคำนวณค่าเฉลี่ยของข้อมูลทั้งหมดใน feature map มาเก็บไว้ใน feature map ขนาด 1*1 แต่จะมี channel เท่ากับ feature map เดิม การรวมข้อมูลที่

Computer Science Department Faculty of Informatics, Mahasarakham University

สำคัญไว้ใน matrix ขนาด 1*1 ทำให้การนำไปคำนวณเพื่อเข้าสู่เลเยอร์ถัดไปจะช่วยให้ neural network สามารถสรุปความสำคัญของ feature map ได้ง่ายขึ้น และไม่มีพารามิเตอร์ที่จะ optimize ภายใน Global Average Pooling จึงหลีกเลี่ยงการเกิด overfitting ได้ใน layer นี้

2.1.7 Dense[11] เป็นชั้น (layer) ของเซลล์ประสาท (neurons) อย่างง่ายที่แต่ละเซลล์นั้นรับข้อมูลมาจากเซลล์ประสาททั้งหมดในชั้นก่อนหน้า การทำงานคือเซลล์ประสาทของ Dense layer จะทำการคูณเมทริกซ์กับเวกเตอร์ โดยที่เวกเตอร์แถวของเอาต์พุตจากชั้นก่อนหน้าเท่ากับเวกเตอร์คอลัมน์ของ Dense layer

2.1.8 Dropout[12] เป็นเทคนิคในการสุ่มเพื่อปิดการทำงานของโหนดชั่วคราวในแต่ละรอบของการทำ Forward Propagation และ Back-propagation ตามอัตราที่กำหนด โดยในระหว่าง train โหนดที่ถูกปิดการทำงานจะทำให้ไม่มีการอัปเดต weight และ bias ซึ่งเป็นการบังคับให้ทุกโหนดที่ไม่ถูกปิดการทำงานต้องมีการอัปเดตค่า weight และ bias ทำให้สามารถเรียนรู้ได้ดีขึ้น และการใช้ Dropout จะทำให้ทุก ๆ การวนซ้ำ neural network จะมีขนาดเล็กลงกว่า network ก่อนหน้า ทำให้เข้าใกล้การเป็นมาตรฐาน

2.1.9 Activation function ที่ใช้ในงานวิจัยนี้จะมีทั้งหมด 3 ฟังก์ชันดังต่อไปนี้

(1) ReLu function[13] ReLu นั้นจะทำให้เอาต์พุตมีค่าอยู่ระหว่าง 0 ถึง infinity ถ้าค่าที่ input เข้าไปต่ำกว่า 0 จะถูกแปลงให้เป็น 0 ทั้งหมด แต่ถ้า input มากกว่า 0 ก็จะเป็น linear function คือไม่มีการเปลี่ยนแปลง สำหรับ ReLu การไล่ระดับสีจะเป็นค่าบวกเสมอ ทำให้ลดการกระจัดกระจายและความน่าจะเป็นที่จะเกิด vanishing gradient

(2) Leaky ReLU function[14] เพื่อแก้ปัญหาของ ReLU ที่ถ้ามีค่าน้อยกว่า 0 เข้าไปหลายๆตัวจะไม่สามารถคำนวณได้ จึงมีการคิดค้นให้มีการรั่วของ output โดยการรั่วนี้จะช่วยเพิ่ม range ของ ReLU ออกไปกลายเป็น (-infinity ถึง infinity) โดยการใช้ค่า a โดยค่า a จะเป็น 0.01 หากเมื่อใดก็ตามค่า a != 0.01 เรียกว่า Randomized ReLU

(3) Softmax function[15] เป็นฟังก์ชันที่เปลี่ยนเวกเตอร์ของจำนวนจริง K ที่รวมกันเป็น 1 ค่า ไม่ว่าจะมีค่าบวกหรือลบ หรือเป็นศูนย์ ฟังก์ชันนี้จะแปลงค่านั้นให้อยู่ระหว่าง 0 ถึง 1 เพื่อให้สามารถคำนวณความน่าจะเป็นได้ โดยฟังก์ชันจะแปลงคะแนนผลลัพธ์ของชั้นสุดท้ายของโครงข่ายประสาทเทียมเป็นการแจกแจงความน่าจะเป็นแบบปกติ ทำให้เป็นเรื่องปกติที่จะให้ฟังก์ชันนี้เป็นเลเยอร์สุดท้ายของโครงข่ายประสาทเทียม และ ฟังก์ชัน softmax นั้นนิยมใช้สำหรับ multi-class classification

2.1.10 Confusion Matrix[16] คือ การวัดประสิทธิภาพการทำนายของโมเดล classification โดย Confusion Matrix จะแสดงค่าความถูกต้องด้วยการเปรียบเทียบคลาสจริง (true class) กับ คลาสคาดการณ์

