**Building a Convolutional Neural Network (CNN) Model for Military Aircraft Detection via Image Processing**

ในบทความนี้เราจะพาไปเรียนรู้เกี่ยวกับการสร้างโมเดล **Convolutional Neural Network (CNN)** สำหรับการจำแนกประเภทของเครื่องบินทหารจากภาพถ่ายทางอากาศ การสร้างโมเดลนี้มีเป้าหมายเพื่อช่วยในการพัฒนาระบบตรวจจับเครื่องบินทหารในสภาพแวดล้อมต่าง ๆ ซึ่งเป็นหนึ่งในงานที่มีความท้าทายสูงในด้าน **Computer Vision** โดยเฉพาะในการทำงานกับชุดข้อมูลที่มีความหลากหลายของประเภทเครื่องบิน

**จุดเริ่มต้นของโครงการ**

ความชอบส่วนตัวของผู้เขียน

โดยส่วนตัวของผู้เขียนบทความนี้ (ผมนั้นเองฮ่า ๆ ) ที่ชื่นชอบเครื่องบินรบต่าง ๆ ของหลาย ๆ ประเทศ อาทิ **F22**, **SU57**, **F15** ด้วยความสวยงามและเทคโนโลยีล้ำสมัยที่มาก่อนกาล สุดยอดนวัตกรรมของ **E n g i n e e r i n g** (เสียง 9Arm) และมนุษยชาติได้ควบรวมอยู่ในนี้แล้ว ซึ่งเครื่องบินเหล่านี้ไม่เพียงแต่เป็นสัญลักษณ์ของความสามารถทางทหาร แต่ยังเป็นผลงานที่สะท้อนถึงความก้าวหน้าทางเทคโนโลยีที่สำคัญในการพัฒนาเครื่องมือและระบบต่าง ๆ ในการตรวจจับและวิเคราะห์ข้อมูลจากภาพ

และเพื่อน ๆ ล่ะครับ อวยเครื่องบินรบลำไหนกันหรือเปล่าฮ่า ๆ

**ชุดข้อมูล (Dataset)** ที่ใช้ในการฝึกโมเดลนี้คือ [**Military Aircraft Detection Dataset**](https://www.kaggle.com/datasets/a2015003713/militaryaircraftdetectiondataset) ซึ่งรวบรวมภาพของเครื่องบินทหารในมุมต่าง ๆ โดยที่แต่ละประเภทจะมีการจัดกลุ่มเป็นคลาสต่าง ๆ ตัวอย่างเช่น เครื่องบินขับไล่, เครื่องบินทิ้งระเบิด, หรือเครื่องบินขนส่ง และสิ่งสำคัญที่สุดคือการจำแนกประเภทเครื่องบินเหล่านี้ให้ถูกต้องจากภาพถ่ายที่มีรายละเอียดสูง

การใช้งาน **Convolutional Neural Networks (CNN)** ในงานด้านนี้ได้รับการพิสูจน์แล้วว่าเป็นทางเลือกที่ดีที่สุดสำหรับการตรวจจับลักษณะต่าง ๆ ในภาพ เช่น รูปทรงและลวดลายที่เครื่องบินแต่ละประเภทมีอยู่

**Demo App & Repo**

ในการแสดงผลการทำงานของโมเดลนี้ ผมได้พัฒนา **Demo App** โดยใช้ **Streamlit** เพื่อให้ผู้ใช้สามารถ Demo การจำแนกประเภทของเครื่องบินทหารได้แบบเรียลไทม์ผ่านเว็บแอปพลิเคชัน

คุณสามารถเข้าไปทดลองใช้งานได้ที่ [**Airforce Detector Demo WebApp**](https://airforce-detector.streamlit.app/)

และส่วนของ Repository: [**Airforce Detector Repo GitHub**](https://github.com/Nukaze/airforce-detector)

