

การจำแนกผลึกน้ำตาลด้วยวิธีการเรียนรู้เชิงลึก



สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2563

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Image classification of sugar crystal with Deep learning



An Independent Study Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Software Engineering

Department of Computer Engineering

FACULTY OF ENGINEERING

Chulalongkorn University

Academic Year 2020

Copyright of Chulalongkorn University

| | |
|----------------------|--|
| หัวข้อสารนิพนธ์ | การจำแนกผลึกน้ำตาลด้วยวิธีการเรียนรู้เชิงลึก |
| โดย | นายสุริยะ ชยะธรรมกุล |
| สาขาวิชา | วิศวกรรมซอฟต์แวร์ |
| อาจารย์ที่ปรึกษาหลัก | ศาสตราจารย์ประภาส จงสฤษดิ์วัฒนา |

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้รับสารนิพนธ์ฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

คณะกรรมการสอบสารนิพนธ์

| | |
|-----------------------------------|----------------------|
| | ประธานกรรมการ |
| (ศาสตราจารย์บุญเสริม กิจศิริกุล) | |
| | อาจารย์ที่ปรึกษาหลัก |
| (ศาสตราจารย์ประภาส จงสฤษดิ์วัฒนา) | |
| | กรรมการ |
| (รองศาสตราจารย์วิวัฒน์ วัฒนาวุฒิ) | |

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สุริยะ ชยะธรรมกุล : การจำแนกผลึกน้ำตาลด้วยวิธีการเรียนรู้เชิงลึก. (Image classification of sugar crystal with Deep learning) อ.ที่ปรึกษาหลัก : ศ.ประภาส จงสกลิตย์วัฒนา

ปัจจุบันระบบควบคุมแบบปัญญาประดิษฐ์เป็นสิ่งสำคัญ ในการทดแทนแรงงานที่มี ประสิทธิภาพ การจำแนกผลึกน้ำตาลระหว่างกระบวนการผลิตอย่างถูกต้อง เป็นพื้นฐานสำหรับ การควบคุมกระบวนการเคี้ยวตผลึกน้ำตาล งานวิจัยนี้นำวิธีการเรียนรู้เชิงลึกมาจำแนกภาพการ ตผลึกน้ำตาลจากกระบวนการน้ำตาลเคี้ยวตผลึก โดยได้ทำการทดลองจริงที่โรงงานน้ำตาลใน ประเทศไทย โดยทำการสร้างโมเดลที่แตกต่างกัน และนำมาฝึกฝนผ่านชุดข้อมูลภาพผลึกน้ำตาล จากกระบวนการผลิตนี้ แล้วทำการวัดและเปรียบเทียบประสิทธิภาพ โดยโมเดล VGG16 ที่ผ่านการ ปรับแต่ง มีความแม่นยำในการจำแนกผลึกน้ำตาลระหว่าง 80% ถึง 94% ของภาพผลึกน้ำตาลทั้ง 4 ประเภท ที่แยกตามสภาวะการเกิดผลึก ผลการทดลองนี้ยังแสดงให้เห็นว่าโมเดลนี้มีความแม่นยำ มากกว่าโมเดลอื่น และสามารถใช้ตรวจสอบการตผลึกของกระบวนการผลิตน้ำตาล อีกทั้งระบบ ควบคุมแบบปัญญาประดิษฐ์โดยอาศัยการถ่ายทอดความชำนาญจากมนุษย์



สาขาวิชา วิศวกรรมซอฟต์แวร์
ปีการศึกษา 2563

ลายมือชื่อนิสิต
ลายมือชื่อ อ.ที่ปรึกษาหลัก

6171006221 : MAJOR SOFTWARE ENGINEERING

KEYWORD: Deep learning / Image Classification / Crystal Formation

Suriya Chayatummagoon : Image classification of sugar crystal with Deep learning. Advisor: Prof. PRABHAS CHONGSTITVATANA

Nowadays, artificial intelligent control is essential in the situation where are shortage of experienced workers. The correct classification of sugar crystals during the production process is the basis for the control of the sugar crystallization. Correct Classification of sugar crystals is necessary for automatic control. This research uses the principles of deep learning with a neural network to identify the crystallization of sugar from the actual production process of sugar factories in Thailand. Performance of the proposed system was measured and compared with the Fine-tuning VGG16 model. It was accurate between 80% and 94% in identifying sugar crystals in four classes. The results of this study also show that this model is more accurate than other models. It can be used with respect to monitoring the crystallization of sugar production process. This system can be employed for artificial intelligent control system which is based on transcribing human expertise.



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Field of Study: Software Engineering

Student's Signature

Academic Year: 2020

Advisor's Signature

กิตติกรรมประกาศ

ขอกราบขอบพระคุณ ศ.ดร. ประภาส จงสฤษดิ์วัฒนา อาจารย์ที่ปรึกษาโครงการ เป็นอย่างยิ่ง ที่ได้สละเวลาให้คำปรึกษา คำแนะนำ รวมถึงแนวทางสำหรับการทำโครงการมหำบัณฑิตนี้ รวมทั้งเป็นผู้ประสานงานให้ความช่วยเหลือแก่นิสิตที่ทำโครงการทุกคน

ขอกราบขอบพระคุณ ศ.ดร.บุญเสริม กิจศิริกุล และ รศ.ดร.วิวัฒน์ วัฒนาวุฒิ คณะกรรมการคุมสอบโครงการมหำบัณฑิตเป็นอย่างยิ่ง ที่ได้กรุณาแนะนำแนวทาง รวมถึงการ ตรวจสอบและแก้ไขโครงการมหำบัณฑิตนี้ ขอกราบขอบพระคุณ ผศ. ดร. ทวีติย์ เสนีวงศ์ ณ อยุธยา อาจารย์ผู้ประสานงานโครงการ มหำบัณฑิตที่ได้สละเวลาให้ความช่วยเหลือแก่นิสิตที่ทำโครงการทุกคน

ขอขอบคุณ คณาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้คำแนะนำ ความรู้และแนวทางการทำโครงการนี้



สุริยะ ชยะธรรมกุล

สารบัญ

| | หน้า |
|--|------|
| | ค |
| บทคัดย่อภาษาไทย..... | ค |
| | ง |
| บทคัดย่อภาษาอังกฤษ..... | ง |
| กิตติกรรมประกาศ..... | จ |
| สารบัญ..... | ฉ |
| 1.1 ความเป็นมาและความสำคัญของปัญหา..... | 1 |
| 1.2 วัตถุประสงค์ของโครงการ..... | 1 |
| 1.3 ขอบเขตของโครงการ..... | 1 |
| 1.4 ขั้นตอนการดำเนินโครงการ..... | 2 |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับ..... | 2 |
| 1.6 ผลงานตีพิมพ์..... | 2 |
| 2.1 แนวคิดและทฤษฎี..... | 3 |
| 3.1 การรวบรวมข้อมูล..... | 17 |
| 3.2 การเตรียมข้อมูล Data Preparation and Image augmentation..... | 18 |
| 4.1 Simple DCNNs..... | 22 |
| 4.3 Principal Component Analysis: PCA..... | 24 |
| 4.4 Custom head..... | 27 |
| บรรณานุกรม..... | 35 |
| ประวัติผู้เขียน..... | 38 |

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

กระบวนการเคี้ยวตกผลึกยังคงจะต้องอาศัยพนักงาน ช่างเคี้ยวน้ำตาลที่มีประสบการณ์ สำหรับควบคุมกระบวนการเคี้ยวตกผลึกให้มีคุณภาพ แต่การฝึกฝนช่างเคี้ยวน้ำตาลนั้นต้องอาศัย ระยะเวลาอันยาวนานประกอบกับช่างเคี้ยวน้ำตาล แต่ละคนนั้นมีความสามารถและความชำนาญที่ไม่เท่ากัน ดังนั้นจึงส่งผลต่อประสิทธิภาพในการผลิตของโรงงาน อีกทั้งค่าแรงของช่างเคี้ยวน้ำตาลที่มี ประสบการณ์นั้นมีราคาสูงกว่า เมื่อเทียบกับพนักงานควบคุมกระบวนการในตำแหน่งอื่น ช่างเคี้ยว น้ำตาลที่มีประสบการณ์นั้น จะเป็นที่ต้องการของตลาดแรงงานในอุตสาหกรรมนี้ ทำให้โรงงานน้ำตาล ต่างประสบปัญหาขาดแคลนทั้งด้านบุคลากรในตำแหน่งนี้เสมอมา อีกทั้งได้รับแรงบันดาลใจจาก กระบวนการควบคุมเคี้ยวตกผลึกน้ำตาลที่กำลังทดสอบในญี่ปุ่นเมื่อต้นปี ค.ศ. 2020 โดยใช้ Digital Image Processing (1) ภาพผลึกที่คมชัดจะถูกส่งไปยังคอมพิวเตอร์อย่างต่อเนื่องในห้องควบคุม ซึ่ง ซอฟต์แวร์เฉพาะจะใช้อัลกอริทึมเฉพาะกับแต่ละภาพและคำนวณ รวมถึงค่าสัมประสิทธิ์การแปรผัน และค่าเฉลี่ยของช่องว่างระหว่างผลึก อัตราส่วนคิดเป็นเปอร์เซ็นต์ของอนุภาคน้ำตาลเล็กที่ไม่ใช่ผลึก และจำนวนเม็ดในแบบเรียลไทม์ เกณฑ์ที่ปรับได้สำหรับตัวแปร จะถูกใช้เพื่อตรวจจับและเตือนให้ ทราบถึงความไม่สอดคล้องเช่น การปรากฏตัวของสิ่งปนเปื้อน การสร้างผลึกที่ไม่ถูกต้อง ขนาดผลึกที่ ไม่ดีในเวลาที่กำหนด

1.2 วัตถุประสงค์ของโครงการ

- เพื่อศึกษากระบวนการและประยุกต์ใช้ Image classification โดย Deep learning
- เพื่อเรียนรู้ขั้นตอนวิธีในการสร้างโมเดล DCNNs, Transfer learning และ Fine-tuning
- เพื่อสร้างโมเดลเพื่อใช้ในการทำนายรูปผลึกน้ำตาลในกระบวนการผลิตได้อย่างแม่นยำ
- เพื่อนำเอาความรู้ที่ได้เรียนมาพัฒนาใช้งานในอุตสาหกรรม

1.3 ขอบเขตของโครงการ

- โมเดลที่สร้างขึ้นนี้จะสามารถแยกรูปผลึกน้ำตาล 4 classes
- การทดลองรูปถ่ายจากกระบวนการเคี้ยวตกผลึก เป็นน้ำตาลชนิดรีเฟน อยู่นอกฤดูเก็บอ้อย)

- ชนิดหม้อเคี้ยวตกผลึกที่ใช้เป็นแบบ Batch type โดยใช้สภาวะการเคี้ยวภายใต้สุญญากาศ
- การเก็บรูปผลึกจะเลือกเฉพาะ Batch type ที่กระบวนการปรกติ โดยคุณสมบัติ (Fine syrup input 200 IU, brix 65° and color product 20 IU)
- ทดสอบและประเมินผลความแม่นยำของโมเดลนี้ จะใช้รูปถ่ายการผลึกน้ำตาลชนิดเดียวกัน และหม้อเคี้ยวเดียวกันในสภาวะที่กล่าวมาข้างต้นเท่านั้น

1.4 ขั้นตอนการดำเนินโครงการ

- ศึกษาและทำความเข้าใจเกี่ยวกับ Image classification และงานวิจัยที่เกี่ยวข้อง
- ศึกษาและทำความเข้าใจเกี่ยวกับการตกผลึกน้ำตาล และงานวิจัยที่เกี่ยวข้อง
- ออกแบบการทดลองเก็บรูปผลึกที่มีคุณภาพให้ได้ตามจำนวนรูปที่ต้องการ
- เลือกเครื่องมือที่ใช้ในการทำ Deep learning ที่เหมาะสม
- ทดสอบและประเมินผลวิธีวิจัยระบบที่พัฒนาขึ้นเพื่อสนับสนุนแนวคิดในงานวิจัยนี้
- จัดทำบทความทางวิชาการและนำเสนอ
- สรุปผลแนวทางการวิจัย ข้อเสนอแนะและจัดทำเล่มโครงการ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- ได้รับโมเดล ที่สามารถทำนายรูปถ่ายผลึกจากสถานะจากกระบวนการระหว่างการเคี้ยวตกผลึกน้ำตาลตรงตามที่ช่างเคี้ยวสังเกตได้
- ได้วิธีการรับแต่งค่ารูปผลึก เพื่อให้มีความเหมาะสมในการสร้างโมเดล
- ได้วิธีการสร้างโมเดล การออกแบบโครงสร้างโมเดลจากการทดลองนี้
- สามารถนำวิธีการนี้ไปปรับใช้ในการทำ Digital image processing กับกระบวนการตกผลึกชนิดอื่นได้

1.6 ผลงานตีพิมพ์

Suriya Chayatummagoon, Prabhas Chongstitvatana, "Image classification of sugar crystal with Deep learning", 2021 - 13th International Conference on Knowledge and Smart Technology (KST), Jan 2021

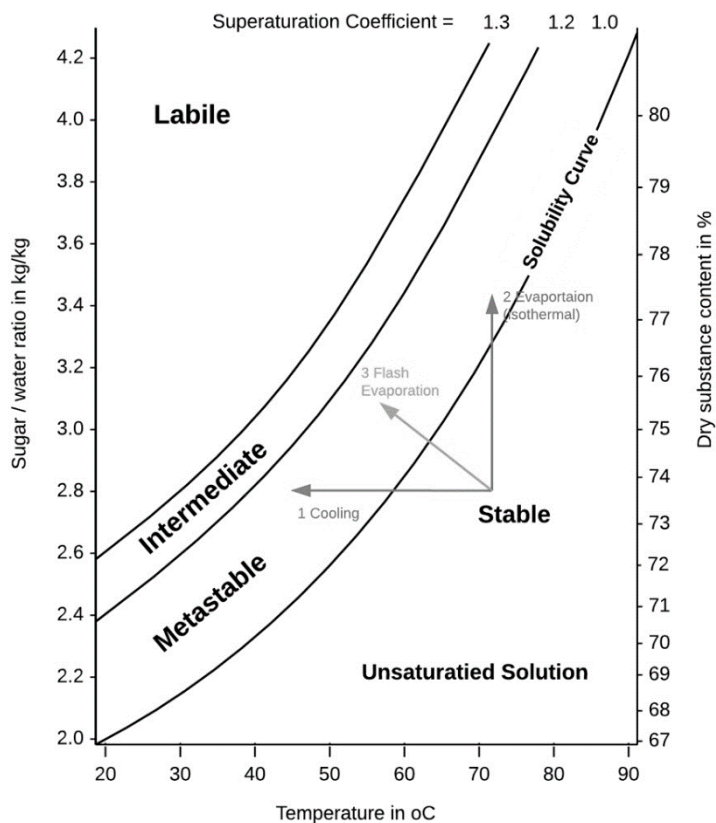
บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดและทฤษฎี

ในการต้มเคี้ยวน้ำเชื่อมและการตกผลึกน้ำตาล เป็นกระบวนการสำคัญที่อาจถือได้ว่าเป็นกระบวนการหลัก ที่จะเปลี่ยนรูปของน้ำตาลที่อยู่ในรูปสารละลายน้ำเชื่อมที่เป็นของเหลวเป็นผลึกน้ำตาลในรูปที่เป็นผลึกแข็ง โดยหลักการทั่วไปของกระบวนการตกผลึกนั้น จะต้องดึงความร้อนแฝงของการกลายเป็นผลึกของแข็ง ออกจากสารละลายที่ต้องการตกผลึก เพื่อทำให้เกิดการเปลี่ยนแปลงสถานะจากของเหลวเป็นของแข็งที่อุณหภูมิของการตกผลึกนั้น ดังนั้นกระบวนการตกผลึกจึงต้องทราบถึงความสัมพันธ์ของการกระจายตัวของมวลสาร ที่ต้องการตกผลึกในเฟสของเหลวและเฟสของแข็ง ขั้นตอนวิธีการทำให้เกิดสภาวะความอิ่มตัวยิ่งยวดของสารละลายเข้มข้น และกระบวนการเลี้ยงผลึกให้ได้ขนาดตามมาตรฐานข้อกำหนด ทั้งนี้ต้องควบคุมทั้งขั้นตอนการต้มเคี้ยว ขั้นตอนการตกผลึก ขั้นตอนการเลี้ยงผลึก และขั้นตอนการปั่นแยกผลึก เพื่อให้ได้ปริมาณผลึกน้ำตาลออกมาให้มากที่สุด ทั้งนี้ในขั้นตอนการต้มเคี้ยวและตกผลึกน้ำตาลทรายนี้จะควบคุมได้ยากกว่าสภาวะปกติทั่วไป

เนื่องจากของเหลวที่เกี่ยวข้องในขั้นตอนนี้จะอยู่ในรูปของน้ำเชื่อมที่มีความเข้มข้นสูงมาก(2) มีความขุ่นหนืดสูงมากการควบคุมการไหล หรือการเคลื่อนย้ายผ่านระบบท่อจึงทำได้ยากทั้งยังต้องคำนึงการถ่ายเทความร้อนในระบบด้วย ซึ่งในแต่ละช่วงระดับของความเข้มข้นอิ่มตัวยิ่งยวดนี้ แบ่งออกตามค่าสัมประสิทธิ์ของความเข้มข้นอิ่มตัวยิ่งยวด (2) คือ ค่า SC. ตั้งแต่ 1.00 – 1.20 เรียกว่าช่วง metastable zone ค่า SC. ระหว่าง 1.20 – 1.30 เรียกว่าช่วง intermediate zone และตั้งแต่ค่า SC. 1.30 – 1.40 เรียกว่าช่วง labile zone ซึ่งแต่ละช่วงสามารถนำมาประยุกต์ใช้ในการควบคุมกระบวนการตกผลึกน้ำตาลได้ซึ่งขึ้นอยู่กับเทคนิคของแต่ละโรงงาน แต่ละช่วงที่มีการแบ่งระดับของความเข้มข้นอิ่มตัวยิ่งยวดนี้มีลักษณะเฉพาะของน้ำเชื่อมดังรูปที่ 1



รูปที่ 1 ความสัมพันธ์ของสภาวะโซนของการเกิดผลึกน้ำตาล

ช่วง metastable zone เป็นช่วงที่น้ำเชื่อมมีความเข้มข้นเกินจุดอิ่มตัวเล็กน้อยจะมีจำนวนโมเลกุลของน้ำตาลมากเกินจุดอิ่มตัว แต่ยังไม่มีความหนาแน่นมากพอที่โมเลกุลของน้ำตาลจะมาเกาะเกี่ยวกันเป็นนิวคลีโอไนด์ ในช่วงนี้ น้ำเชื่อมไม่สามารถเกิดนิวเคลียสหรือเกิดผลึกขึ้นได้เอง แต่ถ้ามีการเติมนิวเคลียสลงไปให้กับน้ำเชื่อมแล้ว

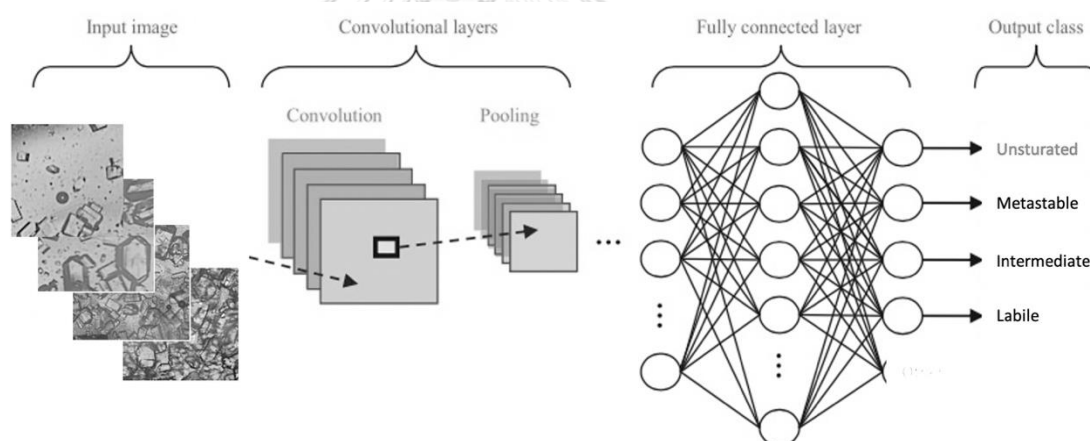
ช่วง intermediate zone เป็นช่วงที่น้ำเชื่อมมีความหนาแน่นของโมเลกุลน้ำตาลมากขึ้นจนทำให้บางจุดของน้ำเชื่อมสามารถเกิดนิวคลีโอไนด์ขึ้นมาได้เองและจะค่อยๆมีขนาดใหญ่ขึ้น ซึ่งในขณะที่เกิดผลึกไปแล้วนั้นอาจจะมีนิวคลีโอไนด์ใหม่เกิดขึ้นมาทำให้ผลึกที่ได้จะมีขนาดต่างกัน แต่ถ้ามีการควบคุมและกระตุ้นให้เกิดนิวคลีโอไนด์อย่างเหมาะสมแล้วจะทำให้เกิดผลึกได้อย่างรวดเร็ว

ช่วง labile zone เป็นช่วงที่น้ำเชื่อมมีความเข้มข้นมากจนมีจำนวนโมเลกุลของน้ำตาลสูงโครหนาแน่นและอยู่ชิดกันมากจนเกิดนิวคลีโอไนด์อย่างอิสระมากมาย จนไม่อาจควบคุมปริมาณและขนาดของผลึกที่จะเกิดขึ้นได้เลย ในช่วงความเข้มข้นอิมตัวยิ่งยวด และเป็นช่วงที่สารละลายน้ำเชื่อมมีความหนืดมากจนควบคุมการตกผลึกได้ยากมากหรืออาจไม่สามารถควบคุมได้เลย

2.2 ทฤษฎีการเรียนรู้เชิงลึก Deep learning

ความรู้เกี่ยวกับการเรียนรู้เชิงลึก Deep learning (3) เป็นแนวคิดในการให้คอมพิวเตอร์เรียนรู้เข้าใจข้อมูลที่ได้รับ โดยมีสถาปัตยกรรมการเรียนรู้ข้อมูลของคอมพิวเตอร์มากมาย Convolutional neural networks นำมาใช้เพื่อสกัดเอาลักษณะเด่นจนทำให้ทราบถึงผลลัพธ์หรือรูปแบบจากภาพถ่าย การทำงานของ Convolutional neural networks(4) เป็นกระบวนการแบบ feed-forward มีส่วนของ hidden layer นั้นจะมี Convolutional layers เพิ่มขึ้นมาเป็นส่วนที่ใช้ในการฟิลเตอร์ภาพ โดยมี kernel function เพื่อใช้แปลงคุณลักษณะและเพื่อแยกองค์ประกอบออกมา เช่น ขอบรูป สี ลักษณะรูปทรง เป็นต้น ก่อนจะนำมาผ่าน activation function ที่ช่วยแปลงค่าให้อยู่ในรูปที่ผลให้ได้ผลลัพธ์ดียิ่งขึ้น รวมถึงส่วนที่เรียกว่าพูลลิ่ง (pooling) โดยส่วนนี้จะทำหน้าที่ปรับขนาดของข้อมูลให้มีขนาดเล็กลง โดยที่รายละเอียดของข้อมูลนั้นยังคงเดิม และในส่วนสุดท้ายก็จะ เป็น Fully connected layer ที่จะคอยเชื่อมต่อในแต่ละชั้นเข้าด้วยกัน ดังรูปที่ 2

รูปที่ 2 กระบวนการ Convolutional neural networks ในการจำแนกผลึก outputs 4 classes



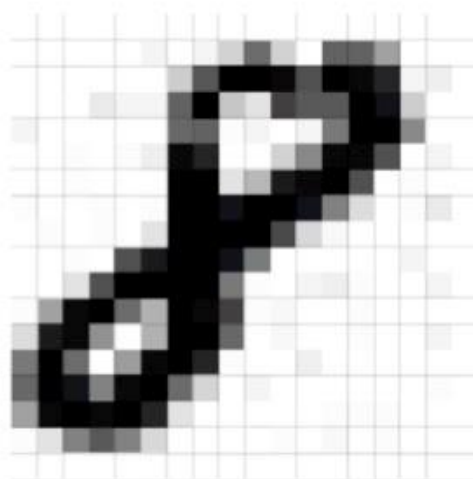
โดยกระบวนการที่กล่าวมาข้างต้นนี้จะแบ่ง ขั้นตอนออกเป็น 4 ขั้นตอนหลัก

- การทำคอนโวลูชัน (Convolution, channel, kernel)
- การขจัดความเป็นเชิงเส้น (relu: Activation function rectified linear)
- การทำพูลลิ่ง (Max, average pooling)
- การเชื่อมต่อกันของแต่ละเลเยอร์อย่างสมบูรณ์ (Fully connected)

2.2.1 คอนโวลูชัน (Convolution)

คอนโวลูชันจะเป็นกระบวนการที่ให้เราสามารถสกัดเอาลักษณะเด่นของรูปถ่ายออกมา โดยใช้ค่าพิกเซล และค่าพิกเซลจะประกอบด้วยกันใน 3 Channel คือ สีแดง (Red), สีฟ้า (Blue), สีเขียว (Green)

| | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|---|---|
| 0 | 2 | 0 | 2 | 0 | 5 | 2 | 5 | 3 | 0 | 5 | 0 | 5 | 3 | 5 | 3 | 0 | 5 | |
| 0 | 2 | 0 | 2 | 1 | 12 | 2 | 11 | 33 | 137 | 37 | 0 | 152 | 147 | 34 | 3 | 0 | 2 | |
| 0 | 2 | 1 | 2 | 0 | 5 | 41 | 160 | 253 | 255 | 255 | 252 | 255 | 233 | 256 | 21 | 15 | 2 | |
| 0 | 2 | 0 | 15 | 0 | 5 | 152 | 251 | 45 | 21 | 124 | 129 | 154 | 255 | 233 | 40 | 0 | 2 | |
| 10 | 2 | 0 | 2 | 0 | 5 | 145 | 146 | 3 | 10 | 2 | 11 | 124 | 253 | 235 | 107 | 0 | 2 | |
| 0 | 2 | 3 | 2 | 4 | 15 | 235 | 216 | 2 | 0 | 26 | 126 | 247 | 240 | 126 | 3 | 11 | 2 | |
| 1 | 2 | 2 | 2 | 0 | 5 | 253 | 252 | 23 | 22 | 224 | 241 | 255 | 161 | 2 | 5 | 0 | 2 | |
| 0 | 2 | 0 | 4 | 0 | 3 | 252 | 250 | 223 | 235 | 255 | 234 | 112 | 23 | 2 | 2 | 17 | 2 | |
| 0 | 2 | 1 | 4 | 0 | 23 | 255 | 252 | 251 | 255 | 172 | 31 | 8 | 3 | 1 | 3 | 0 | 2 | |
| 0 | 2 | 4 | 2 | 163 | 235 | 251 | 255 | 223 | 180 | 2 | 0 | 2 | 3 | 2 | 11 | 0 | 2 | |
| 0 | 2 | 21 | 12 | 255 | 235 | 254 | 235 | 121 | 6 | 2 | 10 | 14 | 5 | 2 | 3 | 0 | 2 | |
| 3 | 73 | 242 | 235 | 141 | 06 | 255 | 235 | 123 | 7 | 2 | 0 | 2 | 5 | 2 | 3 | 0 | 2 | |
| 26 | 221 | 237 | 19 | 0 | 67 | 251 | 235 | 144 | 0 | 0 | 0 | 2 | 7 | 2 | 3 | 11 | 2 | |
| 125 | 255 | 141 | 2 | 87 | 234 | 255 | 206 | 3 | 0 | 2 | 13 | 2 | 1 | 2 | 1 | 0 | 2 | |
| 145 | 248 | 235 | 16 | 235 | 235 | 141 | 34 | 3 | 1 | 2 | 1 | 2 | 3 | 2 | 1 | 5 | 2 | |
| 85 | 257 | 253 | 245 | 235 | 216 | 21 | 1 | 2 | 1 | 2 | 0 | 2 | 2 | 4 | 3 | 0 | 2 | |
| 6 | 23 | 112 | 157 | 124 | 32 | 2 | 2 | 0 | 2 | 0 | 2 | 0 | 2 | 3 | 7 | 3 | 0 | 2 |
| 0 | 2 | 0 | 2 | 0 | 5 | 2 | 5 | 3 | 0 | 5 | 0 | 5 | 3 | 5 | 3 | 0 | 5 | 2 |



รูปที่ 2 การแทนค่าพิกเซลของรูปสีขาว-ดำ นั้นด้วยตัวเลขตั้งแต่ 0 - 255

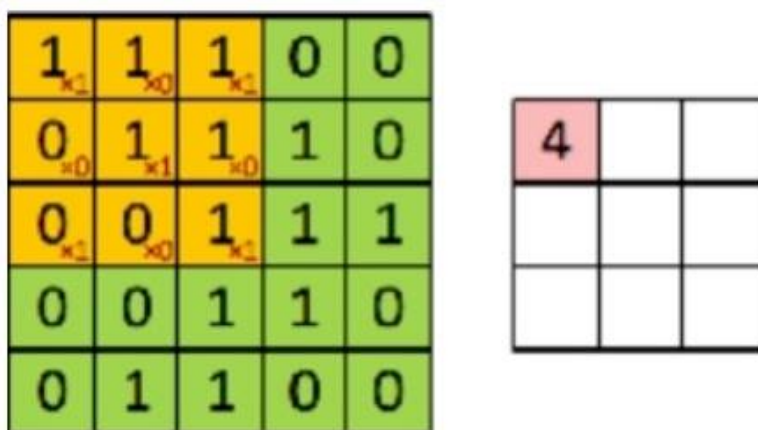
แทนค่าด้วยตัวเลขที่บอกความเข้มของสีนั้นๆ โดยมีค่าตั้งแต่ 0 ถึง 255 (แทนความเข้มจากน้อยไปหามาก) โดยในการทำรูปขาวดำเช่นแนลของรูปนั้นจะมีเพียงหนึ่งแนลเท่านั้นคือแนลแนลของสีดำซึ่งตัวเลข 0 คือสีขาวแล้วจะไล่ระดับความเข้มไปจนถึงตัวเลข 255 คือสีดำ ดังรูปที่ 2

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

รูปที่ 3 จำลองเมทริกซ์ที่ได้จากชุดข้อมูลรับเข้าและเมทริกซ์ Filter

ซึ่งแต่ละชุดการคำนวณนี้จะทำการเก็บค่าไว้เป็นข้อมูลในแบบจำลองเมทริกซ์ของคอนโวลูชัน ซึ่งในแต่ละรูปที่ถ่ายได้นั้นจะมีชุดค่าเมทริกซ์ที่ต่างกันไป โดยข้อมูลนำเข้า 5×5 pixels ที่เป็นรูปขาวดำ โดยที่นี้กำหนดค่าเป็น 0 พิกเซลเป็นสีขาวส่วนค่า 1 เป็นพิกเซลสีดำและมีการกำหนดเมทริกซ์อีกชุดหนึ่งขึ้นมาเพื่อใช้เป็นตัวกรองค่า (Filter) ไปเก็บไว้ในเมทริกซ์ชุดที่เล็กกว่า เราเรียกเมทริกซ์กรองค่าชุดนี้ว่า เคอร์เนล (Kernel) เพื่อทำหน้าที่ตรวจจับลักษณะและสกัดข้อมูลที่สำคัญ feature detector ดังรูปที่ 3



