

การระบุสถานะความเครียดของพีชในระยะเริ่มต้น โดยใช้การเรียนรู้เชิงลึกที่พัฒนาบน FPGA



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า  
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย  
ปีการศึกษา 2563  
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Early stress identification in plant using deep learning implemented on FPGA



Mr. Krit Rojanarungruengporn

A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Engineering in Electrical Engineering

Department of Electrical Engineering

FACULTY OF ENGINEERING

Chulalongkorn University

Academic Year 2020

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การระบุสถานะความเครียดของพืชในระยะเริ่มต้นโดยใช้
	การเรียนรู้เชิงลึกที่พัฒนาบน FPGA
โดย	นายกฤต โรจน์รุ่งเรืองพร
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.สุรีย์ พุ่มรินทร์

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัยรับเป็นส่วน  
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

.....	คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)	
คณะกรรมการสอบวิทยานิพนธ์	ประธานกรรมการ
.....	
(รองศาสตราจารย์ ดร.วันเฉลิม โปรา)	
.....	อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.สุรีย์ พุ่มรินทร์)	
.....	กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.สัญญา มิตรเอม)	

กฤต โรจน์รุ่งเรืองพร : การระบุสถานะความเครียดของพืชในระยะเริ่มต้น โดยใช้การเรียนรู้เชิงลึกที่พัฒนาบน FPGA. ( Early stress identification in plant using deep learning implemented on FPGA) อ.ที่ปรึกษาหลัก : ผศ. ดร.สุริย์ พุ่มรินทร์

เพื่อการเพิ่มผลผลิตของพืชในอนาคต การระบุสถานะความเครียดของพืชถือว่าเป็นงานหนึ่งที่มีสำคัญในการเก็บคุณลักษณะของพืชเพื่อใช้ในพัฒนาสายพันธุ์ ในงานวิจัยนี้ ผู้วิจัยได้ใช้แบบจำลอง CNN และแบบจำลอง CNN+LSTM ในการตรวจหาสถานะเครียดของพืชในระยะเริ่มต้น ( Early stress detection ) เนื่องจากการขาดสารอาหาร ชุดข้อมูลที่ใช้ในงานวิจัยประกอบด้วยรูปภาพมากกว่า 40,000 ภาพของต้นข้าวฟ่างที่ถูกถ่ายในด้านหน้า ด้านข้าง และด้านบน โดยต้นข้าวฟ่างที่ถูกเลี้ยงโดยการจำกัดสารอาหารในการทดลองนี้ จะถูกแบ่งออกเป็น 3 กลุ่ม ประกอบด้วย 100/100 (100% ammonium/100% nitrate), 50/10, และ 10/10 แบบจำลอง CNN ( Convolution neural network ) แรกจะถูกสร้างอยู่บนบอร์ด PYNQ-Z1 ซึ่งเป็นบอร์ด System On Chip (SOC) ที่ส่วน FPGA สามารถโปรแกรมการใช้งานได้ด้วยภาษา Python ร่วมกับ High level synthesis tool [Vivado HSL] ซึ่งจะสามารถทำให้การประมวลผลการตรวจจับลักษณะผิดปกติ ได้เร็วกว่า CPU ARM Cortex-A9 ที่ติดตั้งอยู่บน PYNQ ได้ถึง 38 เท่า ในส่วนที่สอง แบบจำลอง CNN+LSTM จะถูกแบ่งออกเป็น 2 ส่วน คือ ส่วนสกัดคุณลักษณะ (Feature Extraction) และส่วนแยกแยะ (Classification Network) VGG16 ที่ถูกฝึกกับชุดข้อมูล ImageNet จะถูกใช้ในการสกัดคุณลักษณะ LSTM จะถูกใช้เป็นส่วนแยกแยะระบุสถานะเครียด ผลการทดลองแสดงให้เห็นว่า VGG16+LSTM สามารถระบุสถานะความเครียดของข้าวฟ่างได้ด้วยความแม่นยำมากกว่า 85% หลังจากต้นพืชถูกทำให้ขาดสารอาหารไปแล้ว 2 วัน

สาขาวิชา วิศวกรรมไฟฟ้า

ปีการศึกษา 2563

ลายมือชื่อนิติต .....

ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

## 6070419521 : MAJOR ELECTRICAL ENGINEERING

KEYWORD: convolutional neural network (CNN), FPGA, LSTM, Plant stress

Krit Rojanarungruengporn : Early stress identification in plant using deep learning implemented on FPGA. Advisor: Asst. Prof. SUREE PUMRIN

In order to ensure the availability of food in the future, plant stress identification is one of the crucial tasks used in plant phenotyping to develop better crops. In this research, we use a convolution neural network (CNN) and convolution neural network combined with LSTM to identify the early state of plant stress caused by a deficiency of nutrients. We use a treatment study dataset of sorghum (*S. bicolor*) which consists of more than 40,000 images of growing sorghum images captured in the phenotyping facility in 3 views. The experiment studies plant growing under 3 treatment conditions: 100/100 (100% ammonium/100% nitrate), 50/10, and 10/10. The first CNN will be used on PYNQ-Z1 board which is System On Chip (SOC) board. The FPGA of this board can be programmed with Python language together with High level synthesis tool [Vivado HSL] to accelerate processing time upto 38 times compared to its CPU part, ARM Cortex-A9. The second network is divided into two parts: the features extraction and classification network. VGG16 with pre-trained weights from the ImageNet dataset is used as the feature extractor. LSTM cell with multi-layer perceptron (MLP) is used to classify extracted features to determine the stress of the plants after subjected to the stressor. The result revealed that the network can detect the stress at the accuracy of more than 85% at 2 days after plants subjected to the stressor treatment.

Field of Study: Electrical Engineering

Student's Signature .....

Academic Year: 2020

Advisor's Signature .....

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีด้วยความช่วยเหลือจากผู้ช่วยศาสตราจารย์ดร. สุริย์ พุ่มรินทร์อาจารย์ที่ปรึกษา ที่ให้คำปรึกษาแนวทางการดำเนินงานวิจัยและแนวทางในการคิดทั้งในเรื่องวิชาการ การศึกษาต่อและการดำเนินชีวิต

ขอขอบคุณอาจารย์ พี่ และเพื่อนในห้องปฏิบัติการวิจัยการออกแบบและประยุกต์วงจรรวมที่ให้การสนับสนุน รับฟังปัญหาและช่วยให้คำปรึกษาทั้งในเรื่องวิชาการและการดำเนินชีวิต

สุดท้ายนี้ข้าพเจ้าขอขอบคุณครอบครัวของข้าพเจ้า คุณพ่อ คุณแม่ ภรรยา ที่เป็นกำลังใจ ดูแลเอาใจใส่และให้การสนับสนุนในทุกการตัดสินใจของข้าพเจ้าตลอดมา

กฤต โรจน์รุ่งเรืองพร



# สารบัญ

## หน้า

.....	ก
บทคัดย่อภาษาไทย .....	ก
.....	ง
บทคัดย่อภาษาอังกฤษ .....	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญตาราง .....	ฉ
สารบัญรูปภาพ .....	ญ
บทที่ 1 บทนำ .....	12
1.1 ความเป็นมา .....	12
1.2 การทบทวนบทประพันธ์ .....	13
1.2.1 การระบุรู้และการตัดเฉพาะส่วนที่สนใจของภาพ (Identification and Segmentation).....	13
1.2.2 การจำแนกกลุ่ม (Classification).....	15
1.2.3 การเก็บลักษณะของพืช (Plant Phenotyping) .....	19
1.3 วัตถุประสงค์.....	21
1.4 ขอบเขต .....	21
1.5 ประโยชน์ .....	21
บทที่ 2 ทฤษฎีและหลักการ .....	21
2.1 การเก็บลักษณะพืช (Plant Phenotyping) .....	21
2.1.1 ประเภทของการเก็บลักษณะพืชที่ใช้อยู่ในปัจจุบัน .....	22
2.1.2 กระบวนการตัดภาพเฉพาะส่วนใบ (Leaf Segmentation).....	23

2.2 การระบุสถานะความเครียดของพืช (Stress Identification in Plant) .....	26
2.2.1 Deep learning (DL).....	27
2.2.2 Convolution Neural Network (CNN) .....	27
2.2.3 LSTM (Long Short-Term Memory) Network .....	29
2.3 Deep learning บน FPGA.....	29
2.3.1 High-Level Synthesis (HSL) .....	30
2.3.2 Vivado HLS .....	31
2.3.3 System-on-Chip กับ Zynq.....	31
2.3.4 PYNQ Platform.....	33
2.3.5 CNN บน PYNQ.....	34
บทที่ 3 การทดลอง .....	37
3.1 ชุดข้อมูล [Dataset] ที่ใช้ในการทดลอง.....	37
3.2 ขั้นตอนการทดลอง .....	38
3.3 อุปกรณ์ในการทดลอง.....	48
3.4 โปรแกรมในการทดลอง .....	49
บทที่ 4 ผลการทดลอง.....	50
บทที่ 5 อภิปรายผลการทดลอง .....	57
บทที่ 6 สรุปผลการทดลอง .....	60
บรรณานุกรม .....	61
ประวัติผู้เขียน .....	64
ภาคผนวก .....	65
โปรแกรมเตรียมข้อมูลสำหรับฝึก .....	66
เตรียม Data script สำหรับฝึกแบบจำลองแบบต่างๆ.....	66
Crop และ Resize รูปภาพ .....	70



เตรียม Data Script สุดท้ายสำหรับชุดข้อมูลรูปภาพสำหรับฝึกแบบจำลอง CNN.....	74
เตรียม Time series Data Script สุดท้ายสำหรับชุดข้อมูลรูปภาพสำหรับฝึกแบบจำลอง CNN+LSTM.....	75
เตรียมชุดข้อมูล HDF5 สำหรับการฝึกแบบจำลอง.....	79
ฝึกแบบจำลอง.....	88
ฝึกแบบจำลอง CNN สำหรับ FPGA.....	88
ฝึกแบบจำลอง VGG16+LSTM.....	93
ฝึกแบบจำลอง CNN สำหรับเปรียบเทียบ (Baseline Model).....	103
วิธีนำแบบจำลอง CNN ไปใช้บน FPGA [43].....	108



## สารบัญตาราง

หน้า

ตารางที่ 1 งานวิจัยที่มีการนำ Machine learning algorithm มาใช้ในการเก็บลักษณะพืชขณะเกิด ภาวะเครียด[17] .....	16
ตารางที่ 2 งานวิจัยที่มีการนำ Deep learning algorithm มาใช้ในการเก็บลักษณะพืชขณะเกิดภาวะ เครียด[24] .....	18
ตารางที่ 3 รายละเอียดโครงสร้างแบบจำลองตัวอย่างที่ถูกนำเสนอในงานวิจัย [43].....	39
ตารางที่ 4 การใช้ทรัพยากร FPGA บนบอร์ด PYNQ ของโมเดล LeNet-5 และ CIFAR-10 [43] .	39
ตารางที่ 5 การปรับโครงสร้างแบบจำลองและการใช้ทรัพยากรของ FPGA.....	41
ตารางที่ 6 เวลาที่ใช้ในการฝึกแบบจำลองกับชุดข้อมูลทั้ง 18 ชุด.....	50
ตารางที่ 7 ค่าความแม่นยำในการทำนาย (Accuracy) และค่าเฉลี่ยเวลาในการทำนาย Test Data ด้วย อุปกรณ์ต่างๆ .....	51
ตารางที่ 8 ค่าความแม่นยำในการระบุสถานะเครียดบนข้อมูลทดสอบของแบบจำลองต่างๆ.....	54
ตารางที่ 9 เวลาในการฝึกแบบจำลอง และเวลาเฉลี่ยในการทดสอบของแบบจำลอง VGG16+LSTM และแบบจำลอง Baseline.....	56

## สารบัญรูปภาพ

หน้า

รูปที่ 1 ภาพตัวอย่างของปัญหาการระบุใบไม้ที่ประกอบด้วยใบไม้เป็นองค์ประกอบหลัก .....	13
รูปที่ 2 ภาพตัวอย่างการเก็บลักษณะของพืชที่มีการระบุและแยกใบแต่ละใบ[3] .....	14
รูปที่ 3 ลักษณะการนำ Machine Learning Algorithm มาใช้ในการแยกลักษณะของพืชแบบต่างๆ [17].....	17
รูปที่ 4 ตัวอย่างภาพถ่าย Hyperspectral ของใบไม้ (a) แถวของรูปภาพของช่วงความยาวคลื่นต่างๆ ที่รวมกันเป็น 3D Hypercube (b) กราฟแสดงการสะท้อนของแสงในช่วงความยาวต่างๆ ของ Pixel หนึ่งๆ บนภาพถ่าย Hyperspectrum [30] .....	19
รูปที่ 5 แสดงกล้องที่ใช้ในการบันทึกภาพในช่วงความยาวคลื่นแสงช่วงต่างๆ[29] .....	19
รูปที่ 6 ภาพที่ถูกใช้ในการฝึกของกระบวนการ IPK ในการแยกพื้นหลังออกจากต้นพืช เรียงลำดับ จาก ซ้าย กลาง ขวา คือ ชุดข้อมูล A1 A2 และ A3[41].....	24
รูปที่ 7 ผลจากการตัดภาพเฉพาะส่วนที่สนใจ (บนซ้าย) EDM(บนขวา) ภาพ EDM ที่วงกลม ค่าสูงสุดแต่ละจุดซึ่งถือว่าเป็นจุดศูนย์กลางของใบ(ล่างซ้าย) Skeleton image(ล่างขวา) [41] .....	26
รูปที่ 8 ReLU Activation Function (ซ้าย) กราฟแสดงการเปรียบเทียบระหว่างการใช้ ReLU [เส้นทึบ] และ tanh [เส้นปะ] พบว่า ReLU ลู่เข้าเร็วกว่าถึง 6 เท่า (ขวา) [42] .....	28
รูปที่ 9 ระดับของการออกแบบ FPGA.....	30
รูปที่ 10 ภาพรวมของการออกแบบด้วย Vivado HLS .....	31
รูปที่ 11 แสดง โครงสร้างของ Zynq.....	32
รูปที่ 12 Zynq Processing System.....	33
รูปที่ 13 ภาพรวมการทำงานของ CNN บน PYNQ.....	34
รูปที่ 14 IP Blocks สำหรับ Convolution layer [43].....	34
รูปที่ 15 IP Block สำหรับ Pooling layer [43].....	35
รูปที่ 16 IP Block สำหรับ Fully connected layer [43].....	35

รูปที่ 17 IP Block สำหรับลด Timing .....36

รูปที่ 18 ภาพถ่ายของข้าวฟ่างที่ใช้ในงานวิจัย [40] ในมุมมองด้านหน้า ด้านข้าง และด้านบน ตามลำดับ  
.....37

รูปที่ 19 ภาพต้นข้าวฟ่าง ในการทดลองวันที่ 2 10 และ 18 ที่ความละเอียด 32 64 และ 128 Pixel...40

รูปที่ 20 โครงสร้าง CNN ที่ออกแบบด้วย Vivado IP แต่ละชั้น.....43

รูปที่ 21 Overlay ที่แก้ไขและเพิ่ม CNN IP เข้าไป.....43

รูปที่ 22 ขั้นตอนในการเติมชุดข้อมูลสำหรับการฝึกแบบจำลอง CNN+LSTM .....45

รูปที่ 23 ภาพรวมขั้นตอนการฝึกแบบจำลอง CNN+ LSTM.....47

รูปที่ 24 ค่า Accuracy/Loss ของการฝึกแบบจำลองตัวเลือก 2 เพื่อใช้กับ FPGA เทียบ Epoch โดย  
เส้นสีแดง Train\_Loss น้ำเงิน คือ Val\_Loss ม่วง คือ Train\_Acc และ เทาเข้ม คือ Val\_Acc .....52

รูปที่ 25 ค่า Accuracy/Loss ของการฝึกแบบจำลอง VGG16+LSTM เทียบ Epoch โดยเส้นสีแดง  
Train\_Loss น้ำเงิน คือ Val\_Loss ม่วง คือ Train\_Acc และ เทาเข้ม คือ Val\_Acc .....53

รูปที่ 26 ค่าความแม่นยำในการทำนายสถานะเครียดของข้าวฟ่างเทียบกับวันที่ในการทดลอง .....54

## บทที่ 1 บทนำ

### 1.1 ความเป็นมา

หากพิจารณาจากข้อมูลการเพิ่มขึ้นของประชากรตั้งแต่อดีตถึงปัจจุบัน จะสามารถประมาณได้ว่า ในปี 2050 จำนวนประชากรบนโลกจะมีมากกว่า 9 พันล้านคน ซึ่งการผลิตอาหารสำหรับจำนวนประชากรขนาดนี้ถือว่าเป็นสิ่งที่ไม่สามารถทำได้ในปัจจุบัน [1] และเป็นเรื่องที่ทำหายนามากหากจะผลิตได้อย่างเพียงพอในอนาคต หากพิจารณาในแง่ผลผลิตทางการเกษตรเพื่อการบริโภค จะพบว่าค่าเฉลี่ยความสามารถในการผลิตนั้นเพิ่มขึ้นเพียงปีละ 1.3% เท่านั้น ซึ่งเป็นไปไม่ได้เลยที่จะสามารถผลิตได้เพียงพอต่อความต้องการในอนาคต จากข้อเท็จจริงนี้เอง การเพิ่มผลผลิตทางการเกษตรเพื่อการบริโภคจึงเป็นปัญหาที่สำคัญมากอย่างหนึ่งที่ต้องเตรียมความพร้อมก่อนที่ ประชากรทั่วโลกจะต้องเจอกับปัญหาการขาดแคลนอาหารในอนาคต จากงานวิจัยในปัจจุบันพบว่า การหาความสัมพันธ์ระหว่างลักษณะส่วนผสมของยีน (Genotype) และลักษณะที่ปรากฏออกมาของพืช (Phenotype) ทั้งภายนอก เช่น สีของดอก ขนาดของผล และ ลักษณะภายใน เช่น หมู่เลือด ระดับน้ำตาลในผล จะทำให้ผู้ผลิตสามารถที่จะคัดเลือกต้นพืชที่สามารถให้ผลผลิตได้มาก มีความต้านทานโรค หรือสภาพอากาศ ได้รวดเร็วและมีประสิทธิภาพมากขึ้น ซึ่งหากใช้ร่วมกับเทคโนโลยีที่สามารถจัดเรียงลำดับนิวคลีโอไทด์ภายในโครงสร้าง DNA อย่างเช่น NGS (Next Generation DNA Sequencing) ซึ่งมีความก้าวหน้าอย่างมากในปัจจุบัน จะทำให้ผู้ผลิตสามารถที่จะพัฒนาสายพันธุ์พืชที่ให้ผลผลิตและคุณภาพมากขึ้นผ่านการผสมกันในระดับโมเลกุล (Molecular Breeding) ได้ [2] นอกจากนี้การระบุได้ถึงต้นพืชที่มีความต้านทานต่อโรคและสภาพแวดล้อมต่างๆ จะทำให้ผู้พัฒนาสายพันธุ์ นำยีนที่มีความต้านทานนี้ไปพัฒนาสายพันธุ์พืชที่มีความต้านทานมากขึ้นได้ ซึ่งจะทำให้ลดความสูญเสียของผลผลิตที่เกิดจากโรคหรือสภาพแวดล้อมต่างๆ ได้

อย่างไรก็ตาม ในปัจจุบัน โดยส่วนมากการเก็บลักษณะที่ปรากฏของพืชทั้งภายนอกและภายใน (Plant phenotyping) ยังเป็นการเก็บข้อมูลที่ใช้แรงงานคนเป็นหลัก ไม่ว่าจะเป็น การเก็บลักษณะใบ จำนวนใบที่เกิดขึ้น การติดตามการเจริญเติบโต ระยะออกดอกหรือออกผล ความเข้มของสีใบ มากไปกว่านั้น บางลักษณะของพืชจำเป็นที่จะต้องใช้ผู้เชี่ยวชาญ หรือ แม้กระทั่งจำเป็นต้องส่งตัวอย่างเข้าห้องทดลองเพื่อเก็บข้อมูล ซึ่งในการเก็บข้อมูลในแต่ละรอบนั้น จำเป็นต้องใช้ตัวอย่างทดลองเป็นจำนวนมาก และพืชยังเป็นสิ่งมีชีวิตที่เจริญเติบโตอย่างต่อเนื่อง ในการเก็บข้อมูลแต่ละครั้ง จึงจำเป็นต้องใช้ช่วงเวลาก่อนข้างนาน จึงทำให้การเก็บข้อมูลด้วยแรงงานคนนั้น มีความยากลำบากและมีค่าใช้จ่ายสูง แม้ว่าในภาคของการทำวิจัยจะมีการพัฒนาระบบการเก็บลักษณะพันธุ์พืชโดยใช้เทคโนโลยีทางด้านการประมวลผลภาพ

(Computer vision) และหุ่นยนต์แล้ว แต่ก็ยังเป็นระบบที่มีค่าใช้จ่ายสูง และยังต้องการการพัฒนาอีกหลากหลายด้าน ด้วยเหตุนี้ในปัจจุบันการเก็บลักษณะของพืชจึงเป็นข้อจำกัดในการแยกลักษณะทางพันธุกรรมของพืช ที่เกี่ยวกับการเจริญเติบโต ความต้านทานสภาวะความเครียด (Stress) หรือ ผลผลิต

ด้วยความก้าวหน้าของเทคโนโลยีทางการประมวลผลภาพ Computer vision และอุปกรณ์การถ่ายภาพที่มีราคาต่ำลง ทำให้การพัฒนาการเก็บข้อมูลลักษณะของพืช (Plant Phenotyping Platform) ที่สามารถทำงานได้ดีโดยไม่จำเป็นต้องใช้เงินทุนสูงมากเป็นไปได้มากขึ้น ในวิทยานิพนธ์เล่มนี้จะนำเสนอระบบการเก็บข้อมูลลักษณะของพืชพร้อมทั้งระบุสภาวะเครียดของพืชที่เกิดขึ้นจากความผิดปกติของสภาพแวดล้อม โดยใช้การเรียนรู้เชิงลึก (Deep learning) บนอุปกรณ์ลอจิกแบบควบคุมได้ (FPGA)