**1. การเตรียมข้อมูล (Data Preprocessing)**

การเตรียมข้อมูลเป็นขั้นตอนแรกที่สำคัญในการสร้างโมเดลใด ๆ โดยเฉพาะในงานด้านการจำแนกประเภทภาพ การใช้ **ImageDataGenerator** ของ Keras ช่วยให้เราสามารถทำการ **rescale** ค่า pixel ของภาพจาก 0–255 ให้เป็นค่าในช่วง [0, 1] ซึ่งช่วยให้โมเดลเรียนรู้ได้ดีขึ้น



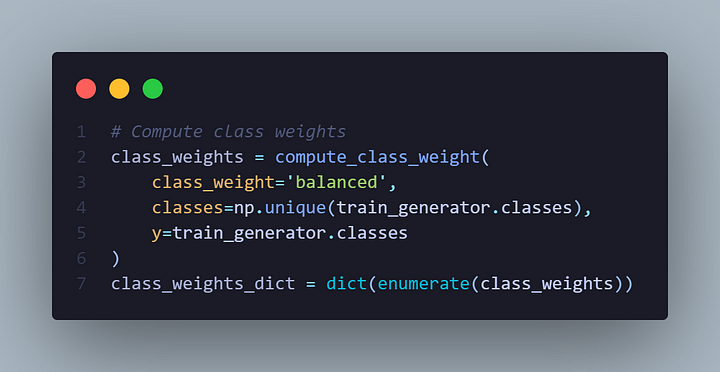
**Data Preparation**

ภาพในชุดข้อมูลถูกปรับขนาดให้มีขนาด 150x150 พิกเซล ซึ่งเป็นขนาดที่เหมาะสมกับการฝึก CNN แบบพื้นฐาน ขนาดนี้จะไม่ใหญ่เกินไปจนทำให้ใช้เวลาฝึกนานเกินไป และไม่เล็กเกินไปจนสูญเสียข้อมูลสำคัญที่อาจจะมีผลต่อการจำแนก

นอกจากการ rescale แล้ว ยังมีการแบ่งข้อมูลออกเป็น **training set** และ **validation set** โดยใช้ validation\_split=0.3 ซึ่งหมายความว่า 30% ของข้อมูลจะใช้ในการทดสอบ และ 70% จะใช้ในการฝึก

**2. การคำนวณ Class Weight**

เมื่อทำการจำแนกภาพออกเป็นหลายประเภท เราอาจจะพบปัญหาที่ **class imbalance** หรือปัญหาคลาสที่มีจำนวนข้อมูลน้อยกว่าคลาสอื่น ๆ ซึ่งในกรณีนี้อาจทำให้โมเดลเกิดการ bias ต่อคลาสที่มีข้อมูลมากกว่าได้



**compute\_class\_weight**

เพื่อลดปัญหานี้ เราจึงใช้ **class weights** ซึ่งช่วยให้โมเดลให้ความสำคัญกับคลาสที่มีข้อมูลน้อยมากขึ้น โดยใช้ฟังก์ชัน compute\_class\_weight จาก **sklearn** เพื่อคำนวณน้ำหนักที่เหมาะสมสำหรับแต่ละคลาส ก่อนที่จะนำไปใช้ในระหว่างการฝึกโมเดล

**3. การออกแบบโมเดล CNN**

โมเดลที่เราเลือกใช้ในการจำแนกประเภทเครื่องบินทหารคือ **Convolutional Neural Network (CNN)** ซึ่งได้รับความนิยมอย่างมากในงานด้านการประมวลผลภาพ ด้วยโครงสร้างที่มีความลึกและการใช้ฟังก์ชัน **ReLU** ในแต่ละเลเยอร์ของ **Conv2D** โมเดลนี้สามารถเรียนรู้ลักษณะสำคัญจากภาพได้ดี



