

รหัสคำสั่งในส่วนการประมวลผลภาพ

ในส่วนนี้จะแสดงรหัสคำสั่งตัวอย่าง ที่ใช้ในการประมวลผลภาพของแอปพลิเคชัน Rice Disease Classification โดยแบ่งออกเป็น 4 ขั้นตอน ดังนี้

ขั้นตอนที่ 1 เป็นการนำภาพทดสอบที่ได้จากการส่องกล้องผ่านแอปพลิเคชันมาแปลงไฟล์ภาพให้อยู่ในรูปของไฟล์ bitmap ที่ข้อมูลตำแหน่งพิกเซลจัดเรียงกันอยู่เหมาะสมที่จะใช้ในการประมวลผล รหัสคำสั่ง เป็นดังนี้

```
public void onImage(CameraKitImage cameraKitImage) {  
    Bitmap bitmap = cameraKitImage.getBitmap();  
    bitmap = Bitmap.createScaledBitmap(bitmap, INPUT_SIZE, INPUT_SIZE, false);  
    imageViewResult.setImageBitmap(bitmap);  
    final List<Classifier.Recognition> results = classifier.recognizeImage(bitmap);  
    String display = results.toString();
```

ขั้นตอนที่ 2 หลังจากที่ได้ไฟล์ภาพประเภท bitmap ไฟล์ที่ได้ก็จะถูกนำไปแปลงตำแหน่งพิกเซล เพราะเดิมทีไฟล์ภาพจากที่ได้มาจะมีตำแหน่งพิกเซลเป็นตัวเลขตั้ง 0-255 โดยมีค่าเป็น int ที่ยังไม่สามารถนำไปใช้กับตัวโมเดลได้ จะต้องนำตำแหน่งพิกเซลของไฟล์ไปแปลงให้เป็น float ถึงจะทำให้ไฟล์ภาพที่ได้ไปกับโมเดลนั้นได้

```
Trace.beginSection("preprocessBitmap");  
// Preprocess the image data from 0-255 int to normalized float based  
// on the provided parameters.  
bitmap.getPixels(intValues, 0, bitmap.getWidth(), 0, 0, bitmap.getWidth(),  
bitmap.getHeight());  
for (int i = 0; i < intValues.length; ++i) {  
    final int val = intValues[i];  
    floatValues[i * 3 + 0] = (((val >> 16) & 0xFF) - imageMean) / imageStd;  
    floatValues[i * 3 + 1] = (((val >> 8) & 0xFF) - imageMean) / imageStd;  
    floatValues[i * 3 + 2] = ((val & 0xFF) - imageMean) / imageStd;
```

```
}
Trace.endSection();
```

ขั้นตอนที่ 3 ข้อมูลไฟล์ภาพที่ถูกแปลงจะถูกนำไปประมวลผลโดยทาง TensorFlowInferenceInterface ที่เป็นตัวไลบรารีที่ทำหน้าส่งข้อมูลไฟล์เทสกับโมเดล ซึ่งข้อมูลที่ถูกส่งจะเป็นข้อมูลพิกเซลของไฟล์ภาพที่ได้จากถ่ายภาพและนำมาแปลงตำแหน่งพิกเซลแล้ว

```
c.inferenceInterface = new TensorFlowInferenceInterface(assetManager,modelFilename);
```

ขั้นตอนที่ 4 เมื่อแอปพลิเคชันทำการประมวลผลภาพเสร็จก็จะส่งค่าออกมา คือ ชื่อของไฟล์ภาพและค่าเปอร์เซ็นต์ของไฟล์ภาพนั้นออกมา โดยจะทำการเอาค่าเปอร์เซ็นต์ของไฟล์ภาพที่ได้จากการทดสอบโมเดลที่มีค่าเปอร์เซ็นต์มากที่สุดออกมาแสดงผลลัพธ์

```
PriorityQueue<Recognition> pq =
    new PriorityQueue<Recognition>(
        3,
        new Comparator<Recognition>() {
            @Override
            public int compare(Recognition lhs, Recognition rhs) {
                // Intentionally reversed to put high confidence at the head of the queue.
                return Float.compare(rhs.getConfidence(), lhs.getConfidence());
            }
        });
```