## 1.2 การทบทวนบทประพันธ์

### 1.2.1 การระบุรู้และการตัดเฉพาะส่วนที่สนใจของภาพ (Identification and Segmentation)

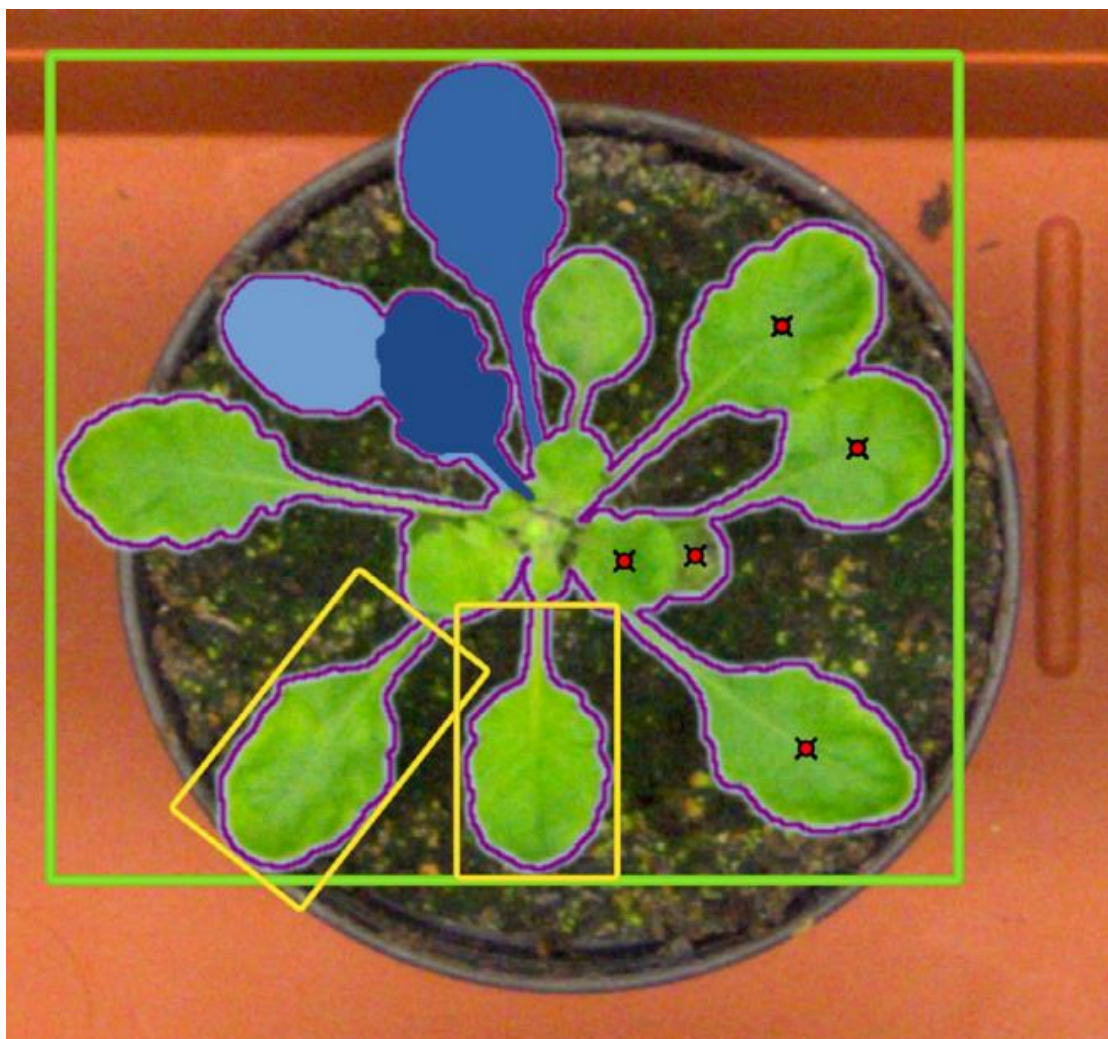
ในการเก็บลักษณะของพืชโดยใช้การประมวลผลภาพนั้น จำเป็นต้องประมาณค่าพารามิเตอร์หลากหลายอย่าง เช่น การระบุรู้ (Detection) การแบ่งภาพเฉพาะส่วน (Segmentation) การติดตามส่วนๆหนึ่งของภาพตลอดช่วงเวลาหนึ่งๆ (Tracking across time) โดยส่วนมากจะเป็นการติดตามพืชทั้งต้น หรือ ติดตามใบแต่ละใบของต้นพืช [3]

หากพิจารณาโดยคร่าวๆแล้ว อาจจะมองได้ว่าการเก็บลักษณะพืชนั้นจะเหมือนกับปัญหาการระบุใบไม้ หรือ การตัดใบไม้ออกจากภาพ ที่ภาพประกอบด้วยใบไม้เพียงใบเดียวเป็นส่วนประกอบหลัก ดังรูปที่ 1



รูปที่ 1 ภาพตัวอย่างของปัญหาการระบุใบไม้ที่ประกอบด้วยใบไม้เป็นองค์ประกอบหลัก ซึ่งมีการทำการวิจัยโดยวิธีการการเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning)[4, 5]

วิเคราะห์จากรูปร่าง (Shape-based)[6, 7] แต่ทว่าในความเป็นจริงแล้วการเก็บลักษณะพืชนั้นมีความยากมากกว่ามาก เพราะ ต้องทำการระบุและตัดภาพเฉพาะใบไม้บนต้นพืช ที่ไม่ได้มีแค่หนึ่งใบหรือที่เรียกว่าปัญหา Multi-instance segmentation ดังรูปที่ 2



รูปที่ 2 ภาพตัวอย่างการเก็บลักษณะของพืชที่มีการระบุและแยกใบแต่ละใบ[3]

ซึ่งไม่เพียงแต่ต้องแยกใบไม้พร้อมกันหลายใบ แต่ใบไม้แต่ละใบยังแตกต่างกันออกไปด้วย รวมถึงรูปร่างที่ซับซ้อน ท่าทางของใบต่อกิ่ง สีสัน ตำแหน่ง และลักษณะต่างๆ โดยเฉพาะในการถ่ายจากมุมมองบน (Top view) ยังมีปัญหาเรื่องของการทับซ้อนของใบอยู่ด้วย หากพิจารณาถึงงานวิจัยที่มีปัจจุบันจะพบว่าวิธีการการแยกแบบมีผู้สอนและไม่มีผู้สอน (Supervised and Unsupervised Plant Segmentation) ถูกใช้อย่างแพร่หลายในการแก้ปัญหานี้ ในงานวิจัย [8] ได้ทำการเปรียบเทียบเทคนิค Feature selection ต่างๆ ประกอบด้วย Correlation-based Feature Selector (CFS), Chi-Squared,

Information Gain และ Gain Ratio ซึ่งทำงานร่วมกับเทคนิค Classification ต่อไปนี้ Naïve Bayes, K-Nearest Neighbors, C4.5, Random forest, ANN, PLS-DA, SVM และ Logistic ในการคัดลักษณะของต้นอ่อน (Seedling) โดยผลของการวิจัยพบว่าการใช้ Feature selection นั้น จะมีโอกาสเพิ่มความแม่นยำในการแยกได้สูงสุดถึง 7.4% และคู่ของกระบวนการที่ใช้ให้ความแม่นยำในการแยกดีที่สุดถึง 80% ซึ่งมาจากการใช้ Chi-Squared ร่วมกับ Logistic ในงานวิจัย [9] ได้ใช้ Adaptive Thresholding algorithm ในการแยกส่วนของใบไม้ออกจากภาพพื้นหลัง ซึ่งสรุปว่า สามารถใช้กับข้อมูลประเภท Real-time video ได้อย่างมีประสิทธิภาพ นอกจากนี้ยังมีการนำเทคโนโลยีทาง Stereo imaging ในการแก้ปัญหาใบที่ทับซ้อนและสร้าง 3D Model ของต้นพืชดังที่พบใน [10-13] ใน [14] ผู้เขียนแยกและติดตามใบของต้น Arabidopsis บนฟลูออเรสเซนซ์วิดีโอ โดยใช้เทคนิค Chamfer-derived energy functional ร่วมกับ แบบของใบไม้ (Leaf Template) ที่เตรียมมาก่อน [15] เสนอเทคนิคการตัดส่วนของภาพ หลากหลายวัตถุ ที่มีการทับซ้อนกัน โดยการเรียนรู้เพื่อแยกวัตถุเต็ม และส่วนของวัตถุและประมาณส่วนที่ขาดขึ้นมา และมีการใช้ Graphical methods ในการแยกวัตถุที่ซ้อนทับกันใน [16]

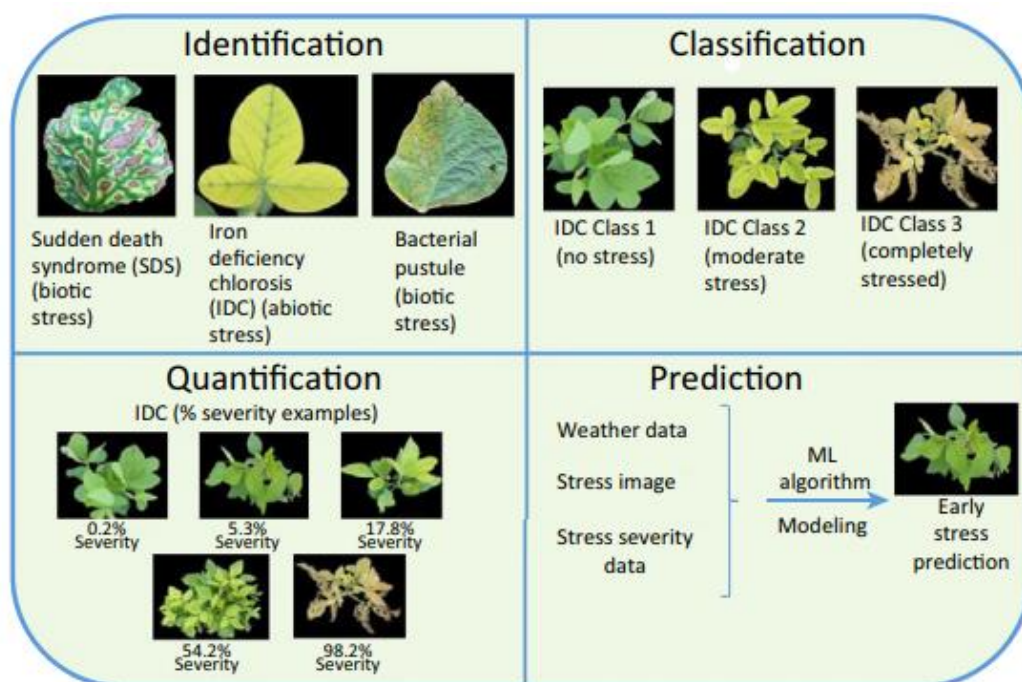
#### 1.2.2 การจำแนกกลุ่ม (Classification)

ในการเก็บลักษณะของพืชและระบุเมื่ออยู่ในสภาวะเจริญนั้นมีการนำเทคโนโลยีทางด้าน Machine learning มาใช้อย่างหลากหลายรูปแบบซึ่งสามารถเห็นได้จากรูปที่ 3 พบได้ในงานวิจัยต่างๆในตารางที่ 1



ตารางที่ 1 งานวิจัยที่มีการนำ Machine learning algorithm มาใช้ในการเก็บลักษณะพืชขณะเกิดภาวะเครียด[17]

ลักษณะการใช้งาน ML Algorithm	ประเภท ML Algorithm	เซ็นเซอร์	ชนิดของพืช	ลักษณะที่เก็บ	ประเภทของ สภาวะความเครียด	อ้างอิง
การระบุ (Identification)	SVM	Hyperspectral	มะเขือเทศ	ลักษณะการขาดน้ำ	สภาวะแห้งแล้ง	[18]
การระบุ (Identification)	SVM, linear kernel, quadratic kernel (QP), radial basis function (RBF), multilayer perceptron (MLP), and polynomial kernel	รูป RGB	มะเขือเทศ	ใบเหลือง	สภาวะติดเชื้อ	[19]
การระบุ (Identification) และการแยก (Classification)	Naïve Bayes (NB), simple logistic (SL), LibSVM (SVM), LibLINEAR (LINE), MLP (BNN), functional trees (FT), random forests (RF), classifier for generating a grafted C4.5 decision tree (J48)	Hyperspectral	Brassica napus L.	Alternaria alternata, Alternaria brassicae, Alternaria brassicicola, and Alternaria dauci	สภาวะติดเชื้อ	[20]
การแยก (Classification)	SVM	รูป RGB	Arabidopsis (Arabidopsis thaliana L.)	แบคทีเรีย Salmonella	สภาวะติดเชื้อ	[21]
การแยก (Classification)	Bayesian classifier	รูป RGB	Arabidopsis	แบคทีเรีย Salmonella	สภาวะติดเชื้อ	[22]
การระบุ (Identification) และการแยก (Classification)	Linear discriminant analysis (LDA) and K-means	รูป RGB	Trifolium subterraneum L.	Ozone	สภาวะมลพิษ	[23]



รูปที่ 3 ลักษณะการนำ Machine Learning Algorithm มาใช้ในการแยกลักษณะของพืชแบบต่างๆ[17]

นอกจากนี้ยังมีการนำการเรียนรู้เชิงลึก (Deep Learning) มาใช้ในการระบุเมื่อพืชเกิดสภาวะเครียด ซึ่งสามารถพบได้ในงานวิจัยในตารางที่ 2

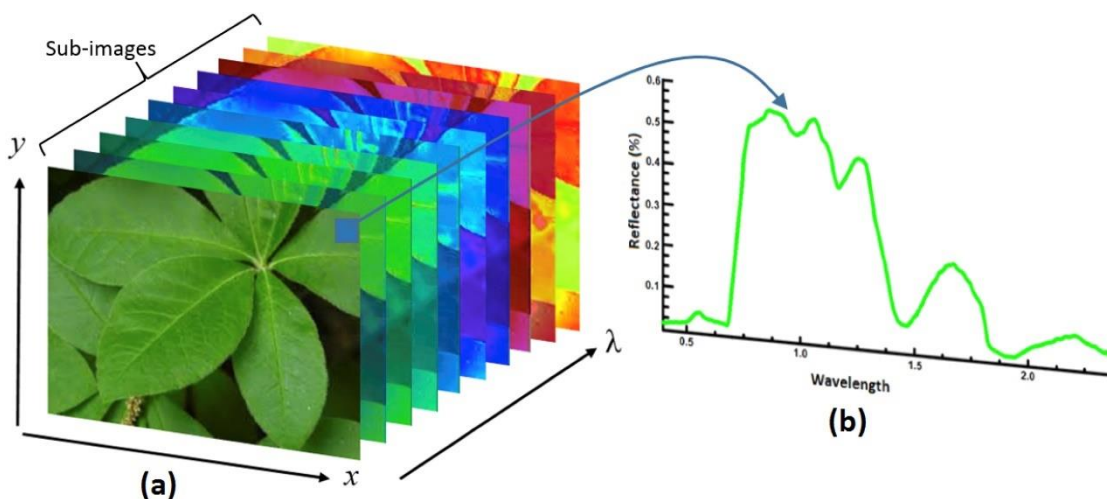
ตารางที่ 2 งานวิจัยที่มีการนำ Deep learning algorithm มาใช้ในการเก็บลักษณะพืชขณะเกิดภาวะเครียด[24]

ลักษณะการใช้งาน DL Algorithm	ประเภท DL Algorithm	ชนิดของพืช	ลักษณะที่เก็บ	ประเภทของสภาวะ ความเครียด	อ้างอิง
Prediction	DNN	มะเขือเทศ	ลักษณะการขาดน้ำ	ความเครียดที่เกิดจาก สิ่งแวดล้อม	[25]
การแยก (Classification)	AlexNet, GoogLeNet	มะเขือเทศ	Tomato yellow leaf curl virus, tomato mosaic virus, target spot, spider mites, Septoria spot, leaf mold, late blight, early blight, bacterial spot	สภาวะติดเชื้อ	[26]
การระบุ (Identification)	LeNet architecture	กล้วยหอม	Early scorch, cottony mold, ashen mold, late scorch, tiny whiteness	สภาวะติดเชื้อ	[27]
การระบุ (Identification)	CNN	แตงกวา	Melon yellow spot virus, zucchini yellow mosaic virus, cucurbit chlorotic yellows virus, cucumber mosaic virus, papaya ring spot virus,	สภาวะติดเชื้อ	[28]

ซึ่งจากงานวิจัยต่าง ๆ นี้พบว่า การนำกระบวนการการเรียนรู้เชิงลึกมาใช้ในการเก็บลักษณะพืช นั้น มีประสิทธิภาพมากในการตรวจพบสภาวะความเครียดที่เกิดขึ้น โดยเฉพาะเมื่อใช้ร่วมกับเทคโนโลยี Hyperspectral Imaging ซึ่งทำให้สามารถระบุถึงสภาวะเครียดที่จะเกิดขึ้นก่อนที่ตามนุษย์จะมองเห็น [24]

หากพิจารณาถึงการใช้งานร่วมกันของกระบวนการเรียนรู้เชิงลึกและข้อมูลที่ได้จาก Hyperspectral Camera แล้วจะพบว่า การประมวลผลข้อมูลภาพ Hyperspectral นั้น จำเป็นอย่างมาก

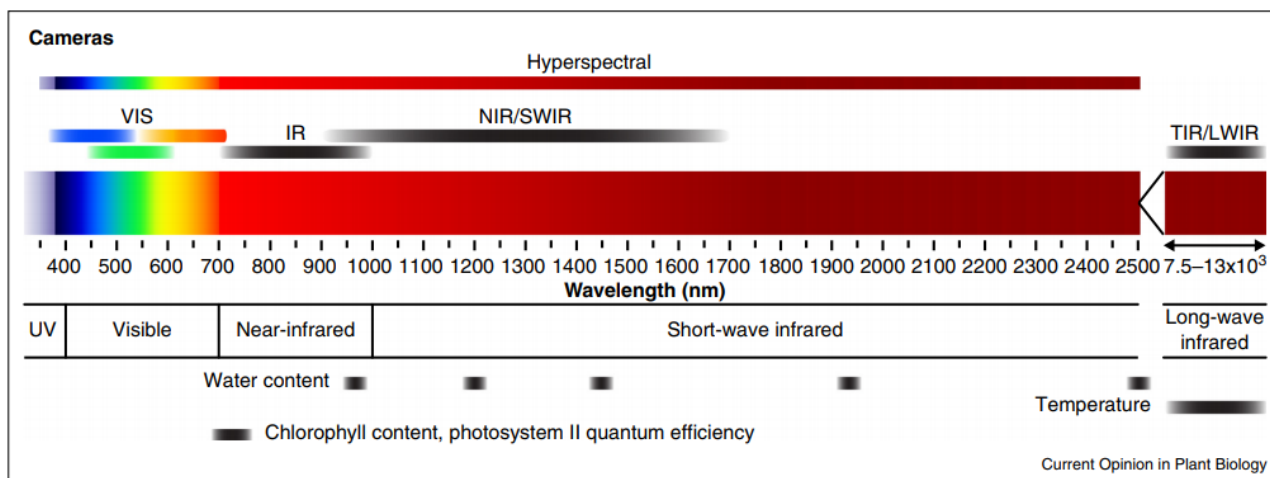
ที่ต้องการอุปกรณ์ที่มีพลังการคำนวณสูง[29] เพราะข้อมูลภาพ Hyperspectral จะถูกเก็บอยู่ในรูปของ 3D Hypercube ดังรูปที่ 4



รูปที่ 4 ตัวอย่างภาพถ่าย Hyperspectral ของใบไม้ (a) แถวของรูปภาพของช่วงความยาวคลื่นต่างๆ ที่รวมกันเป็น 3D Hypercube (b) กราฟแสดงการสะท้อนของแสงในช่วงความยาวต่างๆ ของ Pixel หนึ่งๆ บนภาพถ่าย Hyperspectrum [30]

ประกอบกับการใช้งานการตรวจสอบภาวะความเครียดของพืชที่เกิดขึ้นในระยะเริ่มต้นนั้น จำเป็นที่จะต้องมีการระบุแบบทันที (Real time) และในหลายๆการใช้งานระบบการเก็บลักษณะและตรวจสอบภาวะความเครียดนี้ ถูกใช้งานร่วมกับอากาศยานไร้คนขับ จึงทำให้การวิเคราะห์ภาพ Hyperspectral ต้องการอุปกรณ์ที่มีพลังการคำนวณสูงและกินพลังงานต่ำ [31]

### 1.2.3 การเก็บลักษณะของพืช (Plant Phenotyping)



รูปที่ 5 แสดงกล้องที่ใช้ในการบันทึกภาพในช่วงความยาวคลื่นแสงช่วงต่างๆ[29]

งานวิจัยหลายงานระบุว่า ภาพของต้นพืชที่ถูกถ่ายในช่วง VIS (400-700nm) สามารถที่จะนำมาใช้ประมาณค่าปริมาตรของต้นพืช และพื้นที่ของใบได้ ซึ่งสามารถนำค่าเหล่านี้มาคำนวณหาชีวมวล (Biomass) ของต้นพืชในส่วนที่อยู่เหนือดินได้ ดังพบใน [32-34] และวิดีโอที่ถูกถ่ายในช่วง VIS ยังสามารถใช้ในการคำนวณหาอัตราการเจริญเติบโต (Plant Growth Rate) ได้ ดังพบใน [35, 36] ในช่วง NIR และ SWIR จะเป็นช่วงที่น้ำในใบดูดกลืนแสงได้ดี จึงสามารถใช้ในการวัดค่าปริมาณน้ำในใบได้ (Leaf Relative Water Content) [37] และการสะท้อนของแสงในช่วง NIR นี้ยังสามารถบ่งบอกถึงความหนาของใบ (Leaf Thickness) ได้ด้วยดังปรากฏในงานวิจัย [37] นอกจากนี้ในช่วง TIR/LWIR ยังสามารถสร้างภาพถ่ายความร้อนได้ ซึ่งมีการถูกนำมาใช้ในการวัดอุณหภูมิใบ ซึ่งสามารถนำมาวัดการตอบสนองต่อสภาวะความเครียดที่เกิดจากความเค็มของดินได้ (Salinity Stress) [38] นอกจากนี้ยังมีช่วงคลื่นอีกช่วงที่สามารถวัดความสามารถในการสังเคราะห์ด้วยแสงของใบได้นั้นก็คือ ช่วง Chlorophyll fluorescence (700-800nm) [39]

อย่างไรก็ตาม การถ่ายภาพในช่วง VIS และ NIR นั้นก็ยังมีข้อเสียจากช่วงกว้างของความยาวคลื่น ที่ทำให้เกิดสัญญาณรบกวนได้ง่ายซึ่งบางทีก็จะทำให้สัญญาณภาพในช่วงความยาวคลื่นที่สนใจสูญหายได้ ซึ่งปัญหานี้สามารถแก้ไขได้หากใช้กล้อง Hyperspectrum ในการถ่าย เนื่องจาก กล้องสามารถให้ข้อมูลที่มีความละเอียดของช่วงความยาวคลื่นในระดับ mm สำหรับทุกจุด Pixel บนภาพถ่าย แต่อย่างไรก็ตามกล้องประเภทนี้ยังมีราคาที่สูงมากและจำเป็นต้องใช้ระบบประมวลผลที่มีกำลังการคำนวณสูงมาก[29]

