

## การจำแนกประเภทมวลหินของหน้าอุโมงค์ด้วยระบบประสาทเทียมแบบคอนโวลูชัน

### Rock mass classification prediction on Tunnel face mapping using CNN

สาริต ไกรนรา<sup>1</sup> สิตานัน เสียรักษ์ดิพัฒน์<sup>2</sup> และ รองศาสตราจารย์ ดร. จีรวีตร บุญญะฐิติ<sup>3</sup>

<sup>1,2,3</sup> ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย จ.กรุงเทพฯ

#### บทคัดย่อ

โดยทั่วไปแล้วในการก่อสร้างอุโมงค์ เราจำเป็นต้องตรวจสอบความแข็งแรงของหินตลอดระยะทาง ซึ่งในการตรวจสอบความแข็งแรงนั้น จำเป็นต้องให้ผู้เชี่ยวชาญมาตรวจสอบและการประมวลผลหิน (RMR) ดังนั้นเพื่อให้มีความแม่นยำ และสะดวกมากขึ้น จึงได้นำระบบประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network) มาใช้ปัญหาการจำแนกประเภทมวลหินของหน้าอุโมงค์ โดยการนำข้อมูลที่ได้รับมาแบ่งข้อมูลออกเป็น 2 ส่วน คือ ข้อมูลเพื่อสร้างโมเดล และชุดการตรวจสอบโมเดล โดยได้รับชุดภาพหินหน้าอุโมงค์ จากอุโมงค์ส่งน้ำช่วงแม่แตง-แม่จัด จังหวัดเชียงใหม่ จำนวน 1165 ชุด จากกรมชลประทาน มาใช้ในการสร้างโมเดลที่แตกต่างกัน ซึ่งได้มีการปรับแต่งข้อมูลที่ใช้เทรน และการปรับแต่งโมเดลให้สอดคล้องกับปัญหาของเรา โดยมี pretrained-model ที่ใช้ใน งานวิจัยครั้งนี้ คือ VGG, Resnet และ EfficientNet ทั้งนี้ในการเทรนโมเดลจะเทรนเฉพาะส่วนท้ายของโมเดลหรือที่เรียกว่า การทำ Fine Tuning และการทำ K-Fold cross validation เพื่อหาโมเดลที่เหมาะสมที่สุดในการใช้งาน

คำสำคัญ: ระบบเครือข่ายประสาทเทียมแบบคอนโวลูชัน/ วิธีการโพนินิ่ง/ การจำแนกภาพ/ การเรียนรู้ของเครื่อง

#### Abstract

Normally, constructing a tunnel has to rate rock mass (RMR), a vital score for reinforcement in a tunnel. However, the process must have specialists/geologists rate rock mass with accuracy. Thus, to reduce the time to discover the tunnel and to improve the rating system, we use CNNs (Convolution Neural Network) for Rock classification by using a raw photo of a tunnel face. We

receive the rock tunnel faces from Mae-Daeng Mae-Ngad water tunnel in Chiang-Mai province, then the dataset is divided into two groups which are training datasets and testing datasets. We are using pre-trained model including VGG, Resnet and EfficientNet to train on training dataset before evaluating model performance on testing dataset. The model's weights in last block of convolutional layers and classifier layers will be unfreeze during the training process, which is called Fine-tuning process. We also use K-Fold cross-validation for model selection so that we obtain the best model for the rock classification problem. Finally, we find the most appropriate model which will be discussed on this article.

Keywords: Convolution Neural Network/ Fine Tuning/ Image Classification/ Machine Learning

#### 1. บทนำ

ในการก่อสร้างอุโมงค์โดยทั่วไป จำเป็นต้องมีการใช้ระบบจำแนกมวลหิน เพื่อใช้สำหรับการเสริมความแข็งแรงให้กับอุโมงค์ โดยใช้เกณฑ์วิธีวิทยาที่มีความชำนาญและเชี่ยวชาญ จำแนกมวลหินตามระบบ RMR (Bieniawski, 1989) ในบทความนี้จะได้นำการเรียนรู้ของเครื่อง (Machine Learning) ซึ่งเป็นที่นิยมอย่างมากสำหรับการนำมาใช้แยกประเภทภาพ (Boonkao & Chartlatanagulchai, 2020) และ (Chayatummagoon & Chongstitvatana, 2021) มาใช้สำหรับการจำแนกมวลหิน โดยใช้ภาพแทนที่ใช้ในกรณีวิธีวิทยา ซึ่งจะสามารถลดเวลาในการประเมินหน้ามวลหินในอุโมงค์ ลดจำนวนนักธรณีวิทยาที่ใช้ในการทำงาน และลดปัญหาที่เกิดจากคะแนนที่มีความคลาดเคลื่อนจากตัวผู้ประเมิน และอคติจากนักธรณีวิทยา

ในงานวิจัยนี้จะใช้ภาพหินหน้าอุโมงค์ จากอุโมงค์ส่งน้ำช่วงแม่แตง-แม่จัด จังหวัดเชียงใหม่จากกรมชลประทาน ซึ่งได้รับชุดภาพหินหน้าอุโมงค์ทั้งหมด 1165 รูปเพื่อใช้ในการสร้างโมเดลที่สามารถจำแนกมวลหินได้ โดยเมื่อเปรียบเทียบกับงานวิจัยอื่นพบว่าจำนวนภาพทั้งหมดมีค่อนข้างน้อย

เมื่อเทียบเคียงกับงานวิจัยที่ใกล้เคียงกับ Chen (2021) และ Kim (2019) จากคำแนะนำของ Raschka (2019) พบว่าในกรณีที่มีข้อมูลจำนวนน้อย ควรใช้วิธีการในเลือกและประเมินประสิทธิภาพของโมเดลแบบ K-Fold Cross Validation with independent test set (Repeated) ซึ่งในงานวิจัยครั้งนี้เลือกใช้ 5-Fold Cross Validation โดยการทำซ้ำทั้งหมด 5 รอบ

โดยวิธีที่ใช้ในการดำเนินการทำงานวิจัยในครั้งนี้จะใช้ Pre-trained model ซึ่งเป็นโมเดลอื่นที่ใช้ข้อมูลอื่นในการเทรนก่อนแล้ว เพื่อเพิ่มความเร็วในการเรียนรู้ และช่วยเพิ่มประสิทธิภาพของโมเดลให้มีความแม่นยำมากขึ้น โดยเลือกใช้ VGG (Simonyan & Zisserman, 2014), Resnet (He et al., 2015) และ EfficientNet (Tan & Le, 2019) โดยปรับเปลี่ยนโมเดลเฉพาะส่วนทาง (Classifier Layers) ให้มีจำนวนเท่ากับชนิดของภาพที่นำมา รวมถึงจะใช้วิธีการเทรนโมเดลแบบ Feature Extraction และ Fine-Tuning ตามคำแนะนำของ Chollet (2018) ซึ่งเป็นวิธีการในเทรนโมเดลสำหรับกรณีที่มีจำนวนข้อมูลน้อย โดย Weights ในเฉพาะส่วนสุดท้ายของ Convolutional Layers และ Classifier Layers จะถูก Unfreeze ซึ่งตามคำแนะนำของ Chollet (2018) และ Howard & Ruder (2018) พบว่าจะช่วยลดเวลาในการเทรนโมเดล เนื่องจากจำนวน Weights ที่ใช้ในการเทรนมีจำนวนน้อยลง รวมถึงใช้เพิ่ม Accuracy ของโมเดลให้มีความสูงขึ้นอีกด้วย