Computer Science Department
Faculty of Informatics, Mahasarakham University

(predicted class) ซึ่งใน Confusion matrix จะมีค่าที่ทำนายดังต่อไปนี้

TP (True Positive) ข้อมูลที่ทำนายตรงกับข้อมูลจริงในคลาสที่กำลังพิจารณา

TN (True Negative) ข้อมูลที่ทำนายตรงกับข้อมูลจริงในคลาสที่ไม่ได้พิจารณา

FP (False Positive) ข้อมูลที่ทำนายผิดเป็นคลาสที่กำลังพิจารณา

FN (False Negative) ข้อมูลที่ทำนายผิดเป็นคลาสที่ไม่ได้พิจารณา

2.1.10.1 การคำนวณค่าการวัดผล

(1) Accuracy คือ ค่าความถูกต้อง คำนวณด้วยการนำผลรวมของเส้นทแยงหารด้วยผลรวมของทุกค่าในตาราง คำนวณโดยสมการดังนี้

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

(2) Precision คือ ค่าความแม่นยำ โดยจะแสดงความแม่นยำของคลาสที่เป็นคาดการณ์เป็นผลบวกนั้นมากน้อยเท่าไร คำนวณโดยสมการดังนี้

$$precision = \frac{TP}{TP+FP} \quad (2)$$

(3) Recall คือ ค่าระลึก คำนวณอัตราส่วนของคลาสที่คาดการณ์เป็นผลบวกที่ตรวจพบวัตถุได้อย่างถูกต้อง โดยค่าระลึกนี้จะแสดงให้เห็นว่าตัวโมเดลนั้นสามารถรับรู้ถึงคลาสที่เป็นผลบวกได้ดีแค่ไหน คำนวณโดยสมการดังนี้

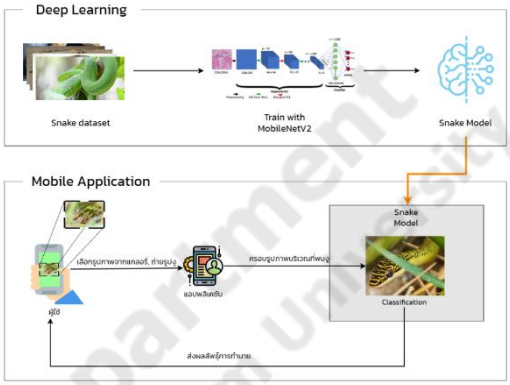
$$recall = \frac{TP}{TP+FN} \quad (3)$$

(4) F1-score คือ เป็นการนำค่าที่ได้จาก precision และ recall มาคำนวณรวมกัน คำนวณโดยสมการดังนี้

$$F1 = \frac{precision \times recall}{precision + recall} \quad (4)$$

3.วิธีดำเนินงานวิจัย

3.1 กรอบการดำเนินงาน



ภาพประกอบที่ 5 กรอบการดำเนินงาน

3.2 Dataset

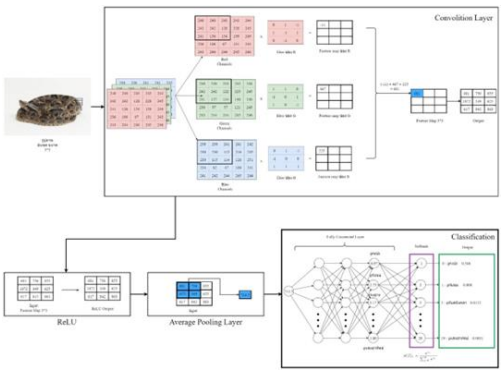
ได้รวบรวมข้อมูลและภาพของงูชนิดต่าง ๆ จากหลากหลายเว็บไซต์บนอินเทอร์เน็ตเช่น สถานเสาวภา สภากาชาดไทย, วิกิพีเดีย, zoothailand เป็นต้น โดยข้อมูลที่รวบรวมมาจะมีแหล่งที่พบ อาการหลังได้รับพิษ ลักษณะของพิษ พฤติกรรม และลักษณะทางกายภาพของงูแต่ละชนิด

โครงงานนี้ได้ใช้ชุดข้อมูลของงู 20 สายพันธุ์(ภาพประกอบที่ 3.5) ในการสร้างโมเดล ซึ่งจะมีรูปภาพทั้งหมด 4,069 รูป ประกอบด้วยชุดข้อมูลสำหรับการ Train 3,269 รูป และ Test 800 รูป รูปภาพอินพุตทั้งหมดจะถูกปรับขนาดเป็น 224*224*3

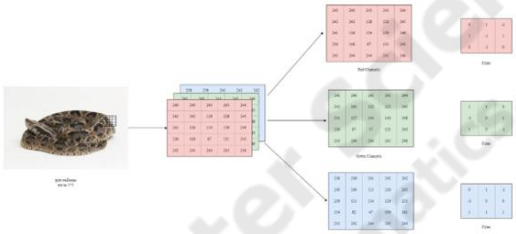
Computer Science Department Faculty of Informatics, Mahasarakham University