**Build Model**

**โครงสร้างของโมเดล**

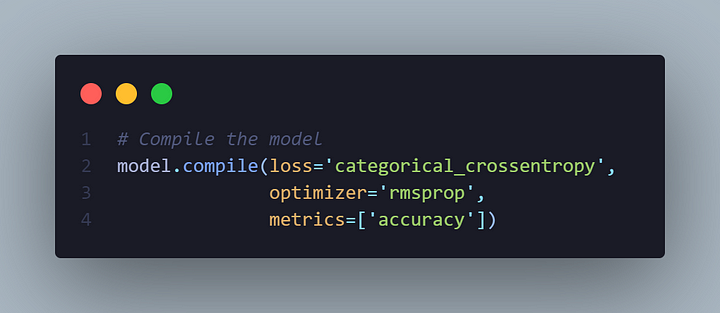
โมเดลนี้มีชั้น **Convolutional Layer** จำนวน 3 ชั้น โดยแต่ละชั้นจะใช้ **ฟิลเตอร์** ขนาด 3x3 เพื่อเรียนรู้ลักษณะของเครื่องบินในภาพ และทำการ **Max Pooling** หลังจากนั้นเพื่อลดขนาดของข้อมูลลง

1. **Conv2D Layer**: แต่ละชั้นจะใช้ **ReLU activation** เพื่อให้โมเดลสามารถจับลักษณะสำคัญได้ เช่น ขอบเขตของเครื่องบินหรือรายละเอียดที่แตกต่างจากพื้นหลัง
2. **MaxPooling Layer**: จะช่วยลดขนาดของข้อมูลและป้องกันการ overfitting โดยการเลือกค่าเฉพาะที่มีความสำคัญที่สุดจากข้อมูล
3. **Fully Connected Layer**: หลังจากนั้นโมเดลจะทำการเชื่อมต่อข้อมูลทั้งหมดที่ผ่านการแปลงรูปแบบมาแล้วจากการใช้ **Flatten Layer** และส่งข้อมูลเข้าสู่ **Dense Layer** เพื่อทำการทำนายผล

สุดท้ายโมเดลจะใช้ **Softmax Activation Function** ที่ชั้นสุดท้ายเพื่อให้ผลลัพธ์เป็นการจำแนกประเภทที่เหมาะสมที่สุดจากคลาสที่มีทั้งหมด

**4. การฝึกและทดสอบโมเดล**

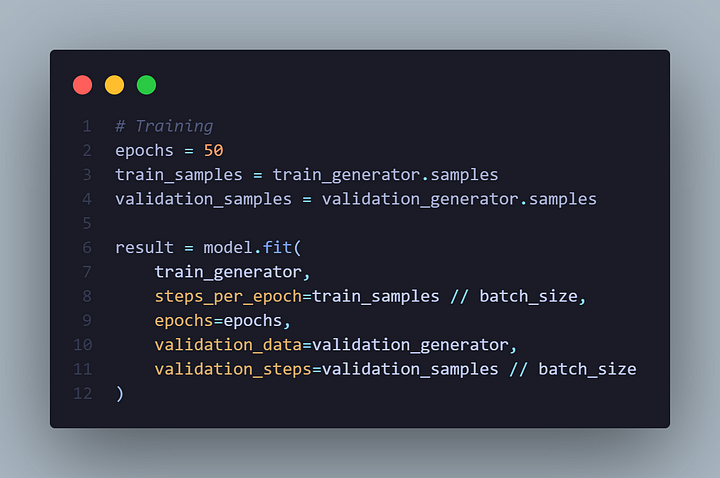
การฝึกโมเดลนี้ใช้ **RMSprop optimizer** ซึ่งเป็น optimizer ที่เหมาะสมสำหรับงานที่มีการเรียนรู้จากข้อมูลที่มีความซับซ้อนสูง เช่น งานด้านภาพ ด้วยการใช้ **categorical crossentropy loss** ซึ่งเป็นฟังก์ชันที่เหมาะสมสำหรับการจำแนกประเภทแบบหลายคลาส