รูปที่ 4 แสดงหลักการทำงานของ filter ค่าและการกำเนิดเมทริกซ์ชุดใหม่หรือพีเจอร์แมพ

ตัวอย่างการคำนวณ

$$\begin{array}{l}
 1 \times 1 = 1, 1 \times 0 = 0, 1 \times 1 = 1 \quad 1, 0, 1 \\
 0 \times 0 = 0, 1 \times 1 = 1, 1 \times 0 = 0 \quad 0, 1, 0 \quad = 1 + 1 + 1 + 1 = 4 \\
 0 \times 1 = 0, 0 \times 0 = 0, 1 \times 1 = 1 \quad 0, 0, 1
 \end{array}$$

เมทริกซ์ที่ทำหน้าที่เป็นตัวกรองค่าจะเคลื่อนไปทั่วภาพและคูณค่าเก็บไว้ในเมทริกซ์ชุดใหม่ดังรูปที่ 4 ซึ่งเราเรียกเมทริกซ์ใหม่นี้ว่าคอนโวลูชันพีเจอร์ convolved feature หรือ พีเจอร์แมพ(1-13)

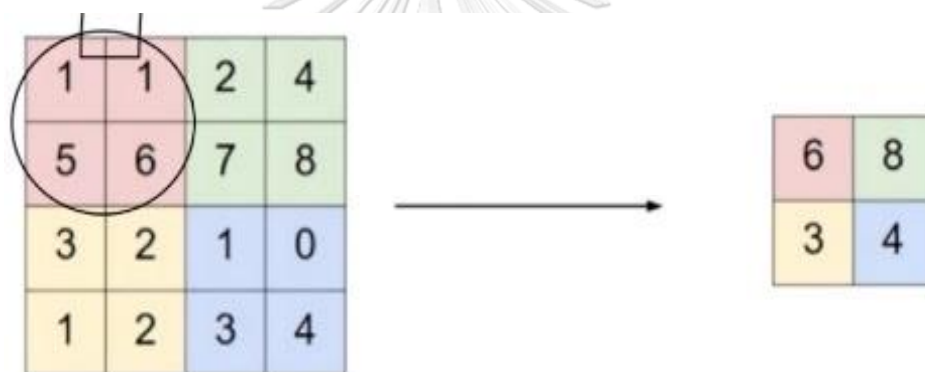
2.2.2 การขจัดความเป็นเชิงเส้น (relu: activation function rectified linear)

หลังจากได้ทำการคอนโวลูชันรูปผลึกและได้พีเจอร์แมพมาแล้ว เราจะนำมาปรับแต่งให้พีเจอร์แมพนั้นไม่มีลักษณะเชิงเส้นด้วยวิธีการ relu ทำการแทนที่ผลของ pixels ที่มีค่าเป็นเชิงลบในพีเจอร์แมพ จุดประสงค์ของการทำ relu นั้นเพื่อให้ Convolutional neural network เรียนรู้ข้อมูลที่ไม่เป็นเชิงเส้นจากรูปดังกล่าว เมื่อนำรูปนั้นเข้าสู่กระบวนการที่เป็นพีเจอร์แมพเข้ามาทำ relu โดยสีดำในรูปเป็นค่าเชิงลบส่วนสีขาวในรูปจะมีค่าเป็นเชิงบวก เมื่อทำการขจัดด้วย relu จะมีค่าที่ได้เพียงค่าที่เป็นเชิงบวกเท่านั้น ซึ่งส่งผลให้ convolutional neural network นี้เกิดการเรียนรู้ข้อมูลที่ได้รับมาจากขั้นตอนการตรวจสอบรูปก่อนหน้า ตลอดจนข้อมูลนั้นจะถูกแปลงมาอยู่ในรูปของ non-linear ซึ่งจะส่งผลให้สามารถเพิ่มประสิทธิภาพรูปของ network นี้ด้วย

2.2.3 การทำพูลลิ่ง (pooling)

การพูลลิ่ง ช่วยลดมิติของพีเจอร์แมพลงแต่ยังคงรักษาข้อมูลสำคัญไว้ ซึ่งการพูลลิ่ง (pooling) สามารถจำแนกเป็นประเภทต่างๆได้เช่น พูลลิ่งด้วยค่าสูงสุด (max pooling), ค่าเฉลี่ย (average pooling) ผลรวม ซึ่งการพูลลิ่งทำให้ผลลัพธ์ที่ได้มีขนาดเล็กและจัดการได้ง่ายขึ้น นอกจากนี้ยังช่วยลดจำนวนของพารามิเตอร์และการคำนวณที่เกินจำเป็นใน network

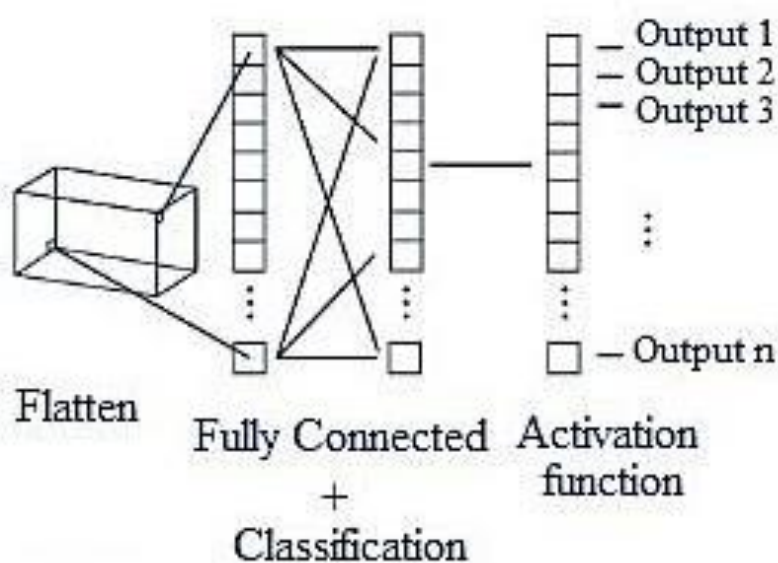
ในกรณีที่ต้องการพูลลิ่งด้วยค่าสูงสุด เราจะกำหนดหน้าต่างหนึ่งขึ้นมา ตัวอย่างในที่นี้สมมติให้หน้าต่างมีขนาด 2×2 pixels และหน้าต่างนี้จะทำการเคลื่อนที่ทีละ 2 pixels ไปจนทั่วเมทริกซ์ของพีเจอร์แมพเพื่อทำการเก็บค่าสูงสุดในทุกๆ 2 pixels ตามรูปที่ 5 ซึ่งนอกจากนี้ การพูลลิ่ง (pooling) จะทำตามจำนวนแชนแนล (channel) ของรูปซึ่งในกรณีรูปสีเราจะได้รับผลลัพธ์ที่ออกมาถึงสามผลลัพธ์ เมื่อพีเจอร์แมพที่ผ่านการทำ relu เข้ามาทำการพูลลิ่ง pooling



รูปที่ 5 แสดงการพูลลิ่งแบบมากที่สุด (max pooling)
CHULALONGKORN UNIVERSITY

2.2.4 การเชื่อมต่อกันของแต่ละเลเยอร์อย่างสมบูรณ์ (Full Connected Layer)

กระบวนการ Convolution, relu และการ pooling กระบวนการทั้งสามนี้ เป็นกระบวนการที่ทำซ้ำได้หลายครั้งและในขั้นสุดท้ายมีการเชื่อมต่อกันของแต่ละชั้นอย่างสมบูรณ์ Fully connected layer ผลลัพธ์จากคอนโวลูชันและพูลลิง นั้นให้ลักษณะเด่น (High-Level Features) ของรูปที่รับเข้ามา และขั้นสุดท้ายเพื่อนำลักษณะเด่นไปทำการคัดกรองรูปที่รับเข้าให้อยู่ในรูปของ classes ดังรูปที่ 6



รูปที่ 6 การเชื่อมต่อกันของแต่ละชั้นอย่างสมบูรณ์ Full connected layer

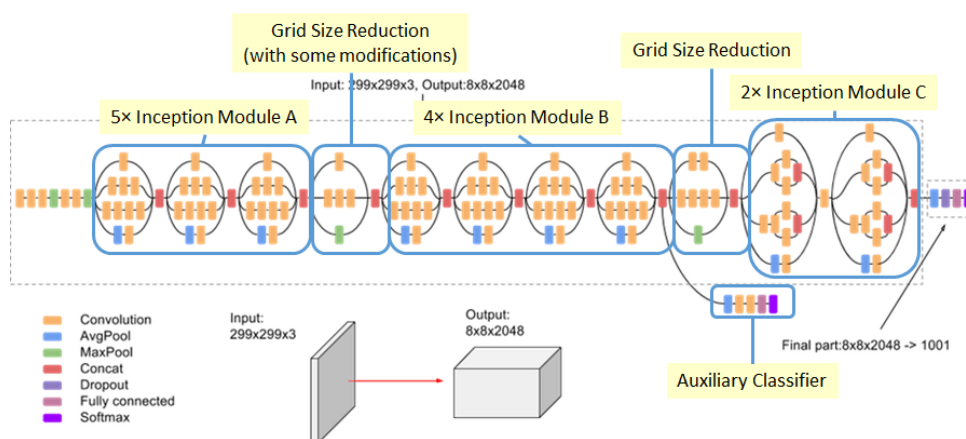
2.3 ทฤษฎีการ Transfer Learning

เนื่องจาก Deep learning จะมีปัญหาเรื่องใช้ระยะเวลาในการฝึก ตั้งแต่ต้นจนจบกระบวนการ ที่นานและมีความซับซ้อนมี ตัวแปร (weight) จำนวนมากซึ่งในการเริ่มต้นฝึกโมเดล Deep learning ที่ซับซ้อนขนาดนี้ตั้งแต่ต้น (Weight initialization ด้วยค่า random) นั้นต้องใช้ทั้งข้อมูล dataset ขนาดใหญ่และใช้พลังการประมวลผลมหาศาลอีกทั้งต้องใช้เวลานานหลายวัน จนอาจถึงนานหลาย สัปดาห์กว่าจะได้โมเดลที่มีประสิทธิภาพ ดังนั้นการใช้ Transfer learning จึงเทคนิคที่ช่วยลดเวลา การฝึกโมเดล Deep learning ด้วยการนำบางส่วนของโมเดลที่ฝึกเรียบร้อยแล้วกับงานที่ใกล้เคียงกัน มาใช้เป็นส่วนหนึ่งของโมเดลใหม่

ImageNet dataset กว่า 15,000,000 ภาพความละเอียดสูงซึ่งมีจำนวน 22,000 classes ส่วน ILSVRC (The ImageNet large scale visual recognition challenge) การแข่งขันที่จัดโดย บริษัท google โดยใช้ข้อมูล subset ของ ImageNet โดยจะมีประเภทของภาพ 1,000 classes ซึ่ง ในแต่ละ classes จะมีรูปภาพอยู่ประมาณ 1,000 ภาพ ทำให้มี dataset ของ ILSVRC รวมประมาณ 1,000,000 ภาพเพื่อใช้ในการฝึกโมเดล โดยโมเดลที่นำมาใช้ในงานวิจัยนี้คือ Inception-v3, ResNet50-v2 และ VGG16

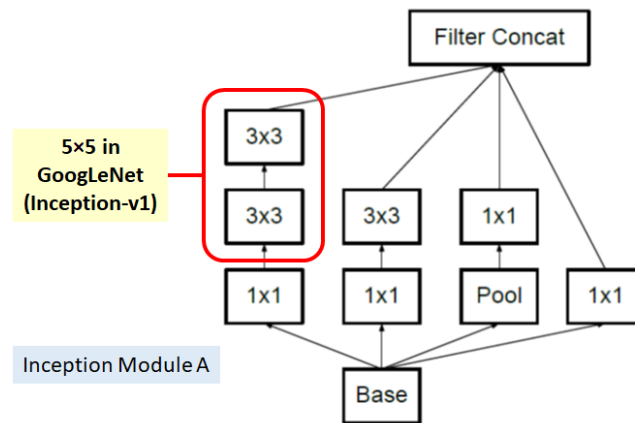
2.3.1 Inception-v3

Inception-v3 (3) (1st ILSVRC-2015) ถูกกล่าวถึงในงานวิจัย (13) ซึ่งเป็นโมเดลที่ได้รับการ พัฒนาโดย Google ซึ่งถูกต่อยอดจาก Inception2,1 (ซึ่งมาจากการพัฒนามาจาก GoogLeNet-2012) โดยการลดโครงสร้างภายในออกเป็น 5 Step คือ Inception Module A จำนวน 5 Module (1), Grid Size of Reduction Step1 จำนวน 1 Module(2), Inception Module B จำนวน 4 Module(3), Grid Size of Reduction Step2 จำนวน 1 Module(4), Inception Module C จำนวน 2 Module(5) และ Head (8x8x2048) สามารถแยก output ได้ 1,000 classes ดังรูปที่ 7



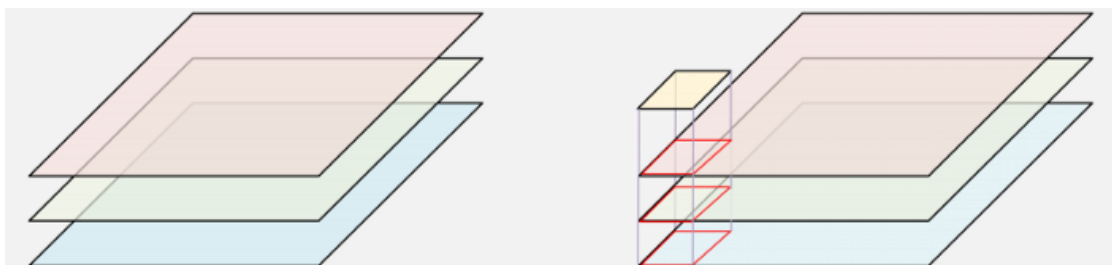
รูปที่ 7 โครงสร้าง Inception-v3

พัฒนาให้ Inception-v3 มี parameter ลดลงจากเดิม แต่ยังคงมีประสิทธิภาพสูงโดยการปรับปรุง convolutions เดิมที่ 5x5 ลงเหลือ 3x3 pixels และ maxpooling เดิมที่ 3x3 ลงเหลือ 2x2 pixels ซึ่งวัตถุประสงค์ของการออกแบบ 1x1 convolution ต้องการให้ shape outputs ขาออกเป็น tensor เช่นกำหนด (N, F, H, W) ซึ่ง N คือ batch size, F คือ จำนวนของ convolution filters ส่วน H,W คือ spatial ของมิติ ดังรูปที่ 8



รูปที่ 8 โครงสร้าง inception module ประกอบด้วย convolution and maxpooling เลเยอร์