3.3 การทำงานของ MobileNetV2

3.3.1 วิธีการทำงานของ MobileNetV2



ภาพประกอบที่ 6 ภาพรวมวิธีการทำงานของ MobileNetV2



ภาพประกอบที่ 7 ตัวอย่างรูปภาพอินพุต

3.5 ขั้นตอนในการพัฒนา Snake Model (Deep learning)

3.5.1 Data Augmentation เทคนิค Data Augmentation เป็นเทคนิคที่ใช้เพื่อเพิ่มข้อมูลใหม่จากข้อมูลที่มีอยู่เดิม ซึ่งส่วนใหญ่จะใช้กับชุดข้อมูลที่ใช้ในการฝึกอบรม (Train) โมเดล โดยเทคนิคนี้จะช่วยแก้ไขปัญหาต่าง ๆ ของตัวโมเดล เช่น Overfitting หรือ ปัญหาการขาดแคลนชุดข้อมูล

3.5.2 เตรียมการชุดข้อมูลสำหรับการฝึกฝนโมเดล ชุดข้อมูลที่ใช้สำหรับการฝึกฝนโมเดลได้มีการเปลี่ยนขนาดภาพให้เป็น 224 * 224 *3 และใช้เทคนิค data augmentation ดังที่กล่าวไว้ในหัวข้อ 3.3.1 ชุดข้อมูลสำหรับการ train ถูกแบ่งไว้สำหรับ validation 20% ซึ่งทั้งสองส่วนมีการกำหนด shuffle = True คือการทำให้ชุดข้อมูลจะถูกสับแบบสุ่มเพื่อหลีกเลี่ยงการเกิด overfitting ในระหว่างฝึกฝนโมเดล และ seed ซึ่งใช้สำหรับใช้การเพิ่มภาพแบบสุ่มและสับเปลี่ยนลำดับของภาพ

4.ผลการทดลอง

4.1 การเปรียบเทียบประสิทธิภาพการรู้จำจากสถาปัตยกรรม CNN

ในการจำแนกสายพันธุ์งูได้ประยุกต์ใช้ร่วมกับสถาปัตยกรรม CNN โดยผู้จัดทำได้เลือกสถาปัตยกรรม InceptionV3 และ MobileNetV2 เพื่อเปรียบเทียบหาโมเดลที่มีประสิทธิภาพมากที่สุด โดยในการประเมินจะใช้ตาราง confusion matrix ในการทดลองครั้งนี้โดยใช้ชุดข้อมูลเดียวกันการประเมินประสิทธิภาพด้วย confusion matrix จะใช้ชุดข้อมูลสำหรับการ Test ของงูทั้งหมด 20 สายพันธุ์น คิดเป็น 20 เปอร์เซ็นต์ของทั้งหมด จากนั้นนำข้อมูลสำหรับประเมินประสิทธิภาพแต่ละสายพันธุ์ หาค่าความแม่นยำ(Precision), ค่าความระลึก(Recall), ด้วย Confusion Matrix เมื่อได้ผลลัพธ์แล้วนำค่าทั้งสองหาค่าความถูกต้อง (Accuracy)

Computer Science Department Faculty of Informatics, Mahasarakham University

	precision	recall	f1-score	support
0	0.74	0.70	0.72	40
1	0.68	0.57	0.62	40
2	0.44	0.40	0.42	40
3	0.80	0.90	0.85	40
4	0.79	0.38	0.51	40
5	0.79	0.75	0.77	40
6	0.38	0.47	0.42	40
7	0.47	0.88	0.61	40
8	0.59	0.55	0.57	40
9	0.69	0.62	0.66	40
10	0.92	0.85	0.88	40
11	0.39	0.68	0.49	40
12	0.56	0.23	0.32	40
13	0.83	0.88	0.85	40
14	0.64	0.57	0.61	40
15	0.95	0.88	0.91	40
16	0.70	0.78	0.74	40
17	0.76	0.85	0.80	40
18	0.79	0.78	0.78	40
19	0.81	0.53	0.64	40
accuracy			0.66	800
macro avg	0.69	0.66	0.66	800
weighted avg	0.69	0.66	0.66	800

ภาพประกอบที่ 8 ผลการประเมินประสิทธิภาพสถาปัตยกรรม MobileNetV2

คลาสที่มีค่า Precision ที่น้อยที่สุด คือ งูปี่แก้วลายแต้ม อยู่ที่ 0.38 อาจเป็นเพราะถึงงูปี่แก้วลายแต้มจะเป็นงูที่มีลวดลายบนตัวชัดเจน แต่รูปภาพส่วนใหญ่ในชุดข้อมูลลวดลายเหล่านี้นั้นค่อนข้างที่จะกลมกลืนไปกับสภาพแวดล้อม เลยอาจทำให้โมเดลไม่สามารถเรียนรู้คุณลักษณะเหล่านี้ได้ดีเท่าที่ควร