**Compile Model**

ในระหว่างการฝึกโมเดล เราจะใช้ **Epochs** จำนวน 50 รอบ และ **Batch Size** ที่ 32 เพื่อลดเวลาในการฝึกและเพิ่มประสิทธิภาพ

เมื่อการฝึกเสร็จสิ้น เราจะทำการทดสอบโมเดลโดยใช้ **Validation Set** และประเมินผลจาก **accuracy** และ **confusion matrix** เพื่อดูผลการทำนายว่าโมเดลสามารถจำแนกประเภทได้ถูกต้องมากน้อยแค่ไหน

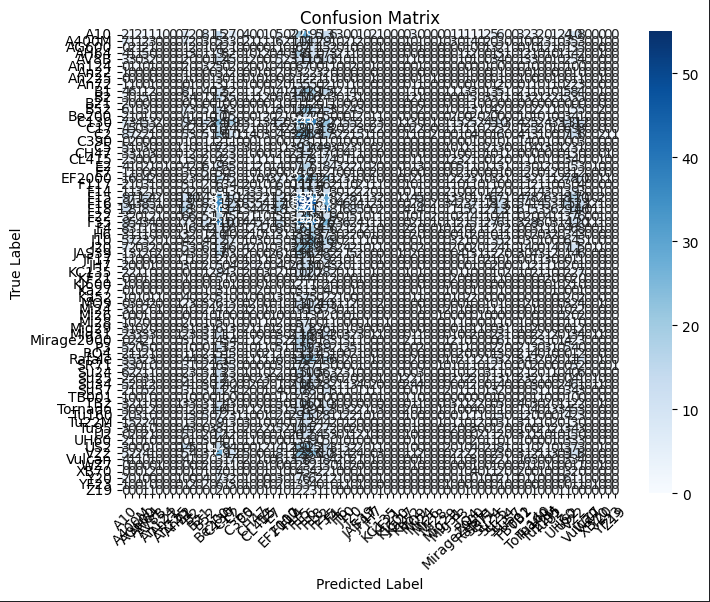


Training Model

**5. การประเมินผลและการแสดงผล**

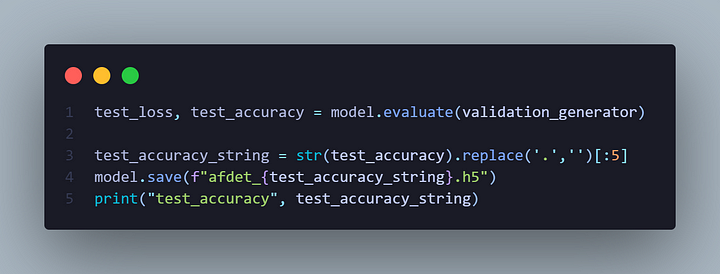
ในขั้นตอนนี้ เราจะทำการวิเคราะห์ประสิทธิภาพของโมเดลโดยการแสดง **Training Accuracy** และ **Training Loss** ผ่านกราฟ ซึ่งจะช่วยให้เราเห็นการพัฒนาของโมเดลในแต่ละรอบการฝึก นอกจากนี้เรายังใช้ **Confusion Matrix** และ **Classification Report** ในการประเมินการทำงานของโมเดลในแง่ของ **Precision, Recall, และ F1-Score** ในแต่ละคลาส

**Confusion Matrix** เป็นส่วนสำคัญส่วนนึงในการดูความสับสนระหว่าง Class ว่ามีมากน้อยแค่ไหน ซึ่งใน Dataset ของเราที่มีอยู่ประมาณ 74 Class ดังภาพด้านล่าง