ในงานวิจัยนี้ผู้เขียนมองเห็นถึงความสำคัญของกล้อง Hyperspectrum ในการเก็บลักษณะของพืชรวมถึงการระบุถึงสภาวะความเครียดของพืช ประกอบกับความก้าวหน้าของเทคโนโลยีทั้งด้าน Computer Vision และ Deep Learning ผู้เขียนจึงเสนอระบบระบุถึงสภาวะความเครียดในระยะเริ่มต้น ที่ใช้กระบวนการการเรียนรู้เชิงลึก ซึ่งถูกติดตั้งอยู่บน FPGA ที่เป็นอุปกรณ์ที่มีพลังการคำนวณสูงและใช้พลังงานต่ำ โดยใช้เครื่องมือการสังเคราะห์ระดับสูง (High Level Synthesis Tool) Vivado HSL ในการสร้าง แต่ด้วยข้อจำกัดในการเข้าถึงของกล้อง Hyperspectrum ซึ่งมีราคาสูงมาก ผู้เขียนจึงจะเสนอระบบที่ใช้กับกล้องในช่วง VIS แทน โดยคาดหวังให้ระบบนี้สามารถนำไปพัฒนาต่อเพื่อใช้ในการวิเคราะห์ภาพจากกล้อง Hyperspectrum ได้โดยง่าย นอกจากนี้เพื่อเพิ่มประสิทธิภาพของแบบจำลองในการระบุสภาวะความเครียดของต้นพืชในระยะเริ่มต้นจึงเสนอ

แบบจำลอง CNN+LSTM โดยใช้แบบจำลอง State of the art ที่ใช้กันอยู่ในปัจจุบันผ่านวิธีการ Transfer Learning เพื่อใช้ในการเรียนรู้ลักษณะสภาวะความเครียดที่มีความสัมพันธ์เชิงเวลา

### 1.3 วัตถุประสงค์

เพื่อพัฒนาระบบระบุถึงสภาวะความเครียดของต้นข้าวฟ่างที่ขาดสารอาหาร ซึ่งใช้กระบวนการทาง Computer vision และ Deep Learning ประเภท CNN (Convolution Neural Network) บน FPGA ด้วย Vivado HLS

### 1.4 ขอบเขต

- ระบบจะรับข้อมูลรูปภาพมาเพื่อดำเนินการวิเคราะห์
- Dataset : Dataset: High-throughput profiling identifies resource use efficient and abiotic stress tolerant sorghum varieties : 96,867 PNG images [40]
- แบบจำลอง CNN จะถูกใช้เพื่อระบุว่าต้นพืชอยู่ในสภาวะเครียดหรือไม่
- แบบจำลอง CNN จะถูกฝึกและปรับค่าบนคอมพิวเตอร์ที่ติดตั้งการ์ดจอ Nvidia
- แบบจำลองที่ถูกฝึกแล้วจะถูกถ่ายไปยังบอร์ด PYNQ-Z1 เพื่อดำเนินการระบุสภาวะเครียดของพืชบน FPGA
- แบบจำลอง CNN+LSTM จะถูกใช้เพื่อระบุว่าต้นพืชอยู่ในสภาวะเครียดหรือไม่
- แบบจำลอง CNN+LSTM จะถูกฝึกและปรับค่าบนคอมพิวเตอร์ที่ติดตั้งการ์ดจอ Nvidia
- ค่าความแม่นยำในการทำนายสภาวะความเครียดของพืชต้องมากกว่า 80%

### 1.5 ประโยชน์

1. ใช้เป็นระบบในการเฝ้าติดตามเฝ้าระวังการเกิดสภาวะเครียดของพืชเพื่อเตือนถึงความผิดปกติในระยะเริ่มต้นแก่ระบบเก็บลักษณะพืช (Plant Phenotyping) เพื่อป้องกันความสูญเสียของผลผลิต และพัฒนาสายพันธุ์พืชต่อไป
2. เป็นระบบที่สามารถพัฒนาต่อยอดเพื่อใช้กับเทคโนโลยีภาพ Hyperspectral Imaging ได้

## บทที่ 2 ทฤษฎีและหลักการ

### 2.1 การเก็บลักษณะพืช (Plant Phenotyping)

การเก็บลักษณะของพืช (Plant Phenotyping) คือ การอธิบายลักษณะทางกายวิภาค การเจริญเติบโต หรือ พัฒนาการ สรีรวิทยา และทางชีวเคมี ของต้นพืชในเชิงปริมาณ ซึ่ง

การเก็บลักษณะของพืชนั้นมีการทำอย่างต่อเนื่องมาตั้งแต่สมัยก่อน โดยนักวิทยาศาสตร์ นักวิจัย หรือผู้ปลูกพืช ซึ่งมีขั้นตอนการเก็บข้อมูลที่หลากหลายและซับซ้อนในบางลักษณะที่เก็บ ซึ่งในปัจจุบันนี้การเก็บข้อมูลส่วนใหญ่ก็ยังคงทำโดยมนุษย์เป็นหลัก

### 2.1.1 ประเภทของการเก็บลักษณะพืชที่ใช้อยู่ในปัจจุบัน

1. เก็บโดยใช้หลักการการสะท้อนและดูดกลืนเชิงสเปกตรัมของ ใบ ลำต้น ของพืช หรือ ต้นไม้
  - a. โดยส่วนมากจะเป็นการอ้างอิงกับสีของใบไม้เป็นหลัก เนื่องจากสีเขียวของใบไม้นั้นจะถูกกำหนดโดยปริมาณและการพัฒนาของสารคลอโรฟิลล์ ซึ่งจะเปลี่ยนแปลงไปตามการเจริญเติบโต สารอาหารที่ได้รับ และสภาวะความเครียดที่เกิดขึ้นกับต้นพืช
  - b. ดัชนีที่ใช้วัด : Normalized difference vegetation index (NDVI)
    - i. หลักการ : ดัชนีนี้จะใช้หลักการเปรียบเทียบระหว่างช่วง VIS (Visible light spectrum) ซึ่งเป็นช่วงที่มีการดูดกลืนแสงของคลอโรฟิลล์มากที่สุด และ NIR (Near Infrared) ซึ่งไม่มีผลกระทบจากการสังเคราะห์ด้วยแสง
    - ii. สมการของ NDVI : 
$$NDVI = \frac{NIR - VIS}{NIR + VIS}$$
2. เก็บข้อมูลผ่านอุณหภูมิของใบ หรือ เรือนยอด (Canopy)
  - a. เนื่องจากเนื้อเยื่อของพืชนั้นจะถูกทำให้อุณหภูมิต่ำลงผ่านการคายน้ำของใบ ข้อมูลอุณหภูมิใบจึงสามารถนำมาหาอัตราการหายใจหรือคายน้ำของพืชได้
    - i. ซึ่งอัตราการหายใจนี้จะสามารถนำมาเฝ้าระวังสภาวะความเครียดที่เกิดจากโรคที่ส่งผลกระทบต่ออัตราการหายใจของพืชได้
    - ii. นอกจากนี้ยังสามารถบ่งบอกถึงความต้องการน้ำของต้นพืชได้อีกด้วย
  - b. ปัญหา
    - i. อุณหภูมิของต้นพืชขึ้นอยู่กับสภาพแวดล้อมรอบๆต้นพืชนั้นมาก หากสภาพแวดล้อมเปลี่ยนแปลงจะทำให้อุณหภูมิเปลี่ยนแปลงไปด้วย จึงยากที่จะคำนวณหาอุณหภูมิที่โดนกระทบโดยอัตราการหายใจได้

### 3. ขนาด รูปร่าง โครงสร้าง และการเจริญเติบโตของพืช

#### a. โดยจะถูกแบ่งการเก็บลักษณะออกเป็น 2 อย่างคือ

i. ส่วนที่อยู่ใต้ดิน : ราก ประสิทธิภาพในการเจริญเติบโตของ ต้นไม้ที่ขึ้นอยู่กับโครงสร้างและการทำงานของระบบราก เป็นสำคัญ

ii. ส่วนที่อยู่เหนือพื้นดิน : ลำต้น ใบ เรือนยอด

1. ความสัมพันธ์ระหว่างการเจริญเติบโตและ พารามิเตอร์ต่างๆของสภาพแวดล้อมจะเป็นตัวแปร สำคัญในการพัฒนาการปลูกในอนาคต

2. ค่าที่สามารถวัดได้

a. โครงสร้างของยอดเรือน

b. รูปทรงของต้นพืช

c. พื้นที่ใบ

d. จำนวนใบ

e. ความสูง

f. อัตราการเจริญเติบโต

g. การครอบคลุมของยอดเรือน (CC : Canopy Cover)

#### 2.1.2 กระบวนการตัดภาพเฉพาะส่วนใบ (Leaf Segmentation)

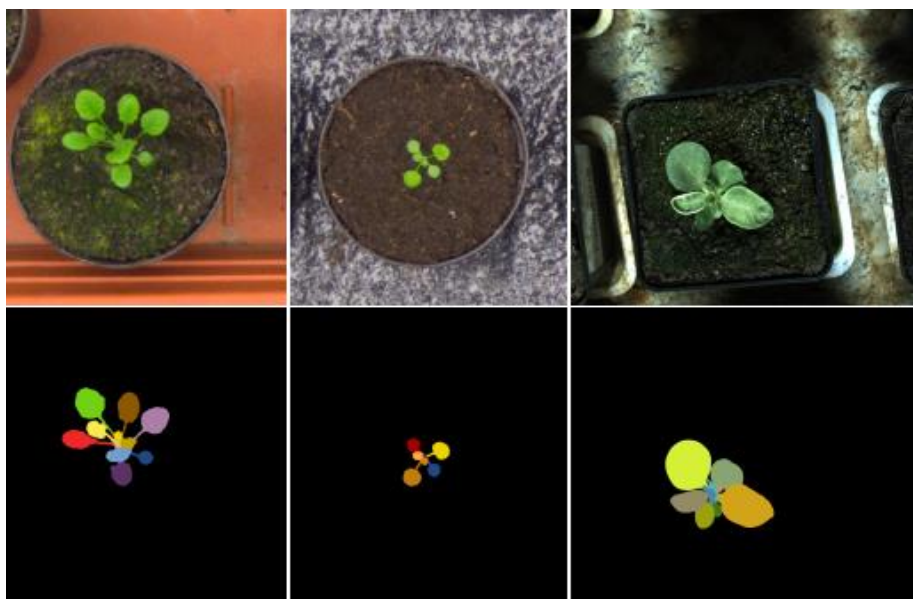
ในงานวิจัยนี้ผู้วิจัยเลือกใช้กระบวนการ IPK Gatersleben : 3-D Histogram-Based Segmentation and Leaf Detection [41] ในการตัดภาพเฉพาะส่วนใบซึ่งมี กระบวนการดังต่อไปนี้

##### 1. ขั้นตอนการรวบรวมข้อมูลภาพ (Image Acquisition)

การใช้กระบวนการนี้ ในขั้นแรกนั้นจำเป็นต้องมีการฝึกโมเดล เพื่อที่จะแยกภาพว่าส่วนไหนของภาพคือตัวต้นพืชและส่วนไหนของภาพนั้น คือพื้นหลัง ดังนั้นจึงจำเป็นต้องเตรียมข้อมูลภาพของต้นพืชและข้อมูลของ ส่วนที่เป็นพื้นหลังที่ถูกระบุมาโดยผู้เชี่ยวชาญ ซึ่งในงานวิจัยของผู้คิด กระบวนการนี้ได้ใช้ข้อมูลตัวอย่างจาก CVPPP 2014 ซึ่งเป็นผู้จัดรายการการ แข่งขันการตัดภาพใบ LSC (Leaf Segmentation Challenge) โดยมีข้อมูลภาพ ของพืช และข้อมูลที่ถูกระบุใบ ต้นพืช จำนวนใบ พื้นหลัง เตรียมไว้ให้



(Ground-truth data) ซึ่งในงานวิจัยเจ้าของกระบวนการ ข้อมูลที่ถูกใช้ฝึก ประกอบไปด้วย ข้อมูลจากชุดข้อมูล A1(500X530 px) จำนวน 95 ภาพ A2(530X565 px) จำนวน 31 ภาพ และ A3(2448X2048 px) จำนวน 27 ภาพ ซึ่งสามารถดูตัวอย่างภาพได้ดังรูปที่ 6



รูปที่ 6 ภาพที่ถูกใช้ในการฝึกของกระบวนการ IPK ในการแยกพื้นหลังออกจากต้นพืช เรียงลำดับ จาก ซ้าย กลาง ขวา คือ ชุดข้อมูล A1 A2 และ A3[41]

## 2. ขั้นตอนเตรียมการ (Preprocessing)

2.1 เพื่อลดผลกระทบของความไม่แน่นอนของแสงสว่าง ภาพทั้งหมดจะถูกเปลี่ยนจาก RGB Color Space เป็น Lab Color Space

2.2 เพื่อลดผลกระทบจากการถ่ายภาพที่ Overexposure และเงา จึงใช้กระบวนการ Simple Color Thresholding

2.4 กระบวนการ Gaussian Blur

2.3 สร้าง 3-D Histogram Color Cubes เพื่อใช้เป็นโมเดลในการแยกภาพว่าส่วนใดเป็นพื้นหลังส่วนใดเป็นต้นพืช โดยจากภาพข้อมูลชุดฝึก จะสำรวจจุด Pixel ของภาพฝึกแต่ละจุด และเปรียบเทียบเพื่อดูว่าจุดนั้นเป็นพื้นหลังหรือตัววัตถุ โดยค่า Lab ของแต่ละจุดนั้นจะถูกนำมาเป็นดัชนีของ 3-D Histogram Cube เมื่อชุดข้อมูลฝึกถูกเรียนรู้ทั้งหมด 3-D Histogram Cube ของต้นพืชที่สนใจและพื้นหลังก็จะถูกสร้างขึ้น

### 3. การตัดภาพ (Segmentation)

3-D Histogram Cube จะถูกใช้เป็นเหมือนตารางค้นหา (Look-up table) ซึ่งเก็บความน่าจะเป็นของจุดสีในระบบ Lab ว่าเป็นพื้นหลัง หรือเป็นต้นพืชที่เราสนใจ โดยพิจารณาจากจำนวน Pixel ทั้งหมดที่มีค่า Lab ณ ตำแหน่งนั้นและจำนวนจุด Pixel ที่ตำแหน่งนั้นที่ถูกแยกโดยข้อมูลสีว่าเป็นพื้นหลังหรือต้นพืชที่สนใจ โดยจุดสีบน 3-D Histogram Cube ที่ไม่พบในข้อมูลสีจะให้มีความน่าจะเป็นเท่ากับ 0 โดย ณ จุดใดๆที่พิจารณาจะใช้วิธีหาค่าเฉลี่ยจากจุดรอบๆ หากจุดๆนั้นไม่มีข้อมูลที่พบในข้อมูลสี

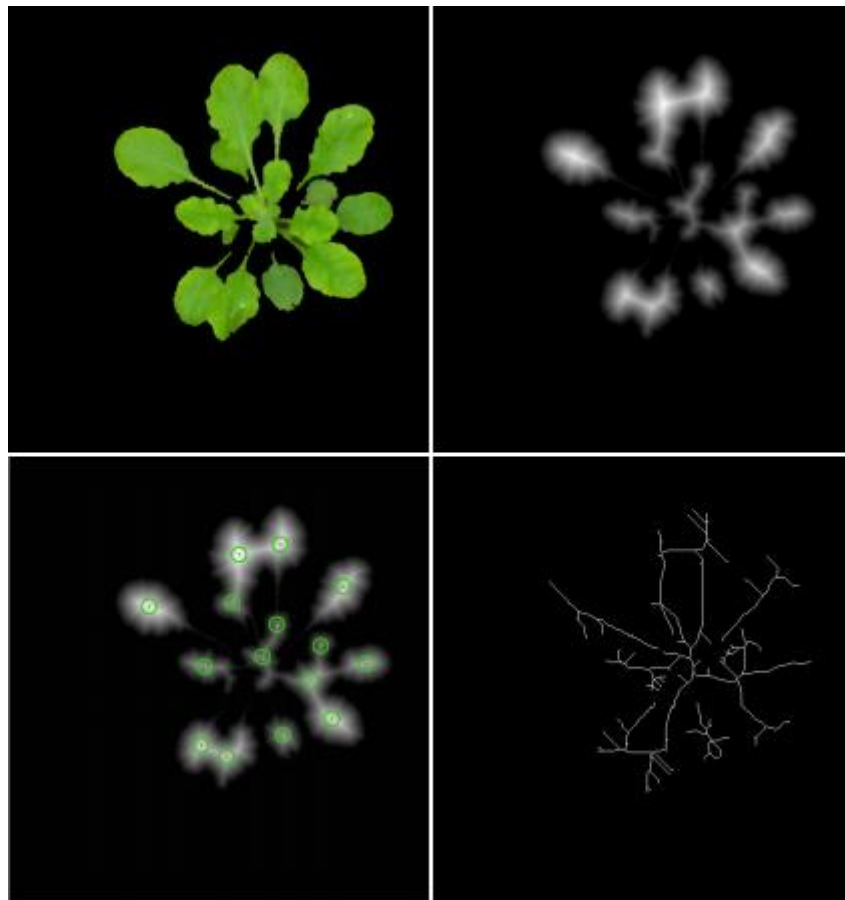
### 4. การสกัดลักษณะไป (Feature Extraction)

หลังจากการตัดเฉพาะภาพต้นพืชที่สนใจออกมาแล้ว จะทำการการระบุไปแต่ละไปบนต้นพืช ซึ่งจะทำตามขั้นตอนดังต่อไปนี้

4.1 ใช้ Morphological erode operation

4.2 คำนวณหา Euclidean distance map (EDM)

4.3 คำนวณหา Skeleton image



รูปที่ 7 ผลจากการตัดภาพเฉพาะส่วนที่สนใจ (บนซ้าย) EDM(บนขวา) ภาพ EDM ที่วางกลมค่าสูงสุดแต่ละจุดซึ่งถือว่าเป็นจุดศูนย์กลางของใบ(ล่างซ้าย) Skeleton image(ล่างขวา) [41]

4.4 Graph Representation คำนวนหา Graph Structure โดยใช้ข้อมูลจุด

ศูนย์กลางใบที่ได้จาก EDM ร่วมกับ Skeleton image ซึ่งจาก Graph Structure ที่ได้นี้จะทำให้สามารถคำนวณหาขอบใบต่างๆได้

## 5. ขั้นสุดท้าย

คำนวณเก็บลักษณะใบที่ต้องการและสร้างภาพหลังจากที่ผ่านกระบวนการเพื่อรายงานผล

## 2.2 การระบุสถานะความเครียดของพืช (Stress Identification in Plant)

การระบุสถานะความเครียดของพืชในงานวิจัยนี้จะทำโดยใช้กระบวนการเรียนรู้เชิงลึก (Deep learning)

### 2.2.1 Deep learning (DL)

DL ถือเป็นกระบวนการที่เป็นส่วนหนึ่งของ Machine Learning ซึ่งใช้หลักการการคำนวณ เป็นชั้นๆมาต่อกัน ซึ่งแต่ละชั้นการคำนวณ (Processing Layer) จะใช้ผลลัพธ์การคำนวณของชั้นก่อนหน้ามาเป็น Input ของชั้นตัวเอง เพื่อที่จะได้เรียนรู้ถึงรูปแบบข้อมูลของชุดข้อมูลนั้นๆ โดยปกติแล้ว DL จะถูกสร้างขึ้นจาก Neural Network หลายๆชั้น ซึ่งแต่ละชั้นก็จะมีค่าพารามิเตอร์ของตัวเองที่สามารถปรับเพื่อเรียนรู้ข้อมูลต่างๆได้ [24]

### 2.2.2 Convolution Neural Network (CNN)

CNN ถือว่าเป็น DL ประเภทหนึ่ง ซึ่งมีลักษณะทั่วไปคล้ายกับ Neural Network หลายชั้น โดยที่ทั้ง 2 กระบวนการนี้คำนวณผลลัพธ์โดยการคูณเวกเตอร์ระหว่างข้อมูล Input และพารามิเตอร์ที่สามารถเรียนรู้ได้ และยังต้องการการเรียนรู้พารามิเตอร์ผ่านกระบวนการ Back propagation เพื่อปรับค่าพารามิเตอร์ของแต่ละชั้นการคำนวณ แต่สิ่งที่ CNN ต่างคือ CNN จะมีโครงสร้างที่ใช้สำหรับการรับข้อมูล Input ในรูปแบบของรูปภาพเท่านั้น จึงทำให้การทำ Forward propagation ของกระบวนการเปรียบเสมือนการทำ 2D Convolution

โดยทั่วไปนั้น CNN จะประกอบด้วยชั้นการคำนวณหลากหลายชั้น ที่ทำหน้าที่ต่างกัน ประกอบด้วย

#### 1. ชั้น 2D Convolution

ในชั้นนี้จะเป็นการทำ Convolution กันระหว่างข้อมูลภาพ Input และ Kernel ซึ่งโดยปกติแล้วขาออกจากชั้น 2D Convolution จะต่อเข้ากับชั้น ReLU ซึ่งขนาดมิติของผลลัพธ์ของชั้นนี้ในกรณีที่ขนาดมิติของภาพกว้างและยาวมีขนาดเท่ากัน สามารถคำนวณได้จาก สมการดังต่อไปนี้

$$O = \left( \frac{I - K + 2P}{s} \right) + 1$$

โดยที่

O คือ ขนาดมิติความกว้างของเมทริกซ์ Output (ด้านกว้างยาวเท่ากัน)

I คือ ขนาดมิติความกว้างของเมทริกซ์ Input (ด้านกว้างยาวเท่ากัน)