จากที่กล่าวไป โมเดลที่ถูกเรียนรู้บน Training Dataset และถูกนำมาวิเคราะห์ทางสถิติ และประเมินประสิทธิภาพของโมเดล โดยมีทั้งหมด 3 ปัจจัยคือความแม่นยำ (Accuracy) ขนาดของโมเดล (Model Size) และเวลาที่ใช้ในการจำแนกมวลหิน (Inference Time) ก่อนที่จะทำการเลือกโมเดลที่เหมาะสมเพื่อมาหาความแม่นยำสุดท้ายบน Testing Dataset ซึ่งจะรายงานผลในรูปแบบของ Confidence Interval ของ Accuracy ที่ 95% รวมถึงค่าอื่นๆที่ใช้ในการวิจารณ์ผลซึ่งคือ Accuracy, Precision, Recall และ F1-Score รวมถึง Confusion Matrix ที่ จะบ่งบอกประสิทธิภาพของโมเดลในภาพมวลหินแต่ละชนิด ว่ามีความสามารถเท่าใดในการจำแนกมวลหิน

**ตารางที่ 1** พารามิเตอร์ต่างๆสำหรับการจำแนกมวลหิน (Rock Mass Rating)

RMR	Parameters
1	Uniaxial Compressive strength (UCS)
2	Rock Quality Designation (RQD)
3	Spacing of discontinuities
4	Condition of discontinuities
5	Groundwater conditions
6	Orientation of discontinuities

## 2. การเตรียมข้อมูล

จากการได้รับข้อมูลตัวอย่างของภาพหน้าตัดหน้าตัดอุโมงค์น้ำ ได้นำไป upload ขึ้น Google drive เป็นจำนวน 1165 รูป และสร้าง spread sheets เพื่อเก็บข้อมูล rating ของแต่ละ station

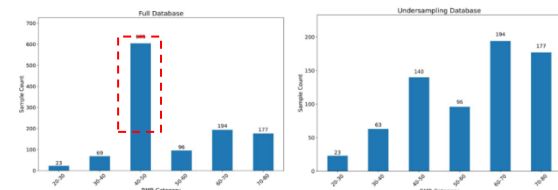


**รูปที่ 1** ตัวอย่างภาพหน้าตัดอุโมงค์ (Tunnel face)

### 2.1 Undersampling

โดยก่อนที่จะไปเทรนโมเดล ข้อมูลต้องได้รับการแยกข้อมูลออกเป็น 6 ชุดข้อมูล ตามคะแนน RMR ที่ได้จากการเก็บข้อมูลที่ได้รับ คือ ชุดข้อมูลที่ 1 = RMR 20-30, ชุดข้อมูลที่ 2 = 30-40, ชุดข้อมูลที่ 3 = 40-50, ชุดข้อมูลที่ 4 = 50-60, ชุดข้อมูลที่ 5 = 60-70 และ ชุดข้อมูลที่ 6 = 70-80 เนื่องจากคะแนน RMR นั้นเป็นคะแนนที่ขึ้นกับปัจเจกบุคคล ทำให้การทำนายในลักษณะของช่วง (Interval) นั้นมีความเหมาะสมมากกว่า

ซึ่งมีจำนวนของภาพที่ใช้ทั้งหมด (dataset) 1165 ภาพ ดังรูปที่ 2 จะสังเกตได้ว่าในช่วงคะแนนระหว่าง 40-50 (ชุดข้อมูลที่ 3) หากเรานำข้อมูลที่ไปเทรนทั้งหมด โมเดลที่ได้ไม่มีประสิทธิภาพ เนื่องจากมีข้อมูลใน class ที่ 3 ใช้เทรนมากเกินไป จะทำให้โมเดลทำนายได้เพียงค่า RMR ที่อยู่ในช่วง 40-50 ดังนั้นจะเหลือรูปภาพที่ใช้ในการดำเนินการ จำนวน 693 ภาพ จากผลการลดรูป/ข้อมูลใน class ที่ 3 เรียกว่า Undersampling Database (Chollet, 2018) เพื่อใช้ในการเตรียมข้อมูลในขั้นตอนถัดไป



**รูปที่ 2** จำนวนข้อมูลในแต่ละชุดข้อมูล (ก่อนและหลังทำ Undersampling ตามลำดับ)

### 2.2 การแบ่งข้อมูล Train Dataset และ Test Dataset

จากภาพทั้งหมด 693 ภาพ แบ่งเป็น Train Dataset และ Test Dataset ใช้สัดส่วน 80:20 คือ 487 ภาพ และ 122 ภาพ ตามลำดับ Train Dataset จะถูกใช้ในการเทรนกับ Test Dataset จะถูกโมเดลให้เรียนรู้เพื่อแยกแยะชนิด RMR ของหน้าตัดหิน และ ใช้สำหรับการประเมินผล (Evaluate) ของโมเดลในตอนสุดท้ายตามที่ Raschka (2018) ได้แนะนำไว้ เพื่อตรวจสอบประสิทธิภาพของโมเดลในการทำงานจริง หากเจอกับข้อมูลใหม่ที่แตกต่างจากที่เรียนรู้ไว้ (ใน Train Dataset) สามารถสรุปข้อมูลรูปทั้งหมดได้สำหรับการดำเนินวิจัยได้ทั้งหมดดังนี้ ตารางที่ 2

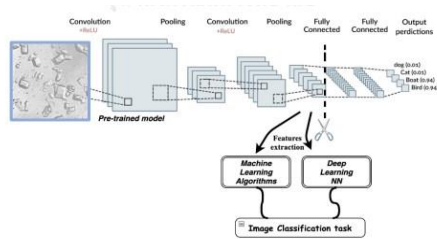
ตารางที่ 2 สรุปรายละเอียดข้อมูลที่ใช้ในการทำโมเดล

ช่วงคะแนน RMR	Train Datasets (ข้อมูล)	Test Datasets (ข้อมูล)	Total (ข้อมูล)
20 - 30	11	2	13
30 - 40	47	12	59
40 - 50	105	26	131
50 - 60	67	17	84
60 - 70	135	34	169
70 - 80	122	31	153
รวมทั้งหมด	487	122	609

### 3. ระเบียบการวิจัย

#### 3.1 ระบบประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network, CNN)

Chollet (2018) ได้เสนอให้ใช้ Convolutional Neural Network เพื่อหาลักษณะเฉพาะของรูปภาพดังรูปที่ 3 โดยใช้หลักการ Convolution เพื่อดึงหาลักษณะเฉพาะ (Feature) ของภาพ ซึ่งจะช่วยลดจำนวน weights บนเครือข่ายประสาทเทียมได้เมื่อเปรียบเทียบกับ Fully Connected Neural Network นอกจากนี้จำนวนมิติของข้อมูลจะถูกบีบอัดลง Pooling layers ซึ่งจะทำการคำนวณมีประสิทธิภาพมากขึ้น เนื่องจากข้อมูลมีขนาดที่ลดลง แต่ยังคงความสำคัญของข้อมูลได้อย่างครบถ้วน โดยมีรายละเอียดดังต่อไปนี้