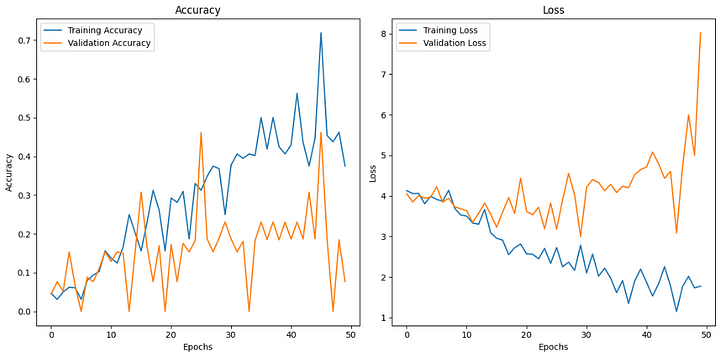


**Confusion Matrix**

เอ่อ.. แต่เอาเป็นว่าอันนี้เราข้ามไปอันถัดไปดีกว่าเนาะ ฮ่าา



**Evaluate Model**



**History Log**

“ผลลัพธ์ที่ได้จากการฝึกโมเดลในขั้นตอนแรกอาจจะยังไม่ได้ดีเท่าที่ควร โดยเฉพาะในส่วนของ **Validation** ซึ่งยังแสดงให้เห็นถึงการ O**verfitting** หรือการเรียนรู้ที่ไม่สามารถ generalize ข้อมูลใหม่ได้ดีเท่าที่ควร ดังนั้น ขั้นตอนถัดไปที่สำคัญคือการปรับจูน **(Tuning)** ตัว Configuration ต่าง ๆ เช่น **Learning Rate**, **Batch Size**, จำนวนของ **Layer** และการเลือก **Optimizer** ให้เหมาะสมกับลักษณะของโมเดลและวัตถุประสงค์ของการใช้งาน **(Purpose)** เพื่อให้โมเดลสามารถทำงานได้ดีที่สุด”

**สรุป**

การสร้างโมเดล **CNN** สำหรับการจำแนกประเภทเครื่องบินทหารจากภาพนั้นถือเป็นงานที่ท้าทาย แต่ด้วยการเตรียมข้อมูลที่ดี การคำนวณ class weight และการออกแบบโมเดล **CNN** ที่เหมาะสม เราสามารถสร้างระบบที่สามารถจำแนกประเภทเครื่องบินทหารได้อย่างแม่นยำและมีประสิทธิภาพ

การใช้ **Convolutional Neural Network (CNN)** สำหรับการจำแนกประเภทภาพเป็นวิธีที่มีประสิทธิภาพสูงในงานด้านการประมวลผลภาพ โดยเฉพาะในการตรวจจับวัตถุที่มีลักษณะซับซ้อนและหลากหลายเช่นเครื่องบินทหารในภาพถ่ายทางอากาศ

**ป.ล.** ใน **Demo** ผมได้แอบใส่ **YOLOv8** ไปด้วย เพราะว่าทำไปทำมาแล้วสนุก ผมก็เลยเพิ่มไปอีกโมเดลนึงซะเลยยอิอิ

**ป.ล. 2.** ใน **Repo** เป็น **GitHub** ของผมเอง พวกคุณสามารถไปกดติดตามกันได้นะ เพราะการมี Follower ใน **GitHub** แล้วรู้สึก **Aura +99999** เบียวสุด ๆ ฮ่า ๆ

**Dataset**

ข้อมูลชุดนี้สามารถเข้าถึงได้ที่ [*Military Aircraft Detection Dataset*](https://www.kaggle.com/datasets/a2015003713/militaryaircraftdetectiondataset) บน Kaggle ซึ่งเป็นแหล่งข้อมูลที่ใช้ในการฝึกโมเดลสำหรับการตรวจจับเครื่องบินทหารจากภาพ

**References**

1. [*Keras Applications API*](https://keras.io/api/applications/)
2. [*Keras Image Classification Example*](https://keras.io/examples/vision/image_classification_from_scratch/)
3. [*TensorFlow Keras Classification Tutorial*](https://www.tensorflow.org/tutorials/keras/classification)
4. [*Image Classification using Keras GitHub Repository*](https://github.com/AvinashNath2/Image-Classification-using-Keras/tree/master)