K คือ ขนาดของ Kernel

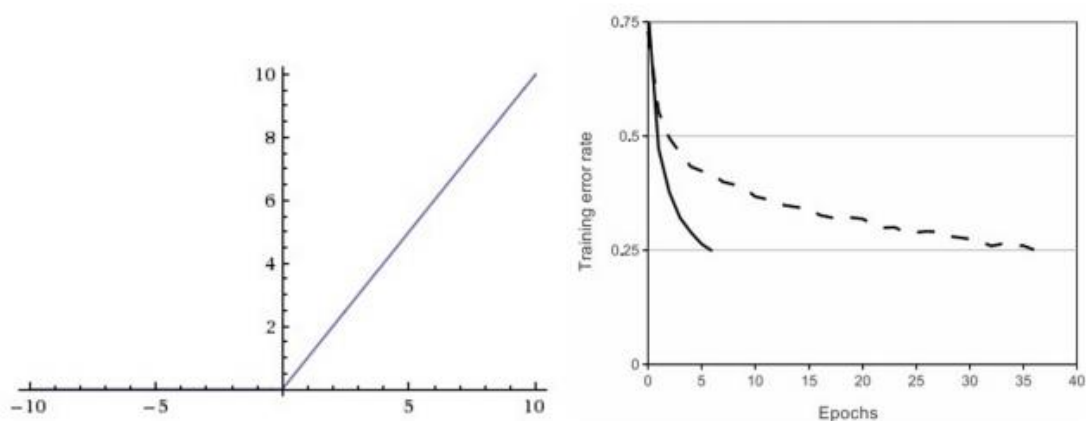
P คือ ขนาด Padding

S คือ ขนาด Stride

## 2. ชั้น Rectified Linear Unit (ReLU)

ReLU คือการทำ Linear Activation รูปแบบหนึ่งที่มีค่า Threshold อยู่ที่ศูนย์ ซึ่ง ReLU ได้รับความนิยมอย่างมากใน CNN เนื่องด้วยมีงานวิจัยระบุว่า การใช้ ReLU นั้น จะทำให้การฝึก CNN ลู่เข้าเร็วขึ้น และการคำนวณ ReLU หากเทียบกับ tanh Sigmoid หรือ Logistic Sigmoid แล้ว สามารถคำนวณได้ง่ายกว่ามาก [42]

$$f_{output} = \text{MAX}(f^{in}, 0)$$



รูปที่ 8 ReLU Activation Function (ซ้าย) กราฟแสดงการเปรียบเทียบระหว่างการใช้ ReLU [เส้นทึบ] และ tanh [เส้นปะ] พบว่า ReLU ลู่เข้าเร็วกว่าถึง 6 เท่า (ขวา) [42]

## 3. ชั้น Max/Average Pooling

การทำ Pooling คือการสุ่มดึงตัวอย่างข้อมูลออกมาจากข้อมูลที่ใหญ่กว่า ซึ่งกระทำโดยการกำหนดขนาดบริเวณที่ต้องการดึงข้อมูล และใช้วิธีหาค่าประมาณหรือค่าสูงสุดของบริเวณนั้น ซึ่งการทำ Pooling จะช่วยลดความซับซ้อนในการคำนวณและยังช่วยป้องกัน Overfitting อีกด้วย

#### 4. ชั้น Fully Connected Layer

ในชั้นนี้จะทำการคูณเวกเตอร์ (Dot Product) ของข้อมูลภาพ Input และค่าถ่วงน้ำหนัก (Weight) ของโมเดล โดยปกติชั้นนี้จะอยู่ในชั้นสุดท้ายของการทำ CNN

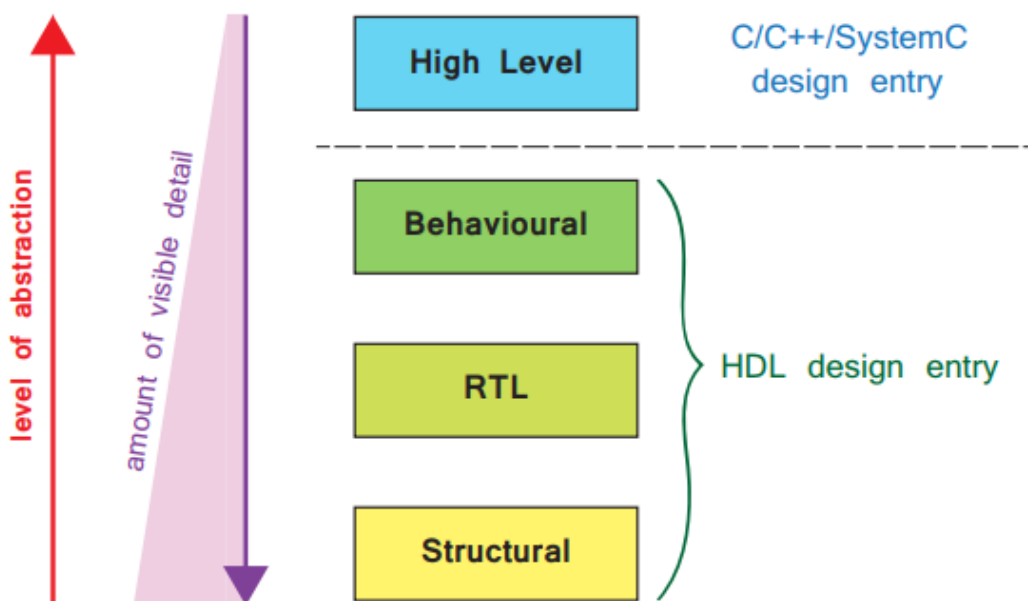
#### 2.2.3 LSTM (Long Short-Term Memory) Network

Recurrent Neural Network หรือที่เรียกว่า RNN เป็น Neural Network ประเภทหนึ่งที่ถูกใช้งานอย่างแพร่หลายในงานทางด้าน Speech Recognition และ NLP (Natural language processing) โดย RNN นั้นถูกออกแบบมาให้ทำงานกับข้อมูลที่มีลักษณะเป็น Sequential Data หรือข้อมูลที่มีความสัมพันธ์กับเวลา (Temporal Data) RNN จะทำงานได้ดีมากกับข้อมูลขาเข้าที่มีความสัมพันธ์กับข้อมูลขาเข้าในอดีต LSTM Network เป็น RNN ประเภทหนึ่งที่ถูกพัฒนาให้แก้ปัญหาเกี่ยวกับ Vanishing Gradient โดยการเพิ่ม Gates เข้าไปควบคุม State ภายในของ Network ให้แต่ละ LSTM Node ใน Network สามารถที่จะเรียนรู้ว่าข้อมูลส่วนไหนควรจำหรือ ข้อมูลส่วนไหนควรลืมได้

### 2.3 Deep learning บน FPGA

ในการพัฒนา FPGA นั้นจำเป็นต้องอาศัยความรู้ในการพัฒนาฮาร์ดแวร์เป็นอย่างมาก ต้องเข้าใจการออกแบบวงจรดิจิทัล และต้องสามารถพัฒนาระบบโดยใช้ Hardware Description Language(HDL) อาทิเช่น VHDL หรือ Verilog ได้ นอกจากนี้การพัฒนา FPGA ยังค่อนข้างใช้เวลามากและมีความซับซ้อนสูงหากเทียบกับการพัฒนาซอฟต์แวร์ ในระบบปฏิบัติการต่างๆ จึงทำให้เกิดการพัฒนากระบวนการออกแบบแบบใหม่ที่เรียกว่า High-Level Synthesis ขึ้นซึ่งเป็นใช้ภาษาระดับสูงในการเขียนโปรแกรมเช่น C/C++ ในการเขียนกระบวนการของซอฟต์แวร์และแปลภาษาเหล่านี้ให้กลายเป็นชุดคำสั่งที่ในภาษา HDL ได้ ซึ่งการออกแบบด้วยวิธีนี้นอกจากจะทำให้ให้นักพัฒนาที่ไม่ได้มีความรู้ในการออกแบบฮาร์ดแวร์สามารถที่จะพัฒนาซอฟต์แวร์ที่ใช้ได้บน FPGA ได้แล้ว ยังสนับสนุนให้เกิดการนำซอฟต์แวร์กลับมาใช้ใหม่ (Design Reuse) ด้วย

### 2.3.1 High-Level Synthesis (HSL)



รูปที่ 9 ระดับของการออกแบบ FPGA

ในระดับการออกแบบ FPGA ในระดับ Structural นั้น จะใช้ HDL ในการออกแบบ ซึ่งการออกแบบจะเกี่ยวข้องกับการสร้าง ปรับค่า และ ต่อกันของ ส่วนต่างๆของฮาร์ดแวร์ระดับล่าง ซึ่งบางครั้งก็ต้องทำกันลงไปถึงระดับ LUTs (Look-up table) หรือ FFs (Flip Flop) ซึ่งการออกแบบลักษณะนี้ผู้ออกแบบจะเป็นผู้ควบคุมรายละเอียดของการออกแบบทุกอย่าง

ในระดับ RTL รายละเอียดในระดับเทคโนโลยีจะถูกซ่อน แต่จะสามารถออกแบบในระดับฮาร์ดแวร์ได้โดยผ่าน Registers และกระบวนการต่างๆที่เกิดขึ้นระหว่าง Registers ซึ่ง Logic synthesis tool นั้นจะถูกใช้งานในระดับนี้เพื่อแปล RTL Code เป็นภาษาฮาร์ดแวร์

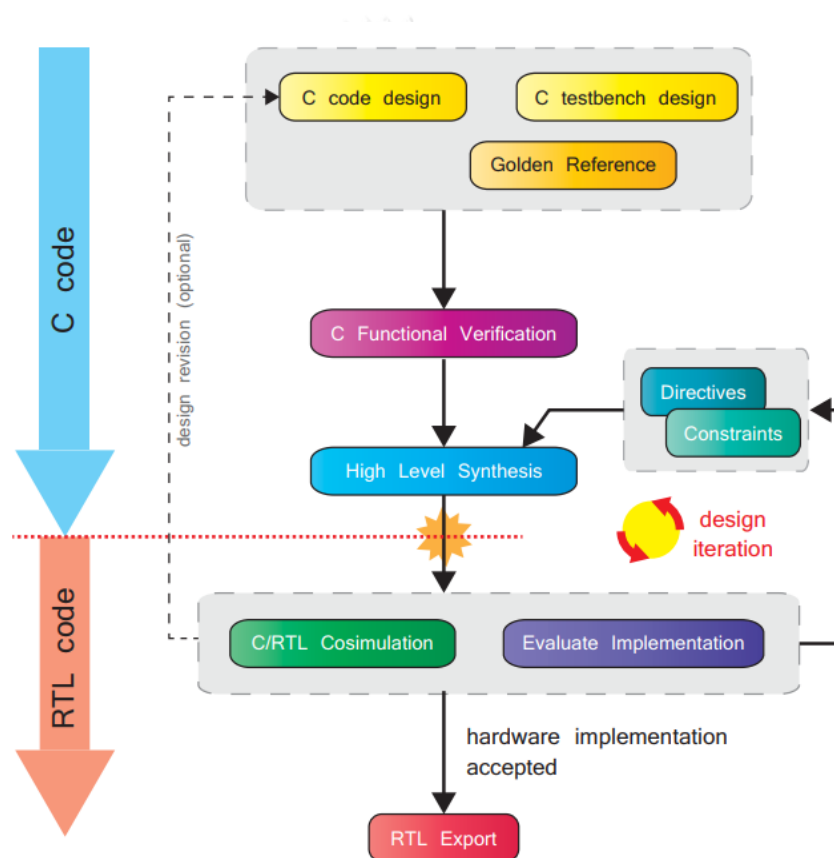
ในระดับ Behavioral นั้นจะใช้ HDL ในการออกแบบพฤติกรรมและกระบวนการของวงจรมากกว่าอธิบายขั้นตอนการทำงานของ Register ซึ่งการออกแบบแบบนี้จะต้องพึ่งความสามารถในสังเคราะห์วงจรออกมาจากการอธิบายพฤติกรรม ของเครื่องมือการสังเคราะห์ ซึ่งผู้ออกแบบจะควบคุมกระบวนการสร้างวงจรได้น้อยลง แต่ก็สามารถพัฒนาได้รวดเร็วขึ้น

ในระดับ High-Level นั้น การออกแบบจะไม่ได้ยุ่งเกี่ยวกับ HDL แต่จะเป็นการใช้ภาษาในการเขียนโปรแกรมระดับสูง อันได้แก่ C++ และ SystemC ในการออกแบบกระบวนการ หากพิจารณาถึงการสังเคราะห์วงจร ในระดับ HDL Design จะพบว่า การสังเคราะห์จะเป็นการสังเคราะห์จาก HDL ไปยังชุดคำสั่งที่อธิบายการเชื่อมต่อ (Netlist) ซึ่งสามารถใช้ในวงจรโลจิกได้เลย หรือเรียก

วิธีการสังเคราะห์แบบนี้ว่า Logic Synthesis แต่ในระดับ High-Level นั้นการสังเคราะห์จะไม่ใช้แบบ Logic Synthesis แต่จะเป็นการสังเคราะห์จาก High-Level ไปเป็น HDL ซึ่งจำเป็นต้องผ่านการสังเคราะห์แบบ Logic Synthesis อีกครั้งก่อนที่จะนำไปใช้ใน FPGA ได้

### 2.3.2 Vivado HLS

Vivado HLS จะทำหน้าที่แปล ภาษา C C++ และ SystemC ไปเป็น RTL ซึ่งสามารถสังเคราะห์และนำไปใช้ใน FPGA Board ได้ ซึ่งภาพรวมของกระบวนการการออกแบบด้วย Vivado HLS นั้นจะสรุปได้ดังรูปที่ 10



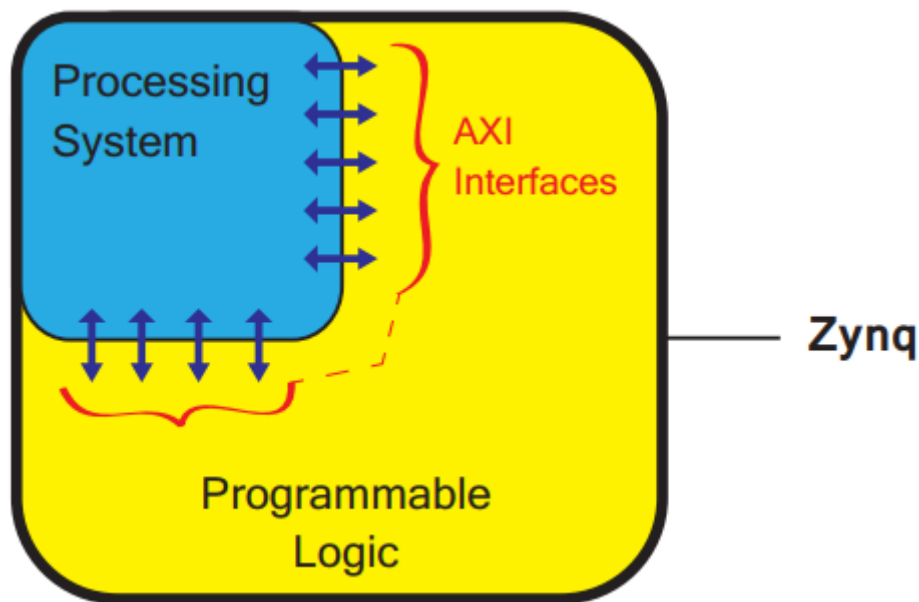
รูปที่ 10 ภาพรวมของการออกแบบด้วย Vivado HLS

### 2.3.3 System-on-Chip กับ Zynq

ในอดีต System-on-Chip (SoC) ส่วนใหญ่จะอยู่ในรูปของ ASIC (Application Specific Integrated Circuit) ซึ่งมีข้อเสียคือ ต้องใช้ทุนและเวลามากในการพัฒนา นอกจากนี้ระบบยังขาดความยืดหยุ่น จึงทำให้เหมาะกับการผลิตในจำนวนมากๆเท่านั้น

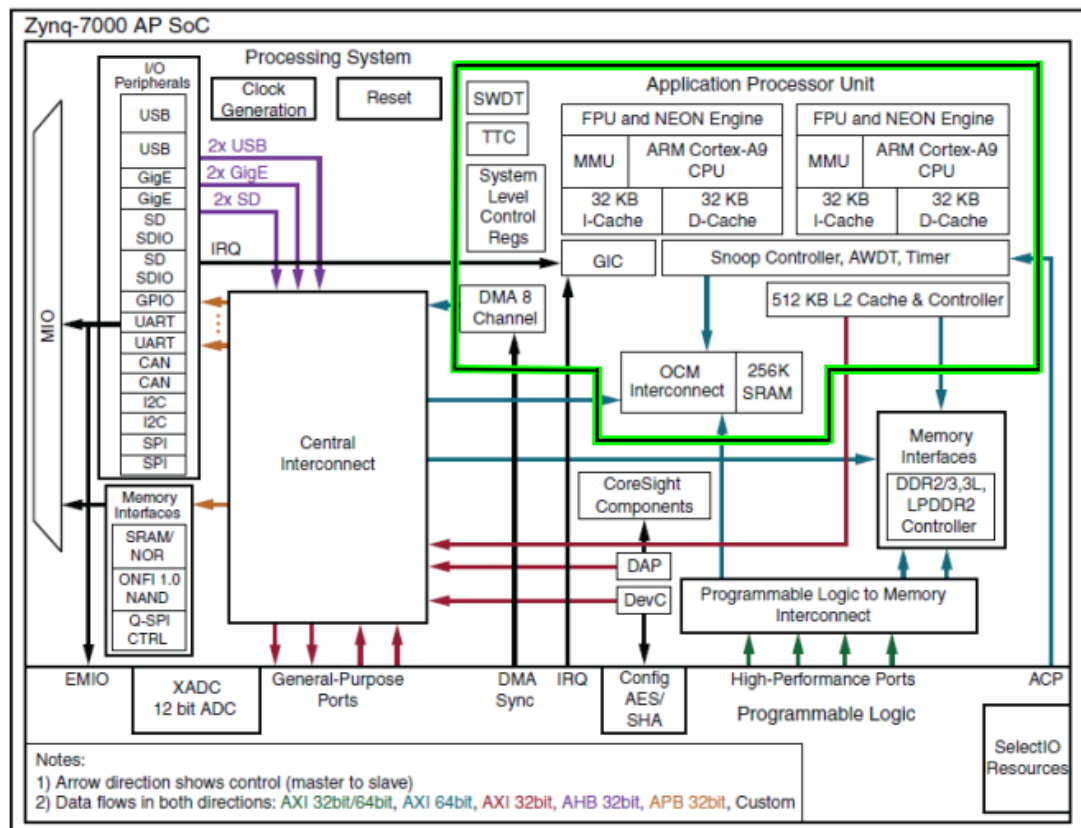


เมื่อมองเห็นถึงข้อเสียเหล่านี้ Xilinx จึงได้พัฒนา Zynq ขึ้นซึ่งเป็นระบบที่นำ SoC มารวมกับ FPGA ดังรูปที่ 11



รูปที่ 11 แสดงโครงสร้างของ Zynq

ซึ่ง Zynq ประกอบไปด้วย 2 ส่วนการทำงานหลักๆ คือ Processing System (PS) และ Programmable Logic (PL) หรือ FPGA ซึ่งทั้ง 2 ส่วนนี้ทำงานโดยสามารถส่งข้อมูลถึงกันได้ผ่าน AXI Interface ทั้ง 9 ตัว ซึ่ง PL นั้นเหมาะกับกระบวนการที่ต้องการความเร็ว เช่น High-speed logic การคำนวณข้อมูลจำนวนมาก การคำนวณแบบขนาน ระบบการเคลื่อนย้ายข้อมูล ในขณะที่ PS นั้นสามารถติดตั้งระบบปฏิบัติการได้ ซึ่งทำให้การออกแบบระบบใดๆสามารถแบ่งออกเป็นแบบฮาร์ดแวร์และซอฟต์แวร์ได้



DS190\_01\_000713  
© Xilinx

รูปที่ 12 Zynq Processing System

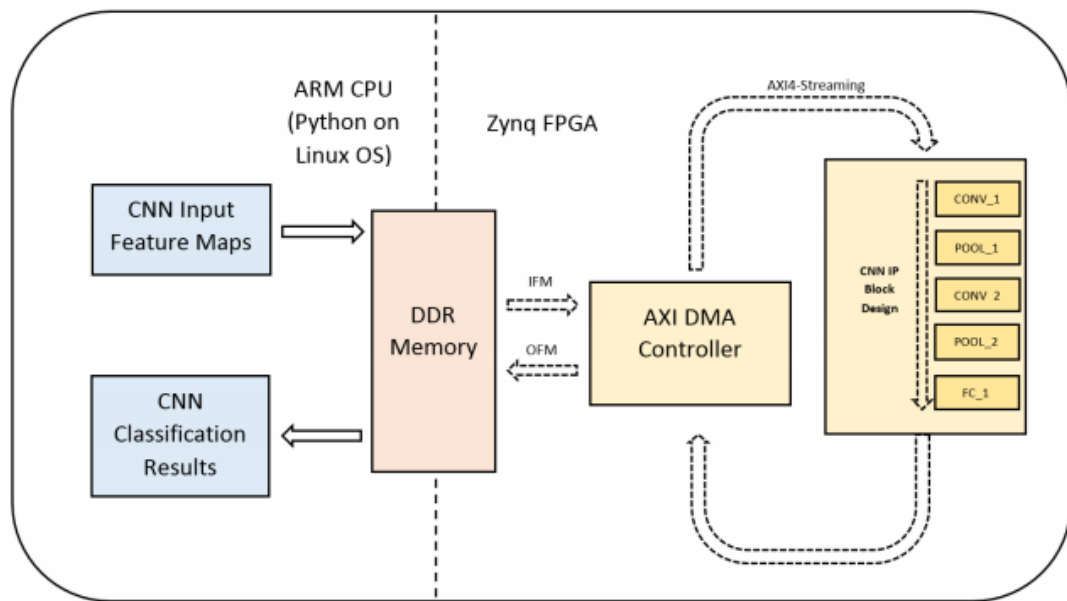
### 2.3.4 PYNQ Platform

ในปัจจุบันแนวโน้มการพัฒนา FPGA มุ่งเน้นความเป็นมิตรต่อนักพัฒนามากขึ้นเรื่อยๆ ด้วยความต้องการที่จะดึงผู้พัฒนาซอฟต์แวร์ให้หันมาใช้ FPGA Xilinx จึงได้ริเริ่มโครงการ PYNQ ซึ่งเป็นระบบการทำงานใหม่ที่ทำให้ผู้พัฒนาสามารถใช้ FPGA IPs ที่ถูกสังเคราะห์มาแล้ว หรือที่เรียกว่า Overlay ในซอฟต์แวร์ที่เขียนขึ้นด้วยภาษา Python ได้

หากพิจารณาถึงการติดตั้ง CNN บน FPGA แล้วระบบ PYNQ มีความเหมาะสมมากเพราะว่า PYNQ นั้นมีระบบปฏิบัติการ Linux Ubuntu 15.10 อยู่ ซึ่งทำให้สามารถใช้ CNN ซอฟต์แวร์อย่างเช่น Theano ในการติดตั้งได้ โดยสามารถที่จะนำการคำนวณของ CNN ย้ายไปคำนวณอยู่บนฝั่ง FPGA ได้ ซึ่งการใช้ PYNQ จะทำให้สามารถใช้ CNN บน FPGA ได้ โดยไม่ต้องลงไปออกแบบในกระบวนการ CNN ในระดับ RTL

