



Report Lab Mini Project AI&ML

Cat Breed Classifier

จัดทำโดย

นาย ชารีฟ มะสาแม็ง

รหัสนักศึกษา 6510110115

Section 02

เสนอ

ดร. อนันท์ ชกสุริวงศ์

รายงานเล่มนี้เป็นส่วนหนึ่งของรายวิชา

240-318 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

ภาคเรียนที่ 1 ปีการศึกษา 2567

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

## ขั้นตอนการดำเนินงาน

### กำหนดโจทย์ปัญหา

ปัญหาคือต้องการแยกแยะสายพันธุ์แมวจากรูปภาพของแมวโดยใช้ AI ในการทำจำแนกสายพันธุ์อัตโนมัติจากภาพถ่ายที่ทำการอัปโหลด ซึ่งจะมีสายพันธุ์ต่างๆ ดังนี้ Persian , sphynx , birman , maine coon , russian blue , bengal , egyptian mau , siamese , abyssinian , bombay , british shorthair และ ragdoll ใช้สำหรับการคัดแยกสายพันธุ์ในระบบจัดเก็บข้อมูลหรือบริการที่เกี่ยวข้องกับสัตว์เลี้ยง เช่น โรงพยาบาลสัตว์หรือการซื้อขายแมวพันธุ์แท้

### แนวทางการแก้ปัญหา

ใช้เทคนิค Deep Learning โดยเฉพาะ Convolutional Neural Network (CNN) ซึ่งเป็นเทคนิคที่มีประสิทธิภาพสูงในการวิเคราะห์และแยกแยะวัตถุในภาพถ่าย ใช้ CNN เพื่อเรียนรู้คุณสมบัติจากภาพของแมว เช่น รูปร่างของหู, ดวงตา, ขน และอื่นๆ ที่ช่วยบ่งบอกสายพันธุ์ สร้างชุดข้อมูลภาพของแมวแต่ละสายพันธุ์เพื่อฝึกโมเดล

### การพัฒนา

พัฒนาโดยการโมเดลจาก Github มาแก้ปัญหาในการจำแนกภาพจะได้รับการ Predict ที่ดีเมื่อใช้เครือข่ายประสาทเทียมแบบ Convolutional (CNN) ซึ่งเป็นหนึ่งในวิธีที่อยู่ใน Deep learning ที่ได้รับความนิยม CNN มีแนวคิดหลักคือการใช้ชุดของเลเยอร์ที่สลับกันระหว่างเลเยอร์ Convolutional (สำหรับการตรวจจับคุณลักษณะในภาพ) และเลเยอร์การสุ่มตัวอย่าง (สำหรับการลดขนาดข้อมูล) เพื่อเปลี่ยนจากการวิเคราะห์รายละเอียดเฉพาะในภาพไปเป็นข้อมูลที่เป็นนามธรรมมากขึ้น เช่น คุณลักษณะสำคัญที่ช่วยในการจำแนกภาพ โมเดล CNN จะเรียนรู้โดยการคัดกรองรายละเอียดที่ไม่สำคัญและเน้นสิ่งที่สำคัญเพื่อนำไปสู่การตัดสินใจขั้นสูง เช่น การระบุวัตถุหรือจำแนกประเภท แต่การฝึกโมเดลนี้ต้องใช้ทรัพยากรจำนวนมาก เช่น CPU หรือ GPU และต้องใช้เวลาในการฝึกกับข้อมูลจำนวนมาก

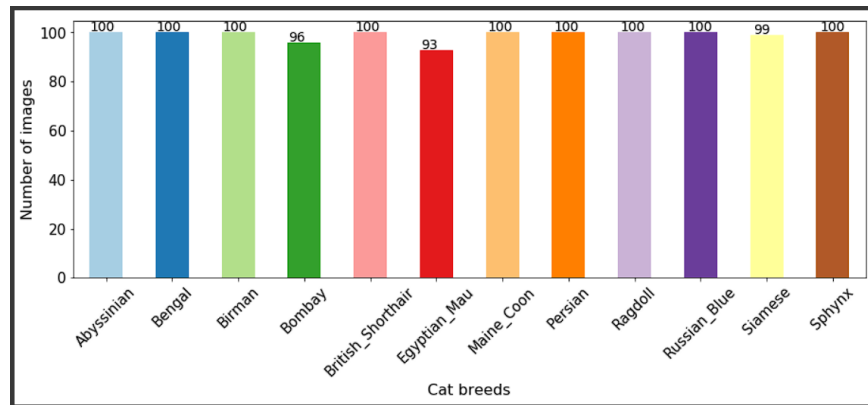
เพื่อประหยัดเวลา เราได้ใช้โมเดล Inception ของ Google ซึ่งเป็น CNN ที่ผ่านการฝึกมาก่อนแล้ว (จากชุดข้อมูลขนาดใหญ่ เช่น ImageNet) และนำมาใช้เทคนิค Transfer Learning หรือการเรียนรู้แบบถ่ายโอน วิธีนี้ช่วยให้เราไม่ต้องเริ่มต้นฝึกโมเดลใหม่ตั้งแต่ต้น แต่สามารถใช้โมเดลที่ฝึกมาแล้วและเพิ่มการฝึกเฉพาะในส่วนสุดท้ายของโมเดลด้วยข้อมูลภาพของเราเองแทน