รูปที่ 3 องค์ประกอบของระบบประสาทเทียมแบบคอนโวลูชัน (Boonkao & Chartlanagulchai, 2020)

##### 3.1.1 Convolution layers

เป็นการดึงส่วนสำคัญ (Features) บนภาพออกมาเช่นขอบของวัตถุ ดวงตา หรือสี เพื่อใช้ในการเทรนตัวโมเดลของแต่ละส่วนต่อไป เป็นการคำนวณระหว่าง kernels เป็น matrix มีขนาดโดยปกติเป็น 3 x 3, 5 x 5 หรือ 7 x 7 กับภาพ ซึ่งจะ Feature map ในเชิงคณิตศาสตร์เป็นการทำ Element wise product ระหว่าง kernels กับภาพบน Channels (Yamashita et al., 2018)

##### 3.1.2 Pooling layers

เป็นการดึงเฉพาะ Feature ที่สำคัญที่สุด เพื่อลดขนาดของภาพให้มีขนาดเล็กลง ส่งผลทำให้ลดทรัพยากรที่ใช้ในการคำนวณลง โดยทั่วไปจะมี

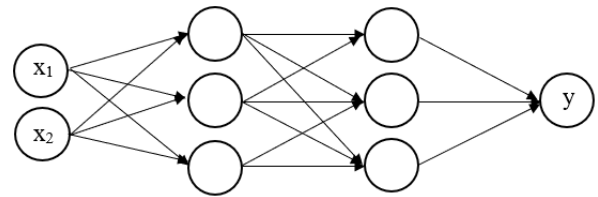
ขนาด kernel ที่ 2 x 2 ซึ่งจะส่งผลให้ Feature ของภาพมีขนาดลดลง (Yamashita et al., 2018)

##### 3.1.3 Activation layers

การทำงานของ Activation layers มีไว้เพื่อเพิ่มความเป็น nonlinearity ของโมเดล ซึ่งจะช่วยให้การคำนวณ และสมการ มีความซับซ้อนมากขึ้น และสามารถหา weights ที่เหมาะสมกับปัญหาได้ โดย Goodfellow et al. (2017) และ Chollet (2018) แนะนำในปัญหาทั่วไป จะใช้ Rectified linear unit หรือ “ReLU”

##### 3.1.4 Classifier layer & Softmax 6 classes

ส่วนท้ายของโมเดลจะเป็นการนำ Node ของหลายๆเซลล์มารวมกัน เพื่อหา weights ที่เหมาะสม เรียกว่า Classifier layers และเรายังอาจเรียกได้อีกชื่อว่า “Fully Connected Neural Network” (FCN) หรือ “fully connected feed-forward network” (FFN) ได้เนื่องจากในทุก node ของเครือข่ายจะมี weights ที่เชื่อมต่อกันทุกตัว (Goodfellow et al., 2017) ซึ่งสามารถดูได้จากรูปที่ 4



รูปที่ 4 Fully Connected Neural Network

ในส่วน Softmax Functions (Goodfellow et al., 2017) ใช้เพื่อในการสร้าง probability distribution บน N Classes ของปัญหา ดังสมการที่ 1 เพื่อใช้ในการคำนวณหาค่า Loss Score เพื่อทำการปรับค่า weights โดยจะใช้หลักการ “Gradient Descent” (Larmarechal, 2012) ในขั้นตอน Backpropagation

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^N e^{z_j}} \quad (1)$$

โดยที่  $\sigma$  คือ Softmax function และ  $z_i$  คือข้อมูลในตำแหน่งที่ i สำหรับ array 1 มิติ

ทั้งนี้จะปรับค่า SoftMax เหลือ 6 classes เพื่อให้สอดคล้องกับปัญหาของเราที่มีทั้งหมด 6 ชุดข้อมูล นอกจากนี้ค่า Loss Score ที่เลือกใช้จะใช้ Categorical Cross Entropy Loss (Goodfellow et al., 2017) ดังสมการที่ 2

$$l_n = -\sum_{i=1}^N q_i \log p_i \quad (2)$$

โดยที่  $l_n$  คือค่า Categorical Cross Entropy Loss ระหว่าง Probability Distribution และ  $p_i$  และ  $q_i$  คือ Probability Distribution ของ P และ Q ใน Class ที่ i

### 3.2 Data augmentation

นอกจากนี้เพื่อเพิ่มประสิทธิภาพในการเทรนโมเดล เราสามารถเพิ่มจำนวนข้อมูลในการเทรนได้ ซึ่งเรียกวิธีนี้ว่า Data augmentation (Chollet, 2018) โดยการปรับลักษณะภาพใน ให้มีความหลากหลายมากขึ้น เช่นการกลับภาพจากซ้ายไปเป็นขวา การหมุนภาพ หรือการปรับความคมชัดของภาพ โดยการปรับลักษณะภาพในครั้งนี้ สำหรับชุดข้อมูลฝึก (Training set) ซึ่งกำหนดค่าการดังนี้

ตารางที่ 3 ขั้นตอนการทำ Data Augmentation สำหรับ Train Dataset

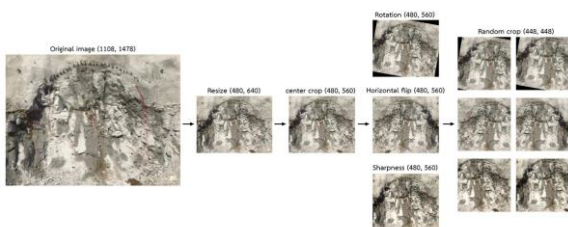
ขั้นตอน	ขนาดของภาพ
Reshape ภาพจากขนาดใดๆ (Aspect Ratio 4:3)	480 x 640 Pixel
Center crop กลางภาพ	480 x 560 Pixel
ทำการหมุนภาพ (Rotation)	-10° ถึง 10°
ทำการกลับรูปภาพ (Horizontal Flip)	โอกาส 50%
ทำการปรับความคมชัดของภาพ (Sharpness)	2
ทำการสุ่มตัดภาพ (Random Crop)	448 x 448 Pixel
ทำการ Normalize ภาพตาม Imagenet Dataset	-

นอกจากนี้สำหรับ ชุดข้อมูลทดสอบ (Test Dataset) จะทำเพียงแค่ลดขนาดภาพ และใช้ Center crop เพื่อจับรายละเอียดภาพตรงกลางแทนโดยไม่มี การ หมุนภาพ กลับภาพ และปรับความละเอียด เนื่องจากต้องการให้ตรงกับภาพรายละเอียดจริงที่นำมาใช้งานมากที่สุด โดยรายละเอียดมีดังนี้