### 2.3.5 CNN บน PYNQ

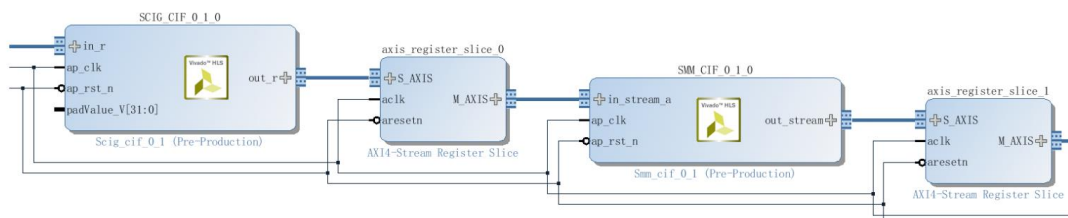
การพัฒนา ระบบ CNN บน PYNQ จะเริ่มจากการติดตั้ง Theano และ Lasagne ก่อน ซึ่งเป็นซอฟต์แวร์ของ Python ที่ใช้ในการออกแบบ CNN หลังจากนั้นจึงจำเป็นต้องสร้าง IP สำหรับแต่ละ Layer ของ CNN โดยใช้ Vivado HLS ในการสร้าง ซึ่งรายละเอียดของการติดตั้งและใช้งานกรอบการทำงานนี้สามารถพบได้ใน [43]



รูปที่ 13 ภาพรวมการทำงานของ CNN บน PYNQ

โดยในงานวิจัย [43] ผู้วิจัยได้สร้าง IP Block พื้นฐานเอาไว้ ซึ่งสามารถนำมาประกอบเป็น CNN ได้ IP Block ที่ออกแบบไว้ประกอบด้วย

- IP Blocks สำหรับ Convolution layer

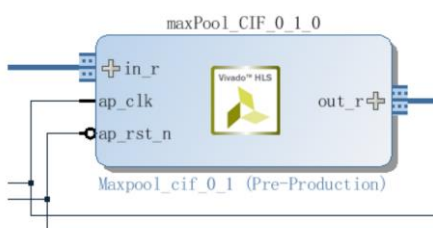


รูปที่ 14 IP Blocks สำหรับ Convolution layer [43]

- SCIG IP Block : ใช้สำหรับการเปลี่ยนรูปภาพให้กลายเป็นเวกเตอร์คอลัมน์ เพื่อใช้สำหรับการคูณกับ Kernel weighs
- SMM IP Block : เป็น IP Block สำหรับการคูณเมทริกซ์

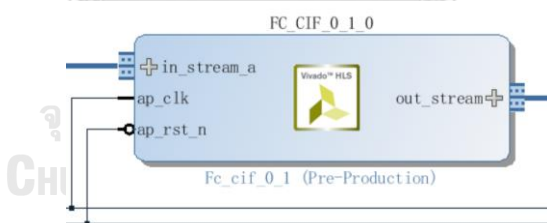
การใช้งาน IP block สองอันที่กล่าวมาพร้อมกันจะทำให้สามารถทำการ Convolution รูปภาพ และ Kernel บน FPGA ได้

- IP Block สำหรับ Pooling layer



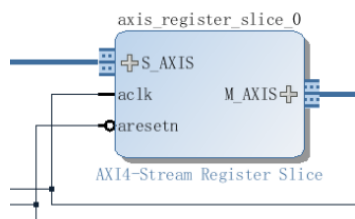
รูปที่ 15 IP Block สำหรับ Pooling layer [43]

- maxPool IP Block : maxPool IP สามารถทำงานได้ 2 Mode คือ การ Pooling โดยใช้ค่าสูงสุดและการ Pooling โดยใช้ค่าเฉลี่ย
- IP Block สำหรับ Fully connected layer



รูปที่ 16 IP Block สำหรับ Fully connected layer [43]

- FC IP Block
- IP Block สำหรับ ลด timing constraint อันเนื่องมาจากการคำนวณแบบ IP Block ที่เชื่อมต่อกันยาวมาก



รูปที่ 17 IP Block สำหรับลด Timing

- Axis\_register\_slice IP Block : IP Block นี้จะถูกใส่อยู่ระหว่างทุก IP Block ที่นำมาสร้าง Network



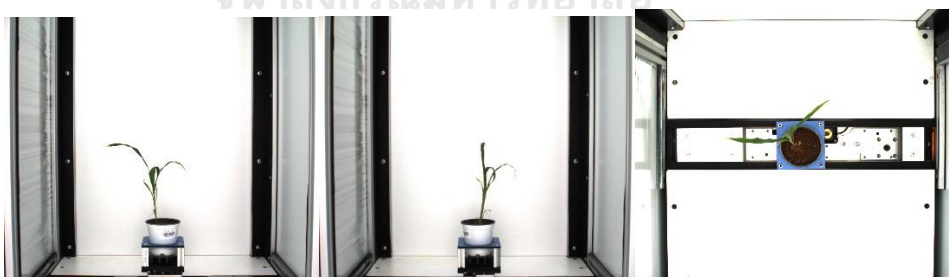
### บทที่ 3 การทดลอง

ในการทดลองนี้จะแบ่งเป็น 2 ส่วนหลักๆดังนี้ คือ

1. ทดลองติดตั้งแบบจำลองบน PYNQ โดยใช้ FPGA ในการคำนวณ CNN เพื่อระบุสถานะความเครียดของพืช
2. ทดลอง ใช้แบบจำลอง LSTM+CNN ในการระบุสถานะเครียดของพืชที่เฝ้าระวัง

#### 3.1 ชุดข้อมูล [Dataset] ที่ใช้ในการทดลอง

ชุดข้อมูลที่ใช้ในการทดลองนี้เป็นชุดข้อมูลที่ได้จากศึกษาลักษณะของข้าวฟ่างภายใต้การจำกัดสารอาหาร หรือที่เรียกว่า Treatment study ซึ่งมาจากการวิจัย [40] ในงานวิจัยนี้ ผู้ทดลองได้นำข้าวฟ่างทั้งหมด 30 สายพันธุ์ จำนวนทั้งหมด 847 ต้น มาเลี้ยงในสภาพแวดล้อมที่ถูกควบคุมและทำการถ่ายรูปทุกต้นในสามมุมมองประกอบด้วย ด้านหน้า ด้านข้าง และด้านบน เป็นระยะเวลา 19 วัน โดยข้างฟางทั้งหมดจะถูกแบ่งออกเป็น 3 กลุ่ม ซึ่งแต่ละกลุ่มจะได้รับสารอาหาร Ammonium และ Nitrate ในอัตราส่วนที่ต่างกัันดังนี้ กลุ่มที่หนึ่งกลุ่มควบคุม ได้รับ Ammonium 100% และ Nitrate 100% [100/100] กลุ่มที่ 2 ได้รับ Ammonium 50% และ Nitrate 10% [50/10] กลุ่มที่ 3 ได้รับ Ammonium 10% และ Nitrate 10% [10/10] รูปที่ถูกถ่ายระหว่างทำการทดลองถูกถ่ายโดยเครื่องมือสำหรับการเก็บลักษณะพืช (Plant phenotyping) โดยเฉพาะ ซึ่งมุมถ่าย ความสว่างแสง และปัจจัยอื่นๆถูกควบคุมให้เหมือนกันทุกครั้งที่ทำการถ่ายภาพ ซึ่งรูปที่ได้มาจะมีความละเอียดอยู่ที่ 2454 x 2056 Pixel ตัวอย่างของภาพถ่ายใน 3 มุมถูกแสดงไว้ในรูปที่ 18



รูปที่ 18 ภาพถ่ายของข้าวฟ่างที่ใช้ในงานวิจัย [40] ในมุมด้านหน้า ด้านข้าง และด้านบน ตามลำดับ

โดยในการฝึก CNN model ให้สามารถแยกสถานะความเครียดของพืชในช่วงระยะเริ่มแรกนั้น เนื่องด้วยข้อจำกัดของระบบคอมพิวเตอร์ที่ใช้ในการฝึก จะใช้ภาพเพียงมุมเดียวในการฝึก ซึ่งก็คือมุมมองด้านหน้า ชุดข้อมูลที่น่ามาฝึกจะถูกแบ่งออกเป็น 18 ชุดข้อมูล คือ

- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 2 ของการทดลอง จำนวน 1,694 ภาพ

- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 3 ของการทดลอง จำนวน 2,541 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 4 ของการทดลอง จำนวน 3,388 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 5 ของการทดลอง จำนวน 4,235 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 6 ของการทดลอง จำนวน 5,082 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 7 ของการทดลอง จำนวน 5,929 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 8 ของการทดลอง จำนวน 6,776 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 9 ของการทดลอง จำนวน 7,623 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 10 ของการทดลอง จำนวน 8,470 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 11 ของการทดลอง จำนวน 9,317 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 12 ของการทดลอง จำนวน 10,164 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 13 ของการทดลอง จำนวน 11,011 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 14 ของการทดลอง จำนวน 11,858 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 15 ของการทดลอง จำนวน 12,705 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 16 ของการทดลอง จำนวน 13,552 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 17 ของการทดลอง จำนวน 14,399 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 18 ของการทดลอง จำนวน 15,246 ภาพ
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 19 ของการทดลอง จำนวน 16,093 ภาพ

ซึ่ง จะถือว่าวันที่ 1 ของการทำการทดลองคือวันแรกที่ต้นข้าวทุกต้น เริ่มได้รับปัจจัยกระตุ้น หรือในการทดลองนี้ คือ สารอาหาร Ammonium และ Nitrate ในอัตราส่วนที่ต่างกัน 3 แบบ เป็นครั้งแรก โดยผู้ทดลองมีสมมุติฐานขั้นต้นว่า ต้นข้าวฟางในแต่ละกลุ่มที่ได้ปัจจัยกระตุ้นที่ต่างกันจะส่งผลให้คุณลักษณะทางกายภาพแตกต่างกันเมื่อต้นข้าวโตขึ้น

ดังนั้น หาก CNN Model สามารถแยกความแตกต่างของทั้ง 3 กลุ่มได้ในช่วงวันต้นๆ ซึ่งถือว่าเป็นช่วงที่ยังมีจำนวนข้อมูลน้อยอยู่ ก็จะสรุปได้ว่า CNN Model สามารถแยกพืชที่อยู่ในสถานะของพืชที่ขาดสารอาหาร Ammonium และ Nitrate ในระยะเริ่มต้นได้

### 3.2 ขั้นตอนการทดลอง

- 3.2.1 ทดลองติดตั้งแบบจำลองบน PYNQ โดยใช้ FPGA ในการคำนวณ CNN เพื่อระบุสถานะความเครียดของพืชในระยะเริ่มต้น

3.2.1.1 ออกแบบโครงสร้าง CNN ในงานวิจัยนี้จะใช้ Vivado HSL IP block ที่ถูกพัฒนาจากงานวิจัย [43] เพื่อใช้งาน CNN บน FPGA ซึ่งในงานวิจัย [43] ผู้วิจัยได้เสนอตัวอย่างการใช้งาน Framework ที่ออกแบบขึ้น 2 ตัวอย่างด้วยกัน ตัวอย่างแรกคือ LeNet-5 Model และชุดข้อมูลลายมือตัวเลข MNIST และ ตัวอย่างที่สองคือ CIFAR-10 Model [Caffe “quick” version] และชุดข้อมูล CIFAR-10 ซึ่งแต่ละ Model มีรายละเอียดโครงสร้างโมเดลดังตารางที่ 3

ตารางที่ 3 รายละเอียดโครงสร้างแบบจำลองตัวอย่างที่ถูกนำเสนอในงานวิจัย [43]

LeNet-5 Model	CIFAR-10 Model [Caffe “quick” version]
<pre>net = {} net['input'] = InputLayer(None, 1, 28, 28) net['conv1'] = ConvLayer(net['input'], num_filters=20, filter_size=5, nonlinearity=linear) net['pool1'] = PoolLayer(net['conv1'], pool_size=2, stride=2, mode='max', ignore_border=False) net['conv2'] = ConvLayer(net['pool1'], num_filters=50, filter_size=5, nonlinearity=linear) net['pool2'] = PoolLayer(net['conv2'], pool_size=2, stride=2, mode='max', ignore_border=False) net['ip1'] = DenseLayer(net['pool2'], num_units=500, nonlinearity = rectify) net['ip2'] = DenseLayer(net['ip1'], num_units=10, nonlinearity = None) net['prob'] = NonlinearityLayer(net['ip2'], softmax)</pre>	<pre>net = {} net['input'] = InputLayer(None, 3, 32, 32) net['conv1'] = ConvLayer(net['input'], num_filters=32, filter_size=5, pad=2, nonlinearity=None) net['pool1'] = PoolLayer(net['conv1'], pool_size=2, stride=2, mode='max', ignore_border=False) net['relu1'] = NonlinearityLayer(net['pool1'], rectify) net['conv2'] = ConvLayer(net['relu1'], num_filters=32, filter_size=5, pad=2, nonlinearity=rectify) net['pool2'] = PoolLayer(net['conv2'], pool_size=2, stride=2, mode='average_exc_pad', ignore_border=False) net['conv3'] = ConvLayer(net['pool2'], num_filters=64, filter_size=5, pad=2, nonlinearity=rectify) net['pool3'] = PoolLayer(net['conv3'], pool_size=2, stride=2, mode='average_exc_pad', ignore_border=False) net['ip1'] = DenseLayer(net['pool3'], num_units=64, nonlinearity = None) net['ip2'] = DenseLayer(net['ip1'], num_units=10, nonlinearity = None) net['prob'] = NonlinearityLayer(net['ip2'], softmax)</pre>

ซึ่งจากผลการทดลองในงานวิจัย [43] พบว่าในการใช้งานโมเดลทั้ง 2 นี้บนอุปกรณ์ PYNQ ใช้ทรัพยากรทั้งหมดดังตารางที่ 4



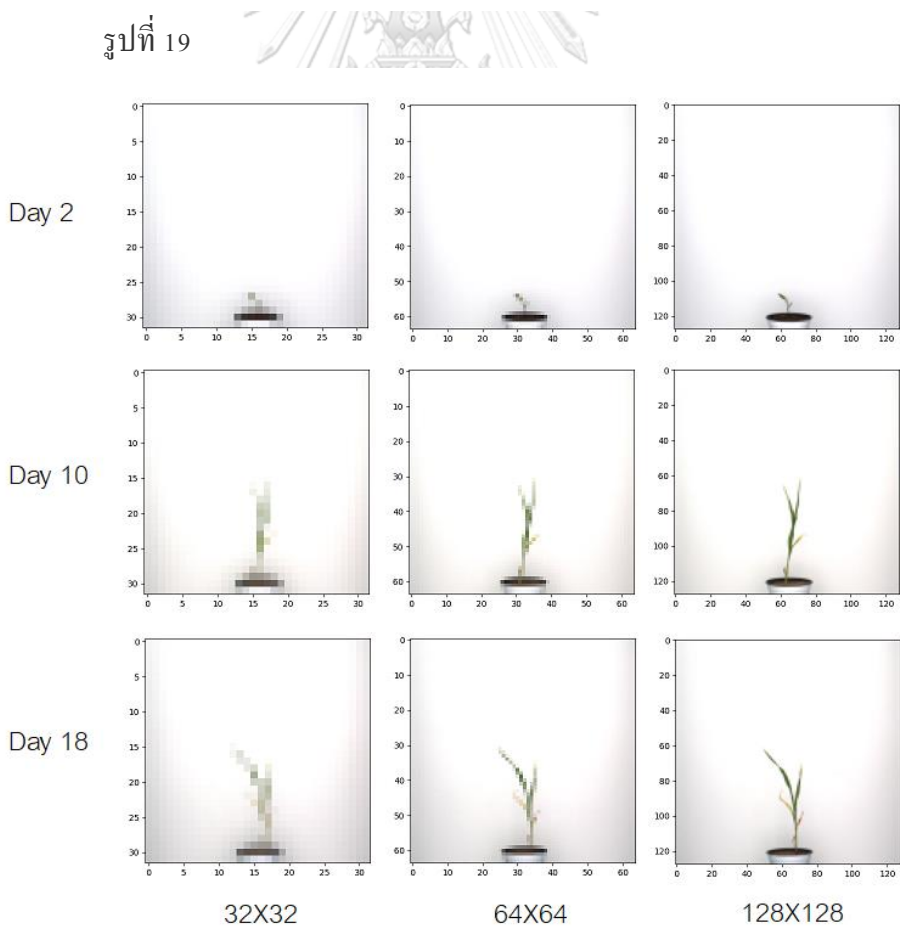
ตารางที่ 4 การใช้ทรัพยากร FPGA บนบอร์ด PYNQ ของโมเดล LeNet-5 และ CIFAR-10 [43]

ทรัพยากร FPGA	ทรัพยากรที่ใช้สำหรับใช้งานโมเดล [คิดเป็น % จากที่มีทั้งหมด]		ทรัพยากรที่มี PYNQ มี
	LeNet-5	CIFAR-10	
BRAM	139 [99%]	102 [73%]	140
DSP48E	130 [59%]	198 [90%]	220
LUT	38304 [72%]	43092 [81%]	53200

ซึ่งหากพิจารณาตามนี้ จะเห็นว่า ตัวบอร์ด PYNQ และ Framework ที่ถูกเสนอนั้น จะใช้งานได้เพียงเฉพาะ CNN โมเดลที่มีความซับซ้อนไม่มากเท่านั้น เพราะ ทั้ง



LeNet-5 และ CIFAR-10 นั้น ถูกจัดว่าเป็นโมเดลขนาดเล็กมากหากเทียบกับ State of the art model ที่มีการใช้งานกันอย่างแพร่หลายในปัจจุบัน ด้วยข้อจำกัดของ ทรัพยากร FPGA ของบอร์ด PYNQ ในการทดลองออกแบบ โครงสร้าง CNN ที่จะนำมาใช้ในการตรวจหาความเครียดของพืชในสภาวะเริ่มต้น พัฒนาโครงสร้าง โมเดลต่อจาก ตัว CIFAR-10 เป็นหลัก เนื่องด้วยเป็น โมเดลที่มีความซับซ้อน มากกว่า LeNet-5 โดยจะอยู่ในเงื่อนไขที่สามารถตรวจจับหาสภาวะความเครียดได้ และใช้ทรัพยากรเท่าที่มีอยู่ใน PYNQ โดยในขั้นต้นเมื่อพิจารณาความละเอียดของ ภาพที่จะใช้ พบความละเอียดภาพในชุดข้อมูลสูงถึง 2454 x 2056 Pixel ซึ่งเป็นไป ไม่ได้ที่จะใช้ในการฝึกบน PYNQ ผู้ทดลองจึงตัดและลดความละเอียดภาพให้อยู่ที่ 128x128 Pixel โดยใช้หลักการที่จำเป็นต้องใช้ภาพขนาดเล็กที่สุดที่เป็นไปได้ อ้างอิงกับตัวอย่างแบบจำลองที่ใช้กับชุดข้อมูล CIFAR-10 ซึ่งมีความละเอียดอยู่ที่ 32x32 Pixel แต่อย่างไรก็ตามชุดข้อมูลรูปภาพต้นพืชนั้นไม่สามารถที่จะใช้ที่ความ ละเอียด 32x32 Pixel ได้หากนำความละเอียดภาพที่ 32 64 128 Pixel มาเทียบดังใน รูปที่ 19



รูปที่ 19 ภาพต้นข้าวฟ่าง ในการทดลองวันที่ 2 10 และ 18 ที่ความละเอียด 32 64 และ 128 Pixel

ในรูปที่ 19 จะเห็นว่าที่ความละเอียด 32 และ 64 Pixel ต้นข้าจะสูญเสียรายละเอียดไปมาก ซึ่งอาจจะส่งผลกระทบต่อความแม่นยำของแบบจำลอง ในการทดลองจึงเลือกที่จะใช้ ภาพขนาด 128x128 Pixel เป็นชุดข้อมูลสำหรับฝึกแบบจำลอง

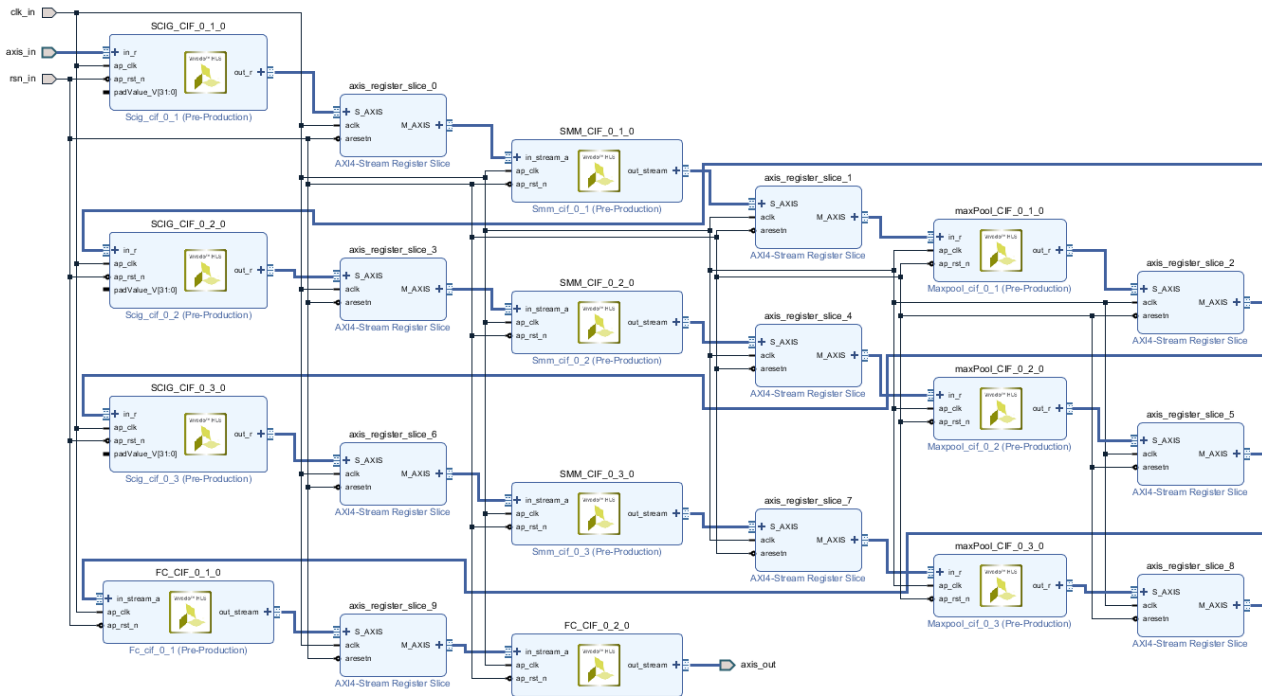
3.2.1.2 ทดลองปรับ โครงสร้างแบบจำลอง CIFA-10 ในการทดลองนี้จะเป็นการปรับโครงสร้างแบบจำลองโดยอ้างอิงตัวอย่างการใช้งานแบบจำลอง CIFA-10 ที่ใช้งานอยู่ในงานวิจัยที่ [43] ซึ่งมีโครงสร้างดังแสดงอยู่ในตารางที่ 3 โดย ใน Input Layer จะทำการเปลี่ยน จาก (3,32,32) เป็น (3,128,128) ซึ่งโครงสร้างที่ทำการปรับนั้นจะมีข้อจำกัดคือต้องใช้ทรัพยากรของ FPGA ไม่เกินกับที่บอร์ด PYNQ มี ในการทดลองนี้ได้เสนอโครงสร้างที่ทดลองปรับ 2 โครงสร้างดังตารางที่ 5