คลาสที่มีค่า Precision ที่สูงที่สุด คือ งูเขียวหางไหม้ท้องเหลือง อยู่ที่ 0.95 เพราะงูเขียวหางไหม้ท้องเหลืองเป็นงูที่มีลักษณะชัดเจนมากที่สุดตัวหนึ่ง ด้วยสีของร่างกายที่เป็นสีโทนเขียวทั้งหมดจึงน่าจะเป็นเรื่องง่ายที่โมเดลจะเรียนรู้ลักษณะเหล่านี้

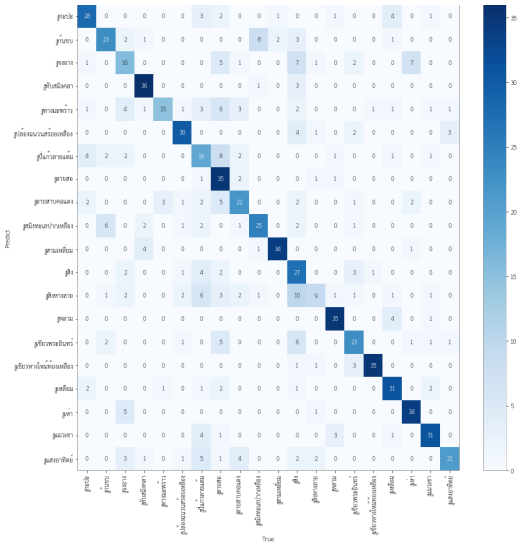
คลาสที่มีค่า Recall ที่น้อยที่สุด คือ งูสิงหางลาย อยู่ที่ 0.23 เพราะงูสิงหางลายนั้นมีลักษณะที่แทบจะไม่ต่างกับงูสิง ที่แตกต่างจะมีแค่ลายบนตัวที่มีสีดำมากกว่าเล็กน้อยและมีลายขีดสีดำอยู่ใต้ปาก ส่งผลให้โมเดลทำนายเป็นงูสิงมากกว่า และด้วยชุดข้อมูลที่รวบรวมมาไม่สามารถแสดงลักษณะเหล่านี้ออกมาได้อย่างชัดเจนทำให้โมเดลไม่สามารถเรียนรู้ได้ดีเท่าที่ควร

คลาสที่มีค่า Recall ที่สูงที่สุด คือ งูทับสมิงคลา อยู่ที่ 0.90 เนื่องจากงูทับสมิงคลานั้นมีสีของร่างกายที่ไม่ซับซ้อน โดยจะเป็นสีขาวสลับดำ ทำให้โมเดลสามารถตรวจจับคุณลักษณะของงูชนิดนี้ได้ง่าย

คลาสที่มีค่า F1-Score ที่น้อยที่สุด คือ งูสิงหางลาย อยู่ที่ 0.32 เพราะงูสิงหางลายมีค่า Precision และ Recall ที่ต่ำ เนื่องจากชุดข้อมูลไม่สามารถแสดงลักษณะเด่นของงูชนิดนี้ออกมาได้ทำให้ไม่สามารถเรียนรู้ข้อแตกต่างระหว่างงูสิงหางลายกับงูสิงหรือกับงูชนิดอื่นที่มีโทนสีคล้ายกันได้ ส่งผลให้การทำนายมีความถูกต้องที่ต่ำ

คลาสที่มีค่า F1-Score ที่สูงที่สุด คือ งูเขียวหางไหม้ท้องเหลือง อยู่ที่ 0.91 งูเขียวหางไหม้มีลักษณะที่เป็นสีเขียวทั่วทั้งร่างกายและบริเวณท้องที่เป็นสีเหลือง ทำให้โมเดลสามารถตรวจจับคุณลักษณะของงูในรูปภาพอินพุตได้ง่าย ด้วยสาเหตุที่กล่าวมาส่งผลให้โมเดลทำนายได้อย่างถูกต้อง ในกรณีที่ตัวของงูนั้นไม่กลืนไปกับสภาพแวดล้อมมากนัก

Computer Science Department
Faculty of Informatics, Mahasarakham University



ภาพประกอบที่ 9 การประเมินด้วย confusion matrix ของ MobileNetV2

	precision	recall	f1-score	support
0	0.59	0.50	0.54	40
1	0.59	0.57	0.58	40
2	0.52	0.55	0.54	40
3	0.79	0.78	0.78	40
4	0.57	0.30	0.39	40
5	0.70	0.70	0.70	40
6	0.48	0.25	0.33	40
7	0.49	0.65	0.56	40
8	0.48	0.57	0.52	40
9	0.67	0.78	0.72	40
10	0.87	1.00	0.93	40
11	0.55	0.65	0.60	40
12	0.14	0.07	0.10	40
13	0.97	0.93	0.95	40
14	0.69	0.60	0.64	40
15	0.92	0.88	0.90	40
16	0.61	0.78	0.68	40
17	0.73	0.90	0.81	40
18	0.67	0.88	0.76	40
19	0.62	0.62	0.62	40
accuracy			0.65	800
macro avg	0.63	0.65	0.63	800
weighted avg	0.63	0.65	0.63	800