## ขั้นตอนการเตรียมข้อมูล

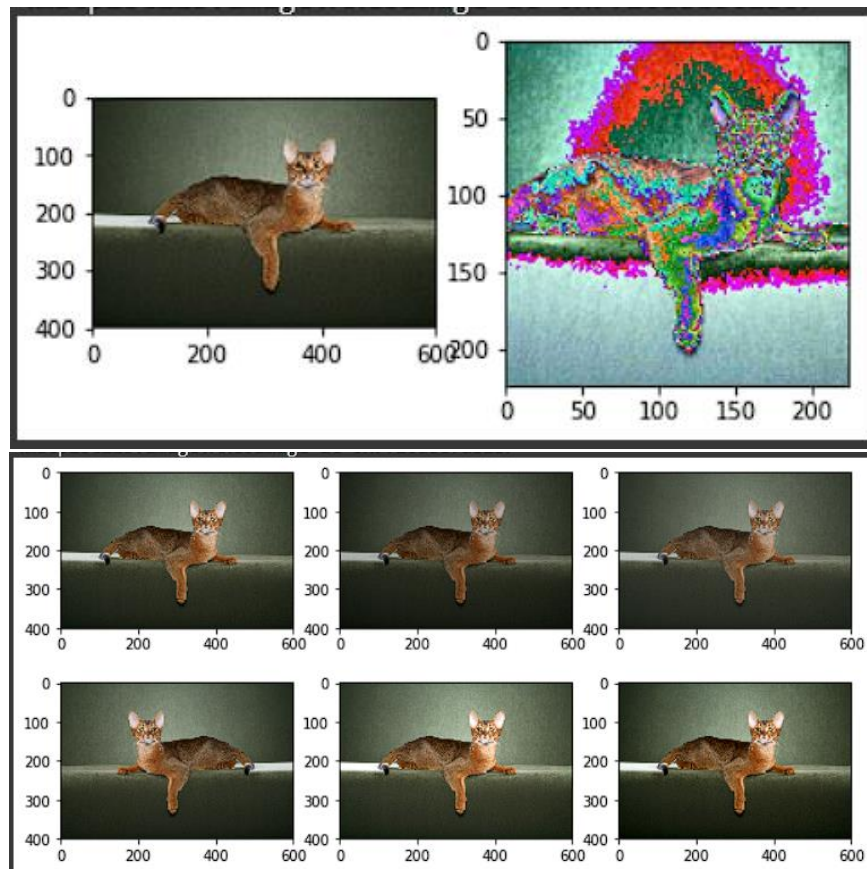
สามารถดูได้ที่ลิงก์ :

[https://colab.research.google.com/drive/165JBje5X78U5\\_yTewX4wpFtZUrw7KF4O?usp=sharing](https://colab.research.google.com/drive/165JBje5X78U5_yTewX4wpFtZUrw7KF4O?usp=sharing)