ตารางที่ 5 การปรับโครงสร้างแบบจำลองและการใช้ทรัพยากรของ FPGA

CIFA-10 [43]		ตัวเลือก 1		ตัวเลือก 2	
โครงสร้างแบบจำลอง	ทรัพยากรที่ใช้ [%]	โครงสร้างแบบจำลอง	ทรัพยากรที่ใช้ [%]	โครงสร้างแบบจำลอง	ทรัพยากรที่ใช้ [%]
Input : (3,32,32)	LUT : 81%	Input : (3,128,128)	LUT : 118%	Input : (3,128,128)	LUT : 89.98%
Cov_1 : (#K:32,K_Size:5,P:2)	LUTRAM : 20.7%	Cov_1 : (#K:32,K_Size:5,P:2)	LUTRAM : 32.4%	Cov_1 : (#K:32,K_Size:5,P:2)	LUTRAM : 24.83%
Pool_max : (size:2,S:2)	FF : 23.92%	Pool_max : (size:2,S:2)	FF : 25.41%	Pool_max : (size:4,S:4)	FF : 23.91%
Cov_2 : (#K:32,K_Size:5,P:2)	BRAM : 73%	Cov_2 : (#K:32,K_Size:5,P:2)	BRAM : 100%	Cov_2 : (#K:32,K_Size:5,P:2)	BRAM : 100%
Pool_max : (size:2,S:2)	DSP : 90%	Pool_max : (size:2,S:2)	DSP : 121%	Pool_max : (size:4,S:4)	DSP : 90.45%
Cov_3 : (#K:64,K_Size:5,P:2)	BUFG : 3.13%	Cov_3 : (#K:32,K_Size:5,P:2)	BUFG : 3.13%	Cov_3 : (#K:64,K_Size:5,P:2)	BUFG : 3.13%
Pool_max : (size:2,S:2)		Pool_max : (size:2,S:2)		Pool_max : (size:4,S:4)	
Dense : (64)		Cov_4 : (#K:64,K_Size:5,P:2)		Dense : (128)	
Dense : (10) [SoftMax]		Pool_max : (size:2,S:2)		Dense : (1)	
		Dense : (64)			
		Dense : (1)			

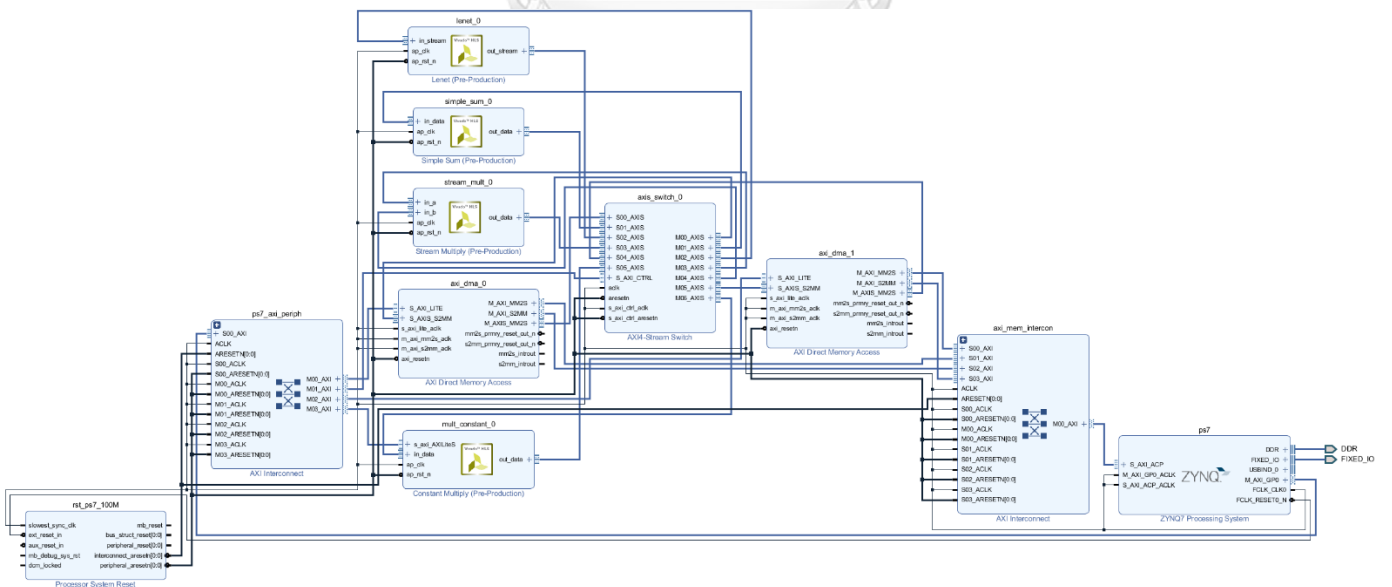
จากตารางที่ 5 จะเห็นว่า Artix-7 ซึ่งเป็นชิป FPGA ของ PYNQ นั้นมีทรัพยากรที่จำกัดอยู่มากทำให้ไม่สามารถใช้ โครงสร้าง CNN ที่มีความซับซ้อนสูงได้ ทั้งนี้หากเพิ่ม ชั้น Convolution เข้าไปจากโครงสร้างตัวอย่าง CIFA-10 ก็ทำให้ใช้ทรัพยากรเกินไปแล้ว ซึ่งในการทดลองนี้จึงเลือกทดลองโดยใช้โครงสร้างตัวเลือกที่ 2 ในการทดลอง

- 3.2.1.3 เตรียมข้อมูลเพื่อใช้ในการฝึกโมเดล ชุดข้อมูลที่จะนำมาใช้ในการทดลองนี้มีจำนวนทั้ง 15,246 ภาพ ซึ่งมาจากการถ่ายรูปต้นข้าวฟ่างทั้งหมด 847 ต้น เป็นเวลา 19 วัน โดยรูปภาพทั้งหมดจะมีการระบุเลขประจำครั้งที่ถ่าย ซึ่งสามารถนำเลขไปเชื่อมโยงกับข้อมูลต่างๆ อาทิ เช่น เลขประจำตัวต้นพืช วันที่ที่ถ่ายรูป กลุ่มทดลองของพืชต้นนั้นๆ น้ำหนักต้นพืช ณ วันที่ถ่าย สายพันธุ์ และ ข้อมูลต่างๆ โดยข้อมูลที่เราจะสนใจในการทดลองนี้เป็นหลักคือ วันถ่ายรูป และกลุ่มทดลองของพืชในรูปถ่าย ซึ่งกลุ่มทดลองจะระบุว่าพืชนั้นได้รับสารอาหารอยู่ในระดับไหนระหว่าง 100/100 50/10 หรือ 10/10 โดยวันที่ 1 ของการทดลองจะถือว่าเป็นวันแรกที่พืชแต่ละต้นได้รับสารอาหารที่แตกต่างกัน ในกลุ่มทดลองที่ได้รับสารอาหาร 100/100 ถือว่าเป็นกลุ่มควบคุมซึ่งไม่ควรที่จะแสดงอาการขาดสารอาหาร และในกลุ่มทดลอง 10/10 ควรจะแสดงอาการขาดสารอาหารมากที่สุดโดยการเตรียมข้อมูลสำหรับฝึกแบบจำลอง CNN จะแบ่งข้อมูลทั้งหมดเป็น 18 ชุด ตามวันที่ 2-19 และใช้แต่ละชุดในการฝึกแบบจำลอง โดยแต่ละครั้งในการฝึกจะแบ่งชุดข้อมูลแต่ละชุดในอัตราส่วน Train : 70 % Validation : 15 % และ Test : 15 % รูปทุกรูปที่นำมาใช้ในการทดลองจะถูกตัดภาพและย่อให้อยู่ในอัตราส่วน 128x128 pixels
- 3.2.1.4 ฝึกแบบจำลองทั้งหมดด้วยชุดข้อมูลทั้ง 18 ชุด โดยการฝึกโมเดลนั้นจะใช้ Theano และ Lasagne ในการสร้างโมเดลและฝึกบนคอมพิวเตอร์ที่ติดตั้งระบบปฏิบัติการ Ubuntu 18.04 ก่อน ซึ่งเมื่อปรับ Hyper parameters ต่างๆ และทดสอบจนได้ Accuracy ของ โมเดลที่พอใจแล้ว แบบจำลองที่ผ่านการฝึกจะถูกนำเอาค่า Weights และ Bias ไปใช้ในแบบจำลองที่สร้างอยู่บน PYNQ
- 3.2.1.5 สร้าง Vivado IP ของแต่ละชั้นของแบบจำลอง CNN เพื่อนำมาใช้ในการระบุสถานะความเครียดของพืชในระยะเริ่มต้น บน PYNQ และรวม IP ทั้งหมดเพื่อสร้าง IP ใหม่สำหรับแบบจำลอง CNN



รูปที่ 20 โครงสร้าง CNN ที่ออกแบบด้วย Vivado IP แต่ละชั้น

### 3.2.1.6 แก้ไขและเปลี่ยน Overlay ของ PYNQ-Z1 และสร้างไฟล์ Bitstream ใหม่ที่สามารถติดตั้งแบบจำลอง CNN บน FPGA ได้



รูปที่ 21 Overlay ที่แก้ไขและเพิ่ม CNN IP เข้าไป

3.2.1.7 ทดลองใช้แบบจำลอง CNN ที่ฝึกด้วยชุดข้อมูลทั้ง 18 ชุด บน PYNQ-Z1 เปรียบเทียบกันระหว่างการใช้แบบจำลองบน Dual-core ARM Cortex-A9 ใน ส่วน Processing System ของชิป Zynq-7020 บนบอร์ด PYNQ-Z1 และ FPGA โดยจะใช้ Accuracy เป็น Metric ในการวัดประสิทธิภาพของ แบบจำลอง โดย Accuracy สามารถนิยามได้ดังต่อไปนี้

$$ACC = \frac{(tp + tn)}{(tp + fp + fn + tn)}$$

tp คือ True Positive มีความหมายว่าสิ่งที่ทำนาย ตรงกับสิ่งที่เกิดขึ้นจริง ในกรณีทำนายว่าต้นพืชไม่เครียด และต้นพืชไม่อยู่ในสภาวะเครียดจริง

tn คือ True Negative มีความหมายว่าสิ่งที่ทำนาย ตรงกับสิ่งที่เกิดขึ้นจริง ในกรณีทำนายว่าต้นพืชเครียด และต้นพืชอยู่ในสภาวะเครียดจริง

fp คือ False Positive มีความหมายว่าสิ่งที่ทำนาย ไม่ตรงกับสิ่งที่เกิดขึ้นจริง ในกรณีทำนายว่าต้นพืชไม่เครียด แต่ต้นพืชอยู่ในสภาวะเครียด

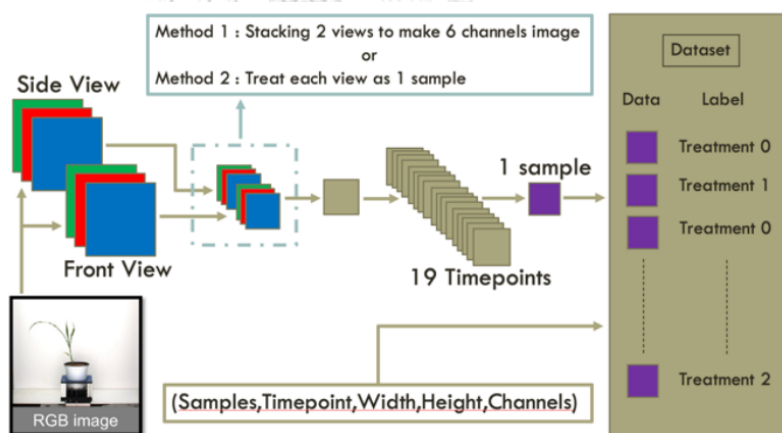
fn คือ False Negative มีความหมายว่าสิ่งที่ทำนาย ไม่ตรงกับสิ่งที่เกิดขึ้นจริง ในกรณีทำนายว่าต้นพืชเครียด แต่ต้นพืชอยู่ในสภาวะไม่เครียด

3.2.2 ทดลองใช้แบบจำลอง CNN+LSTM เพื่อประสิทธิภาพในการระบุสถานะ ความเครียดของข้าวฟ่างที่ขาดสารอาหาร

3.2.1.1 เตรียมข้อมูลรูปภาพให้เป็น Temporal Dataset เพื่อใช้ในการฝึกแบบจำลอง CNN+LSTM โดยการเตรียม Temporal dataset จะเป็นการนำรูปภาพของต้นพืช ต้นเดียวกันที่ถ่ายมาตั้งแต่วันที่ 2 – 19 มาเรียงต่อกัน ซึ่งสุดท้ายจะได้ข้อมูลในการ ฝึกแบบจำลองขนาด (จำนวนตัวอย่าง, จำนวนวันที่ถ่ายภาพ, กว้าง, ยาว, Channel ของภาพ) โดยหากรูปภาพมีการถ่ายหลายมุมมอง สามารถที่จะนำแต่ละมุมมองมา ซ้อนกันในมิติของ Channel ได้ โดยในชุดข้อมูลนี้จะใช้ 2 มุมมองคือ ด้านหน้า และด้านข้าง ซึ่งกระบวนการการเตรียมข้อมูลถูกแสดงในรูปที่ 21 โดยชุดข้อมูลที่ เตรียมจะมีทั้งหมด 18 ชุด ดังต่อไปนี้

- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 2 ของการทดลอง ขนาด (847,2,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 3 ของการทดลอง ขนาด (847,3,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 4 ของการทดลอง ขนาด (847,4,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 5 ของการทดลอง ขนาด (847,5,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 6 ของการทดลอง ขนาด (847,6,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 7 ของการทดลอง ขนาด (847,7,512,512,6)

- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 8 ของการทดลอง ขนาด (847,8,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 9 ของการทดลอง ขนาด (847,9,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 10 ของการทดลอง ขนาด (847,10,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 11 ของการทดลอง ขนาด (847,11,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 12 ของการทดลอง ขนาด (847,12,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 13 ของการทดลอง ขนาด (847,13,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 14 ของการทดลอง ขนาด (847,14,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 15 ของการทดลอง ขนาด (847,15,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 16 ของการทดลอง ขนาด (847,16,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 17 ของการทดลอง ขนาด (847,17,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 18 ของการทดลอง ขนาด (847,18,512,512,6)
- ชุดข้อมูลรูปถ่ายของข้าวของวันที่ 1 ถึง 19 ของการทดลอง ขนาด (847,19,512,512,6)



รูปที่ 22 ขั้นตอนในการเตรียมชุดข้อมูลสำหรับการฝึกแบบจำลอง CNN+LSTM

3.2.1.2 ทดลองฝึกแบบจำลอง CNN+LSTM โดยใช้ VGG16 เป็นส่วน CNN และใช้ Transfer learning จาก ชุดข้อมูล ImageNet โดยจะฝึกเฉพาะ Layer สุดท้ายของ VGG16 และส่วนของ LSTM โดยการฝึกจะใช้ชุดข้อมูล Temporal dataset ทั้ง 18 ชุด ซึ่งสาเหตุที่เลือกใช้ Transfer learning แทนที่การฝึกแบบจำลองใหม่ทั้งหมด เพราะ จำนวนข้อมูลที่ใช้ฝึกที่มีอยู่ไม่เพียงพอต่อการฝึกแบบจำลองขนาดใหญ่ และ ทรัพยากรของคอมพิวเตอร์ที่ใช้ฝึกนั้นไม่เพียงพอต่อการฝึกโมเดล

ขนาดใหญ่ทั้งหมด ซึ่งการฝึกโดยข้อมูลที่ไม่มีเพียงพอนั้นจะส่งผลให้แบบจำลองมีความแม่นยำในการทำนายต่ำ

แบบจำลองที่เสนอมีหน้าที่รับข้อมูลมาทีละตัวอย่างและทายกลุ่มของต้นพืชว่าอยู่ในกลุ่มไหน ซึ่งในการทดลองนี้จะขอทายเพียง 2 กลุ่มคือ กลุ่ม 100/100 และ กลุ่ม 10/10 โดยค่าความแม่นยำคือ Accuracy ของการทายกลุ่มของพืชที่ถูกทดลอง ซึ่งหาก แบบจำลองมี Accuracy สูงในชุดข้อมูลวันต้นๆ ก็จะหมายความว่า แบบจำลองสามารถระบุสถานะเจริญของต้นข้าวฟ้างจากการขาดสารอาหารในสถานะเริ่มต้นได้ดี โดยโครงสร้างของ VGG16+LSTM มีหน้าตา ดังต่อไปนี้

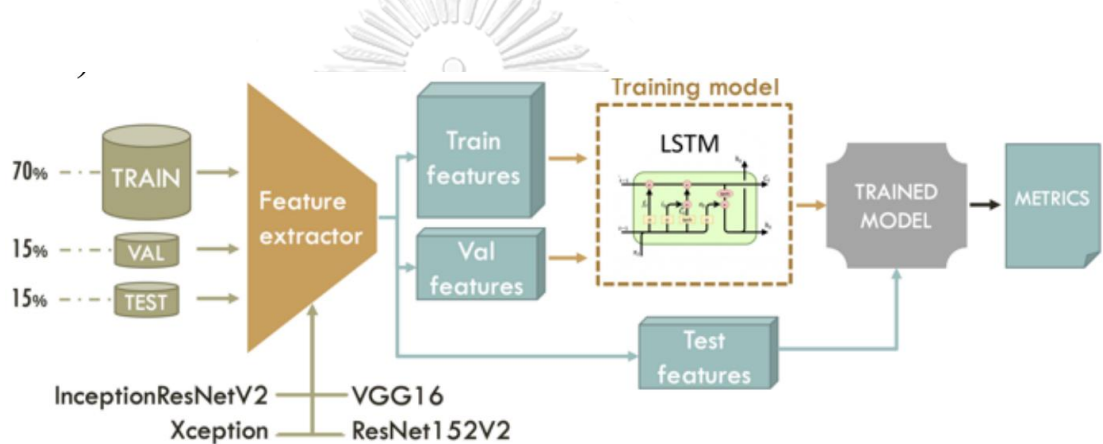
```
lstm_input = layers.Input(input_shape=(16*16*512))//Size of VGG16 Last layer
cnn_base = vgg16.VGG16(input_shape=(img_size,img_size,3),weights="imagenet",include_top=False)
cnn_out = layers.Conv1D(128,3,padding='same',activation='relu')(cnn_base.output)
cnn = tf.keras.Model(cnn_base.input, cnn_out)
cnn.trainable = False
final_model = layers.TimeDistributed(cnn)(lstm_input)
final_model = layers.LSTM(100)(final_model)
final_model = layers.Dense(64, activation="relu")(final_model)
final_model = layers.Dense(32, activation="relu")(final_model)
final_model = layers.Dense(1, activation="softmax")(final_model)
model = tf.keras.Model([lstm_input], final_model)
optimizer = Nadam(lr=0.002,
                  beta_1=0.9,
                  beta_2=0.999,
                  epsilon=1e-08,
                  schedule_decay=0.004)
model.compile(loss="binary_crossentropy",
              optimizer=optimizer,
              metrics=["accuracy"])
```

โดยการฝึกแบบจำลองนี้จะใช้ Adam เป็น Optimizer และมี Learning Rate เริ่มต้นเท่ากับ 0.002 และจะลดลงด้วย Schedule\_Decay = 0.004 ซึ่งการใช้งาน โคตรงสร้าง VGG16+LSTM ที่ถูกเสนอนั้นจะแบ่งเป็น 2 ส่วนหลักๆ คือ VGG16 มีหน้าที่เป็น Feature Extractor ซึ่งจะทำหน้าที่ลดมิติของข้อมูลเพื่อให้เหลือเฉพาะข้อมูลที่ประกอบด้วยคุณลักษณะเฉพาะของข้อมูลนั้นๆ ให้มากที่สุดเท่านั้น ซึ่งข้อมูลที่ถูกลดมิติในลักษณะนี้ในบางงานวิจัยจะเรียกมิติของข้อมูลที่ถูกลดมิติว่า Latent Space โดยหลังจากที่ข้อมูลถูกลดแล้วจะถูกส่งไปยัง ส่วนที่ 2 ซึ่งก็คือ ส่วนของ LSTM Layer เพื่อการฝึกแบบจำลองต่อไป ในการเลือก Pretrained

model เพื่อนำมาใช้เป็น Feature Extractor นั้น ได้มีการเลือกโมเดลมาทดลอง ทั้งสิ้น 4 โมเดล คือ

- VGG16
- InceptionResNetV2
- Xception
- ResNet152V2

แต่เนื่องจากขนาดของ Out layer ของแต่ละแบบจำลอง และข้อจำกัดของ ทรัพยากรคอมพิวเตอร์ [RTX 4000] ทำให้สามารถใช้ VGG16 เป็น Feature Extractor ได้เพียงแบบจำลองเดียว



รูปที่ 23 ภาพรวมขั้นตอนการฝึกแบบจำลอง CNN+ LSTM

นอกจากแบบจำลองที่เสนอ การทดลองนี้จะฝึกแบบจำลอง CNN ตัวอื่นๆที่เป็น State of the art ที่นิยมใช้กันในปัจจุบัน เพื่อนำมาเปรียบเทียบ โดยการฝึกแบบจำลอง CNN ที่นำมาเปรียบเทียบกับแบบจำลองที่เสนอจะฝึกโดยใช้ชุดข้อมูลเดียวกับที่ใช้ในการฝึกแบบจำลอง CNN ที่ใช้นบอร์ด PYNQ แบบจำลองที่นำมาเปรียบเทียบมีดังต่อไปนี้

- VGG16
- ResNet50V2
- GoogLeNet
- DenseNet121

ทั้งหมดเป็น Pretrained model จากชุดข้อมูล ImageNet ทั้งสิ้น ซึ่งในการฝึกจะใช้วิธี Transfer learning กับชุดข้อมูลที่เตรียมไว้ทั้ง 18 ชุด

### 3.2.3 ทดสอบ โมเดลทั้งหมดกับชุดข้อมูล Test เปรียบเทียบผลการทดลอง



### 3.3 อุปกรณ์ในการทดลอง

#### 3.3.1 PYNQ-Z1 1 บอร์ด

- ชิป ZYNQ XC7Z020-1CLG400C
  - 650MHz dual-core Cortex-A9 processor
  - DDR3 memory controller with 8 DMA channels and 4 high performance AXI3 slave ports
  - High-bandwidth peripheral controllers: 1G Ethernet, USB 2.0, SDIO
  - Low-bandwidth peripheral controller: SPI, UART, CAN, I2C
  - Programmable from JTAG, Quad-SPI flash, and microSD card
  - Artix-7 family programmable logic
    - 13,300 logic slices, each with four 6-input LUTs and 8 flip-flops
    - 630 KB of fast block RAM
    - 4 clock management tiles, each with a phase-locked loop (PLL) and mixed-mode clock manager (MMCM)
    - 220 DSP slices
    - On-chip analog-to-digital converter (XADC)
- หน่วยความจำ
  - 512MB DDR3 with 16-bit bus @ 1050Mbps
  - 16MB Quad-SPI Flash
  - MicroSD slot
- USB และ Ethernet
  - USB-JTAG Programming circuitry
  - USB-UART bridge
  - USB OTG PHY (supports host only)
  - Gigabit Ethernet PHY
- Audio and Video
  - Electret microphone with pulse density modulated (PDM) output

- 3.5mm mono audio output jack, pulse-width modulated (PWM) format
    - HDMI sink port (input)
    - HDMI source port (output)
  - Switches, push-buttons, and LEDs
    - 4 push-buttons
    - 2 slide switches
    - 4 LEDs
    - 2 RGB LEDs
- 3.3.2 คอมพิวเตอร์ตั้งโต๊ะ 1 เครื่อง [Workstation]
  - AMD Ryzen 9 3900X
  - RAM 64 GB
  - HDD 8 TB, SSD 2 TB
  - NVIDIA Quadro RTX 4000 GPU.