ภาพประกอบที่ 10 ผลการประเมินประสิทธิภาพสถาปัตยกรรม InceptionV3

จากภาพประกอบที่ 9 คลาสที่มีจำนวนรูปภาพที่ทำนายถูกต้องน้อยที่สุดคืองูสิงหางลาย อยู่ 9 จาก 40 รูปเท่านั้น โดยคลาสที่งูสิงหางลายทำนายผิดส่วนมากคือ งูสิง, งูปี่แก้วลายแต้ม คาดว่าสาเหตุที่โมเดลไม่เรียนรู้ลักษณะของงูสิงหางลายหรือทำนายผิดอาจเป็นเพราะงูสิงกับงูสิงหางลายนั้นมีลักษณะที่เหมือนกัน จะต่างกันตรงที่งูสิงหางลายมีลายตามตัวที่ออกสีดำกว่าเล็กน้อยและมีขีดสีดำที่ใต้ปาก ส่วนที่ทำนายผิดเป็นงูปี่แก้วลายแต้มนั้นอาจเป็นเพราะรูปภาพที่ใช้ฝึกฝนโมเดลนั้นมีโทนสีหรือสภาพแวดล้อมคล้ายกัน

คลาสที่มีค่า Precision ที่น้อยที่สุด คือ งูสิงหางลาย อยู่ที่ 0.14 เพราะงูสิงหางลายนั้นมีลักษณะที่แทบจะไม่ต่างกับงูสิง ที่แตกต่างจะมีแค่ลายบนตัวที่มีสีดำมากกว่าเล็กน้อยและมีลายขีดสีดำอยู่ใต้ปาก ส่งผลให้โมเดลทำนายเป็นงูสิงมากกว่า และด้วยชุดข้อมูลที่รวบรวมมาไม่สามารถแสดงลักษณะเหล่านี้ออกมาได้อย่างชัดเจนทำให้โมเดลไม่สามารถเรียนรู้ได้ดีเท่าที่ควร

คลาสที่มีค่า Precision ที่สูงที่สุด คือ งูหลาม อยู่ที่ 0.97 งูหลามนั้นเป็นหนึ่งงูที่มีขนาดใหญ่ที่สุดและมีสีสันที่หลากหลายในชุดข้อมูล คาดว่าที่โมเดลสามารถเรียนรู้ลักษณะที่หลากหลายเหล่านี้ได้อาจเป็นเพราะจุดร่วมของงูหลามคือเป็นงูที่บนร่างกายนั้นมีลวดลายที่เป็นสีดำ สีน้ำตาล สีขาว และสีเหลือง ซึ่งอาจเป็นลักษณะที่โมเดลเลือกที่จะเรียนรู้

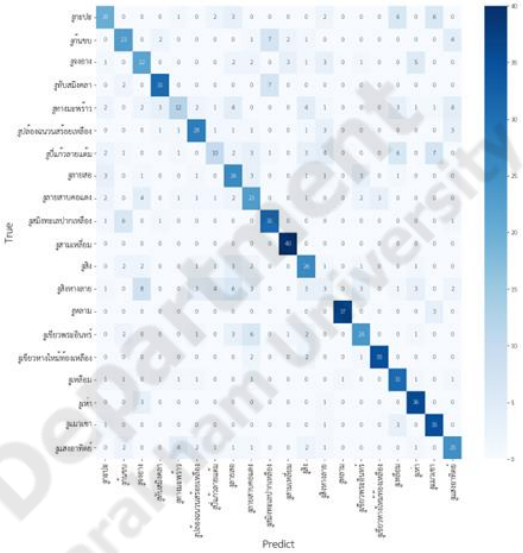
Computer Science Department
Faculty of Informatics, Mahasarakham University

คลาสที่มีค่า Recall ที่น้อยที่สุด คือ งูสิงหางลาย อยู่ที่ 0.07 เพราะงูสิงหางลายนั้นมีลักษณะที่แทบจะไม่ต่างกับงูสิง ที่แตกต่างจะมีแค่ลายบนตัวที่มีสีดำมากกว่าเล็กน้อยและมีลายขีดสีดำอยู่ใต้ปาก ส่งผลให้โมเดลทำนายเป็นงูสิงมากกว่า และด้วยชุดข้อมูลที่รวบรวมมาไม่สามารถแสดงลักษณะเหล่านี้ออกมาได้อย่างชัดเจนทำให้โมเดลไม่สามารถเรียนรู้ได้ดีเท่าที่ควร

คลาสที่มีค่า Recall ที่สูงที่สุด คือ งูสามเหลี่ยม อยู่ที่ 1.00 งูสามเหลี่ยมในด้านของลักษณะแล้วถือว่าเป็นประเภทเดียวกับงูทับสมิงคลาและงูสมิงทะเลปากเหลือง เพราะด้วยร่างกายที่เป็นสีดำสลับกับสีอีกสีหนึ่ง สำหรับงูสามเหลี่ยมคือสีดำสลับเหลือง ด้วยร่างกายที่มีสีที่ไม่ซับซ้อนนี้ทำให้โมเดลสามารถตรวจจับคุณลักษณะของงูชนิดนี้ได้ง่าย