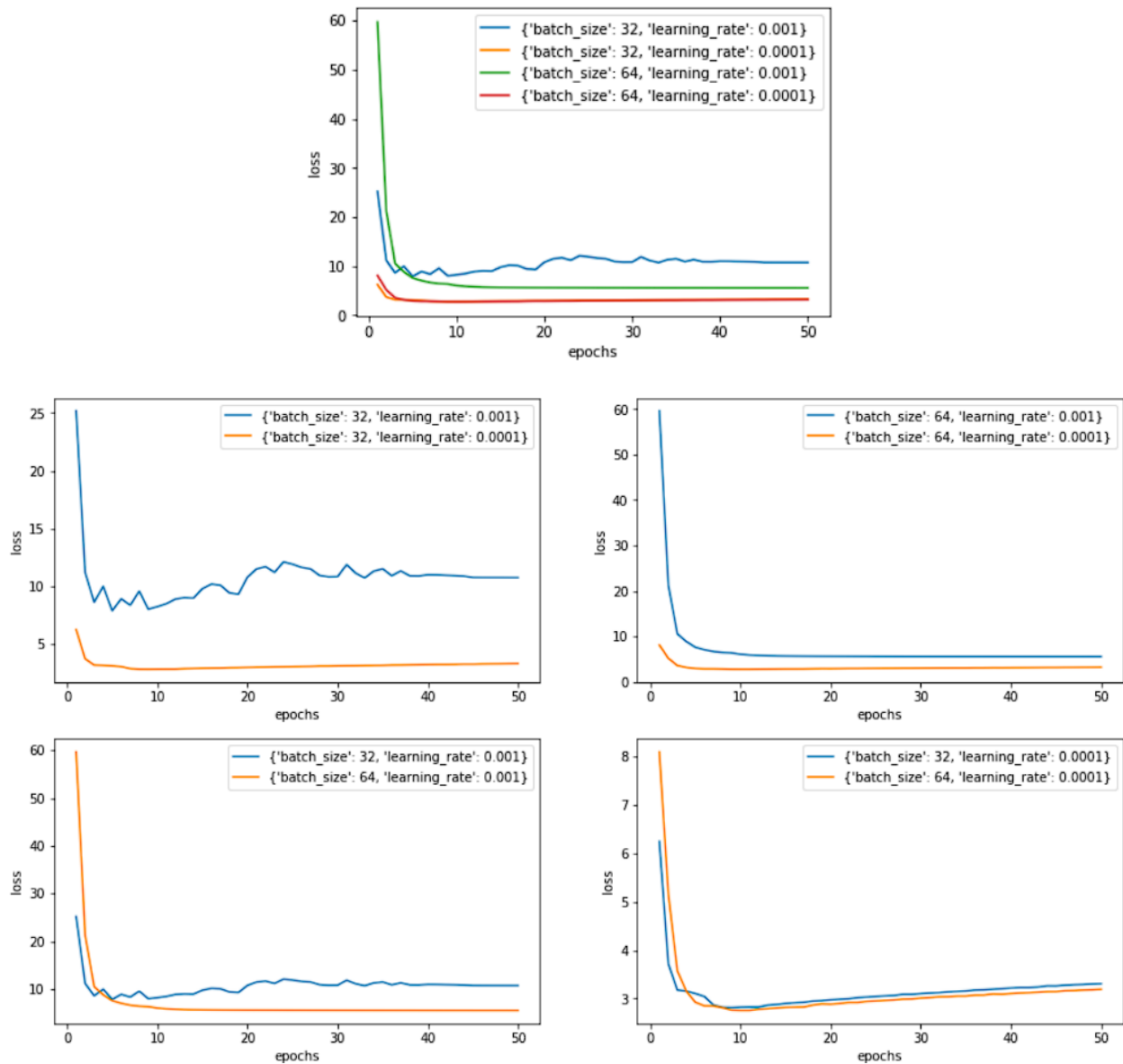
1. การกระจายคลาสเพื่อดูว่าข้อมูลมีความสมดุลหรือไม่สมดุล