### 3.4 โปรแกรมในการทดลอง

- 3.4.1 Ubuntu 18.04
- 3.4.2 Jupyter notebook ติดตั้งอยู่บน Ubuntu OS
- 3.4.3 Theano Python Library
- 3.4.4 Lasagne Python Library
- 3.4.5 TensorFlow library V2.0
- 3.4.6 NVIDIA's CUDA toolkit
- 3.4.7 OpenCV2
- 3.4.8 Docker
- 3.4.9 Vivado 2016
- 3.4.10 Vivado HLS
- 3.4.11 Visual Studio Code

## บทที่ 4 ผลการทดลอง

การทดลองทั้งหมดของงานวิจัยนี้ถูกแบ่งออกเป็น 2 ส่วนหลักๆ คือการทดลองแบบจำลองเพื่อระบุสถานะความเครียดของพืชในระยะเริ่มต้นบน FPGA ในบอร์ด PYNQ และการทดลองแบบจำลองเพื่อระบุสถานะความเครียดของพืชในระยะเริ่มต้น โดยใช้แบบจำลอง CNN+LSTM ซึ่งในส่วนนี้ ผลการทดลองจะถูกนำเสนอเป็น 2 ส่วน

### 4.1 ผลการทดลองการใช้งานแบบจำลองระบุสถานะความเครียดสถานะเริ่มต้นของต้นข้าวฟ่างบน PYNQ

จากการใช้แบบจำลองตัวเลือกที่ 2 ซึ่งมีโครงสร้างดังต่อไปนี้

- Input : (3,128,128)
- Cov\_1: (#Kernel:32, K\_Size:5, P:2)
- Pool\_max: (size:4, S:4)
- Cov\_2: (#Kernel:32, K\_Size:5, P:2)
- Pool\_max: (size:4, S:4)
- Cov\_3: (#Kernel:64, K\_Size:5, P:2)
- Pool\_max: (size:2, S:2)
- Dense: (128)
- Dense: (1)

ในการฝึกกับข้อมูลทั้ง 18 ชุด แบบจำลองใช้เวลาในการตั้งตารางที่ 6

ตารางที่ 6 เวลาที่ใช้ในการฝึกแบบจำลองกับชุดข้อมูลทั้ง 18 ชุด

Day 2	45.94 s	Day 8	385.98 s	Day 14	259.13 s
Day 3	88.37 s	Day 9	226.73 s	Day 15	325.43 s
Day 4	108.04 s	Day 10	246.6 s	Day 16	453.22 s
Day 5	103.7 s	Day 11	344.13 s	Day 17	371.2 s
Day 6	204.88 s	Day 12	291.64 s	Day 18	392.91 s
Day 7	169.18 s	Day 13	282.01 s	Day 19	352.04 s

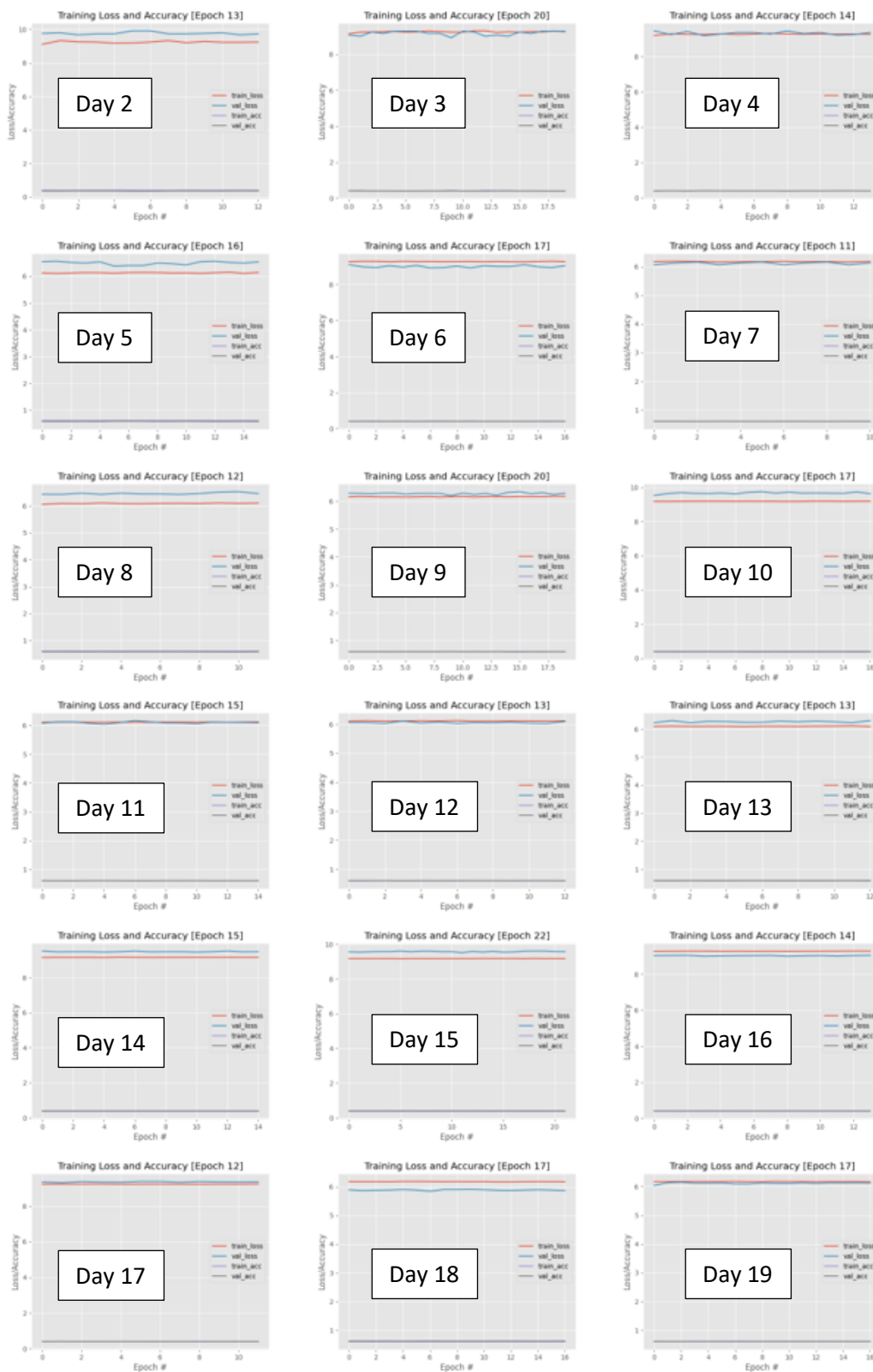
จากตารางที่ 6 จะเห็นว่าเวลาในการฝึกแบบจำลองจะเพิ่มขึ้นเมื่อจำนวนข้อมูลรูปภาพเพิ่มขึ้นแต่จะมีบางช่วงของการฝึกที่ใช้เวลามากกว่าที่ควรจะเป็นเช่น ในวันที่ 6 หรือ 8 ซึ่งเกิดจากตัวโปรแกรมที่

ใช้ฝึกเกิดค้างในบางช่วงขณะฝึก ทำให้เวลารวมเพิ่มขึ้น และหลังจากฝึกแบบจำลองแล้วได้นำค่า Weights ที่ฝึกแล้วมาใช้ทดสอบกับชุดข้อมูล Test ซึ่งได้ผลการทำนายกับ Test data บน PYNQ ดังในตารางที่ 7

ตารางที่ 7 ค่าความแม่นยำในการทำนาย (Accuracy) และค่าเฉลี่ยเวลาในการทำนาย Test Data ด้วยอุปกรณ์ต่างๆ

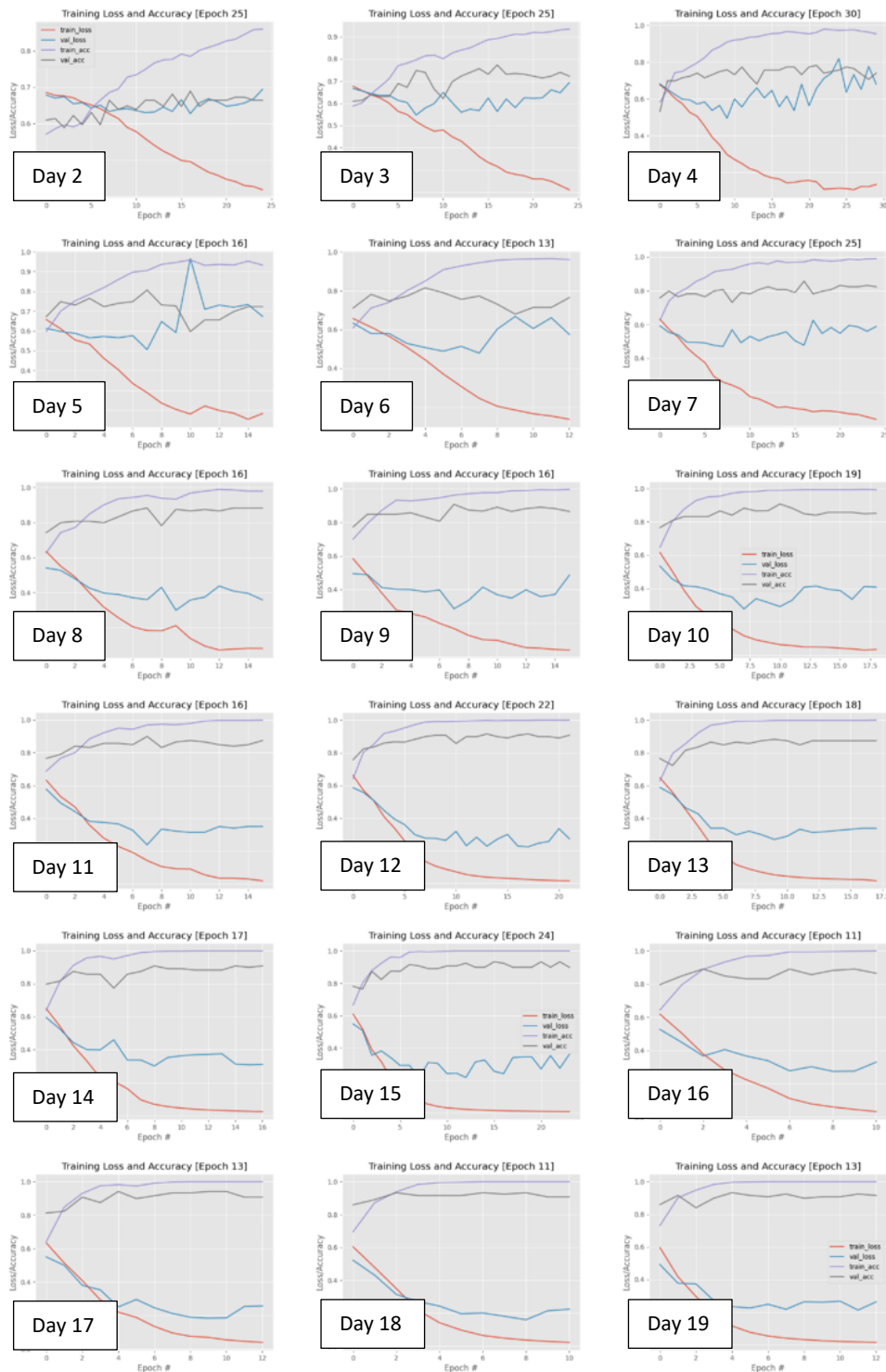
Day	บน Workstation ด้วย GPU RTX 4000		Dual-core ARM Cortex-A9 [PYNQ]		FPGA [PYNQ]	
	Accuracy	ค่าเฉลี่ยเวลาในการทำนาย ต่อ รูป	Accuracy	ค่าเฉลี่ยเวลาในการทำนาย ต่อ รูป	Accuracy	ค่าเฉลี่ยเวลาในการทำนาย ต่อ รูป
Day 2	0.43	5.19 ms	0.43	452.9 ms	0.43	11.73 ms
Day 3	0.40					
Day 4	0.41					
Day 5	0.63					
Day 6	0.39					
Day 7	0.61					
Day 8	0.61					
Day 9	0.62					
Day 10	0.40					
Day 11	0.59					
Day 12	0.59					
Day 13	0.60					
Day 14	0.39					
Day 15	0.40					
Day 16	0.39					
Day 17	0.40					
Day 18	0.60					
Day 19	0.61					

โดยผลการทดลองในตารางที่ 7 มาจากการใช้งานแบบจำลองตัวเลือกที่ 2 และข้อมูล Test ทั้ง 18 ชุด บน เครื่อง Workstation และบนบอร์ด PYNQ ทั้งฝั่ง CPU [Cortex-A9] และ FPGA



รูปที่ 24 ค่า Accuracy/Loss ของการฝึกแบบจำลองตัวเลือก 2 เพื่อใช้กับ FPGA เทียบ Epoch โดยเส้นสีแดง Train\_Loss น้ำเงิน คือ Val\_Loss ม่วง คือ Train\_Acc และ เทาเข้ม คือ Val\_Acc

## 4.2 ผลการทดลองการใช้แบบจำลอง CNN+ LSTM ในการระบุสถานะความเครียดระยะเริ่มต้นของต้นข้าวฟ่าง

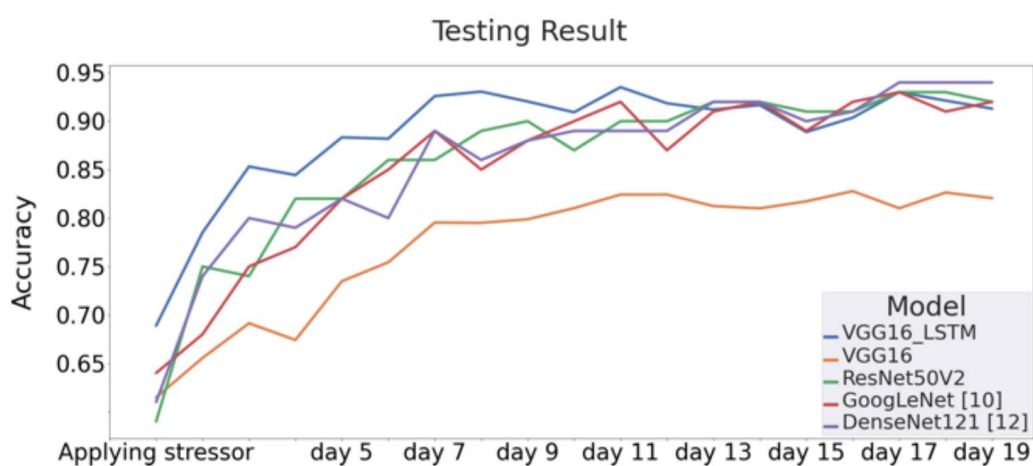


รูปที่ 25 ค่า Accuracy/Loss ของการฝึกแบบจำลอง VGG16+LSTM เทียบ Epoch โดยเส้นสีแดง Train\_Loss น้ำเงิน คือ Val\_Loss ม่วง คือ Train\_Acc และ เทาเข้ม คือ Val\_Acc

ในรูปที่ 25 จะแสดงถึงค่า Loss และ Accuracy ของการฝึก CNN+LSTM แต่ละ Epoch ซึ่ง

- เส้นสีแดงจะหมายถึง ค่า Loss ของแบบจำลองกับชุดข้อมูลฝึก Train
- เส้นสีน้ำเงินจะหมายถึง ค่า Accuracy ของแบบจำลองกับชุดข้อมูลฝึก Train
- เส้นสีม่วงจะหมายถึง ค่า Accuracy ของแบบจำลองกับชุดข้อมูลฝึก Validation
- เส้นสีเทาเข้มจะหมายถึง ค่า Loss ของแบบจำลองกับชุดข้อมูลฝึก Validation

Accuracy ของแบบจำลอง VGG16+LSTM, VGG16, ResNet50V2, GoogLeNet และ DenseNet121 ที่ถูกฝึกและทดสอบกับชุดข้อมูลข้าวฟ่าง จะถูกพล็อตกราฟเทียบกับวันที่ทำการทดลองในรูปที่ 26



รูปที่ 26 ค่าความแม่นยำในการทำนายสถานะเครียดของข้าวฟ่างเทียบกับวันที่ในการทดลอง

ตารางที่ 8 ค่าความแม่นยำในการระบุสถานะเครียดบนข้อมูลทดสอบของแบบจำลองต่างๆ

day	VGG16+LSTM	VGG16	ResNet50V2	GoogLeNet	DenseNet121
1	0.69	0.61	0.59	0.64	0.61
2	0.79	0.66	0.75	0.68	0.74
3	0.85	0.69	0.74	0.75	0.8
4	0.84	0.67	0.82	0.77	0.79
5	0.88	0.73	0.82	0.82	0.82
6	0.88	0.75	0.86	0.85	0.8
7	0.93	0.80	0.86	0.89	0.89
8	0.93	0.80	0.89	0.85	0.86
9	0.92	0.80	0.9	0.88	0.88
10	0.91	0.81	0.87	0.9	0.89
11	0.94	0.82	0.9	0.92	0.89
12	0.92	0.82	0.9	0.87	0.89
13	0.91	0.81	0.92	0.91	0.92
14	0.92	0.81	0.92	0.92	0.92

15	0.89	0.82	0.91	0.89	0.9
16	0.90	0.83	0.91	0.92	0.91
17	0.93	0.81	0.93	0.93	0.94
18	0.92	0.83	0.93	0.91	0.94
19	0.91	0.82	0.92	0.92	0.94

จากรูปที่ 26 จะเห็นว่า VGG16+LSTM เส้นสีน้ำเงินจะสามารถทำนายสภาวะความเครียดของต้นข้าวฟ่างที่ขาดสารอาหาร ได้ค่า Accuracy ที่ 85.32% ที่วันที่ 3 หรือ 2 วันหลังจากต้นข้าวฟ่างเริ่มขาดสารอาหาร และหลังจากนี้ค่า Accuracy ของแบบจำลองที่เสนอจะเพิ่มขึ้นเรื่อยๆเมื่อแบบจำลองได้รับข้อมูลรูปภาพเพิ่มขึ้น จนไปสูงสุดที่วันที่ 8 สำหรับแบบจำลองที่นำมาเปรียบเทียบกับอื่นๆ ถึงแม้ในช่วงต้นแบบจำลองจะทำค่า Accuracy ได้ต่ำกว่าแบบจำลองที่เสนอ แต่เมื่อได้รับข้อมูลที่มากเพียงพอก็จะสามารถทำค่า Accuracy ได้สูงกว่า 80% ทั้งหมด โดยเวลาเฉลี่ยที่ใช้ในการคำนวณ บน CPU มีค่าเท่ากับ 13.36 sec/Step สำหรับแบบจำลอง CNN ที่นำมาเปรียบเทียบ และ 1.72 sec/step สำหรับ VGG16+LSTM และ บน GPU ใช้เวลาคำนวณเฉลี่ยอยู่ที่ 1.67 sec/step สำหรับแบบจำลองที่นำมาเปรียบเทียบ และ 0.46sec/step สำหรับแบบจำลอง VGG16+LSTM โดยที่ 1 Step มีค่าเท่ากับ 1 Batch ซึ่งมีค่าเท่ากับ 32 รูปภาพ สำหรับแบบจำลอง CNN ที่นำมาเปรียบเทียบ และมีค่าเท่ากับ 16 รูปภาพสำหรับแบบจำลอง VGG16+LSTM โดยในตารางที่ 8 เป็น รายละเอียดค่าความแม่นยำของแบบจำลอง VGG16+LSTM บนข้อมูล Temporal Test Data และแบบจำลอง CNN อื่นๆที่นำมาเปรียบเทียบ บนข้อมูลทดสอบ และตารางที่ 9 แสดงเวลาที่ใช้ในการฝึกและทดสอบแบบจำลอง VGG16+LSTM เปรียบเทียบกับเวลาที่ใช้ในการฝึกแบบจำลอง VGG16 ตารางที่ 9 แสดงเวลาที่ใช้ในการฝึกแบบจำลอง VGG16+LSTM และ VGG16 และเวลาเฉลี่ยในการทำนายสภาวะเครียดของแบบจำลอง VGG16+LSTM และ แบบจำลองเปรียบเทียบกับอื่นๆ (Baseline Model)