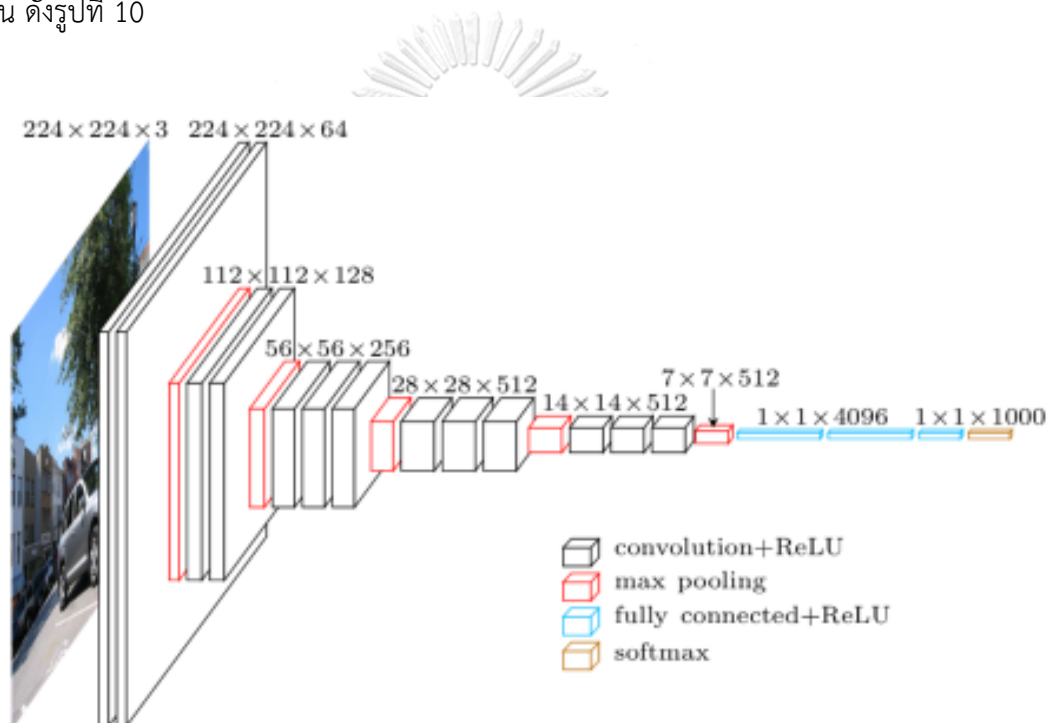
โดยภายในของ inception module กรณี input ข้อมูลเข้ามามี 3 สีคือ RGB ในที่นี่เราสามารถเปรียบเทียบได้กับการทำ feature map ดังรูปด้านซ้าย ซึ่งสมมุติว่าเราต้องการผลลัพธ์ feature map เพียง 1 ค่านั้นถูกทำได้โดยใช้ convolution ขนาด 1x1 pixels ที่มีการขยับที่ 1 stride ดังรูปที่ 9 โดยค่า weight ที่คูณในแต่ละชั้นของ RGB นั้นจะมีค่าที่ไม่เท่ากันส่งผลให้ค่าที่ได้ในแต่ละพิกเซล หลังการทำ feature map นี้จะมีที่เป็นลักษณะเฉพาะ



รูปที่ 9 การทำ feature map ของตัวอย่าง RGB โดยใช้ 1x1 convolution layer

2.3.3 VGG16

โมเดล VGG (5) ย่อมาจาก Visual Geometry Group ซึ่งเป็นกลุ่มนักวิจัยจาก Oxford ทำการพัฒนาสถาปัตยกรรมนี้ขึ้นมา และที่ได้รับความสนใจมากจากการแข่งขัน ILSVR ปี ค.ศ. 2014 และเป็นที่ยอมรับจนถึงปัจจุบัน โดยสิ่งที่เป็นจุดเด่นของ VGG16 คือการแทนที่ hyperparameter จำนวนมาก เน้นไปที่การออกแบบ เลเยอร์ conv2D 3x3 pixels, 1 stride และการใช้ same padding และ maxpooling 2x2 pixels, 2 stride แบบเดียวกันตลอดทั้งโครงสร้าง โดยชื่อของ VGG16 หมายถึงมี 16 ชั้นที่มีน้ำหนักเครือข่ายนี้เป็นเครือข่ายที่ใหญ่และมีพารามิเตอร์ประมาณ 138 ล้าน ดังรูปที่ 10

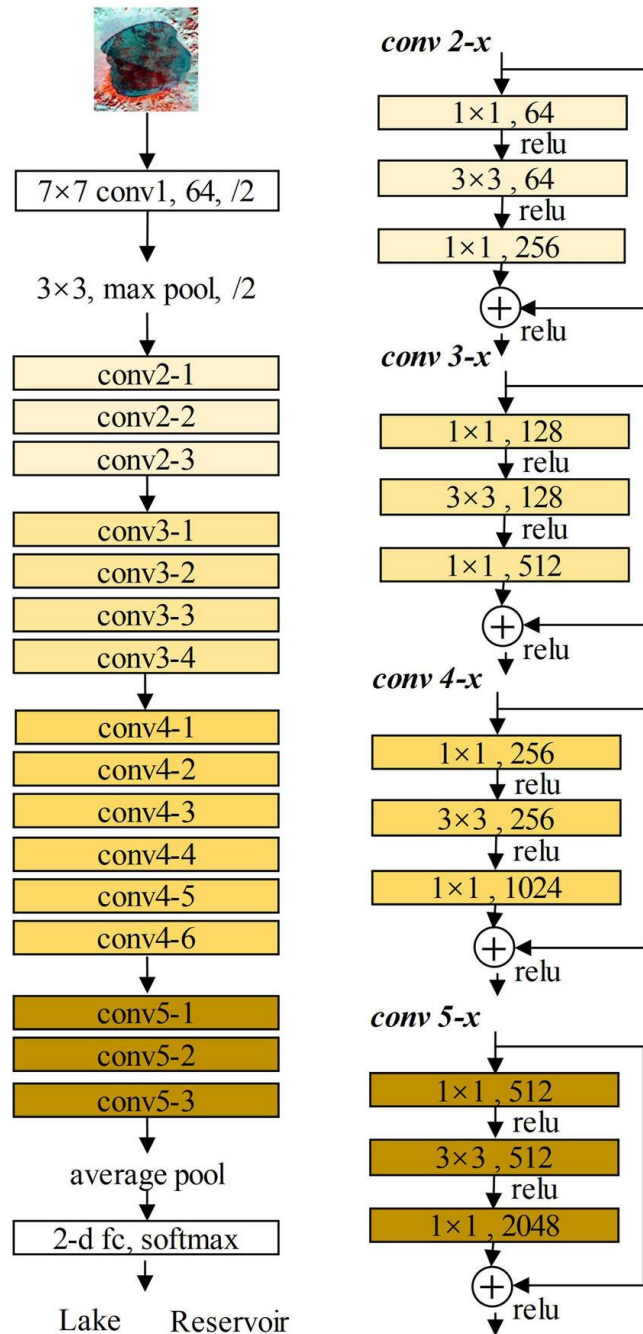


รูปที่ 10 ภาพโครงสร้าง VGG16

2.3.2 ResNet50-v2

ResNet (7) มาจาก Deep residual Network ถูกนำเสนอในงานวิจัย Deep residual learning for image recognition ซึ่งแก้ปัญหาเรื่อง vanishing gradient ซึ่งเกิดขึ้นกับโครงข่ายที่มีความลึกค่อนข้างมากซึ่งมีจำนวนชั้นของ network ถึง 152 เลเยอร์ (8 เท่าของโมเดล VGG16) โดยใช้เทคนิคการออกแบบ module ที่มีลักษณะทางลัดลงใน network ตัวโครงข่ายนี้ประกอบด้วย 4 block ตามรูปที่ 15 โดยจำนวนที่มีพารามิเตอร์สำหรับฝึกทั้งหมดคือชั้นที่เราใช้เรียกชื่อ เช่น

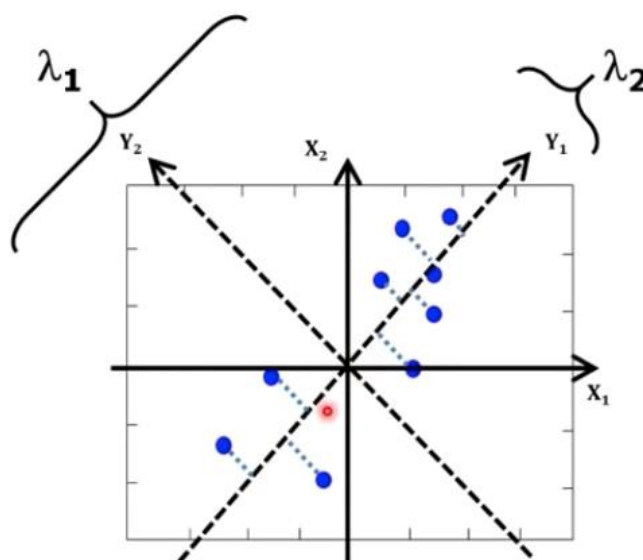
ResNet50 จะหมายถึงจำนวน 50 เลเยอร์ ซึ่งจะอธิบายขนาดว่า [3, 4, 6, 3] ซึ่งคือ $(3 + 4 + 6 + 3) \times 3 = 48$ ชั้น + 2 ชั้น = 50 ซึ่ง ResNet ที่นิยมใช้จะเป็น ResNet18, ResNet34, ResNet50, ResNet101 และ ResNet152 ซึ่งในงานวิจัยนี้ได้ใช้ ResNet50 อ้างอิงงานวิจัย [2] ดังรูปที่ 11



รูปที่ 11 ResNet50-v2 architecture

2.4 ทฤษฎีการวิเคราะห์องค์ประกอบหลัก (Principal Component Analysis : PCA)

เป็นการปรับแกนเพื่อทำให้องค์ประกอบที่สัมพันธ์กันมาอยู่ด้วยกัน (10) เพื่อสามารถลดมิติได้ หมายถึง ทำให้เก็บข้อมูลขนาดเล็กลง และลดสิ่งรบกวนหรือ noise เราเข้าใจมันง่ายขึ้นและส่งผลให้ความกะเน่ เนื่องจากบางแกนมีความแตกต่างของข้อมูลน้อย หรือแกนไหนมีความแตกต่างของข้อมูลที่มาก ทำให้เราสามารถตัดแกนที่มีความแตกต่างข้อมูลที่น้อยนั้นออกไป โดยไม่คำนึงถึงได้ การลดนี้ทำให้ลดความซับซ้อนและขนาดข้อมูลได้ ดังรูปที่ 12



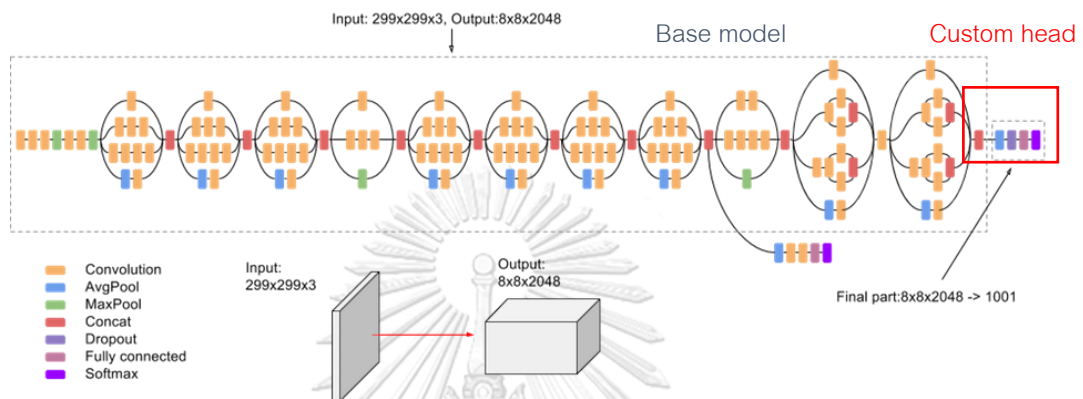
รูปที่ 12 แสดงความสัมพันธ์ PCA

จากภาพแกน x_1 และ x_2 ข้อมูลอยู่ตามแนวทแยงซึ่งถ้าสังเกตข้อมูลเกาะอยู่ทั้งในแนวของ x_1 มาก x_2 ก็จะไม่มากด้วย จะสามารถทำได้โดยการหมุนแกนจาก x_1 ไป y_1 และ x_2 ไปยังแกน y_2 ก็ จะพบว่าแกน y_1 นั้นมีความสำคัญโดยข้อมูลจะมี variance ในแนวที่แคบ $\lambda_2 > \lambda_1$ ซึ่งสามารถสรุปได้ว่า y_2 ไม่มีความสำคัญสามารถตัดทิ้งได้ (เราต้องหมุนแกน เพื่อให้ได้ค่าว่าแกนหนึ่ง มีการกระจัดกระจายมาก ส่วนอีกแกนมีการกระจัดกระจายน้อย โดยเราจะทำการตัดแกนที่มีการกระจัดกระจายน้อยทิ้ง) การคำนวณ coordinate จากการหมุนแกน

ซึ่งการปรับแกนเพื่อทำให้องค์ประกอบที่สัมพันธ์กันมาเกี่ยวพันกัน จะสามารถลดมิติของ ข้อมูลได้ เนื่องจากบางแกนมีความแตกต่างกันข้อมูลน้อยทำให้เราสามารถตัดแกนนั้นออกไปได้ การลดมิติทำให้ให้เราลดความซับซ้อนและขนาดของข้อมูลได้ การตัดสัญญาณรบกวน และเป็นการเพิ่มความถูกต้องของข้อมูลมากขึ้นได้

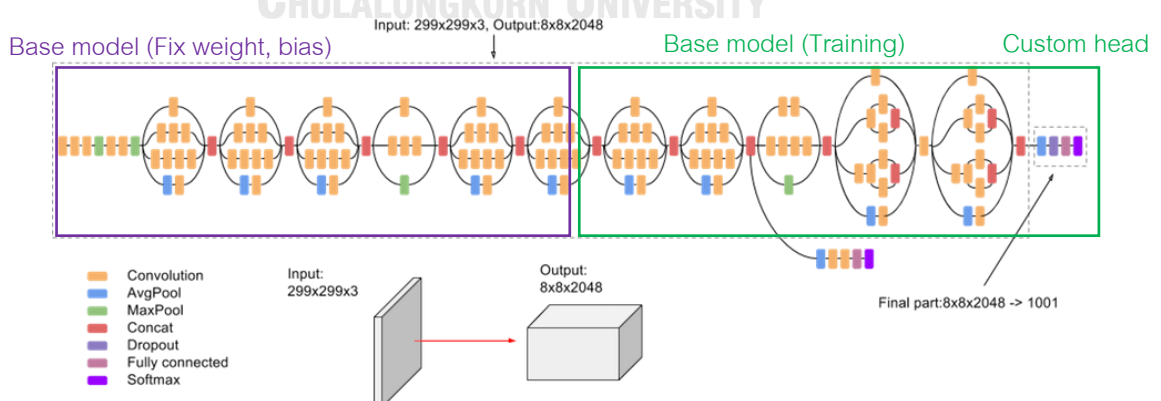
2.5 ทฤษฎีการปรับแต่ง Fine-tuning

ในที่นี้จะขอกล่าวถึงการใช้ตัวอย่างการ Transfer learning โดยใช้ model Inception-v3 ซึ่งสามารถโหลดได้จาก <https://keras.io/api/applications/inceptionv3/> ซึ่งจะประกอบด้วย 2 ส่วนคือ base model และ head model โดยมีคำสั่งที่ใช้ในการโหลด base model inception v3 เข้ามาใช้งานดังนี้ กรณีเปลี่ยน head model ให้เปลี่ยน include top จาก true เป็น ดังรูปที่ 13



รูปที่ 13 โครงสร้าง networks ของโมเดล Inception-v3 ที่มีการ custom head

จากนั้นจึงทำการเพิ่ม layers ที่เราต้องการและทำการฝึกแบบ fix weight และ bias ในส่วนของ base model และทำการตรวจสอบค่าความแม่นยำของโมเดลที่เพิ่ม head layers เข้าไป ซึ่งกรณีที่มีความแม่นยำต่ำเราอาจทำการ Fine-tuning ซึ่งการ Fine-tuning นี้เหมาะกับการใช้งานใหม่และใช้ dataset ใหม่เพื่อให้โมเดลมีประสิทธิภาพที่ดีขึ้น โดยอาจเทรนในบางส่วนของ base model หรือทั้งหมดของ network เลยก็ได้ดังรูปที่ 14