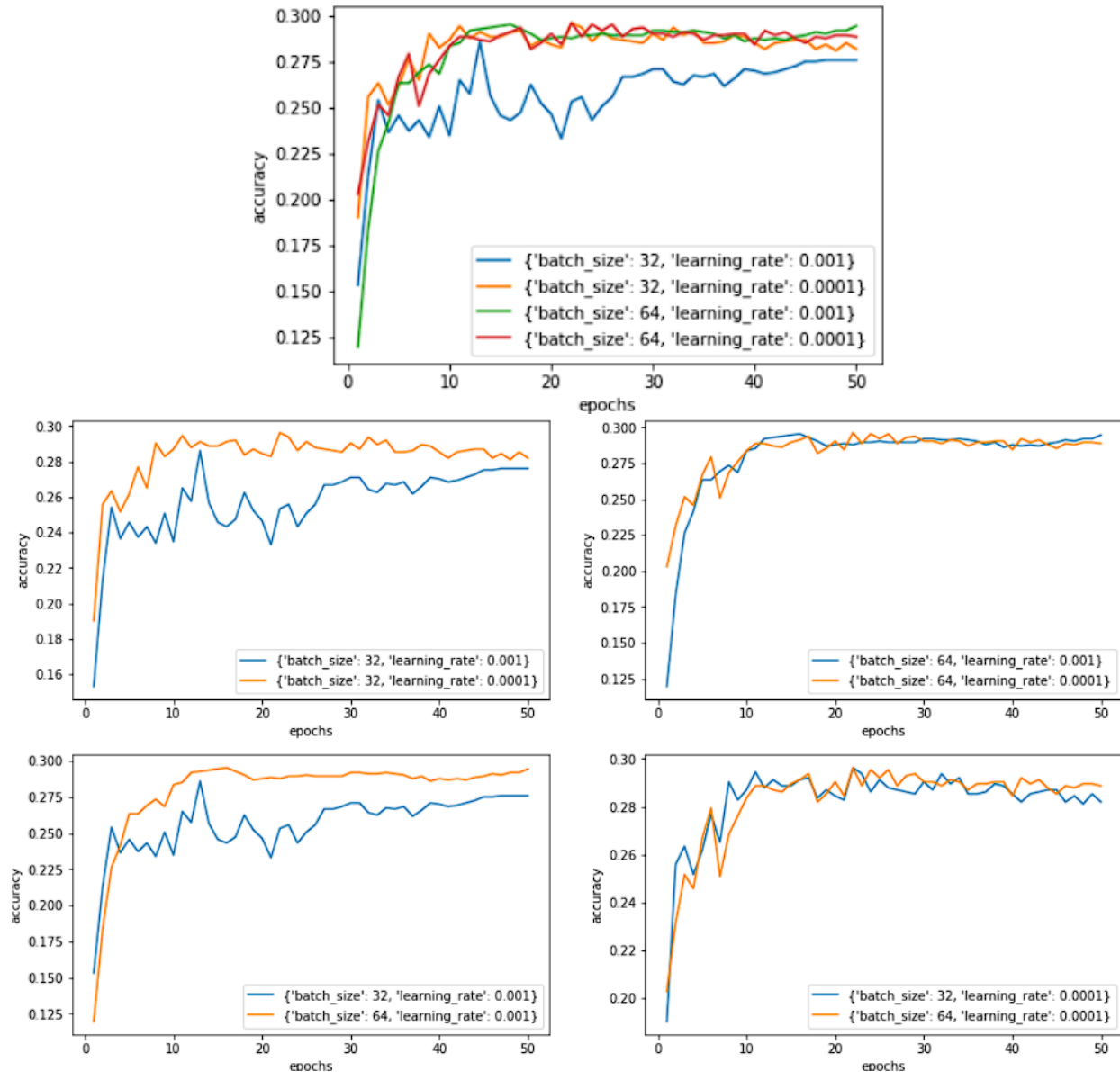
2. แสดงภาพตัวอย่างที่ถูกแปลงแล้ว



## 3. Result of grid search &amp; cross validation



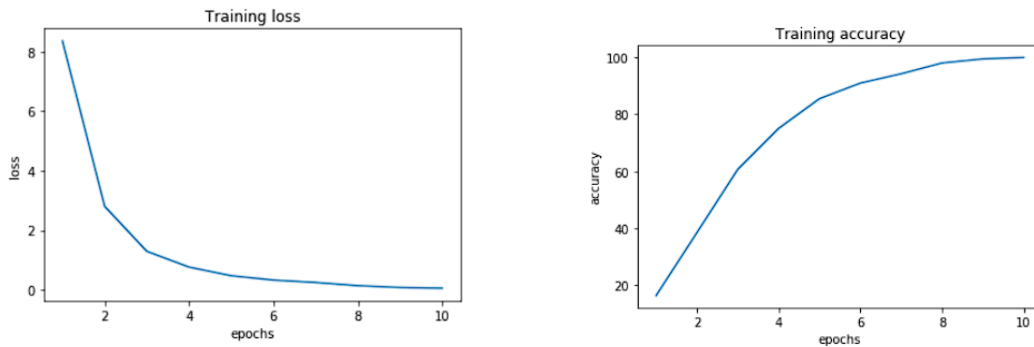
การสูญเสียการตรวจสอบต่อ epoch



### ความแม่นยำในการตรวจสอบต่อ epoch

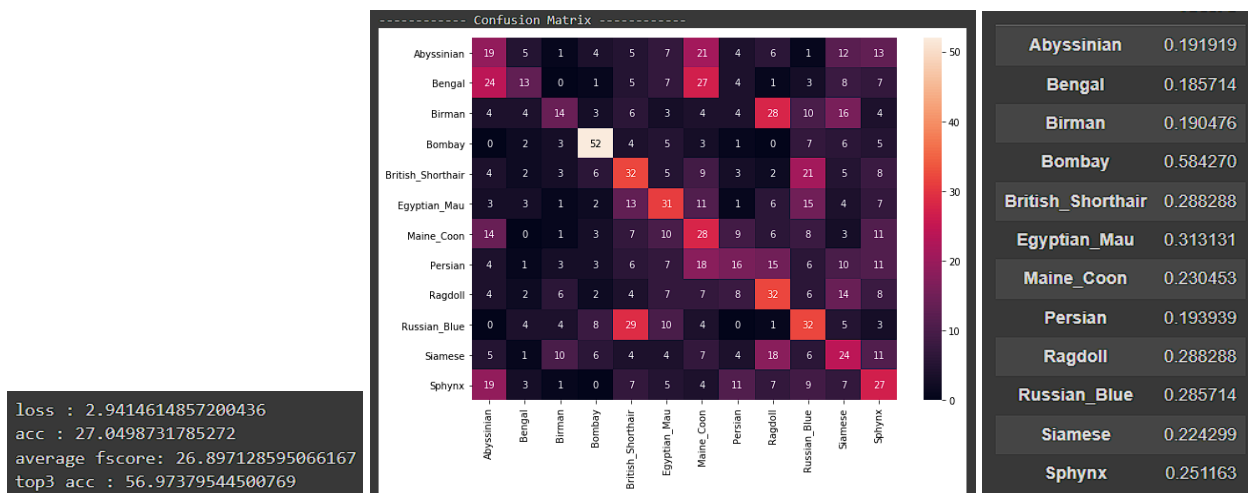
เป็นการดึงเอาผลลัพธ์ที่ดีที่สุดจากการทดสอบโมเดลหลายครั้ง (cross-validation) โดยดูจากประวัติที่บันทึกไว้ใน `cv_history` ตัวอย่างเช่น ถ้าเราทดสอบโมเดล 5 รอบ แล้วแต่ละรอบได้ผลลัพธ์แตกต่างกันจะหาว่ารอบไหนทำงานได้ดีที่สุด แล้วนำผลลัพธ์นั้นมาแสดง หลังจากได้ผลลัพธ์ที่ดีที่สุดแล้วจะแสดงผลออกมาทางหน้าจอให้เราวิเคราะห์ เป็นการแสดงผลทั้งหมดจากการทำ cross-validation ในรูปแบบของกราฟ เพื่อให้เห็นภาพรวมของการทดสอบโมเดลแต่ละรอบ เช่น กราฟอาจแสดงให้เห็นว่าความแม่นยำของโมเดลดีขึ้นหรือแย่ลงในแต่ละรอบ ซึ่งช่วยให้เราเข้าใจแนวโน้มและเลือกโมเดลที่ดีที่สุดได้ง่ายขึ้น

## 4. Result of training



การแสดงผลค่าความสูญเสีย (loss) และความแม่นยำ (accuracy) ของโมเดลในรอบการฝึกครั้งสุดท้าย รวมถึงการวาดกราฟแสดงแนวโน้มของผลลัพธ์การฝึกทั้งหมด