ตารางที่ 4 ขั้นตอนการทำ Data Augmentation สำหรับ Test Dataset

ขั้นตอน	ขนาดของภาพ
Reshape ภาพจากขนาดใดๆ (Aspect Ratio 4:3)	480 x 640 Pixel
Center crop กลางภาพ	448 x 448 Pixel
ทำการ Normalize ภาพตาม Imagenet Dataset	-

สำหรับ EfficientNet ที่เป็นสถาปัตยกรรมรูปแบบใหม่ที่จะกล่าวถัดไปในหัวข้อ 3.4.3 จะมีการ Reshape ภาพสุดท้ายจากการทำ Center crop ภาพใน 448 x 448 Pixel ให้มีขนาดตามที่โมเดลต้องการ เนื่องจาก EfficientNet นั้นจะมีการแนะนำ Resolution ของภาพที่ใช้ให้สอดคล้องกับขนาดของโมเดล



รูปที่ 5 การทำ Data augmentation สำหรับ Training Dataset

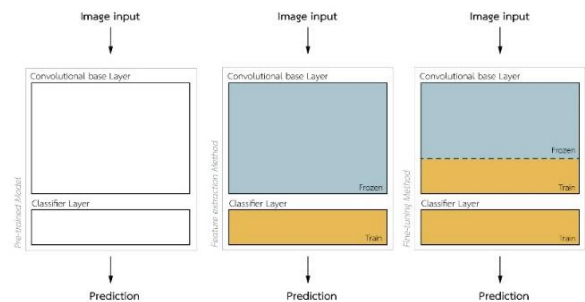


รูปที่ 6 การทำ Data augmentation สำหรับ Test Dataset

### 3.3 Transfer learning

เพื่อประหยัดเวลาในการเทรน และเพิ่มประสิทธิภาพของโมเดล โดยเฉพาะกับปัญหาที่มี dataset จำนวนน้อย เราจึงนำ Pre-trained models เช่น VGG, ResNet, และ EfficientNet เป็นต้น มาปรับแต่งเลเยอร์ส่วนท้ายของโมเดล เพื่อเทรนต่อกับ Dataset ขนาดเล็กในงานเฉพาะทาง หรือ ใช้สกัด Feature สำหรับงานที่ต้องการออกมา โดยมีงานวิจัย Yosinski et Al (2014) ยืนยันว่าการทำ transfer learning ช่วยเพิ่มประสิทธิภาพการทำงานของ deep neural network และยังเข้าไปช่วยในการปรับปรุง generalization performance

ในการเทรน Pre-trained models ให้เข้ากับปัญหาเรา Chollet (2018) ได้อธิบายขั้นตอน ดังนี้ เราจะเริ่มต้นจาก Feature extraction แล้วต่อกับ วิธี Fine-tuning method ซึ่งวิธี Feature extraction เป็นการนำ Convolution Neural Network มาลบ Dense Layer สุดท้ายออกไป แล้วสร้าง New classifier แล้วเทรนแค่ส่วนท้าย (Unfreeze แค่ ส่วนท้ายของโมเดล) แล้วต่อกับ วิธี Fine-tuning method เป็นการเพิ่ม บางส่วนของ Convolutional Based Layer เข้ามาเทรนเพิ่มจากวิธี Feature extraction (Unfreeze ส่วนท้ายของ Convolutional Layer ใน โมเดล) ดังรูปที่ 7



รูปที่ 7 Feature extraction และ Fine-tuning

### 3.4 Models

#### 3.4.1 VGG

เป็นสถาปัตยกรรม Convolutional Neural Network (CNN) ที่เรียบง่ายและใช้กันอย่างแพร่หลาย โดยเริ่มจาก Input รูปภาพ RGB ขนาด 224 x 224 ผ่านไปยัง Convolutional Layers ซึ่งมีฟิลเตอร์ที่มีช่องรับแสงขนาด 3x3 pixel, 1 stride และใช้ same padding และ max-pooling 2x2 pixels, 2 stride แบบเดียวกันตลอดทั้งโครงสร้าง (Simonyan et al., 2014) แต่เพื่อให้สอดคล้องกับปัญหาเราจะให้โมเดลมีการปรับแต่งในส่วนท้ายจะให้ เป็น Adaptive Average pooling 2D ซึ่งจะมีขนาด Output เป็นภาพ 7 x 7 Pixels 512 Channels จากนั้น Output จะถูกแปลงเป็น Fully connected layer 25088 nodes กับ 2 layers 4096 nodes และสุดท้ายปรับ softmax ให้เป็น 6 classes โดยเราจะใช้ทั้ง VGG16 และ VGG19 สำหรับงานวิจัยครั้งนี้

#### 3.4.2 ResNet

ResNet เป็นโมเดลสำหรับการแก้ปัญหาปัญหาการคัดแยกประเภทจากรูปภาพ (Image classification) เช่นเดียวกับ VGG โดยแบบจำลอง ซึ่ง

สถาปัตยกรรมประกอบด้วย 4 Block ในแต่ละ Block จะมี Convolutional Layer, 2 stride กับ Max Pool 3x3 จำนวนแตกต่างกัน

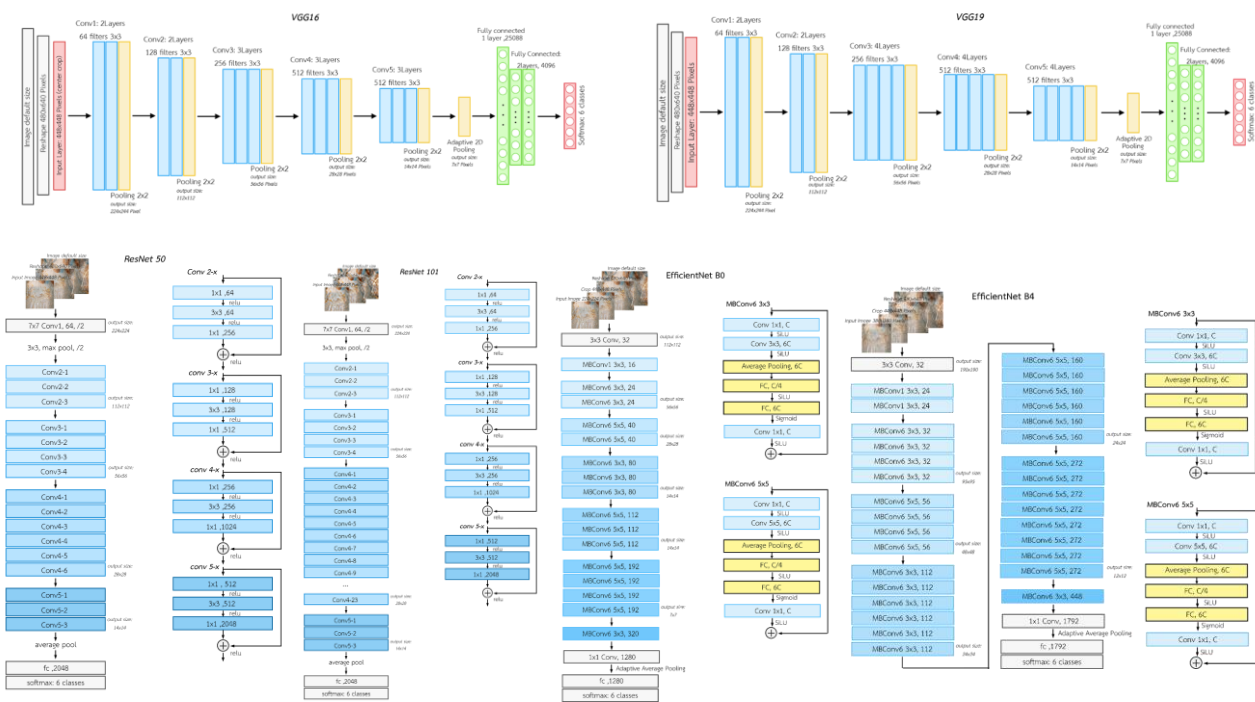
โดย ResNet จะมี Residual learning เป็นการ skip connection (Residual Forward/Backward passing) ไม่เหมือนกับ VGG ที่ไม่มีการ skip layer (Simple Forward/ Backward passing) รวมถึงมี parameter น้อยกว่า VGG แต่ Feature map ของ Resnet มีคุณภาพน้อยกว่า VGG ซึ่งได้ปรับแต่ง layers ในส่วนท้าย เป็น Fully connected layer 2048 nodes และสุดท้ายผ่าน Softmax Layer 6 classes ซึ่งเราจะใช้โมเดล ResNet50 และ ResNet101 ในการศึกษาในครั้งนี้ (He et al., 2016)