คลาสที่มีค่า F1-Score ที่น้อยที่สุด คือ งูสิงหางลาย อยู่ที่ 0.10 เพราะงูสิงหางลายมีค่า Precision และ Recall ที่ต่ำ เนื่องจากชุดข้อมูลไม่สามารถแสดงลักษณะเด่นของงูชนิดนี้ออกมาได้ทำให้ไม่สามารถเรียนรู้ข้อแตกต่างระหว่างงูสิงหางลายกับงูสิงหรือกับงูชนิดอื่นที่มีโทนสีคล้ายกันได้ ส่งผลให้การทำนายมีความถูกต้องที่ต่ำ

คลาสที่มีค่า F1-Score ที่สูงที่สุด คือ งูหลาม อยู่ที่ 0.95 คาดว่าสาเหตุที่งูหลามเป็นงูที่มีค่า f1-score สูงที่สุดสำหรับโมเดล InceptionV3 อาจเป็นเพราะขนาดของตัวที่ใหญ่และโทนสีที่ค่อนข้างเด่นชัดทำให้สามารถทำนายได้อย่างถูกต้องมากกว่าชนิดอื่นๆ



ภาพประกอบที่ 11 การประเมินด้วย confusion matrix ของ InceptionV3

จากผลการประเมินประสิทธิภาพ พบว่าค่า Accuracy ของสถาปัตยกรรม MobileNetV2 ดังภาพประกอบที่ 4.1 ดีกว่า InceptionV3 ดังภาพประกอบที่ 4.3 และได้ทำการทดสอบกับรูปภาพพบว่าโมเดลของ MobileNetV2 ได้ผลลัพธ์ที่ถูกต้องมากกว่า ดังนั้นจึงเลือกใช้สถาปัตยกรรม MobileNetV2 ประยุกต์ร่วมกับโมเดล

จากการทดลองดังกล่าวสามารถแสดงภาพตัวอย่างที่ได้จากการทำนายผลลัพธ์ โดยยกตัวอย่างได้ดังตารางตารางที่ 4.1 ตัวอย่างผลการทำนาย

Computer Science Department
Faculty of Informatics, Mahasarakham University

ตารางที่ 4.1 ตัวอย่างผลการทำนาย

ผลทำนายจากสถาปัตยกรรม		
MobileNet V2	Inception V3	ผลเฉลย
งูแมวเซา	งูแมวเซา	งูแมวเซา
งูเหลือม	งูเหลือม	งูเหลือม
งูแสงอาทิตย์	งูแสงอาทิตย์	งูแสงอาทิตย์
งูเขียวพระอินทร์	งูลายสาบคอแดง	งูเขียวพระอินทร์
งูปี่แก้วลายแต้ม	งูปี่แก้วลายแต้ม	งูปี่แก้วลายแต้ม
งูเห่า	งูเห่า	งูเห่า
งูลายสอ	งูลายสอ	งูลายสอ
งูสามเหลี่ยม	งูสามเหลี่ยม	งูสามเหลี่ยม
งูกะปะ	งูกะปะ	งูกะปะ
งูจงอาง	งูเห่า	งูจงอาง

4.3 สรุปและวิเคราะห์ผลการทดลอง

จากการทดลองในการหาสถาปัตยกรรม CNN ที่ดีที่สุดโดยการทดลองจากชุดข้อมูลจำนวน 4069 รูป จากภาพประกอบที่ 4.1 และภาพประกอบที่ 4.3 พบว่าสถาปัตยกรรม MobileNetV2 มีค่า accuracy เป็น 66 เปอร์เซ็นต์ InceptionV3 มีค่า accuracy เป็น 65 เปอร์เซ็นต์ การเปรียบเทียบประสิทธิภาพจากการคำนวณของ Confusion Matrix และทำการทดสอบกับภาพจริงและเปรียบเทียบผลทำนาย จึงทำให้สรุปได้ว่าสถาปัตยกรรม MobileNetV2 มีประสิทธิภาพในการทำนายที่ดีกว่า และเมื่อเทียบกับขนาดโมเดลที่ได้จากการฝึกฝนแล้ว โมเดลที่ได้จากสถาปัตยกรรม MobileNetV2 มีขนาดที่เล็กมากเมื่อเทียบกับโมเดลที่ได้จากสถาปัตยกรรม InceptionV3 ที่มีขนาดใหญ่ ดังนั้นในงานนี้จึงเลือกใช้สถาปัตยกรรม MobileNetV2