## 5. Result of testing



Loss : ค่า loss เท่ากับ 2.941 ซึ่งบ่งบอกถึงความผิดพลาดของโมเดลในระดับหนึ่ง

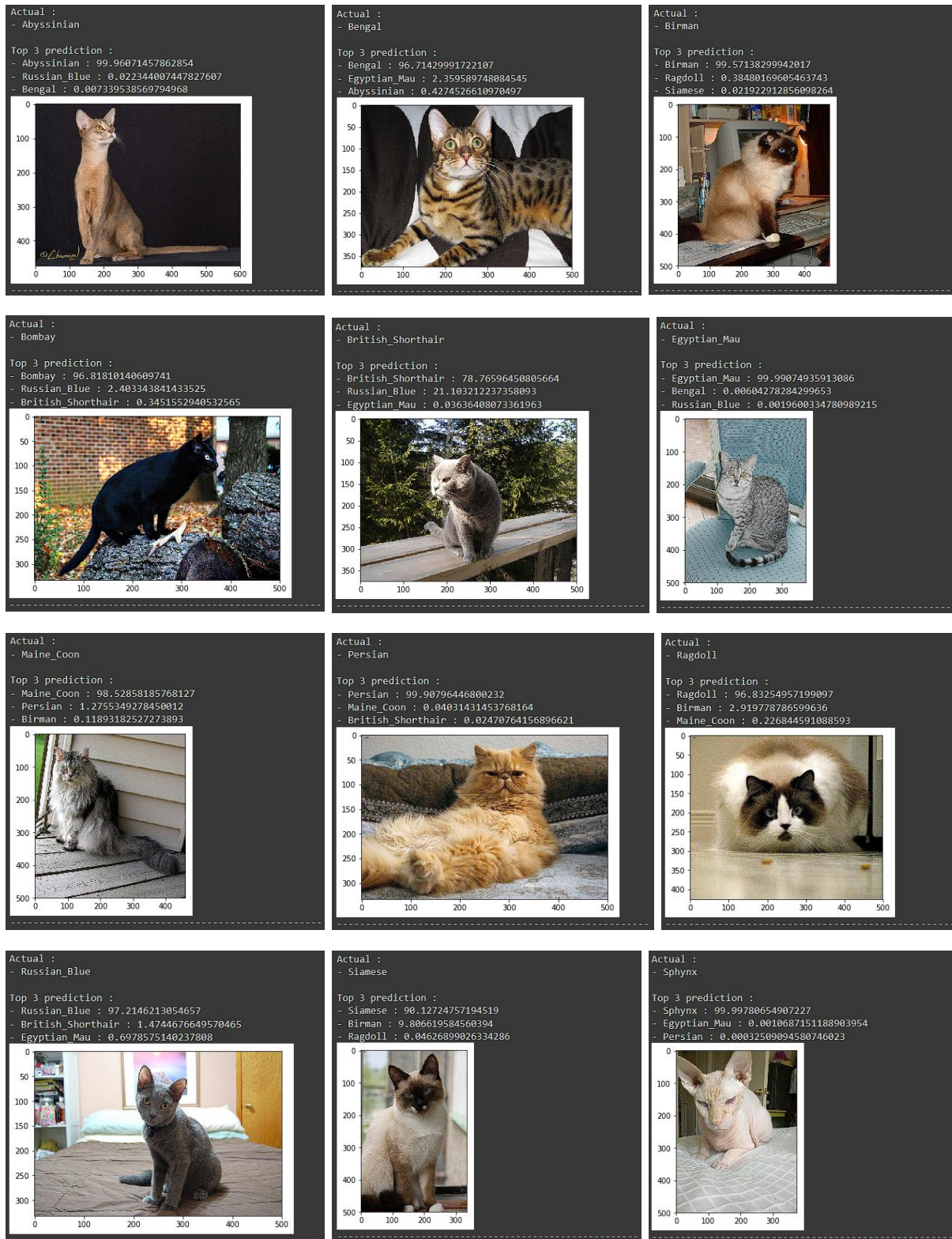
Accuracy : ความแม่นยำของโมเดลในการทดสอบเท่ากับ 27.05% แสดงว่าทำนายได้ถูกต้องประมาณ 27% ของข้อมูลทดสอบทั้งหมด

average fscore : ค่าเฉลี่ยของ F-score เท่ากับ 26.9% ซึ่งบ่งบอกถึงความสมดุลของความแม่นยำและการครอบคลุมในแต่ละคลาส

top3 acc : ความแม่นยำของ Top-3 อยู่ที่ 56.97% หมายความว่าใน 3 อันดับแรกของค่าทำนาย โมเดลมีคำตอบที่ถูกต้องอยู่มากกว่า 56% ของครั้ง



## 6. สร้างภาพการทำนายที่ถูกต้องสำหรับแต่ละสายพันธุ์



การประมวลผลภาพ : ภาพจะต้องถูกแปลงเป็นรูปแบบที่โมเดลสามารถเข้าใจได้ เช่น การปรับขนาด, การทำ Normalization, และการจัดรูปเป็นเทนเซอร์

การทำนายโดยโมเดล : โมเดลจะคำนวณความน่าจะเป็นของภาพว่าจะเป็นคลาสใดบ้าง แล้วทำการดึงค่าทำนายที่ น่าจะเป็นมากที่สุดออกมา

การแสดงผลลัพธ์ : ผลลัพธ์จะถูกแสดงในรูปแบบของป้ายชื่อคลาสพร้อมกับความน่าจะเป็นของแต่ละค่าทำนาย และ แสดงภาพที่ทำนายควบคู่ไปด้วย

ใช้เพื่อทำนายภาพว่าเป็นประเภทใด โดยการแสดงผลลัพธ์หลายอันดับ (top-k) พร้อมกับแสดงภาพที่ใช้ทำนายเพื่อ ช่วยในการตรวจสอบผลลัพธ์

### ขั้นตอนการพัฒนาเว็บไซต์

สามารถดูได้ที่ลิงก์ : <https://github.com/sareefhub/Web-Cat-Breeds-Classifer>

เราได้ทำการพัฒนาโดยใช้ Flask ในการสร้างเว็บไซต์โดยจะมีไฟล์หลักในการทำงานดังนี้

1. predict.py ใช้ในการจำแนกสายพันธุ์ของแมว
2. app.py ใช้ในการรันหน้าเว็บไซต์และสามารถเข้าผ่าน localhost : <http://127.0.0.1:5000/>
3. index.html การแสดงหน้าเว็บไซต์
4. styles.css ใช้ในการตกแต่งหน้าตาเว็บไซต์ให้สวยงามยิ่งขึ้น