รูปที่ 14 โครงสร้าง networks ของโมเดล Inception-v3 ที่มีการ fine-tuning

2.6 เอกสารและงานวิจัยที่เกี่ยวข้อง

งานวิจัย Classification of crystallization outcomes using deep convolutional neural networks (1) ได้ทำการค้นคว้าภาพผลึกจาก 5 แหล่งข้อมูล โดยใช้ข้อมูลประมาณเกือบครึ่งล้านรูปมาทำการจำแนกประเภทโดยใช้ Convolutional neural network การใช้เครื่องจักรมาเรียนรู้จัดจำผลึก ได้รวบรวมรูปการทดลองผลึกโมเลกุลขนาดโมเลกุลประมาณครึ่งล้านรูปจากหลายแหล่ง โดยอัลกอริทึมการเรียนรู้ของเครื่องที่ทันสมัยได้รับการเทรน และทวนสอบในส่วนต่างของชุดข้อมูลนี้ พบว่ามีรูปภาพการทดสอบมากกว่า 94% สามารถทำนายประเภทของผลึกได้อย่างถูกต้อง

งานวิจัย Monitoring sugar crystallization with deep neural networks (13) ทำการค้นคว้ารูปผลึกจากโรงงานน้ำตาลในประเทศจีน ซึ่งเข้าได้ใช้ภาพผลึกจำนวน 1,000 ภาพ จากนั้นให้ช่างเคียวที่มีประสบการณ์ 3 คนลงความเห็นในการแบ่ง classes ผลึกน้ำตาลเป็น 5 classes โดยใช้โมเดล Transfer learning ทั้งหมด 4 โมเดล ซึ่งโมเดล ResNet50-v2. มีความแม่นยำสูงสุดคือสามารถทำนายผลึกทั้ง 5 classes อยู่ระหว่าง 83% ถึง 99% โดยนำมาประยุกต์ใช้ในงานวิจัยคือรูปแบบงานวิจัยโดยรวมเหมือนกัน ซึ่งมีการนำใช้ model มาเทรนซึ่งเป็นโมเดลสร้างเองไว้เปรียบเทียบ 1 โมเดล simple DCNNs ส่วน 3 โมเดลเป็นการโหลดมาจาก keras ใช้ข้อมูลภาพผลึก 1,000 ภาพ แบ่ง classes โดย training set 80% และ testing set 20%

งานวิจัย Techniques for classifying sugar crystallization images based on spectral and the use of neural networks (6) วิธีการเก็บตัวอย่างของผลึกน้ำตาล โดยใช้การถ่ายภาพสเปกตรัม จากกระบวนการผลิตจริงที่โรงงานในประเทศสเปน โดยได้นำค่าสเปกตรัมมาเข้า Neural network แบบ Learning vector quantization เพื่อแยกประเภทผลึก output 4 classes (two sizes, deformed crystals, homogeneous and three or more size

บทที่ 3

Method of Data Collection

3.1 การรวบรวมข้อมูล

ทำการเก็บข้อมูลตัวอย่างของผลึกน้ำตาล โดยใช้บันทึกภาพผลึกน้ำตาลการถ่ายภาพส่องผ่าน sight glass และผ่านแว่นขยายขนาด 10 เท่า ที่ระดับกึ่งกลางของหม้อเคี้ยวตผลึกน้ำตาล (อ้างอิงตำแหน่งจากทดสอบของประเทศญี่ปุ่นซึ่งใช้ระบบส่องผลึกแบบ digital image processing) จากนั้นใช้ช่างเคี้ยวที่ชำนาญในการเคี้ยวตผลึกน้ำตาลจำนวน 5 คน มาทำการติด label เพื่อแยกประเภทภาพถ่ายผลึกน้ำตาล โดยมีจำนวนของภาพผลึกน้ำตาลที่ใช้ทั้งหมด (data set) 1,000 ภาพ แบ่งเป็นชุดข้อมูลฝึก (training set) 800 ภาพ และชุดข้อมูลทดสอบ (testing set) 200 ภาพโดยการถ่ายรูปผลึกจากนั้นนำภาพผลึกนั้น มาคัดแยกโดยช่างเคี้ยวที่มีประสบการณ์เพื่อแบ่งผลึกออกมา 4 classes โดยแยกตามโซนของสภาวะการเกิดผลึกสัมประสิทธิ์ความอิ่มตัวยิ่งยวด (supersaturation coefficient) คือ unsaturated, metastable, intermediate และ labile จำนวนภาพถ่ายผลึกที่เก็บได้ในแต่ละโซนนั้นควรจะเก็บอย่างน้อยโซนละประมาณ 250 ภาพ ดังแสดงในตารางที่ 1

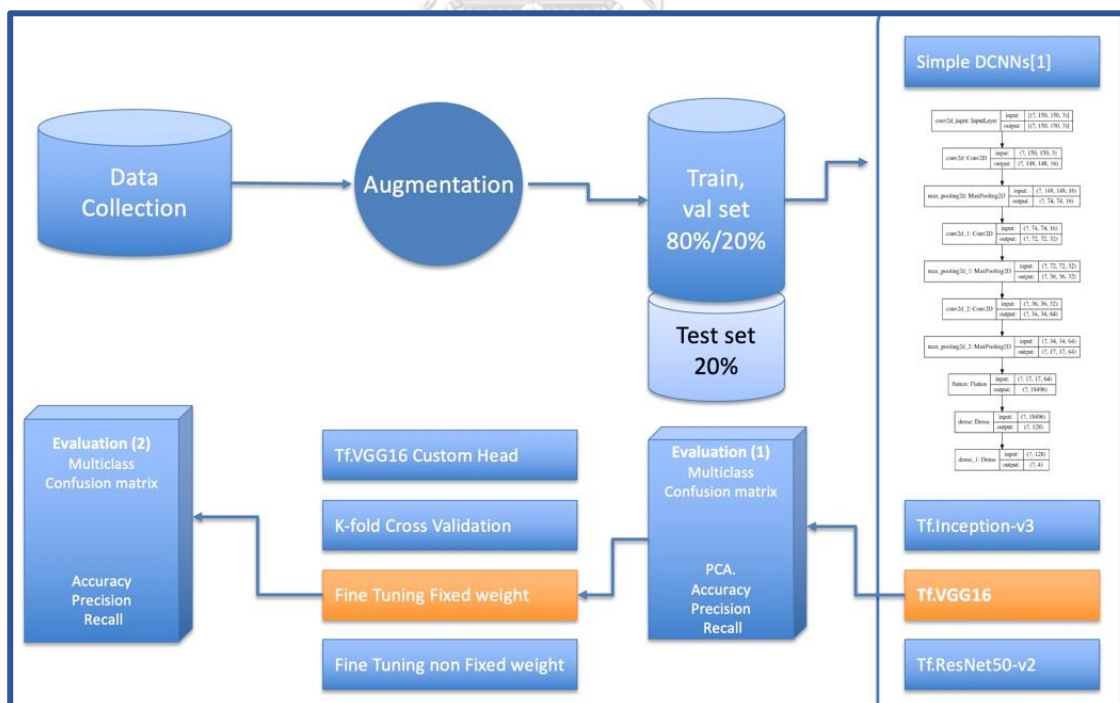
ตารางที่ 1 ข้อมูล class ของประเภทผลึกที่ทำการแยกและจำนวนรูปผลึกที่ทำการเก็บข้อมูล

| Class | No. Images | |
|-------------------|------------------|-----------------|
| | Training Dataset | Testing Dataset |
| Unstaturated Zone | 200 | 50 |
| Metastable Zone | 200 | 50 |
| Intermediate Zone | 200 | 50 |
| Labile Zone | 200 | 50 |

3.2 การเตรียมข้อมูล Data Preparation and Image augmentation

เพื่อเพิ่มประสิทธิภาพในชุดข้อมูลฝึก (Training set) 80% โดยขั้นตอนดังรูปที่ 15 มาปรับเพิ่มคุณภาพรูปภาพเพื่อเพิ่มประสิทธิภาพของชุดข้อมูลฝึกโดยใช้ Tensorflow import keras ผ่านฟังก์ชัน image data generator ซึ่งเป็นการทำ real-time augmentation โดยลดภาระ Ram ของเครื่องระหว่างการ augmentation ซึ่งกำหนดค่าการดังนี้

- ทำการปรับ Rotation range = 45 degree
- ทำการปรับ Width shift range = 20%
- ทำการปรับ Height shift range = 20%
- ทำการปรับ Brightness range = 20-40%
- ทำการปรับ Shear range = 20%
- ทำการปรับ Horizontal flip = True
- ทำการปรับ Vertical flip = True



รูปที่ 15 ภาพรวมขั้นตอนการ augmentation และดำเนินการทดลอง

3.3 ขั้นตอนการสร้างชุดข้อมูลสำหรับการทำ data set

3.3.1 นำข้อมูลรูปภาพทั้งหมด upload ขึ้น google drive จากนั้นทำการสร้าง file dataset สร้างชุดข้อมูลโดยใช้สร้างโปรแกรมในการคัดแยกภาพผลึกทั้งหมดโดยใช้ IDE colab และทำการ mount google drive โดยสั่งให้สร้าง directory = sugar_dataset_kaggle และ subdirectory = train/ และ test/ จากนั้นทำการสร้าง part ใน train และ test folder แยกตาม classes ทั้ง 4 ดังรูปที่ 16

```
# Create_directories

dataset_home = '/content/drive/My Drive/master_project/sugar_dataset_kaggle/'
subdirs = ['train/', 'test/']

for subdir in subdirs:
    #create label subdirectories
    labeldirs = ['intermediate', 'unsaturated', 'Metastable', 'labile']

    for labldir in labeldirs:
        newdir = dataset_home + subdir + labldir
        makedirs(newdir, exist_ok=True)
```

รูปที่ 16 แสดงขั้นตอนการแยก path data set ในขั้นตอนการเตรียมข้อมูล

```
# define ratio of picture to use for validation
val_ratio = 0.2

# copy training dataset images into subdirectories
src_directory = '/content/drive/My Drive/master_project/labeling_data2/'

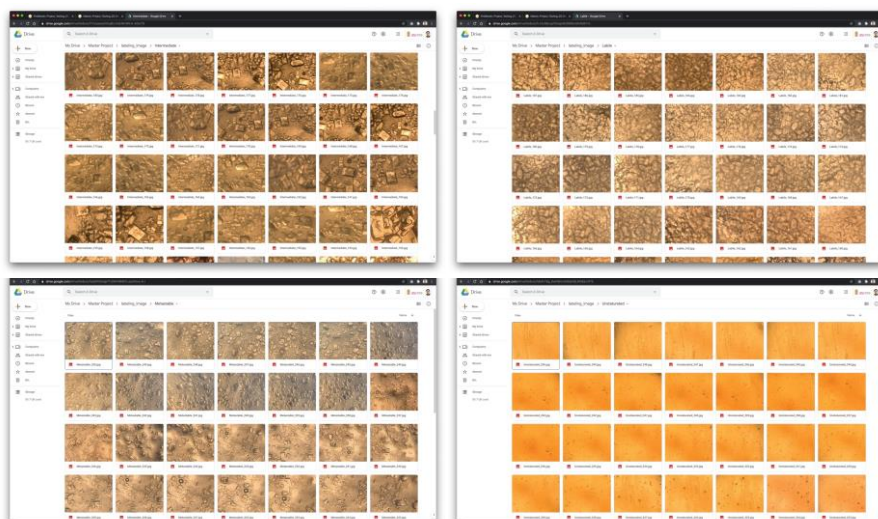
for file in listdir(src_directory):
    src = src_directory + '/' + file
    dst_dir = 'train/'

    if random() < val_ratio:
        dst_dir = 'test/'

    if file.startswith('Intermediate'):
        dst = dataset_home + dst_dir + 'Intermediate/' + file
        copyfile(src, dst)
    elif file.startswith('Unstaturated'):
        dst = dataset_home + dst_dir + 'Unstaturated/' + file
        copyfile(src, dst)
    elif file.startswith('Metastable'):
        dst = dataset_home + dst_dir + 'Metastable/' + file
        copyfile(src, dst)
    elif file.startswith('Labile'):
        dst = dataset_home + dst_dir + 'Labile/' + file
        copyfile(src, dst)
```

รูปที่ 17 แสดงโปรแกรมการแบ่งข้อมูลโดยการสุ่มเพื่อสร้าง data set สำหรับฝึกและทดสอบ

ซึ่งจะทำการไปดึงข้อมูลมาจาก path labeling_data2 ซึ่งทำการอ่านไฟล์ .jpg [name_classes] . 1-250.jpg มาสร้างเป็น data set โดยกำหนด seed เพื่อป้องกันภาพซ้ำกันในการสุ่มชุดข้อมูล และทำการแบ่ง train set 80% และ test set 20% ดังรูปที่ 17 เมื่อแบ่งชุดข้อมูลเรียบร้อยแล้วจะได้ข้อมูลดังภาพด้านล่าง ซึ่งเป็นตัวอย่างของ train set ดังรูปที่ 18

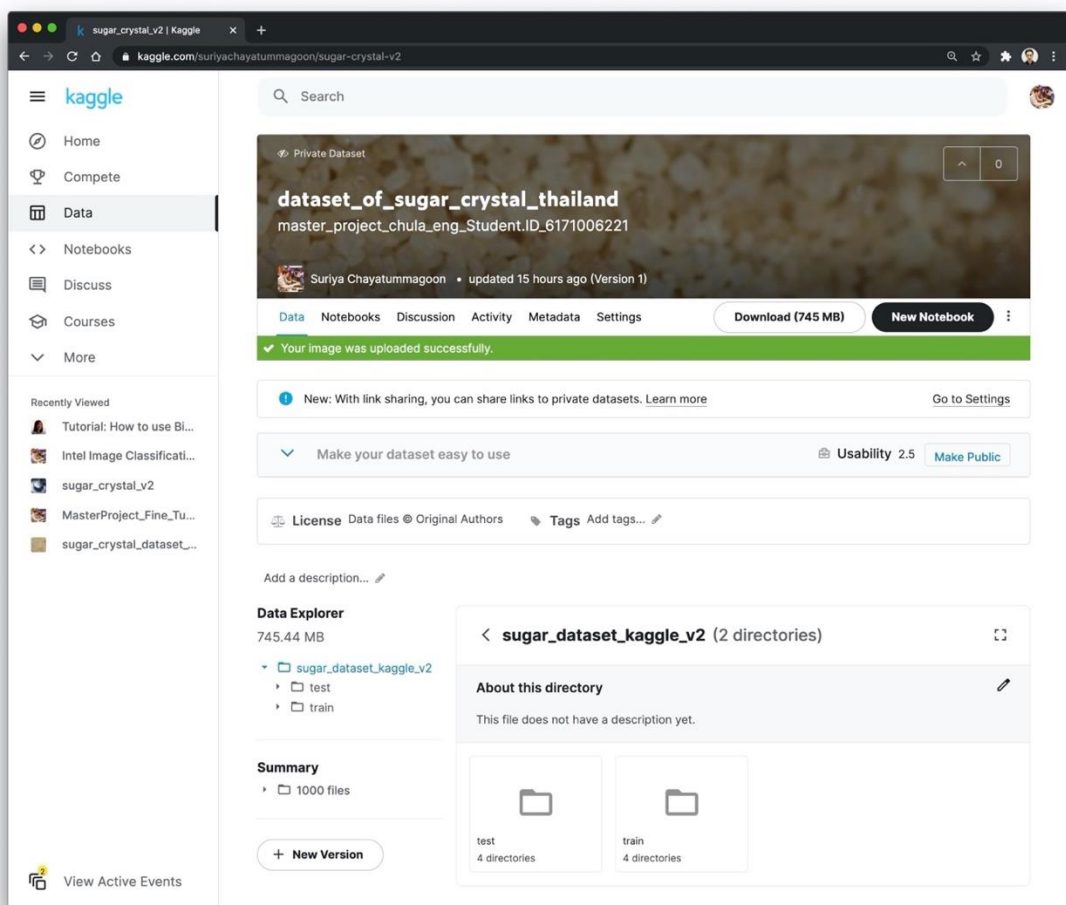


รูปที่ 18 แสดงข้อมูล train set ภายในทั้ง 4 classes จากโปรแกรมสุ่มแบ่งภาพผลึก



3.3.2 การนำข้อมูลเข้า Kaggle Competition

โดยทำการ zip และ upload ขึ้นบน Cloud ของ Kaggle Private เพื่อสร้าง Project และ IDE. เนื่องจาก Kaggle มีการเก็บข้อมูลที่มีความไวในการอ่านสูงเพราะข้อมูลถูกเก็บในลักษณะ cloud storage ได้ด้วยเพื่อสะดวกต่อการดู notebook และการตามรอยเอกสารรวมถึงรองรับการทำงานเป็นทีมด้วย โดยใน project ดังรูปที่ 19 และเมื่อเสร็จ Project สามารถ deploy environment ผ่านการ download ทั้ง Docker image ซึ่งสามารถได้จาก link ของ Project ของเราได้ทันทีเพื่อนำไปใช้งานต่อไป