### 3.4.3 EfficientNet

เป็น Convolutional neural networks โดยใช้หลักการ Compound Scaling (Tan et al., 2019) โดยเสนอวิธีการขยายขนาดโมเดลโดยใช้ 3 ปัจจัยหลัก คือ ความกว้างของโมเดล (width) หรือจำนวน Channel ที่ใช้ในการทำ Convolution ใน Convolutional Layers, ความลึกของโมเดล (Depth) หรือจำนวนชั้น Layers และความละเอียดของภาพขาเข้าที่ใช้

โดย Model ที่เป็น Baseline ในที่ใช้ในการทำ Compound Scaling จะใช้ EfficientNet-B0 Baseline network ที่ในแต่ละ block ของ Layers จะใช้ Inverted bottleneck MBConv (Sandler et. al, 2018) โดยจะมี โครงสร้างของ Convolutional Layer Block ซึ่งจะช่วยลดขนาดการคำนวณของโมเดล โดย Convolutional Layers จะใช้ Filter ขนาดใหญ่ และมีจำนวน Channel ที่เยอะทำการดึง Features ที่สำคัญของภาพออกมา ก่อนที่จะใช้ Filter ขนาด 1x1 เพื่อลดจำนวน Channel นอกจากนี้มีการใช้ MBConv และ Squeeze-and-Excitation Optimization (Hu et al., 2018) ทำให้เพิ่ม Accuracy ของโมเดลให้มีประสิทธิภาพสูงขึ้น

เมื่อเปรียบเทียบกับ CNN ในแบบเก่าที่จำนวน Channel จะยิ่งเพิ่มขึ้นในชั้น Layers ที่มากขึ้น นอกจากนี้ยังลดขนาดของโมเดล เนื่องจากจำนวน weights ที่ใช้ยังลดลง ส่งผลให้มีจำนวน parameters และชั้น Layer ที่น้อยกว่า VGG และ Resnet และในส่วนของ FCN จะถูกลดจาก 1000 Nodes เหลือเพียง 6 Nodes ตามจำนวน Class เท่านั้น ซึ่งโมเดลที่เลือกใช้ในการศึกษาครั้งนี้คือ EfficientNet-B0 และ EfficientNet-B4



รูปที่ 8 องค์ประกอบของสถาปัตยกรรมทั้งหมด

## 4. ผลการดำเนินงานวิจัย

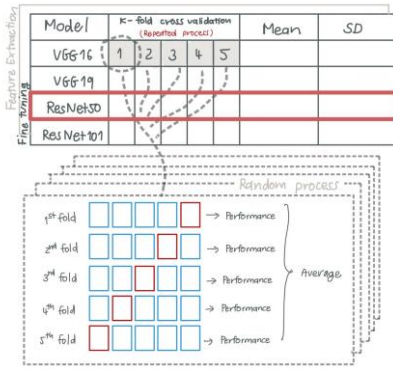
จากข้อมูลที่ได้แบ่งออกเป็น 2 ส่วนคือ Training Dataset และ Testing Dataset โดยอัตราส่วน 80% และ 20% โมเดลทั้งหมดถูกเทรนบน Pytorch Frame-work โดยใช้เครื่องจาก Google Colab โดยใช้ GPU เพื่อให้เวลาในการเทรนโมเดลนั้นมีความรวดเร็วมากขึ้น

### 4.1 การเทรนโมเดล

ในการเทรนจะข้อมูลจาก Training Dataset ใช้ K-Fold Cross Validation with independent test set (Repeated) ตามคำแนะนำ

ของ Raschka (2019) โดยเลือกใช้ 5-Fold Cross Validation โดยเป็นไปดังรูปที่ 9 ซึ่งเหมาะกับ Dataset ที่มีข้อมูลที่น้อย เมื่อเปรียบเทียบกับงานวิจัยอื่นๆ (Chen, 2021) โดยข้อมูลจะแบ่งออกเป็น Training Dataset และ Validation Dataset ซึ่งในแต่ละ Fold โมเดลจะถูกเทรนบนข้อมูล Training Dataset และวัดค่าความแม่นยำ (Accuracy) บน Validation Dataset ซึ่งสุดท้ายเมื่อครบทั้ง 5 Fold แล้วจะมาหาค่าเฉลี่ยเพื่อให้ความแม่นยำจากการทำ 5-Fold Cross Validation นอกจากนี้จะทำการทำซ้ำทั้งหมด 5 รอบเพื่อหาค่าเฉลี่ย (Mean) และค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) เพื่อทำการวิเคราะห์ทางสถิติในขั้นตอนถัดไป





รูปที่ 9 การทำ 5-Fold Cross Validation (Repeated 5 Times)

โดยการเทรนโมเดลจะเทรนทั้งหมด 50 Epoch ในขั้นตอน Feature Extraction และ 10 Epoch ในขั้นตอน Fine Tuning ซึ่งใช้ Categorical Cross Entropy Loss (Good Fellow et al., 2017) ที่ใช้ในการปรับค่า Weights ในขั้นตอน Back propagation นอกจากนี้ยังใช้ Adam optimizer สำหรับการปรับค่า weights ในแต่ละรอบโดยรายละเอียด parameters ต่างๆที่ใช้ในการเทรนสามารถดูได้ดังตารางที่ 5

ตารางที่ 5 ค่า parameters ต่างๆที่ใช้ในการเทรนโมเดล

Parameters	Value
Loss	Categorical Cross Entropy Loss
Optimizer	Adam Optimizer
Batch size	32
Epochs	50 / 10
Learning rate	$7 \times 10^{-4} / 7 \times 10^{-5}$
Weight Decay	$1 \times 10^{-5}$