### หลักการทำงานของโค้ด

```
def predict_image(image_path):
    # Read in the image_data
    image_data = tf.io.gfile.GFile(image_path, 'rb').read()

    # Loads label file, strips off carriage return
    label_lines = [line.rstrip() for line in
tf.io.gfile.GFile("nn_files/retrained_labels.txt")]

    # Unpersists graph from file
    with tf.io.gfile.GFile("nn_files/retrained_graph.pb", 'rb') as f:
        graph_def = tf.compat.v1.GraphDef()
        graph_def.ParseFromString(f.read())
        _ = tf.import_graph_def(graph_def, name='')

```



```

with tf.compat.v1.Session() as sess:
    # Feed the image_data as input to the graph and get first prediction
    softmax_tensor = sess.graph.get_tensor_by_name('final_result:0')

    predictions = sess.run(softmax_tensor, {'DecodeJpeg/contents:0':
image_data})

    # Sort to show labels of first prediction in order of confidence
    top_k = predictions[0].argsort()[-len(predictions[0]):][::-1]

    # Get top label and score
    top_label = label_lines[top_k[0]]
    top_score = predictions[0][top_k[0]]
    return top_label, top_score

```

โค้ดนี้เป็นฟังก์ชันที่ใช้ในการทำนายประเภทของภาพ (Image Classification) โดยใช้โมเดล TensorFlow ที่ถูกฝึกมาแล้ว กระบวนการทำงานของโค้ดสามารถอธิบายได้ดังนี้ :

โหลดข้อมูลภาพ : ฟังก์ชันจะอ่านภาพจากไฟล์ที่กำหนดและแปลงเป็นข้อมูลดิบเพื่อใช้เป็นข้อมูลนำเข้าในการทำนาย

โหลดโมเดลและป้ายชื่อ (Label): มันจะโหลดโมเดล TensorFlow (ไฟล์ .pb) และไฟล์ที่เก็บป้ายชื่อของคลาสต่างๆ ที่โมเดลทำนายได้ (ไฟล์ .txt)

ทำนายภาพ : โค้ดนี้จะเรียกใช้โมเดลเพื่อทำนายผลลัพธ์ โดยใช้ภาพที่ให้เป็นข้อมูลนำเข้า และคำนวณความน่าจะเป็นสำหรับแต่ละคลาสที่โมเดลสามารถทำนายได้

เรียงลำดับผลลัพธ์ : ผลลัพธ์จะถูกจัดเรียงตามความมั่นใจ (confidence score) เพื่อแสดงคลาสที่มีความน่าจะเป็นมากที่สุดในการทำนาย

คืนค่าผลลัพธ์ : ฟังก์ชันจะคืนค่าคลาสที่โมเดลทำนายว่าเป็นไปได้มากที่สุด พร้อมกับความน่าจะเป็นของคลาสนั้น

โดยสรุป ฟังก์ชันนี้จะอ่านภาพ โหลดโมเดล TensorFlow ที่ฝึกมาแล้ว และใช้โมเดลนั้นเพื่อทำนายว่าภาพนั้นอยู่ในคลาสใด พร้อมคืนค่าป้ายชื่อและคะแนนความมั่นใจที่โมเดลคำนวณได้

```

import os
from flask import Flask, request, redirect, url_for, render_template
from werkzeug.utils import secure_filename
from predict import predict_image # Import predict_image from predict.py

app = Flask(__name__)
app.config['UPLOAD_FOLDER'] = 'static/uploads'

ALLOWED_EXTENSIONS = {'jpg', 'jpeg', 'png'}

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

@app.route('/', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        if 'file' not in request.files:
            return redirect(request.url)

        file = request.files['file']

        if file and allowed_file(file.filename):
            # ลบไฟล์เก่าทั้งหมดใน UPLOAD_FOLDER
            for filename in os.listdir(app.config['UPLOAD_FOLDER']):
                file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
                try:
                    os.remove(file_path)
                except Exception as e:
                    print(f"Error deleting file {file_path}: {e}")

            # บันทึกไฟล์ใหม่
            filename = secure_filename(file.filename)
            filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
            file.save(filepath)

            # Predict the image using predict_image function
            label, score = predict_image(filepath)

            return render_template('index.html', label=label, score=score,
image_url=filepath)

        # การรีโหลดเรจคัลกลับไปหน้าแรกเมื่อรีเฟรช
        return render_template('index.html', label=None, score=None)

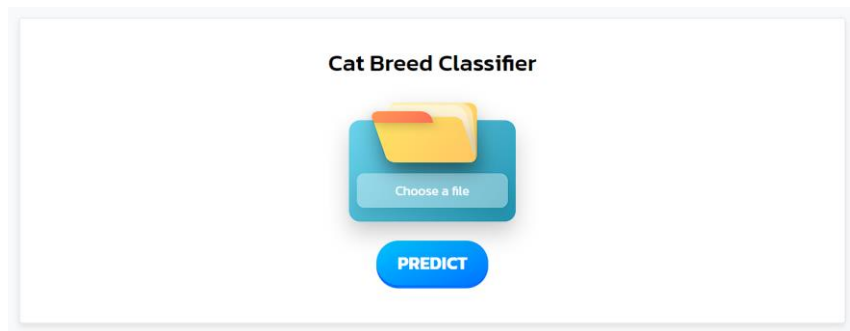
```

```
if __name__ == '__main__':
    app.run(debug=True)
```