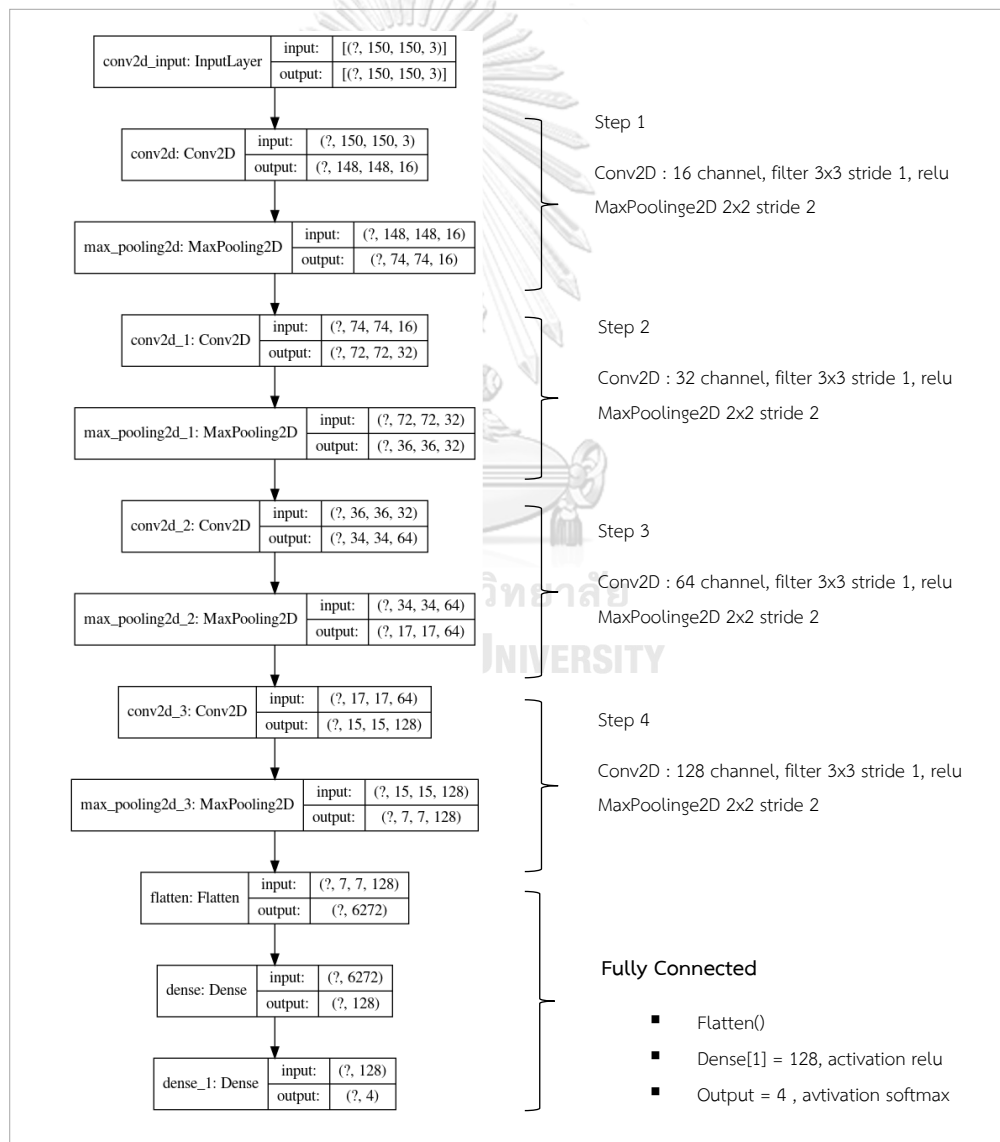
รูปที่ 19 แสดงข้อมูลการสร้าง Project บน Kaggle หลังจาก upload data set

บทที่ 4

Create Model Simple DCNNs and Transfer Learning

4.1 Simple DCNNs.

ใช้ภาษา python 3.8 , tensorflow 2.3, keras 2.4.3 บน Cloud Kaggle โดยใช้ TPU v3-8 (tensor core processor unit 16 GB HBM twin MXUs) การออกแบบโมเดล simple DCNNs (4) มี Input layer = 150x150x3 จากนั้นเข้ากระบวนการ convolution โดยออกแบบเป็น 4 step ซึ่งใช้ conv2d มี filter ขนาด 3x3 โดยขั้นที่ละ strids=1 จำนวน layer 16 ,32, 64,128ตามลำดับ โดยทุก step จะทำการลดขนาดมิติผ่าน MaxPooling2D ที่มีขนาด filter=2x2

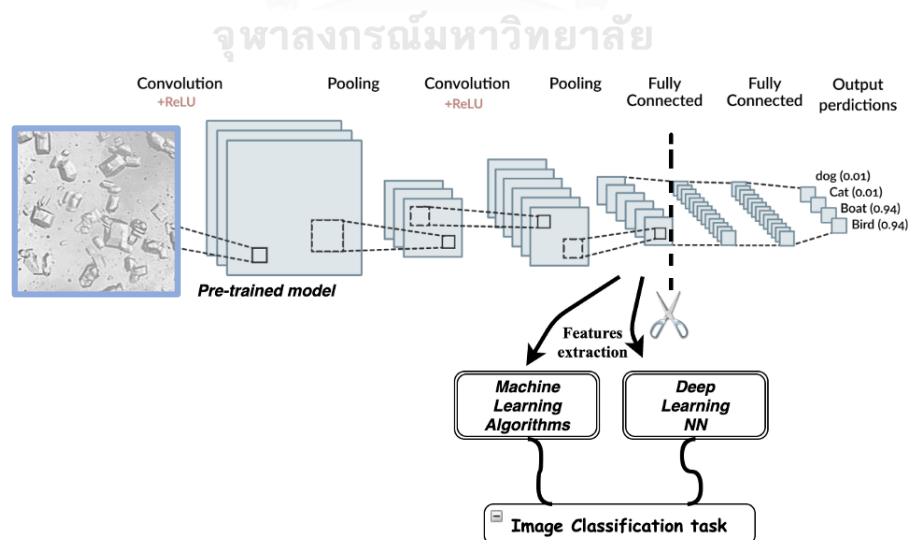


รูปที่ 20 ภาพรวมของโครงสร้างโมเดล Simple DCNNs

ในส่วนของ Fully connected จะทำโดยผ่านการ flatten() ซึ่งจะเปลี่ยนชนิดข้อมูลให้เป็น vector ที่เป็น array ขนาด 1 มิติ และส่งเข้า network โดยออกแบบ layer แรกเป็น dense 128 โดยใช้ activation = relu และ output layer เป็น dense 4 โดยใช้ activation softmax ในส่วนของการ compile model โดยใช้ optimizer = adam และ batch size = 64 และมีการคิด loss แบบ sparse categorical crossentropy เพื่อให้ค่า output เป็น onehot encoding วัดประสิทธิภาพระหว่างการฝึก แต่ในรอบโดยใช้ metrics = accuracy และ learning rate ที่ 0.001 และการ fit โดยแบ่งข้อมูลแบบ validation set = 20% และ batch size = 64 กำหนดการฝึก 50 epochs และกำหนดสิ้นสุดการฝึกโดยใช้ early stopping โดยดูจากค่า loss ใน validation set ไม่ลดลงใน 3 รอบสุดท้ายของการฝึก (patience = 3)

4.2 Pre-trained 3 models

การทดลองที่สองใช้การถ่ายทอดการเรียนรู้[8] การฝึกโมเดลการเรียนรู้เชิงลึกที่ซับซ้อน (transfer learning) ตั้งแต่เริ่มต้น (การเริ่มต้นด้วยน้ำหนักด้วยค่าสุ่ม) ต้องใช้ทั้งข้อมูลการฝึกอบรมขนาดใหญ่และพลังในการคำนวณสูง ดังนั้นการใช้การเรียนรู้แบบถ่ายทอดจึงเป็นเทคนิคที่ได้รับความนิยมเนื่องจากจะช่วยลดเวลาในการฝึก โมเดลขนาดใหญ่ที่ผ่านการฝึกแล้วบางส่วนกับงานที่คล้ายกันเป็นส่วนหนึ่งของโมเดลใหม่ ผลลัพธ์ของโมเดลที่ได้รับการ pre-trained จะถูกใช้เพื่อดึงคุณสมบัติผ่านตัวแปรใหม่แบบเชื่อมทุกเลเยอร์ (fully dense) จะถูกดึง latent feature (สร้างโดยโมเดล transfer learning ผ่านการ load weight จาก imagenet) คุณสมบัติเหล่านี้ใช้เป็นอินพุตสำหรับโมเดล machine learning ในชั้น fully connected ดังรูปที่ 21



รูปที่ 21 การวิเคราะห์ latent feature จาก pre-trained model

4.3 Principal Component Analysis: PCA

เมื่อดึงคุณสมบัติแฝง latent feature ออกมาเปรียบเทียบกับข้อมูลที่เราทำการ label ทั้ง 4 classes และจากแกน decomposition : n_components แบบ 2 มิติ PCA จึงถูกนำไปใช้เพื่อหลีกเลี่ยง (ลดมิติอย่างน้อยลงของแกนที่ปรับแล้ว) จากนั้นเราทำการสังเกตและเลือกโมเดลที่เราต้องการผ่านการเทียบประสิทธิภาพของการทำนายของทั้ง 3 เมื่อเราใช้ transfer learning ที่แตกต่างกันสามแบบ คือ inception-v3 ,VGG16 และ ResNet50-v2

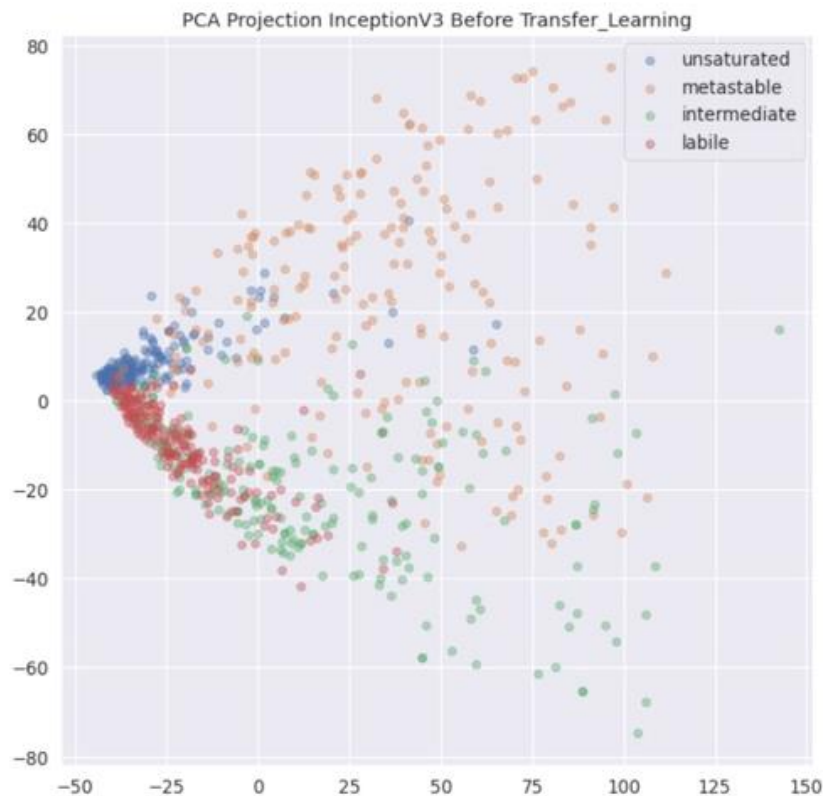
4.3.1 Inception-v3 pre-trained model and PCA.

เอาค่าน้ำหนักจาก ImageNet แล้วนำมา pretrained model ส่งค่า feature vector ออกของ model inception-v3 โดยคิดจาก width, height, channel คือค่า x, y และ z ตามลำดับนำมาคูณกับ $3 \times 3 \times 2048 = 18,432$ มาทำการวิเคราะห์ PCA ก่อนเข้าจากการโมเดล pretrained มาทำการ custom head ดังรูปที่ 22

```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
87916544/87910968 [=====] - 1s 0us/step
shape of train,test_image input : (800, 150, 150, 3) (200, 150, 150, 3)
shape of train,test_feature output : (800, 3, 3, 2048) (200, 3, 3, 2048)

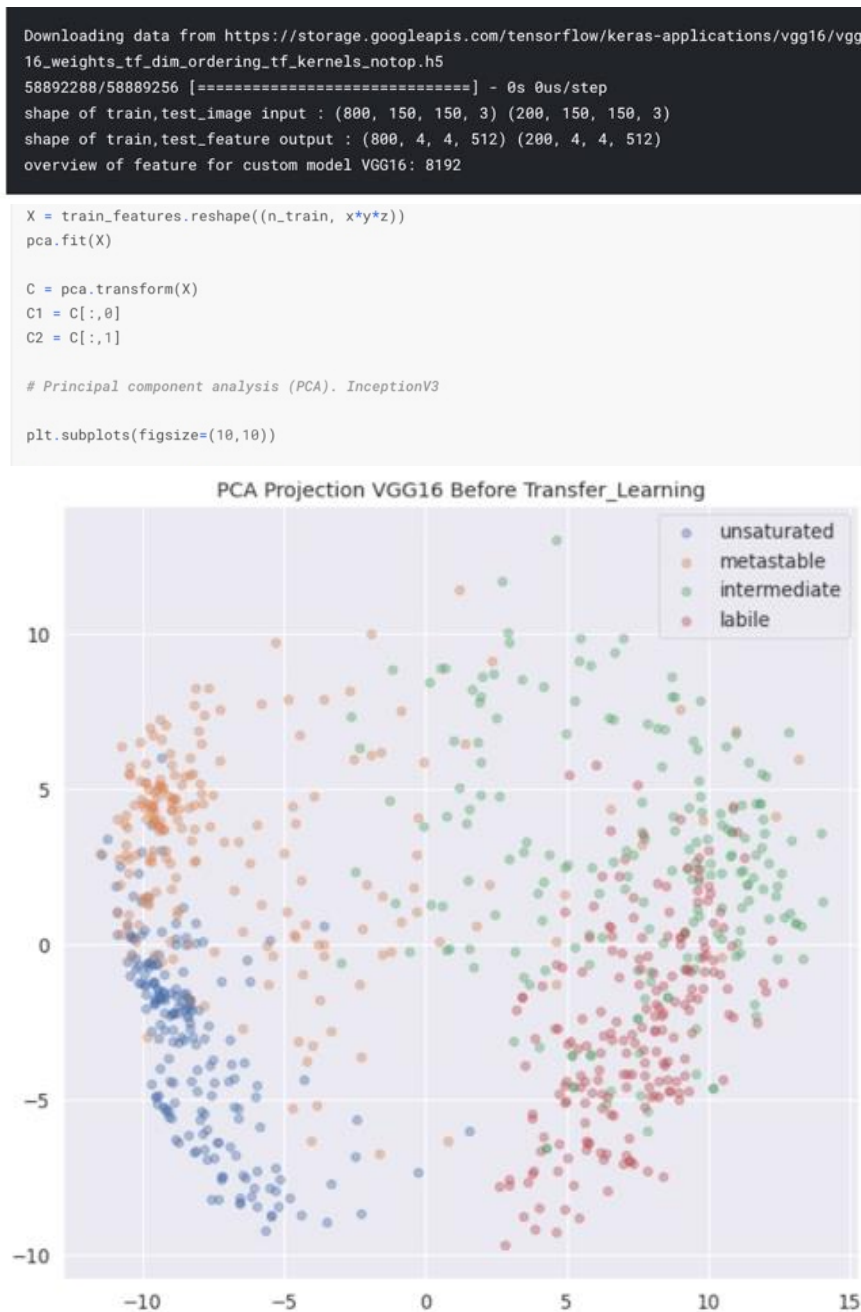
```