#### 4.2 ผลการดำเนินงานวิจัย

จากการประเมินประสิทธิภาพของโมเดลพบว่าเราจะใช้ปัจจัยในการเลือกโมเดลที่เหมาะสมที่สุดนั้น เราจะพิจารณาปัจจัย 3 อย่างดังนี้

##### 4.2.1 ความแม่นยำ (Accuracy)

ความแม่นยำ (Accuracy) โดยจะใช้การวิเคราะห์ทางสถิติ One-way ANOVA เพื่อหาความแตกต่างอย่างมีนัยสำคัญ โดยความแม่นยำจะหาจากค่าเฉลี่ยของการทำ 5-Fold Cross Validation ซึ่งจะซ้ำทั้งหมด 5 รอบ โดยค่า Accuracy จะบ่งบอกประสิทธิภาพของโมเดลโดยตรง

##### 4.2.2 ขนาดของโมเดล (Model Size)

เป็นขนาดของ Weights ที่อยู่บนความจุบนเครื่องคอมพิวเตอร์ โดยหากโมเดลมีขนาดที่ซับซ้อน ก็จะมีขนาดใหญ่ นอกจากนี้ยังส่งผลต่อความจุบน RAM ของคอมพิวเตอร์ หากโมเดลมีขนาดใหญ่เกิน อาจทำให้ไม่สามารถโหลดขึ้นไปบน RAM ได้พอ ทำให้จำเป็นต้องใช้ RAM มีความจุสูงขึ้น

##### 4.2.3 เวลาที่ใช้ในการจำแนกผลของโมเดล (Inference Time)

เป็นเวลาของโมเดลที่ใช้ในการจำแนกผลโดย จะนับเพียงเวลาที่รูปภาพเข้า (Input Image) จนถึงให้ผลลัพธ์ออกมา โดยไม่นับเวลาที่ใช้ใน

การแปลงรูป (Image Transformation) ซึ่งจะทำงานบน CPU เท่านั้น โดยเครื่องคอมพิวเตอร์ที่ใช้มี CPU รุ่น Intel i7-8700

ตารางที่ 6 ตารางสรุปผลการเทรนโมเดล

Model	Accuracy	Model Size [MB]	Inference Time [ms]
VGG16	$77.63 \pm 0.56^a$	512	494.18
VGG19	$76.84 \pm 0.59^b$	532	583.45
Resnet50	$76.06 \pm 0.32^c$	90	302.66
Resnet101	$75.00 \pm 0.42^d$	162	480.84
EfficientNet-B0	$73.59 \pm 0.41^e$	16	162.71
EfficientNet-B4	$74.41 \pm 0.39^d$	68	465.96

#### 4.3 การวิจารณ์ผลดำเนินงานวิจัยและการเลือกโมเดลที่เหมาะสมที่สุด

ในปัจจุบัน 3 ปัจจัยที่ใช้ในการเลือกโมเดลที่เหมาะสมที่สุดที่กล่าวไปใน จะไม่นำ Inference Time มาใช้ในการพิจารณาเนื่องจากเวลาที่ใช้นบน CPU นั้นต่ำกว่า 1 วินาที ซึ่งเป็นเวลาที่น้อยมากสำหรับปัญหาประเภท Image Classification

ส่วนในแง่ของความแม่นยำพบว่า VGG มีความแม่นยำสูงที่สุด (VGG16 ที่ 77.63%) เนื่องจากจำนวน parameters ใน Classifier Layers มีเยอะกว่าสถาปัตยกรรมอื่นๆ รวมถึงใน EfficientNet มีการใช้ Input Resolution ที่น้อยกว่าสถาปัตยกรรมอื่นตามคำแนะนำของ Tan & Le (2019) (224 Pixels สำหรับ B0 และ 380 Pixels สำหรับ B4) ทำให้มีประสิทธิภาพต่ำกว่า VGG และ Resnet

ในแง่ของขนาดของโมเดล พบว่า EfficientNet มีขนาดโมเดลน้อยที่สุด โดยเฉพาะ B0 ที่มีขนาดเล็กที่สุดอยู่ที่ 16 MB ในขณะที่ VGG มีขนาดอยู่ที่ 512 MB ซึ่งเป็นสาเหตุจาก EfficientNet ใช้การขยายโมเดลแบบ Compound Scaling (Tan & Le, 2019) ทำให้การจำนวนชั้น Layers นั้นมีการขยายอย่างมีประสิทธิภาพ แทนที่จะเพิ่มจำนวนชั้น Layers จำนวนมากใน VGG และ Resnet

ในส่วนของการจำนวน Layers ในแง่ของ Model Accuracy นั้นไม่สามารถคาดเดาได้ เนื่องจากจำนวน Layers ที่ใช้นั้น หากใช้ Layers ที่มากเกินไป บางครั้งอาจทำให้เกิด Overfitting (Chollet, 2018) กล่าวคือโมเดลนั้นมีความซับซ้อนมากกว่า ความยากของปัญหาที่นำมา ส่งผลให้โมเดลนั้นใช้วิธีการจำมากกว่าการเรียนรู้ (Generalization) รวมถึงอาจทำให้ Accuracy ต่ำลงด้วยในบางครั้ง ส่วนในแง่ของ Model Size และ Inference Time นั้น ยิ่งจำนวน Layers มีจำนวนที่มาก มีจำนวน parameters ที่ใช้ในการเทรนมากขึ้นส่งผลให้มีขนาด และเวลาในการรันที่มากขึ้นด้วยเช่นกัน

อย่างไรก็ดีการเลือกโมเดลให้เหมาะสมกับงานนั้นขึ้นอยู่กับการใช้งาน และการนำโมเดลไปพัฒนาเป็นโปรแกรมในขั้นถัดไป หากต้องการโมเดลที่มีความแม่นยำสูงสุด โดยไม่คำนึงถึงขนาด และระยะเวลาในการรันนั้น ควรเลือกใช้ VGG16 เนื่องจากมี Accuracy ที่สูงที่สุด แม้ว่าจะมีขนาดของ Model ที่ใหญ่ก็ตาม (512 MB) ซึ่งหากต้องการพัฒนาให้ตัวโปรแกรมอยู่บน Cloud นั้นอาจเลือกใช้ VGG16 ที่มี Accuracy ที่ 77.63%

แต่หากต้องการโมเดลที่นำไปพัฒนาเป็นโปรแกรมที่ให้กับผู้ใช้โดยตรง (Application) การใช้โมเดลขนาดเล็กอาจมีความเหมาะสมมากกว่า เนื่องจากพื้นที่บรรจุข้อมูลของผู้ใช้งานมีจำนวนจำกัด และส่งผลต่อความจุที่มีอยู่บน RAM อีกด้วยหากโมเดลมีขนาดใหญ่เกินไป อาจทำให้จำเป็นต้องใช้อุปกรณ์ที่มีพลังในการคำนวณที่สูงกว่า มีความจุสูงกว่า ซึ่งในกรณีนี้ ควรเลือกใช้ EfficientNet-B0 ที่แม้มี Accuracy เพียง 73.59% แต่ขนาดของโมเดลมีขนาดเล็กกว่า VGG และ Resnet อย่างมีนัยยะสำคัญ (16 MB)