5.สรุปและอภิปรายผลการทดลอง

5.1 สรุปผลและอภิปรายผล

ในการทดลองสร้างโมเดล MobileNetV2 ได้ทำการทดลองสองแบบ คือ 1.โมเดลที่เทรนแค่ครั้งเดียว 2.โมเดลที่เทรนซ้ำหลายครั้ง โดยโมเดลที่เทรนครั้งเดียวจะเป็นโมเดล MobileNetV2 แบบธรรมดาที่รับค่าน้ำหนัก(Weight) จาก imagenet แล้วนำมาปรับแต่ค่าไฮเปอร์พารามิเตอร์ต่างๆแล้วเทรนจนเสร็จหนึ่งครั้งแล้วบันทึกโมเดลเก็บไว้ ผลที่ได้คือค่า Accuracy ที่เกิน 60% แต่จะไม่มากกว่า 70% และค่า loss ที่สูงกว่า 1.2 ทั้งหมด ทางผู้จัดทำโครงงานจึงทดสอบนำโมเดลที่เทรนครั้งเดียวนี้มาปลดล็อกเลเยอร์ที่ถูกล็อกไว้และลดค่า Learning Rate ลงด้วยการหารด้วย 10 ทุกครั้งที่มีการเทรนใหม่ โดยจะทำการเทรนซ้ำไปเรื่อยๆจนกว่าค่า Accuracy และ Loss จะไม่มีการเปลี่ยนแปลง ผลที่ได้คือโมเดลมีค่า Accuracy ที่มากกว่า 70% และ Loss ที่ต่ำกว่า 1.0 รวมถึงจำนวนรูปภาพที่มีผลการทำนายถูกต้องใน Confusion Matrix ก็ดีกว่าโมเดลที่เทรนครั้งเดียว แต่ในการทดสอบทำนายผลรูปภาพจริง โมเดลที่เทรนครั้งเดียวกลับสามารถทำนายผลลัพธ์ได้ถูกต้องมากกว่า ซึ่งสาเหตุอาจมาจากความเป็นไปได้ที่โมเดลที่เทรนซ้ำหลายครั้งมีความเชื่อมโยงกับข้อมูลที่ใช้ในการเทรนมากขึ้น ซึ่ง

Computer Science Department
Faculty of Informatics, Mahasarakham University

อาจทำให้โมเดลสามารถทำนายกับข้อมูลที่ใช้ในการเทรนได้ดี แต่อาจมีประสิทธิภาพในการทำนายกับรูปภาพที่แตกต่างหรือมีความหลากหลายได้น้อยลง ทางผู้จัดทำโครงงานจึงเลือกใช้โมเดลที่เทรนแค่ครั้งเดียวสำหรับแอปพลิเคชั่นจำแนกสายพันธุ์งู

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

1. เนื่องจากงูไม่ใช่สัตว์ที่พบเห็นได้ทั่วไปและไม่เป็นที่นิยมถ่ายรูปเพื่ออัปโหลดลงบนอินเทอร์เน็ต ทำให้มีช่องทางการหารูปภาพที่จำกัด ส่งผลให้การหารูปภาพที่มีความหลากหลายและจำนวนที่เพียงพอในการเทรนโมเดลนั้นทำได้ยากและส่งผลต่อประสิทธิภาพของโมเดลนั้นไม่ดีเท่าที่ควรจะเป็น

2. ในการฝึกฝนโมเดลแต่ละครั้งต้องใช้เวลาที่นาน และในการปรับเปลี่ยนค่าไฮเปอร์พารามิเตอร์หรือการปรับแต่งเลเยอร์ต่างๆ ภายในโมเดลต้องมีการเทรนใหม่ทุกครั้ง รวมถึงปัญหาอื่นๆที่มีเวลาเป็นปัจจัยหลัก

3. ด้วยข้อจำกัดทางด้านฮาร์ดแวร์ทำให้ไม่สามารถฝึกฝนโมเดลหลายๆตัวพร้อมกันได้ เพราะจะทำให้เครื่องทำงานหนักและมีความร้อนสูงซึ่งจะทำให้เครื่องดับและต้องฝึกฝนโมเดลใหม่ตั้งแต่ต้น

4. การฝึกฝนโมเดลบน Google Colab จะสามารถรันได้ 1 โมเดลต่อหนึ่งอีเมลเท่านั้น และจำกัดสิทธิ์การใช้ GPU หรือ TPU เป็น 1 ครั้งต่อหนึ่งอีเมล ทำให้หากในการเทรนแบบธรรมดาโดยไม่ใช้ GPU หรือ TPU จะส่งผลทำให้การเทรนโมเดลนั้นกินเวลานานมากยิ่งขึ้นไปอีก

5.3 ข้อเสนอแนะ

1. การเลือกรูปภาพต้องมีความหลากหลาย อยู่ใน background ที่มีสภาพแวดล้อมหลายๆแบบถึงแม้จะเป็นสายพันธุ์เดียวกัน เพื่อให้โมเดลได้รู้จักงูรูปแบบต่างๆมากขึ้น ซึ่งจะเพิ่มประสิทธิภาพและความถูกต้องให้กับโมเดลที่ต้องที่ได้รับรูปภาพอินพุตที่หลากหลาย