รูปที่ 22 PCA Projection of pretrained model inception-v3

4.3.2 VGG16 pre-trained model and PCA.

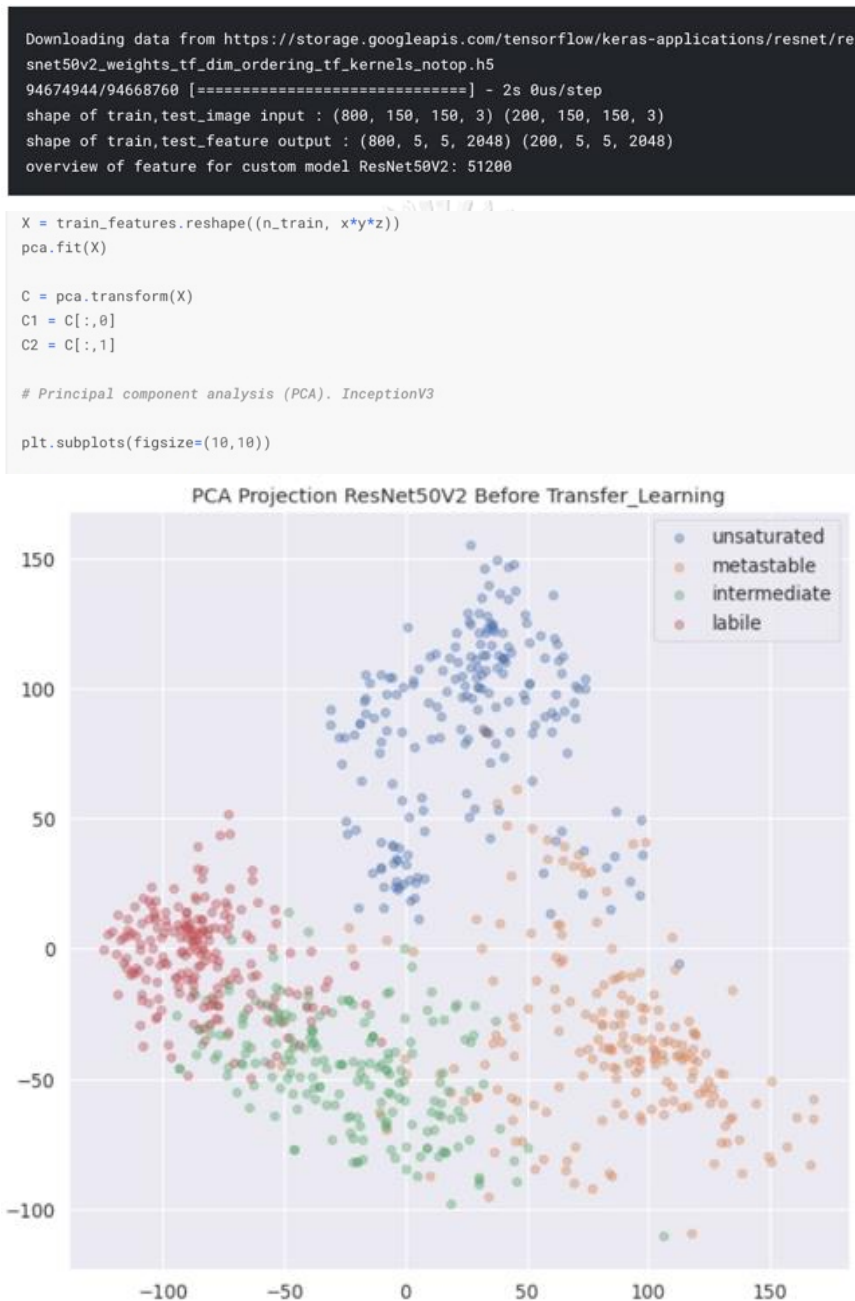
เอาค่าน้ำหนักจาก ImageNet แล้วนำมา pretrained model ส่งค่า feature vector ขาออกของ model inception โดยคิดจาก width, height, channel คือค่า x, y และ z ตามลำดับ นำมาคูณกับ $4 \times 4 \times 8192 = 8,192$ มาทำการวิเคราะห์ PCA ก่อนเข้า Custom head ดังรูปที่ 23



รูปที่ 23 PCA Projection of pretrained model VGG16

4.3.3 ResNet50-v2 pre-trained model and PCA.

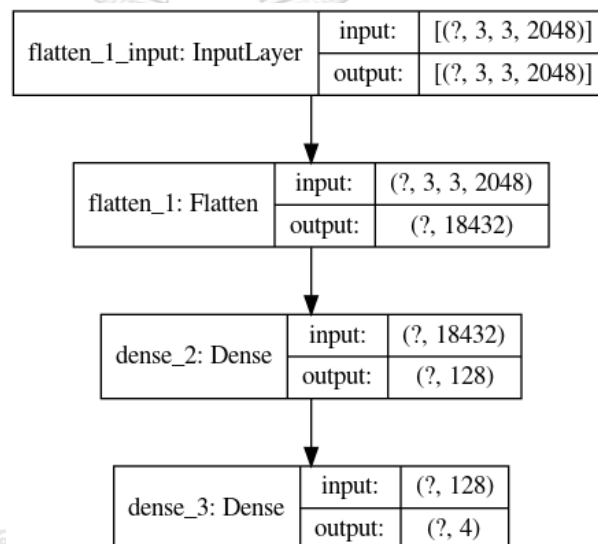
เอาค่าน้ำหนักจาก ImageNet แล้วนำมา pretrained model ส่งค่า feature vector ขาออกของ model inception โดยคิดจาก width, height, channel คือค่า x, y และ z ตามลำดับ นำมาคูณกับ $5 \times 5 \times 2048 = 51,200$ มาทำการวิเคราะห์ PCA ก่อนเข้า Custom head ดังรูปที่ 24



รูปที่ 24 PCA Projection of pretrained model ResNet50-v2

4.4 Custom head

การนำโมเดล inception-v3, ResNet50-v2 และ VGG16 จากขั้นตอน pre-trained โดยใช้ weight จาก ImageNet มาทำการ custom head และทำการ flatten โดยเพิ่มชั้น Dense 128 โดยใช้ activation = relu และกำหนดชั้นสุดท้ายเป็น dense 4 เพื่อกำหนด output โดยใช้ activation softmax ทำการ validation split = 20% และ compile model โดยใช้ optimizer = adam และ batch size = 64 และมีการใช้ loss แบบ sparse categorical_crossentropy เพื่อให้ค่า output เป็น onehot encoding วัดประสิทธิภาพระหว่างการฝึก แต่ละรอบโดยใช้ metrics = accuracy และ learning rate ที่ 0.001 และ ทำการฝึก 50 epochs และกำหนดสิ้นสุดการฝึกโดยใช้ earlyStopping โดยดูจากค่า loss ใน validation set ไม่ลดลงใน 3 รอบสุดท้ายของการฝึก (patience = 3) ดังรูปที่ 25



รูปที่ 25 แสดงโครงสร้าง custom head ที่เป็น fully connected ของ Inception-v3

บทที่ 5

ผลการวิเคราะห์ข้อมูล Simple DCNNs และ Transfer Learning

เมื่อทำการฝึกจนครบทั้ง 4 Simple DCNNs, inception-v3, VGG16 และ ResNet50-v2 นำค่า accuracy และ loss ระหว่างการฝึกมาแสดงความสัมพันธ์ในแต่ละ Step ของ epoch ที่ทำการฝึกระหว่างค่า accuracy และค่า loss ในชุด train และ validation ดังรูปที่ 26

5.1 สร้างสร้าง function ที่ใช้ในการ plot accuracy และ loss ดังนี้

```
def plot_accuracy_loss(history):
    """
        Plot the accuracy and the loss during the training of the nn.
    """
    fig = plt.figure(figsize=(24,16))

    # Plot accuracy
    plt.subplot(221)
    plt.plot(history.history['accuracy'], 'bo--', label = "acc")
    plt.plot(history.history['val_accuracy'], 'ro--', label = "val_acc")
    plt.title("train_acc vs val_acc")
    plt.ylabel("accuracy")
    plt.xlabel("epochs")
    plt.legend()

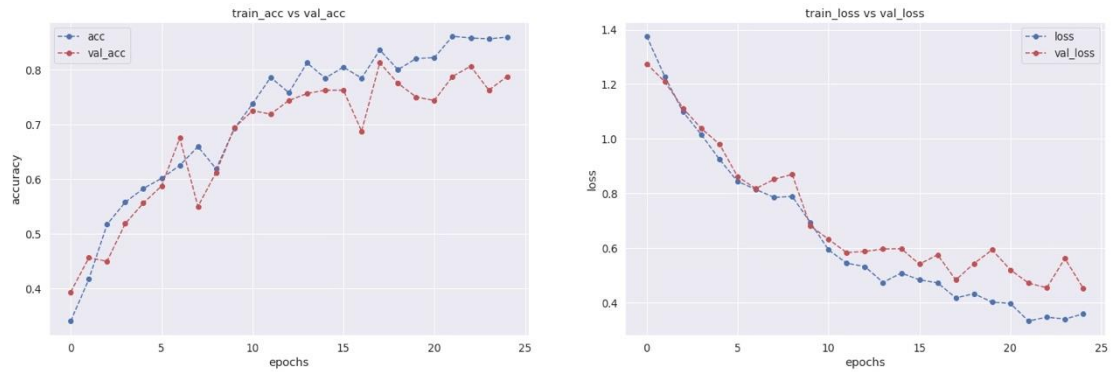
    # Plot loss function
    plt.subplot(222)
    plt.plot(history.history['loss'], 'bo--', label = "loss")
    plt.plot(history.history['val_loss'], 'ro--', label = "val_loss")
    plt.title("train_loss vs val_loss")
    plt.ylabel("loss")
    plt.xlabel("epochs")

    plt.legend()
    plt.show()
```

รูปที่ 26 แสดง function ในการ plot กราฟแสดงความสัมพันธ์ accuracy และ loss

5.2 วิเคราะห์ความสัมพันธ์จากข้อมูลหลังทำการฝึก

5.2.1 Simple DCNNs ดังรูปที่ 27



รูปที่ 27 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ epochs ของ Simple DCNNs

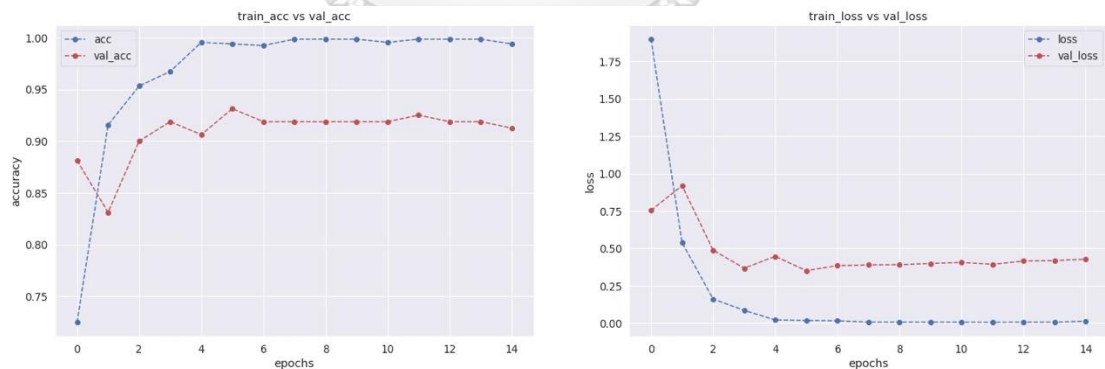
Simple DCNNs มีค่า

Accuracy_validation (ความแม่นยำ = 0.7875)

Loss_validation (ค่าความผิดพลาด = 0.4544)

5.2.2 inception-v3-tf custom head

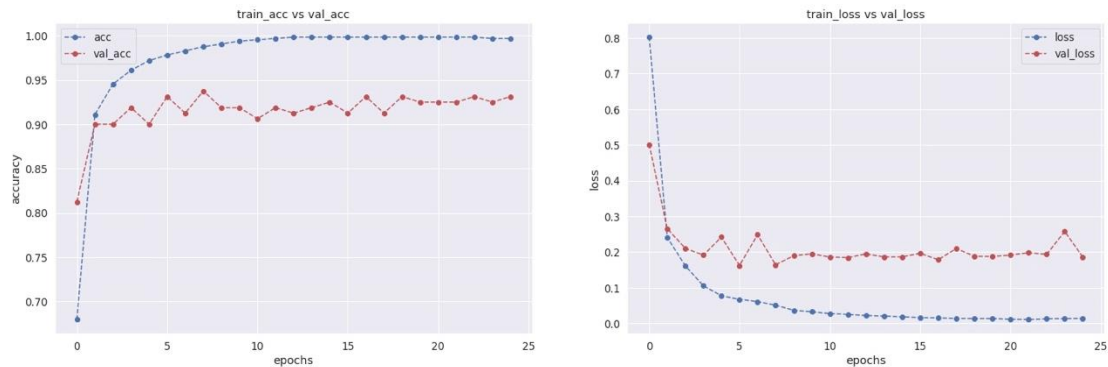
Accuracy_val (ความแม่นยำ = 0.9125), Loss_val (ค่าความผิดพลาด = 0.4276) ดังรูปที่ 28



รูปที่ 28 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ epochs ของ inception-v3

5.2.3 กราฟ VGG16-tf custom head

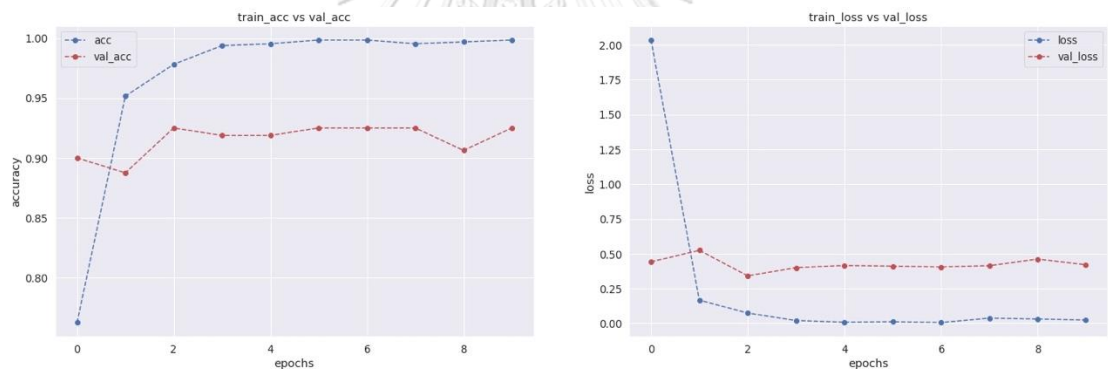
Accuracy_val(ความแม่นยำ = 0.9312), Loss_val (ค่าความผิดพลาด = 0.1860) ดังรูปที่ 29