#### 4.4 การประเมินประสิทธิภาพของโมเดลสุดท้าย

สำหรับการเลือกโมเดลนั้น ขึ้นอยู่การเป้าหมายในการพัฒนาเป็นโปรแกรม โดยเลือกใช้โมเดล VGG16 และ EfficientNet-B0 ที่มีจุดประสงค์ที่ต่างกัน ทำการเทรนบนข้อมูล Training Dataset ทั้งหมด ก่อนที่จะทำการวัดประสิทธิภาพโดยการทำ Classification Report, การหา Confidence Intervals ของ Accuracy และ Confusion Matrix ของแต่ละโมเดล โดยสูตรหาค่าของ Confidence Intervals ของ Accuracy (Raschka, 2018) ระบุไว้สมการที่ 3

$$ACC_s = \frac{1}{n} \sum_{i=1}^n \delta(L(\hat{y}_i, y_i)) \quad (3)$$

โดยที่  $ACC_s$  คือ ค่า Accuracy จากการคำนวณค่า Performance  $n$  คือ จำนวนชุดข้อมูลทั้งหมดและ  $\delta(L(\hat{y}_i, y_i))$  คือ 0-1 Loss โดยมีค่าเท่ากับ 1 เมื่อโมเดลทายถูกและ 0 เมื่อทายผิด

นอกจากนี้ Classification Report และ Confusion Matrix สามารถอธิบายประสิทธิภาพของโมเดลในตัวชี้วัดอื่นๆ เช่น Precision และ Recall ดังสมการที่ 4 และ 5 ตามลำดับซึ่งอาจใช้งานสำหรับ Tuning Model งานอื่นๆถัดไปให้มีค่า False positive (Miss) และ False negative (False Alarm) ที่มีค่าน้อยลง

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

รูปที่ 10 Confusion Matrix สำหรับ 2 Class

$$Precision = \frac{TP}{TP+FN} \quad (4)$$

$$Recall = \frac{TP}{TP+FP} \quad (5)$$

โดยที่  $TP$  คือ True Positive,  $FP$  คือ False Positive,  $FN$  คือ False Negative และ  $TN$  คือ True Negative นอกจากนี้ยังมีค่า F1-Score ที่เป็น Harmonic mean ระหว่าง Precision และ Recall โดยค่าต่างๆเหล่านี้เป็นตัวชี้วัดอื่นในงานถัดไปได้ และเปรียบเทียบกับโมเดลอื่นๆหากใช้ตัวชี้วัดที่แตกต่างจากเดิม รวมถึงสามารถรู้ความสามารถของโมเดล

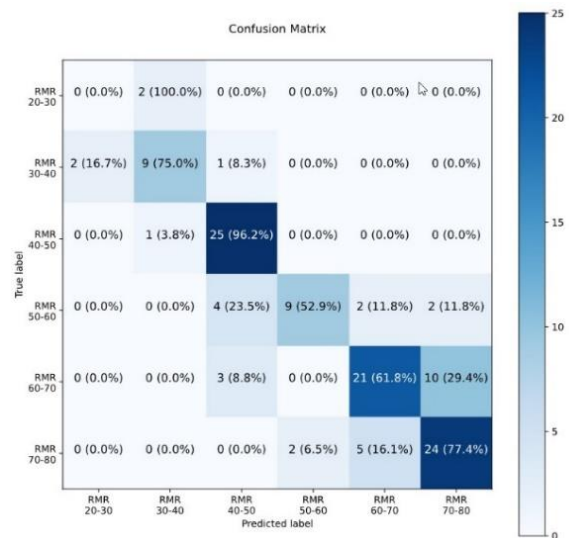
ในแต่ละ Class ของข้อมูลได้จาก Confusion Matrix ที่มีความละเอียดและแสดงถึงประสิทธิภาพที่แท้จริงของโมเดลได้มากกว่าใช้เพียง Accuracy

#### 4.4.1 VGG16

สำหรับ VGG16 เหมาะสำหรับจุดประสงค์ที่ต้องการ Accuracy ที่สูงที่สุดโดยไม่คำนึงถึงขนาดของโมเดลโดยมี Classification Report ตารางที่ 7 และ Confusion Matrix ดังรูปที่ 11

ตารางที่ 7 Classification Report ของ VGG16

Class	Accuracy	Precision	Recall	F1-Score	Data
RMR 20-30	0.0000	0.0000	0.0000	0.0000	2
RMR 30-40	0.7500	0.7500	0.7500	0.7500	12
RMR 40-50	0.9615	0.7576	0.9615	0.8475	26
RMR 50-60	0.5294	0.8182	0.5294	0.6429	17
RMR 60-70	0.6176	0.7500	0.6176	0.6774	34
RMR 70-80	0.7742	0.6667	0.7742	0.7164	31
Total	0.7213	0.7276	0.7213	0.7148	122



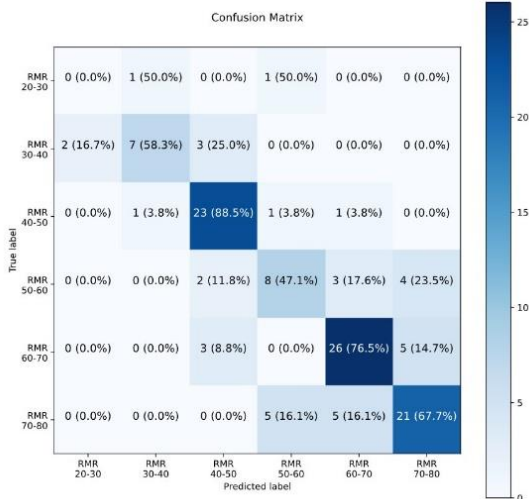
รูปที่ 11 Confusion matrix ของ VGG16

#### 4.4.2 EfficientNet-B0

สำหรับ EfficientNet-B0 เหมาะสำหรับจุดประสงค์ที่ต้องการโมเดลขนาดเล็กแต่แลกกับ Accuracy ที่ลดลงมาโดยมี Classification Report ตามตารางที่ 8 และ Confusion matrix ในแต่ละ Class ในรูปที่ 12

ตารางที่ 8 Classification Report ของ EfficientNet-B0

Class	Accuracy	Precision	Recall	F1-Score	Data
RMR 20-30	0.0000	0.0000	0.0000	0.0000	2
RMR 30-40	0.5833	0.7778	0.5833	0.6667	12
RMR 40-50	0.8846	0.7419	0.8846	0.8070	26
RMR 50-60	0.4706	0.5333	0.4706	0.5000	17
RMR 60-70	0.7647	0.7429	0.7647	0.7536	34
RMR 70-80	0.6774	0.7000	0.6774	0.6885	31
Total	0.6967	0.6938	0.6967	0.6922	122



รูปที่ 12 Confusion matrix ของ EfficientNet-B0

เมื่อทำการหาค่า Confidence Interval ของ Accuracy ดังสมการที่ 3 พบว่าได้ค่าตามตารางที่ 9 ดังนี้