2. ปริมาณของชุดข้อมูลต้องเพียงพอสำหรับการฝึกฝนและทดสอบ เพราะคุณภาพของโมเดลขึ้นอยู่กับปริมาณของชุดข้อมูล เนื่องจากทางผู้จัดทำโครงงานไม่สามารถหาชุดข้อมูลเพิ่มจากสายพันธุ์ละ 200 รูปได้ ทำให้โมเดลนั้นมีประสิทธิภาพไม่ดีเท่าที่ควร จึงขอแนะนำว่าถ้าสามารถหาควรจะมีรูปภาพอย่างน้อย 500 ภาพต่อคลาส

เอกสารอ้างอิง

1. World Health Organization, (2021), Snakebite envenoming, Retrieved 18 August 2022 from https://www.who.int/news-room/fact-sheets/detail/snakebite-envenoming
2. Sumit Saha, (2018), A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way, Retrieved 12 August 2022 from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
3. Sakib Mostafa, Fang-Xiang Wu, in Neural Engineering Techniques for

Computer Science Department Faculty of Informatics, Mahasarakham University

Autism Spectrum Disorder, (2021), Diagnosis of autism spectrum disorder with convolutional autoencoder and structural MRI images, Retrieved 12 August 2022 from https://www.sciencedirect.com/science/article/pii/B978012822822700003X?via%3Dihub

4. Natthawat Phongchit, (2018), Convolutional Neural Network (CNN) คืออะไร, Retrieved 12 August 2022 from https://medium.com/@natthawatphongchit/มาลองดูวิธีการคิดของ-cnn-กัน-e3f5d73eebaa#:~:text=Convolutional%20Neural%20Network%20(CNN)%20หรือ,อยู่เป็นอะไรกันแน่
5. Prabhu, (2018), Understanding of Convolutional Neural Network (CNN) — Deep Learning, Retrieved 19 August 2022 from https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148
6. DataRobot, (2019), Multiclass classification in machine learning, , Retrieved 10 August 2022 from https://www.datarobot.com/blog/multiclass-classification-in-machine-learning/
7. Mark Sandler, (2019), MobileNetV2: Inverted Residuals and Linear Bottlenecks, Retrieved 9 December 2022 from https://doi.org/10.48550/arXiv.1801.04381
8. Surapong Kanoktipsatharporn, (2020), Transfer Learning คืออะไร สอน Transfer Learning จากโมเดล MobileNet JSON ไป Retrain เทรนต่อ ภาพจากกล้อง Webcam ด้วย TensorFlow.js – tfjs ep.10, Retrieved 20 January 2023 from https://www.bualabs.com/archives/3493/what-is-transfer-learning-build-headless-mobilenet-model-transfer-learning-machine-learning-mobilenet-json-retrain-webcam-tensorflow-js-tfjs-ep-10/
9. Saurav Singla, (2020), Why is Batch Normalization useful in Deep Neural Networks?, Retrieved 20 January 2023 from https://towardsdatascience.com/batch-normalisation-in-deep-neural-network-ce65dd9e8dbf
10. David Landup, (2022), Don't Use Flatten() - Global Pooling for CNNs with TensorFlow and Keras, Retrieved 20 January 2023 from https://stackabuse.com/dont-use-flatten-global-pooling-for-cnns-with-tensorflow-and-keras/
11. Minsuk Heo 허민석, (2020), [TensorFlow 2 Deep Learning] Dense

Computer Science Department
Faculty of Informatics, Mahasarakham University

Layer[Video], Retrieved 20 January 2023 from https://www.youtube.com/watch?v=lor2LnEVn8M

12. Baeldung, (2022), How ReLU and Dropout Layers Work in CNNs, Retrieved 20 January 2023 from https://www.baeldung.com/cs/ml-relu-dropout-layers

13. Danqing Liu, (2017), A Practical Guide to ReLU Start using and understanding ReLU without BS or fancy equations, Retrieved 20 January 2023 from https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7

14. Mr.P L, (2019), Deep Learning แบบฉบับสามัญชน EP 2 Optimization & Activation Function เรียนกันสบายๆ สไตส์ชิลๆ, Retrieved 20 January 2023 from https://medium.com/mmp-li/deep-learning-แบบฉบับสามัญชน-ep-2-optimization-activation-function-เรียนกันสบายๆสไตส์ชิลๆ-9feb5a87e3b2

15. ECoding, (2021), Softmax Function คืออะไร, Retrieved 20 January 2023 from https://medium.com/super-ai-engineer/softmax-function-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-eae1f1bbef63

16. Deniel Johnson, (2022), Confusion Matrix in Machine Learning with EXAMPLE, Retrieved 4 September 2022 from https://www.guru99.com/confusion-matrix-machine-learning-example

Computer Science Department
Faculty of Informatics, Mahasarakham University