รูปที่ 29 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ epochs ของ VGG16

5.2.4 กราฟ ResNet50-v2-tf custom head

Accuracy_val (ความแม่นยำ = 0.9250), Loss_val (ค่าความผิดพลาด = 0.4215) ดังรูปที่ 30



รูปที่ 30 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ epochs ของ ResNet50-v2

ตารางที่ 2 เปรียบเทียบประสิทธิภาพการ Simple DCNNs และ Transfer learning 3 models

| Models | Accuracy | Precision | Epochs |
|-----------------|----------|-----------|--------|
| Simples DCNNs | 60.5 | 6.0 | 24 |
| Inception-v3-tf | 78.5 | 78.8 | 14 |
| VGG16-tf | 85.0 | 87.4 | 24 |
| ResNet50-v2-tf | 83.0 | 84.7 | 9 |

บทที่ 6

Improvement Fine-tuning model การเพิ่มประสิทธิภาพโมเดลโดยการปรับแต่ง

6.1 k-fold cross validation

โดยนำโมเดล Transfer Learning VGG16 จากขั้นตอนเบื้องต้นนั้นมาใช้เทคนิคที่นิยมในการสร้างและทดสอบโมเดล Machine Learning โดยใช้การ Resampling ซึ่งจะทำการเป็นข้อมูลเป็นจำนวน k เท่าๆ กันเพื่อทำการสร้างและทดสอบโมเดล (train และ validate) คำนวณค่าเฉลี่ย accuracy (11) ก่อนนำไปทำนายในชุดข้อมูล test set เช่นการแบ่งข้อมูลออกเป็น $k = 10$ ชุด จากนั้นทำการ แบ่งข้อมูลชุดฝึก max_samples = 80% และ validation_set = 20% ซึ่งการศึกษาวิธีการนี้เพื่อจะดูว่าสามารถเป็นการช่วยลด bias ของเวลาที่ฝึกโมเดลตลอดยังลดปัญหาการเกิด overfitting เทียบกับการวิธีอื่นดังรูปที่ 31 และ 32

```
# Number of estimators
n_estimators = 10

# Proportion of samples to use to train each training
max_samples = 0.8

max_samples *= n_train
max_samples = int(max_samples)

models = list()

# ทำการสุ่ม iterate 1-10
for i in range(n_estimators):

    model = tf.keras.Sequential([ tf.keras.layers.Flatten(input_shape = (x, y, z)),
                                  # One layer with random size
                                  tf.keras.layers.Dense(128,activation=tf.nn.relu),
                                  tf.keras.layers.Dense(nb_classes, activation=tf.nn.softmax)
                                ])

    model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics=['accuracy'])

    models.append(model)
```

รูปที่ 31 แสดง Estimator 10 fold การสร้างโมเดลและการตั้งค่า

```

histories = []

for i in range(n_estimators):

    print ("interate",i+1)
    train_idx = np.random.choice(len(train_features), size = max_samples)
    histories.append(models[i].fit(train_features[train_idx], train_labels[train_idx], batch_size=BatchSize, epochs=50, validation_split = 0.2 , callbacks=[callback]))

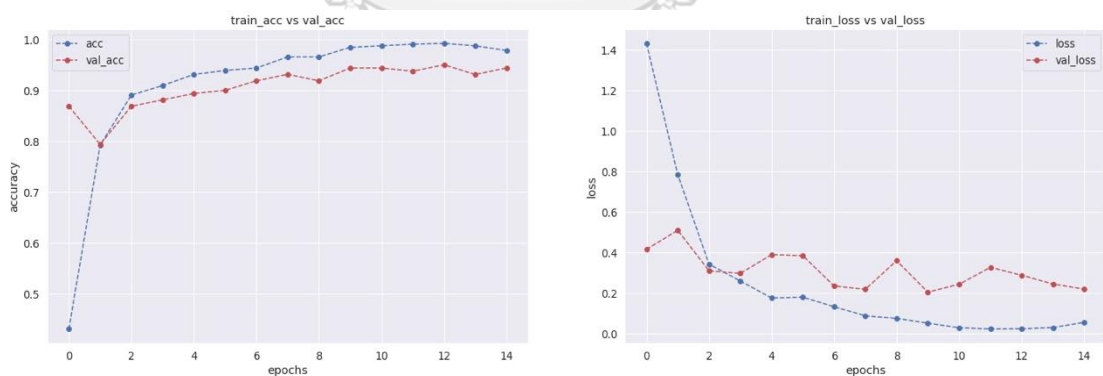
```

รูปที่ 32 คำสั่งวน loop ของ estimator 10 fold model

6.2 Fine-tuning custom model latest 5 layers

เนื่องจากโมเดล VGG16 มีจำนวนเลเยอร์ทั้งหมด 19 เลเยอร์จึงทำการปรับ 5 เลเยอร์สุดท้ายของโมเดลโดยการ ปรับแต่งเลเยอร์นี้จะแปลง 5 เลเยอร์ดั้งเดิมของ VGG16 เป็นขนาด Conv2d = 3x3 pixels และมี 256 channel (layer 1), maxpooling2d ขนาด 2x2 pixels (layer 2), flatten (layer 3), dense = 128 และ activation = relu (layer 4) และ outputs dense = 4 และ activation = softmax โดยใช้ optimizer = adam และมี learning rate ที่ 0.001 และ batch size ขนาด 64 ทำการฝึก 50 epochs และกำหนดสิ้นสุดการฝึกโดยใช้ early stopping ซึ่งในการทดลองจะทำทั้งแบบ fixed-weight และแบบไม่ fixed-weight เพื่อดูประสิทธิภาพของโมเดล

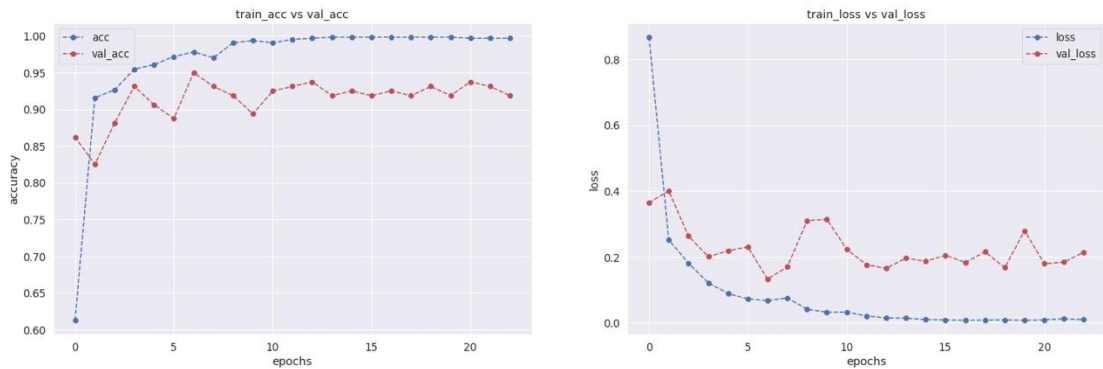
ผลลัพธ์การฝึก ดังรูปที่ 33 Accuracy_val(ความแม่นยำ = 0.9187), Loss_val (ค่าความผิดพลาด = 0.2145)



รูปที่ 33 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ epochs ของ VGG16 fixed weight

6.3 ไม่ Fixed-weight in base model 14 layers

เหมือน 6.2 แต่ไม่ fixed weight ระหว่างการฝึก โดยใช้คำสั่ง `model.trainable = True` โดยผลลัพธ์การฝึกดังรูปที่ Accuracy_val (ความแม่นยำ = 0.9438), Loss_val (ค่าความผิดพลาด = 0.2204) ดังรูปที่ 34



รูปที่ 34 กราฟแสดงความสัมพันธ์ระหว่าง accuracy, loss กับ epochs VGG16 ไม่ fixed weight

โดยสามารถสรุปภาพรวมของการเพิ่มประสิทธิภาพของโมเดล VGG16 จากขั้นตอนแรกเพิ่มหาโมเดลที่มีประสิทธิภาพดีที่สุดของ 4 โมเดลนี้ได้ค่าดังตารางที่ 3

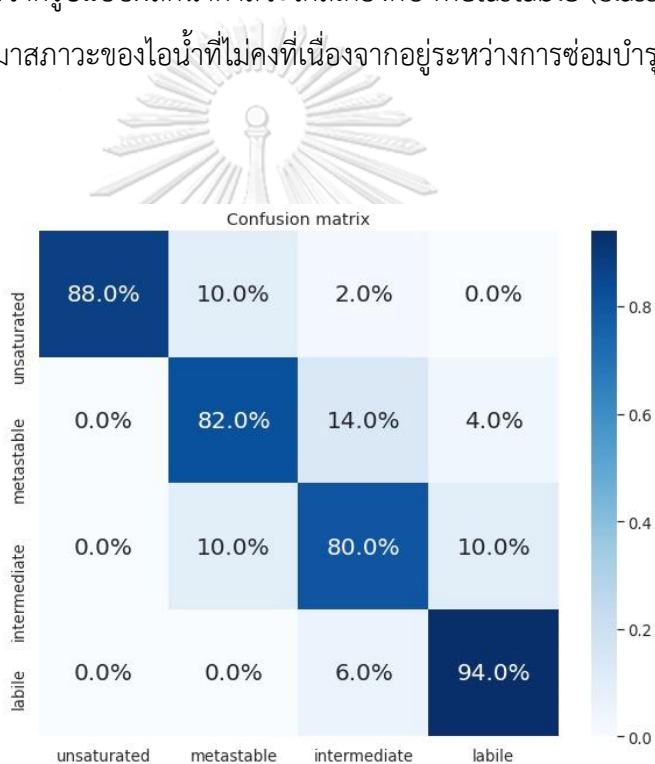
ตารางที่ 3 เปรียบเทียบประสิทธิภาพการ Fine tuning

| Fine-tuning VGG16 models | Accuracy | Precision | Epochs |
|-----------------------------------|----------|-----------|--------|
| a) Custom head | 85.0 | 87.4 | 24 |
| b) Estimator 10 folds | 84.0 | 86.8 | 41 |
| c) Custom last 5 fixed weight | 86.5 | 89.0 | 14 |
| d) Custom last 5 fixed non-weight | 86.0 | 86.5 | 22 |

บทที่ 7

บทสรุป

การทดลองความแม่นยำในการทำนายแม่นยำสุดที่คลาส 4 คือ 94% และคลาส 1 คือ 88% ส่วนคลาส 2 และ 3 เท่ากับ 82% และ 80% ตามลำดับ มีข้อสังเกตคือ unsaturated (class 1) ในช่วงแรกของการทดสอบการเคี้ยวผลึกน้ำตาลนั้นจะมีขนาดผลึกที่แตกต่างจากช่วงที่ metastable (class 2) ส่งผลให้มีการทำนายที่สูงขึ้นเมื่อเทียบ intermediate (class 2) โดยมีค่าความแม่นยำในการทำนายต่ำสุด เนื่องจากรูปแบบผลึกน้ำตาลจะใกล้เคียงกับ metastable (class 3) ดังรูปที่ 35 ซึ่งการเคี้ยวน้ำตาลนี้อาจมาสถานะของไอน้ำที่ไม่คงที่เนื่องจากอยู่ระหว่างการซ่อมบำรุงหม้อไอน้ำหลัก



รูปที่ 35 แสดง confusion matrix multiclass ของ VGG16 fixed weight (c)

ในงานวิจัยนี้ใช้การเรียนรู้เชิงลึกเพื่อตรวจสอบการตกผลึกของน้ำตาล ข้อมูลนี้รวบรวมจากเหตุการณ์ที่บ๊วยปี 2562/2563 ของโรงงานน้ำตาลในประเทศไทย แบบจำลองนี้มาจากข้อมูลภาพของการตกผลึกของน้ำตาลจากกระบวนการผลิตจริง มีการใช้ความเชี่ยวชาญของช่างจำลองประสบการณ์ 5 คนเพื่อตรวจสอบความถูกต้องของชุดข้อมูล ที่ผ่านการวิเคราะห์คุณสมบัติเฉพาะ เวกเตอร์ในกระบวนการเรียนรู้การถ่ายโอนโมเดลที่มีความแม่นยำและแม่นยำสูงสุดคือ VGG16 กำหนด 5 ชั้นสุดท้ายโดยใช้น้ำหนักคงที่ ความแม่นยำในการทำนายของทั้งสี่คลาสอยู่ในช่วงระหว่าง 80% ถึง 94% โมเดลนี้สามารถใช้เพื่อการควบคุมกระบวนการอัตโนมัติต่อไป (9)

บรรณานุกรม

1. A. Bruno PC, J.Newman, E. Snell and D. So. Classification of crystallization outcomes using deep convolutional neural networks. PLoS ONE 13(6). June 2018.
2. Pieter H. Principles of Sugar Technology. Bibliography of Crystallized Elsevier.234-57.
3. al. CSe. Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015:1-9.
4. Aydin FEaG. Data classification with deep learning using Tensorflow. Int Conf on Computer Science and Engineering. 2017:755-8.
5. Andrew SKZ. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv. 2014;1409.1556.
6. B.G. Mertzios and P. Liatsis. Techniques for classifying sugar crystallization images based on spectral analysis and the use of neural networks. Elsevier. Nov 1996:641-5.
7. H. Kaiming ZX, R. Shaoqing and J. Sun. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2016:770-8.
8. Jenkins G. Introduction to Cane Sugar Technology. Science Direct. 1966:267-375.
9. L. Suarez PG, S.F. de Azevedo. Computationally efficient process control with neural networkbased predictive models. Int Joint Conf on Neural Networks. 2009:2990-7.
10. S. Dhruva RR, K. Matlockl, S. Ghosh and R. Pal. Dimensionality Reduction based Transfer Learning applied to Pharmacogenomics Database. Int Conf of the IEEE Engineering in Medicine and Biology Society. 2018:1246-9.
11. Shukla SYaS. Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. IEEE Int Conf on Advanced Computing. 2016:78-83.
12. Iteca A. New Process Control During The Crystal Growth: Real Time Image Analysis. Sugar Asia. Jan 2020.

13. J. Zhang YM, J. Wu and S. Yu. Monitoring sugar crystallization with deep neural networks,. Food Engineering. Feb 2020;280,109965:267-375.





จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ประวัติผู้เขียน

| | |
|-------------------|---|
| ชื่อ-สกุล | suriya chayatummagoon |
| วัน เดือน ปี เกิด | 01 March 1987 |
| สถานที่เกิด | Bangkok |
| วุฒิการศึกษา | Chemical Engineer - KMITL. |
| ที่อยู่ปัจจุบัน | 14 Naradhiwat Rajanagarindra Rd, Thung Wat Don, Sathon, Bangkok 10120 |
| ผลงานตีพิมพ์ | Suriya Chayatummagoon, Prabhas Chongstitvatana, “Image classification of sugar crystal with Deep learning”, 2021 - 13th International Conference on Knowledge and Smart Technology (KST), Jan 2021 |
| รางวัลที่ได้รับ | Smart energy hackathon south east asia 2nd |