ตารางที่ 9 Confidence Interval ของ Accuracy ในโมเดลต่างๆ

Accuracy	Guessing	Lower Bound	Mean	Upper Bound
VGG16	16.67	64.17	72.13	80.09
EfficientNet-B0	16.67	61.51	69.67	77.83

## 5. สรุปผลการวิจัย และข้อเสนอแนะ

### 5.1 สรุปผลการวิจัย

ในผลการดำเนินงานวิจัยนี้พบว่าเราใช้ Convolution Neural Network สำหรับการจำแนกประเภทของหินได้โดยใช้ข้อมูลชุดจุดเจาะอุโมงค์จากอุโมงค์ส่งน้ำช่วงแม่แตง-แม่จัด จังหวัดเชียงใหม่ โดยข้อมูลจะถูกแบ่งเป็น 2 ส่วนคือ Training Set และ Testing Set โดยใช้ Pretrained Model ซึ่งใช้ Fine-Tuning Method ในการเทรนโมเดลในส่วนของ Training Set ได้ความแม่นยำ (Accuracy) ระหว่าง 73 – 77% ในการทำ 5-Fold Cross Validation ทั้งหมด 5 ครั้ง ซึ่งในการพิจารณาเลือกโมเดลนั้นพบว่าหากต้องการโมเดลที่ให้ความแม่นยำสูงสุด เลือกใช้ VGG16 ที่มีความแม่นยำอยู่ที่  $72.13 \pm 7.96\%$  แต่จะมีขนาดของโมเดลอยู่ 512 MB แต่หากต้องการโมเดลที่มีขนาดเล็กกว่า แต่มีความแม่นยำที่น้อยกว่า เลือกใช้ EfficientNet-B0 ที่มีความแม่นยำอยู่ที่  $69.67 \pm 8.16\%$  และมีขนาดของโมเดลอยู่ 15.8 MB

### 5.2 ข้อเสนอแนะ

ในงานวิจัยครั้งนี้ถือเป็นการริเริ่มการนำปัญญาประดิษฐ์เข้ามาใช้งานในการจำแนกมวลหินแทนมนุษย์ซึ่งจำนวน และคุณภาพของข้อมูลนั้นมีความสำคัญอย่างมาก ถ้ามีการเก็บข้อมูลที่มีคุณภาพ กล่าวคือมี Resolution มุมองศาที่แน่นอน รวมถึงมีจำนวนที่มากขึ้นให้โมเดลมีการ

เรียนรู้ภาพที่หลากหลายมากขึ้น ก็จะทำให้โมเดลมีความสามารถมากขึ้น โดย อาจจะใช้โมเดลนี้กับโครงการชุดเจาะอุโมงค์อื่นได้ นอกจากนี้ในปัจจุบัน ยังมีวิธีการเก็บข้อมูลชุดเจาะอุโมงค์โดยใช้ Point Cloud ซึ่งเป็นการเก็บชุดหน้าเจาะอุโมงค์โดยเก็บลักษณะในข้อมูล 3 มิติทำให้ สามารถเก็บรอยแตกร้าวของหินที่มีมากกว่าข้อมูลในลักษณะของภาพโดยใช้ CNNs ซึ่งปัจจุบันสามารถทำได้โดยใช้กล้องจากโทรศัพท์มือถือ

ในแง่ของการพัฒนาโมเดล อาจเลือกใช้โมเดลที่มีการพัฒนามากกว่าปัจจุบัน หรือโมเดลอื่นๆเช่น MobileNetV2 และ Wide-Resnet ซึ่งอาจให้ความแม่นยำมากกว่าโมเดลที่นำมาพิจารณา อีกทั้งอาจนำโมเดลที่มีอยู่มาทำ Hyperparameter-Tuning ซึ่งคือการหาค่า Learning Rate ที่ดีที่สุด และค่า parameters อื่นๆที่ใช้ในการเทรนโมเดล จะส่งผลต่อความแม่นยำให้มีความมากขึ้น

## กิตติกรรมประกาศ

รายงานวิจัยฉบับนี้สำเร็จได้ด้วยดีด้วยความกรุณาจากรองศาสตราจารย์ ดร. จีรวัตร บุญญะฐิติ อาจารย์ที่ปรึกษาโครงการที่ได้ช่วยให้ความรู้ คำเสนอแนะ รวมทั้งข้อแก้ไขบกพร่องต่างๆมาโดยตลอด จนรายงานเล่มนี้เสร็จสมบูรณ์ จึงอยากกราบขอบพระคุณเป็นอย่างสูง รวมถึงพี่อ้น จากบริษัท Right Tunneling PLC. ที่ให้คำแนะนำรวมถึงข้อมูลชุดเจาะอุโมงค์จากอุโมงค์ส่งน้ำช่วงแม่แตง-แม่จัด สัญญาที่ 1 โครงการเพิ่มปริมาณน้ำในอ่างเก็บน้ำเขื่อนแม่กวงอุดมธารา จังหวัดเชียงใหม่ จากกรมชลประทาน ที่ใช้ในการดำเนินงานวิจัยครั้งนี้ และสุดท้ายนี้ขอขอบคุณเพื่อนๆ ที่คอยให้คำปรึกษาและให้กำลังใจตลอดการทำงานวิจัยครั้งนี้

## เอกสารอ้างอิง

- [1] Boonkao, S., & Chartlatanagulchai, W. (2020). Classification of Human Facial Expression using Artificial Intelligence Techniques (Doctoral dissertation, Kasetsart University).
- [2] Bieniawski, Z.T. (1989), Engineering Rock Mass Classification, John Wiley & Sons, New York.
- [3] Chayatummaoon, S., & Chongstitvatana, P. (2021, January). Image classification of sugar crystal with deep learning. In 2021 13th International Conference on Knowledge and Smart Technology (KST) (pp. 118-122). IEEE.
- [4] Chen, J., Yang, T., Zhang, D., Huang, H., & Tian, Y. (2021). Deep learning based classification of rock structure of tunnel face. Geoscience Frontiers, 12(1), 395-404.
- [5] Chollet Francois. (2018). Deep learning with python. Manning Publications Co.
- [6] Goodfellow, I., Bengio, Y., & Courville, A. (2017). Deep learning. The MIT Press.
- [7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE



- conference on computer vision and pattern recognition (pp. 770-778).
- [8] Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146.
- [9] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7132-7141).
- [10] Kim, H., Cho, L., & Kim, K. S. (2019). Rock classification prediction in tunnel excavation using CNN. *Journal of the Korean geotechnical society*, 35(9), 37-45.
- [11] Lemaréchal, C. (2012). "Cauchy and the Gradient Method" (PDF). *Doc Math Extra*: 251–254.
- [12] Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. arXiv preprint arXiv:1811.12808.
- [13] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).
- [14] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- [15] Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.
- [16] Yamashita, R., Nishio, M., Do, R. K., & Togashi, K. (2018). Convolutional Neural Networks: An overview and application in Radiology. *Insights into Imaging*, 9(4), 611–629. <https://doi.org/10.1007/s13244-018-0639-9>
- [17] 16. Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. *Advances in neural information processing systems*, 27.