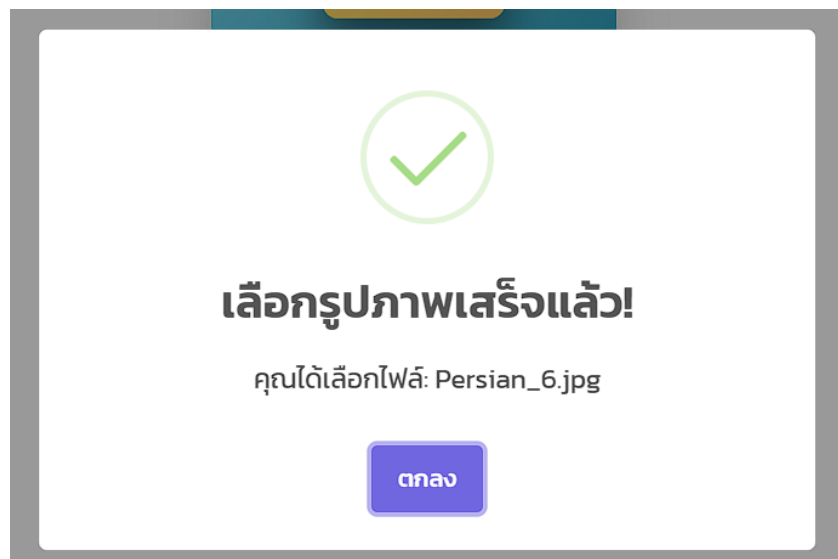
โค้ดนี้เป็นเว็บแอปพลิเคชันที่สร้างด้วย Flask ซึ่งให้ผู้ใช้อัปโหลดภาพผ่านหน้าเว็บ จากนั้นนำภาพที่อัปโหลดไปผ่านกระบวนการทำนายด้วยโมเดลที่ถูกฝึกมามาก่อนหน้านี้ โดยจะแสดงผลการทำนายประเภทของภาพและความมั่นใจในการทำนายบนหน้าเว็บ ผู้ใช้สามารถอัปโหลดภาพใหม่ได้เรื่อยๆ และทุกครั้งที่ย่ออัปโหลด ระบบจะลบภาพเก่าออกจากโฟลเดอร์ที่เก็บไฟล์อัปโหลด

### พัฒนา UI และ ทดสอบการใช้งาน

หน้าตาเว็บไซต์

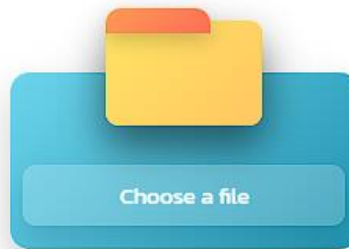


สามารถกดเลือกไฟล์รูปภาพแมวที่ต้องการ Predict



เมื่ออัปโหลดแล้วจะแสดงว่าได้ทำการเลือกไฟล์สำเร็จแล้ว

## Cat Breed Classifier



### Prediction Result

**Breed:** persian

**Confidence:** 56.75%



เมื่อทำการ Predict จะแสดงผลการทำนายสายพันธุ์ของแมวและเปอร์เซ็นต์

วิดีโอแนะนำเสนอ

สามารถดูได้ที่ลิงก์ : <https://youtu.be/2p7oJXEWp6Q?si=Alzka5klnxkFAkq6>

**บรรณานุกรม**

**อ้างอิง**

Github Colab : <https://github.com/willyfh/cat-breed-classification-using-cnn>

Github Predicted : <https://github.com/batogov/cat-breeds-classifier?tab=readme-ov-file>