ตารางที่ 9 เวลาในการฝึกแบบจำลอง และเวลาเฉลี่ยในการทดสอบของแบบจำลอง VGG16+LSTM และแบบจำลอง Baseline

Timepoints	Average training time (h:min:sec)		Timepoints	Average training time (h:min:sec)		Average testing time per sample (millisec)	
	VGG16 LSTM	VGG16		VGG16 LSTM	VGG16	Model	Mean $\pm$ SD
1	0:21:44	0:2:24	11	0:24:50	6:32:32	VGG16 LSTM	185.25 $\pm$ 6.38
2	0:21:20	0:13:57	12	0:23:38	4:43:30		
3	0:22:53	0:31:2	13	0:25:13	4:47:32		
4	0:21:2	1:31:12	14	0:26:12	7:54:57		
5	0:20:55	1:36:32	15	0:27:38	7:13:20		
6	0:22:56	3:3:36	16	0:30:41	7:44:10		
7	0:26:46	3:53:20	17	0:25:32	7:20:44	Baseline	8.94 $\pm$ 0.86
8	0:22:30	5:50:16	18	0:32:8	5:17:30		
9	0:22:44	4:38:12	19	0:29:1	5:22:56		
10	0:23:21	4:49:6	Only Feature Extraction	0:18:32			



## บทที่ 5 อภิปรายผลการทดลอง

จากผลการทดลองใช้งานแบบจำลอง CNN เพื่อใช้ระบุสถานะความเครียดระยะเริ่มต้น บน FPGA โดยใช้บอร์ด PYNQ จะเห็นว่า จากตารางที่ 7 ค่าความแม่นยำ [Accuracy] ในการทำนายของแบบจำลอง CNN นั้น ต่ำมาก ในทุกช่วงของชุดข้อมูลทดลองตั้งแต่วันที่ 2 – 19 และไม่สามารถนำไปใช้งานจริงได้ซึ่งค่าความแม่นยำสูงสุดที่ได้นั้นอยู่ที่ 0.61 บน FPGA และ 0.62 บน CPU ซึ่งสาเหตุหลักที่ทำให้ค่าความแม่นยำน้อย มาจากข้อจำกัดทางด้านทรัพยากรของบอร์ด PYNQ และ Framework ที่ใช้ในการใช้งาน CNN บนบอร์ด PYNQ ที่ทำให้ไม่สามารถใช้งานแบบจำลองที่มีความซับซ้อนสูงบน FPGA ได้ โดยหากพิจารณาจากตารางที่ 5 จะพบว่า การเพิ่มขึ้น Convolution Layer เข้าไปที่แบบจำลองตัวอย่าง CIFAR-10 เพียงชั้นเดียว ก็ทำให้การใช้ทรัพยากร LUT [118%] และ DSP [112%] เกินจากที่ FPGA บนบอร์ด PYNQ มีแล้ว ซึ่งหากดูจากการเรียนรู้ของแบบจำลองดังรูปที่ 24 จะเห็นว่าในการฝึกแบบจำลอง CNN ตัวเลือกที่ 2 ที่ฝึกกับชุดข้อมูลทั้ง 18 ชุดนั้น ค่าความแม่นยำ (Accuracy) และ ค่า Loss ของการฝึกแทบจะไม่ลดลงเลยเมื่อจำนวน Epoch เพิ่มในทุกๆชุดข้อมูล ซึ่งบ่งบอกได้ว่าแบบจำลองไม่สามารถเรียนรู้คุณลักษณะของข้อมูลได้ หรือที่เรียกว่า Underfit ซึ่งวิธีแก้ไขคือต้องเพิ่มจำนวนชั้นของแบบจำลองหรือเพิ่มจำนวน Weights ของแต่ละชั้นซึ่งเป็นไปได้ในบอร์ด PYNQ เนื่องจากข้อจำกัดทางด้านทรัพยากร หากพิจารณาเพิ่มเติมถึงค่าความแม่นยำ Accuracy ในตารางที่ 7 จะเห็นว่าค่าความแม่นยำ Accuracy ของแบบจำลองที่ใช้งานบน GPU และ CPU ARM Cortex-A9 นั้นมีค่าเท่ากัน แต่บางชุดข้อมูลที่นำไปใช้งานใน FPGA ของบอร์ด PYNQ นั้นมีค่าไม่เท่าซึ่ง เหตุผลหนึ่งที่เป็นไปได้คือการใช้งาน CNN บน FPGA โดย Framework ในงานวิจัย [43] มีการใช้ FIXED-POINT DATA QUANTISATION ซึ่งจะทำให้ค่าต่างๆคลาดเคลื่อนไปในการคำนวณและอาจจะส่งผลให้มีการลดลงของค่า Accuracy อย่างไรก็ตามหากพิจารณาถึงความเร็วในการทำนายแล้วจะพบว่าการทำนายโดยแบบจำลอง CNN ซึ่งใช้งานอยู่บน FPGA นั้นมีความเร็วสูงมาก โดยใช้เวลาในการประมวลผลแต่ละภาพอยู่ที่ 11.73 ms ซึ่งแม้จะช้ากว่าการทำนายผลโดยใช้ GPU Nvidia Quadro RTX 4000 ถึง 2 เท่า แต่ก็เร็วกว่า CPU Dual-core ARM Cortex-A9 บนบอร์ด PYNQ ถึง 38 เท่า ซึ่งหากเทียบว่าเป็นการใช้งานในระบบฝังตัวถือว่า FPGA สามารถประมวลผลได้เร็วกว่าการใช้ CPU ในระบบฝังตัวเป็นอย่างมาก

ด้วยเหตุผลที่ไม่สามารถทำแบบจำลองที่สามารถใช้งานได้จริงบน FPGA ในงานทดลองนี้ จึงมีการทำการทดลองอีกส่วนซึ่งก็คือ การใช้แบบจำลอง CNN ขนาดใหญ่ในการระบุสถานะความเครียดของต้นข้าวฟ่างในสภาวะเริ่มต้น โดยจากรูปที่ 25 จะเห็นว่าในการฝึกแบบจำลองที่นำเสนอซึ่งก็คือ VGG16+LSTM ในทุกชุดข้อมูลที่ใช้ฝึกทั้ง 18 ชุด แบบจำลองที่นำเสนอสามารถ

เรียนรู้ข้อมูลจากรูปภาพได้ทั้งสิ้น โดยดูจากค่าความแม่นยำในการทำนายที่สูงขึ้น และค่า Loss ที่ลดลง เมื่อจำนวน Epoch ในการฝึกมากขึ้น แต่อย่างไรก็ตามในช่วง 2 – 6 วันแรกนั้นค่าความแม่นยำและค่า Loss ในชุดข้อมูล Validation จะยังไม่สามารถลู่เข้าหาค่าความแม่นยำและค่า Loss ของชุดข้อมูล Train ได้ เนื่องจากจำนวนข้อมูลที่ยังไม่มากพอ แต่ถึงแม้จะยังไม่ลู่เข้าค่าความแม่นยำในช่วงนี้ก็อยู่ในช่วง 70 -80 % ทั้งหมด ซึ่งจากรูปที่ 26 จะเห็นว่า การใช้แบบจำลองขนาดใหญ่ในการทำนายสภาวะความเครียดของพืชนั้นหากมีจำนวนข้อมูลภาพถ่ายที่เพียงพอและโมเดลนั้นมีขนาดใหญ่เพียงพอที่จะเรียนรู้ ก็จะสามารถทำนายสภาวะความเครียดได้ดีทั้งสิ้น โดยจะเห็นได้จาก แบบจำลอง ResNet50V2 GoogLeNet และ DenseNet121 ในช่วง 3 วันแรกนั้นเนื่องจากข้อมูลรูปภาพยังไม่เพียงพอทำให้ค่าความแม่นยำที่ทำได้ยังไม่ถึง 80% โดยหลังจากวันที่ 5 แบบจำลอง CNN ทั้งหมดนั้นสามารถทำนายสภาวะความเครียดได้ความแม่นยำมากกว่า 80% แล้วทั้งสิ้น และเมื่อใช้ชุดข้อมูลทดสอบของวันที่ 13 แบบจำลอง CNN ทั้งสามสามารถทำนายสภาวะเครียดของข้าวฟ่างได้ความแม่นยำในระดับเท่ากับหรือมากกว่าแบบจำลอง VGG16+LSTM ที่เสนอโดยแบบจำลองที่เสนอมีความแม่นยำอยู่ที่ 91% ซึ่งอีกด้านหนึ่งหากเปรียบเทียบแบบจำลอง VGG16 กับแบบจำลอง CNN อื่นๆจะพบว่าความแม่นยำในการทำนายจะน้อยกว่าแบบจำลองอื่นๆในทุกๆช่วงของชุดข้อมูล แต่หากนำมาทำงานร่วมกับส่วนของ LSTM ดังโครงสร้าง VGG16+LSTM ดังที่เสนอก็จะทำงานได้ดีขึ้นมาก โดยเฉพาะในช่วงวันแรกๆ หลังจากที่เกิดสภาวะความเครียดและขาดสารอาหาร โดยสามารถทำนายสภาวะความเครียดได้สูงถึง 85% ในวันที่ 3 ของการทดลอง ซึ่งหากพิจารณาเปรียบเทียบกับแบบจำลอง CNN อื่นๆ ที่ต้องการทำนายสภาวะความเครียดของต้นข้าวฟ่างในระยะเริ่มต้นจะเห็นว่าแบบจำลอง CNN นั้นมีความแม่นยำในการทำนายสภาวะความเครียดของต้นข้าวฟ่างในช่วงต้น วันที่ 1 ถึง 4 ไม่สูงนัก อันเนื่องมาจากในวันที่ 1 ถึง 4 ต้นพืชที่ขาดสารอาหารนั้นยังไม่แสดงอาการที่ชัดเจนนั้น ทำให้แบบจำลอง CNN ที่ไม่สามารถรับรู้ข้อมูลที่มีความสัมพันธ์เชิงเวลา อาทิเช่น อัตราการเจริญเติบโตที่ควรจะเป็น หรือ ขนาด หรือ สี ของใบที่ค่อยๆเปลี่ยนแปลงแต่ยังไม่ได้เปลี่ยนอย่างชัดเจน ซึ่งหากเปรียบเทียบกับผลความแม่นยำในการทำนายของแบบจำลอง VGG16+LSTM ที่ผู้วิจัยเสนอจะพบว่า แบบจำลองสามารถเพิ่มความแม่นยำในการทำนายในระยะเริ่มต้นได้สูงมากกว่า แบบจำลอง CNN ปกติ ทั้งนี้เนื่องจากส่วน LSTM ของแบบจำลองสามารถที่จะรับรู้ถึงข้อมูลที่มาจากรูปภาพก่อนหน้าได้ด้วยทำให้ภาพสามารถรับรู้ข้อมูลที่มีความสัมพันธ์เชิงเวลาต่างๆได้

อย่างไรก็ตามหากเปรียบเทียบกันในความเร็วในการทำนายของแบบจำลอง VGG16+LSTM และแบบจำลอง CNN อื่นๆที่นำมาเปรียบเทียบ จากตารางที่ 9 จะพบว่า เวลาที่ใช้ในการทำนายโดยใช้ VGG16+LSTM นั้น ใช้เวลามากกว่า ค่าเฉลี่ยเวลาการทำนายของแบบจำลอง

CNN ต่างๆที่นำมาเปรียบเทียบถึง 20.72 เท่า เนื่องจากการทำนาย 1 ครั้งนั้นแบบจำลอง VGG16+LSTM จะใช้ข้อมูลลักษณะเป็น Time series ซึ่งข้อมูลจะใหญ่ขึ้นเรื่อยๆทุกครั้งที่มีการถ่ายรูปเพิ่มทุกวันๆ นอกจากข้อมูลที่ใหญ่กว่าแล้วการเพิ่มของชั้น LSTM เข้าไปด้วยจะทำให้แบบจำลองมีขนาดใหญ่กว่ามาก อย่างไรก็ตามในการใช้งานถึงการระบุสถานะความเครียดของต้นไม้ที่นั่น ไม่ได้ต้องการความเร็วในการประมวลผลที่สูงนัก ความเร็วในการประมวลผลภาพที่ 185.25 ms ถือได้ว่าสามารถใช้งานได้จริง



## บทที่ 6 สรุปผลการทดลอง

ในการทดลองนี้ผู้ทดลองได้ทดลองออกแบบและฝึกโครงสร้าง CNN และ CNN+LSTM เพื่อระบุรัฐสภาวะความเครียดของต้นข้าวฟ่างในระยะเริ่มต้น โดยใช้ชุดข้อมูลการทดลอง Treatment Study การขาดสารอาหาร Ammonium และ Nitrate ของต้นข้าวฟ่างเป็นข้อมูลฝึกแบบจำลอง และได้ทดลองนำแบบจำลอง CNN ไปใช้บนบอร์ด PYNQ ทดลองใช้งานแบบจำลองทั้งบนด้าน CPU Dual-core ARM Cortex-A9 และ FPGA ของ PYNQ ซึ่งได้ผลการทดลองว่า บอร์ด PYNQ มีทรัพยากรของ FPGA ไม่เพียงพอที่จะรองรับ CNN ที่ใหญ่พอที่จะฝึกแบบจำลองให้ระบุสภาวะความเครียดของพืชได้ โดยค่าความแม่นยำ (Accuracy) ที่ได้สูงสุด อยู่ที่ 0.61 เท่านั้น ค่า Accuracy และ Loss ในการฝึกบ่งชี้ชัดเจนว่า CNN ที่ใช้ใน FPGA ซึ่งเป็นขนาดที่ใหญ่ที่สุดที่สามารถใช้งานได้บน PYNQ ยัง Underfit กับข้อมูลอยู่ อย่างไรก็ตามหากเปรียบเทียบความเร็วในการประมวลผลระหว่าง Dual-core ARM Cortex-A9 และ FPGA แล้ว พบว่า FPGA มีความเร็วมากกว่าถึง 38 เท่าด้วยกัน แต่ก็ยังช้ากว่า GPU Nvidia Quadro RTX 4000 ถึง 2 เท่า แต่หากเทียบในปัจจุบันที่ใช้งานเป็นระบบฝังตัว เรื่องของต้นทุน และการใช้พลังงาน FPGA ก็ยังคงเป็นตัวเลือกที่เหมาะสม นอกจากนี้ผู้ทดลองยังได้ทดลองออกแบบและฝึกโครงสร้าง VGG16+LSTM เพื่อให้ได้โครงสร้างที่มีประสิทธิภาพความแม่นยำสูงในการระบุสภาวะความเครียดของต้นข้าวฟ่าง ซึ่งผลการทดลองแสดงออกมาชัดเจนว่า การใช้แบบจำลอง LSTM ร่วมกับ CNN ขนาดใหญ่จะทำให้ค่าความแม่นยำของแบบจำลองระบุความเครียดในสภาวะเริ่มต้นสูงกว่าการใช้งาน CNN เพียงอย่างเดียว ซึ่งค่า Accuracy ที่ได้คือ 85.32% ที่วันที่ 3 หรือ 2 วันหลังจากต้นข้าวฟ่างเริ่มขาดสารอาหาร วิธีการระบุสภาวะเครียดในระยะเริ่มต้นนี้สามารถใช้กับพืชชนิดอื่นได้โดยการจัดเก็บชุดข้อมูลใหม่ในลักษณะเดียวกัน

บรรณานุกรม



จุฬาลงกรณ์มหาวิทยาลัย  
**CHULALONGKORN UNIVERSITY**

1. Division, U.N.D.o.E.a.S.A.P. *Population growth*. 2014 [cited 2014 10 October]; Available from: <http://www.unpopulation.org>.
2. Phillips, R.L., *Mobilizing Science to Break Yield Barriers*. Crop Science, 2010. **50**(Supplement\_1): p. S-99.
3. Minervini, M., et al., *Finely-grained annotated datasets for image-based plant phenotyping*. 2015.
4. V. B. Soares, J. and D. W. Jacobs, *Efficient segmentation of leaves in semi-controlled conditions*. Vol. 24. 2013.
5. Yanikoglu, B., E. Aptoula, and C. Tirkaz, *Automatic plant identification from photographs*. Vol. 25. 2014.
6. Cerutti, G., et al., *ReVeS Participation - Tree Species Classification Using Random Forests and Botanical Features*. Vol. 1178. 2012.
7. Kalyoncu, C. and O. Toygar, *Geometric Leaf Classification*. Vol. 133. 2014.
8. Silva, L.O.L.A., et al., *Comparative assessment of feature selection and classification techniques for visual inspection of pot plant seedlings*. Vol. 97. 2013. 47-55.
9. Wang, J., et al., *An Adaptive Thresholding algorithm of field leaf image*. Vol. 96. 2013. 23-39.
10. Song, Y., et al., *Surface Modelling of Plants from Stereo Images*. Vol. 312-319. 2007. 312-319.
11. Biskup, B., et al., *A stereo imaging system for measuring structural parameters of plant canopies*. Vol. 30. 2007. 1299-308.
12. Jin, J. and L. Tang, *Corn plant sensing using real-time stereo vision*. Vol. 26. 2009. 591-608.
13. Alenyà, G., B. Dellen, and C. Torras, *3D modelling of leaves from color and ToF data for robotized plant measuring*. 2011. 3408-3414.
14. Yin, X., et al., *Multi-leaf tracking from fluorescence plant videos*. 2014.
15. Wu, B., R. Nevatia, and Y. Li, *Segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses*. 2008.
16. He, X. and S. Gould, *An Exemplar-Based CRF for Multi-instance Object Segmentation*. 2014. 296-303.
17. Singh, A., et al., *Machine Learning for High-Throughput Stress Phenotyping in Plants*. Vol. 21. 2015.
18. Römer, C., et al., *Early drought stress detection in cereals: Simplex Volume Maximization for hyperspectral image analysis*. Vol. 39. 2012. 878-890.
19. Mokhtar, U., et al. *Identifying Two of Tomatoes Leaf Viruses Using Support Vector Machine*. in *Information Systems Design and Intelligent Applications*. 2015. New Delhi: Springer India.
20. Baranowski, P., et al., *Hyperspectral and Thermal Imaging of Oilseed Rape (Brassica napus) Response to Fungal Species of the Genus Alternaria*. 2015.
21. Schikora, M., et al., *An image classification approach to analyze the suppression of plant immunity by the human pathogen SalmonellaTyphimurium*. BMC Bioinformatics, 2012. **13**(1): p. 171.
22. Schikora, M., et al., *Probabilistic Classification of Disease symptoms caused by Salmonella on Arabidopsis Plants*. Vol. 2. 2010. 874-879.
23. Kruse, O.M.O., et al., *Pixel classification methods for identifying and quantifying leaf surface injury from digital images*. Computers and Electronics in Agriculture, 2014. **108**: p. 155-165.
24. Singh, A.K., et al., *Deep Learning for Plant Stress Phenotyping: Trends and Future Perspectives*. Trends in Plant Science, 2018. **23**(10): p. 883-898.

25. Kaneda, Y., S. Shibata, and H. Mineno, *Multi-modal sliding window-based support vector regression for predicting plant water stress*. Knowledge-Based Systems, 2017. **134**: p. 135-148.
26. Brahim, M., K. Boukhalfa, and A. Moussaoui, *Deep Learning for Tomato Diseases: Classification and Symptoms Visualization*. Applied Artificial Intelligence, 2017. **31**(4): p. 299-315.
27. Amara, J., B. Bouaziz, and A. Algergawy, *A Deep Learning-based Approach for Banana Leaf Diseases Classification*. 2017. 79-88.
28. Fujita, E., et al., *Basic Investigation on a Robust and Practical Plant Diagnostic System*. 2016. 989-992.
29. Fahlgren, N., M.A. Gehan, and I. Baxter, *Lights, camera, action: high-throughput plant phenotyping is ready for a close-up*. Current Opinion in Plant Biology, 2015. **24**: p. 93-99.
30. Mishra, P., et al., *Close range hyperspectral imaging of plants: A review*. Biosystems Engineering, 2017. **164**: p. 49-67.
31. González, C., et al., *Use of FPGA or GPU-based architectures for remotely sensed hyperspectral image processing*. Integration, 2013. **46**(2): p. 89-103.
32. Golzarian, M.R., et al., *Accurate inference of shoot biomass from high-throughput images of cereal plants*. Plant Methods, 2011. **7**(1): p. 2.
33. Honsdorf, N., et al., *High-Throughput Phenotyping to Detect Drought Tolerance QTL in Wild Barley Introgression Lines*. PLOS ONE, 2014. **9**(5): p. e97047.
34. Hairmansis, A., et al., *Image-based phenotyping for non-destructive screening of different salinity tolerance traits in rice*. Rice, 2014. **7**(1): p. 16.
35. Tessmer, O.L., et al., *Functional approach to high-throughput plant growth analysis*. BMC Systems Biology, 2013. **7**(6): p. S17.
36. Neilson, E., et al., *Utilization of a high-throughput shoot imaging system to examine the dynamic phenotypic responses of a C4 cereal crop plant to nitrogen and water deficiency over time*. Vol. 66. 2015.
37. Seelig, H.D., et al., *The assessment of leaf water content using leaf reflectance ratios in the visible, near-, and short-wave-infrared*. International Journal of Remote Sensing, 2008. **29**(13): p. 3701-3713.
38. Sirault, X.R.R., R.A. James, and R.T. Furbank, *A new screening method for osmotic component of salinity tolerance in cereals using infrared thermography*. Functional Plant Biology, 2009. **36**(11): p. 970-977.
39. Baker, N.R., *Chlorophyll Fluorescence: A Probe of Photosynthesis In Vivo*. Annual Review of Plant Biology, 2008. **59**(1): p. 89-113.
40. Veley, K.M., et al., *High-throughput profiling and analysis of plant responses over time to abiotic stress*. bioRxiv, 2017: p. 132787.
41. Pape, J.-M. and C. Klukas, *3-D Histogram-Based Segmentation and Leaf Detection for Rosette Plants*. 2014. 61-74.
42. Krizhevsky, A., I. Sutskever, and G. E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*. Vol. 25. 2012.
43. Wang, E., J.J. Davis, and P.Y.K. Cheung, *A PYNQ-Based Framework for Rapid CNN Prototyping*. IEEE.



## ประวัติผู้เขียน

ชื่อ-สกุล	Krit Rojanarungruengporn
วัน เดือน ปี เกิด	18 Jun 1990
สถานที่เกิด	Thailand
วุฒิการศึกษา	Chulalongkorn university
ที่อยู่ปัจจุบัน	59/15 Moo.2 Bangrugpattana Bangbuatong Nonthaburi 11110



จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY