

รายงานประจำวิชา

หลักพื้นฐานของป8ญญาประดิษฐ์ (Fundamentals of Artificial Intelligence)

รหัสวิชา 01418261 หมู[เรียน 870

เรื่อง

Face Mask Detection

กลุ่ม Chill Guys

จัดทำโดย

นาย ปัณณวัฒน์ นิ่งเจริญ รหัสนิสิต 6630250231

นาย พันธุ์ธัช สุวรรณวัฒนะ รหัสนิสิต 6630250281

นาย วรินทร์ สายปัญญา รหัสนิสิต 6630250435

นางสาว อัมพุชินี บุญรักษ์ รหัสนิสิต 6630250532

นาย ปุณณภพ มีฤทธิ์ รหัสนิสิต 6630250591

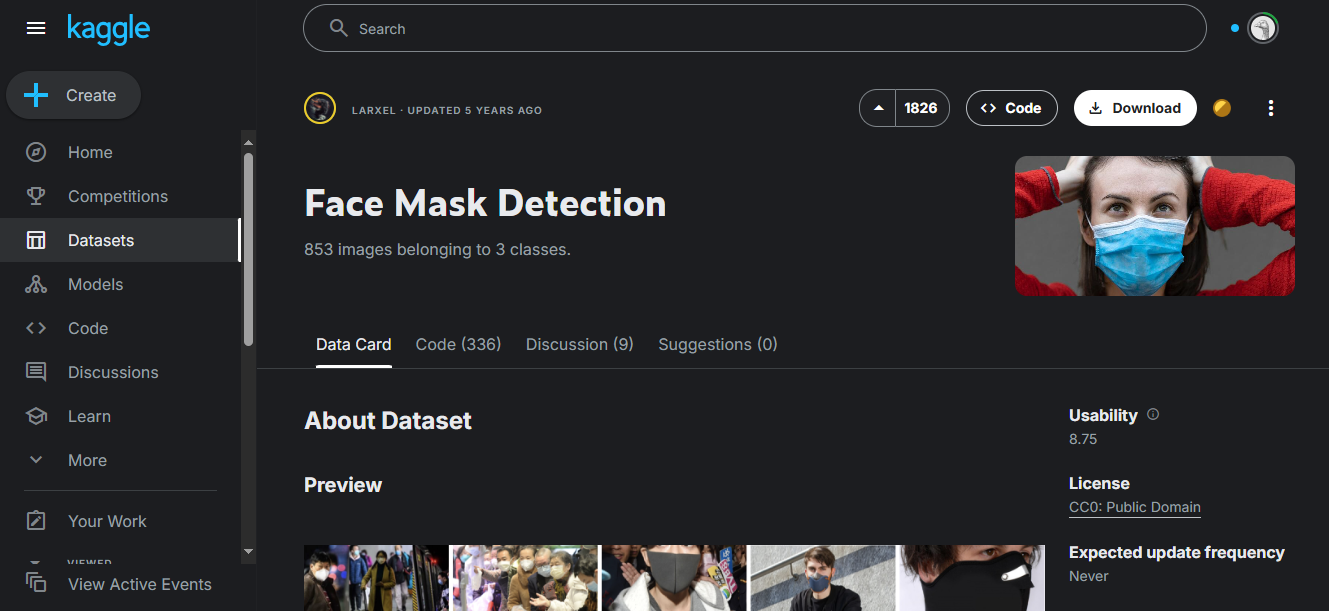
เสนอ

อาจารย์ ดร.ชโลธร ชูทอง

อาจารย์ประจำสาขาวิชาวิทยาการคอมพิวเตอร์

สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา

**อธิบายข้อมูล Dataset**



ชื่อข้อมูล dataset คือ **Face Mask Detection**

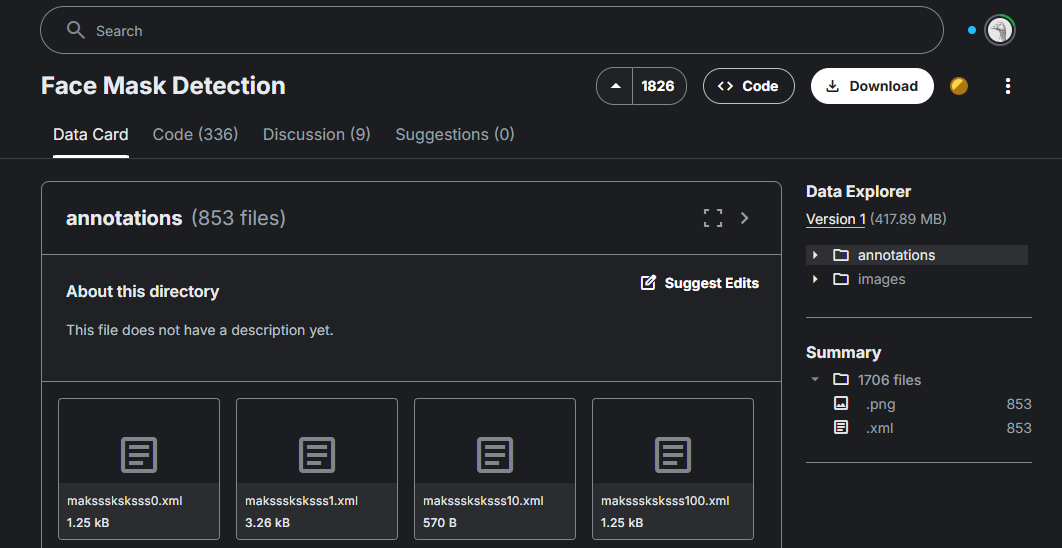
ชุดข้อมูลนี้เป็นชุดข้อมูลสำหรับการตรวจจับการสวมหน้ากากที่ใช้ในการป้องกันโรคระบาดทางเดินหายใจ เช่น COVID-19 หรือโรคอื่นๆ โดยมีรูปภาพทั้งหมด 853 ภาพ

ภาพแบ่งออกเป็น 3 ชนิด (3 classes) ได้แก่

* คนที่สวมหน้ากาก (With mask)
* คนที่ไม่สวมหน้ากาก (Without mask)
* คนที่สวมหน้ากากไม่ถูกต้อง (Mask worn incorrectly)

แต่ละภาพจะมีข้อมูล bounding boxes ที่บ่งชี้ถึงตำแหน่งของใบ้หน้าของคนในภาพ โดยใช้รูปแบบของ PASCAL VOC dataset ซึ่งเป็นรูปแบบที่ใช้กันอย่างแพร่หลายในการให้ข้อมูลการตรวจจับวัตถุในภาพ โดย บ่งชี้ถึงพิกเซลของภาพที่มีการตรวจจับและประเภทของวัตถุในพิกเซลนั้น ๆ ตามลำดับ

**โครงสร้าง Dataset**

****

มี 2 folders ประกอบด้วย folder

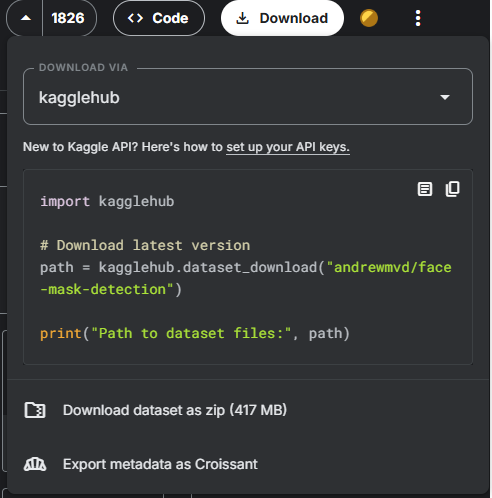
* images ที่เก็บรูปภาพข้อมูลคนใส่ mask และ ไม่ใส่ mask
* annotations ที่เก็บข้อมูลคำอธิบายของรูปภาพ

โดย folder images เก็บข้อมูลรูปภาพอยู่ที่ 853 รูปเป็นไฟล์นามสกุล .png

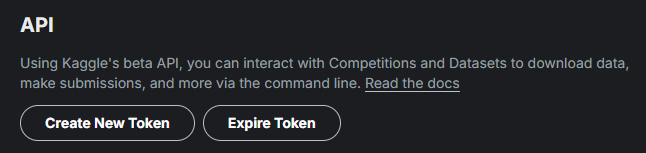
และ folder annotations เก็บข้อมูลคำอธิบายของรูปภาพที่ 853 ไฟล์เป็นไฟล์นามสกุล .xml

**ขั้นตอนการเตรียม Dataset**

ขั้นตอนการเตรียมการให้ทำการติดตั้ง library ของ kagglehub ใน Google Colab ให้เรียบร้อยย ทำตามขั้นตอนของเว็บไซต์ kaggle



จากนั้นให้ไปที่หน้า settings บัญชี kaggle ของเราทำการเปิดใช้งาน Kaggle API เพื่อจะได้ใช้ฟังก์ชันในการ download ข้อมูล datasets หลังจากเปิดใช้งานเสร็จจะได้ download ไฟล์ kaggle.json ซึ่งเป็นไฟล์ที่เก็บชื่อ username กับ api key เพื่อเอาไปกรอกในหน้า login ของ kaggle



**AI Models ที่สร้าง**

โมเดล AI ที่เราสร้างเป็นโมเดลที่ใช้สำหรับการจำแนก คนใส่แมส และ คนไม่ใส่แมส ตัวโมเดลที่เราสร้างมีด้วยกัน 4 โมเดล ประกอบด้วย Neural Network 1 โมเดล และ Machine Learning 3 โมเดล ได้แก่

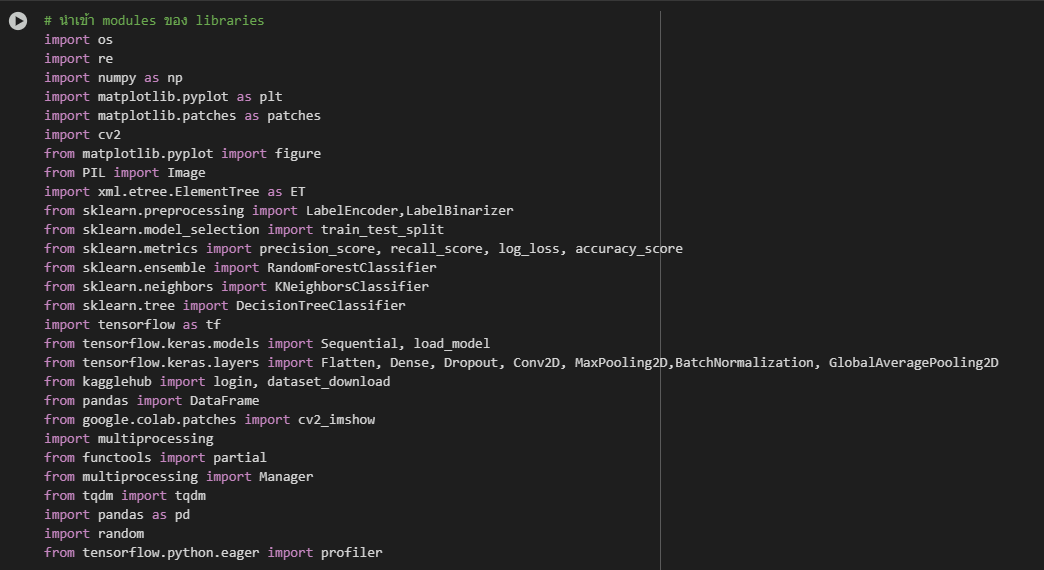
* Neural Network Model
* KNN Model
* Decision Model
* Random Forest Model

**อธิบายโค้ด**

1. ติดตั้ง libraries ที่จำเป็น

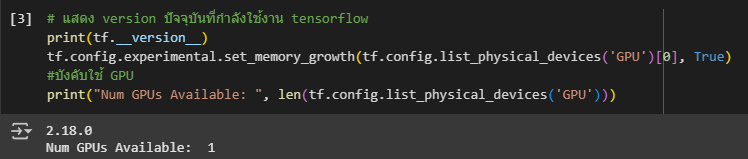


2. นำเข้า modules ของ Iibraries



3. แสดง Version ปัจจุบันที่กำลังใช้งานใน tensorflow

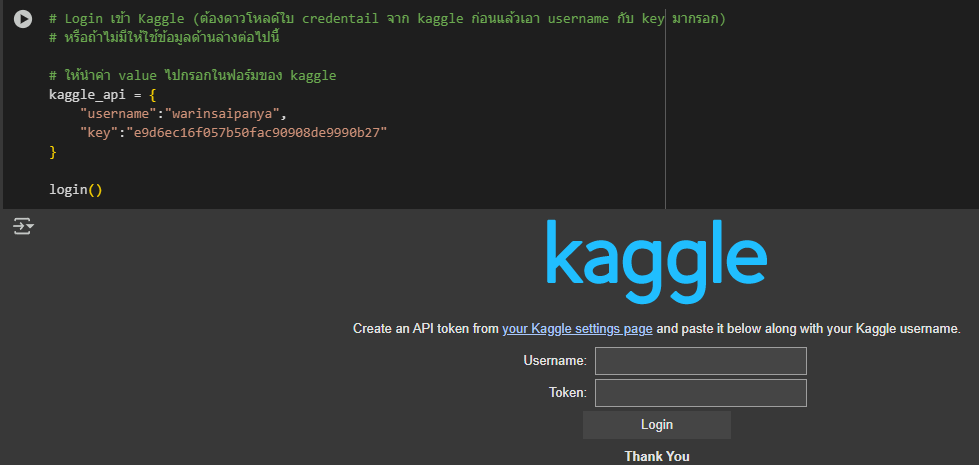
บังคับใช้ GPU



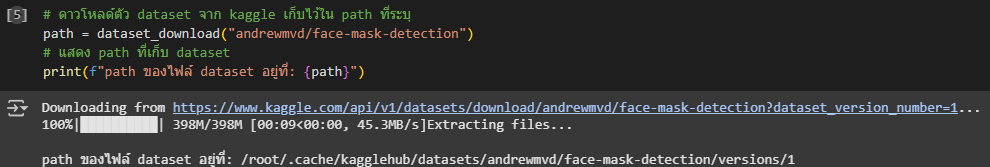
4. Login เข้า Kaggle (ต้องดาวโหลด์ใบ credentail จาก kaggle ก่อนแล้วเอา username กับ key มากรอก)

หรือถ้าไม่มีให้ใช้ข้อมูลด้านล่างต่อไปนี้

ให้นำค่า value ไปกรอกในฟอร์มของ kaggle



5ดาวโหลด์ตัว dataset จาก kaggle เก็บไว้ใน path ที่ระบุ และแสดง path ที่เก็บ dataset



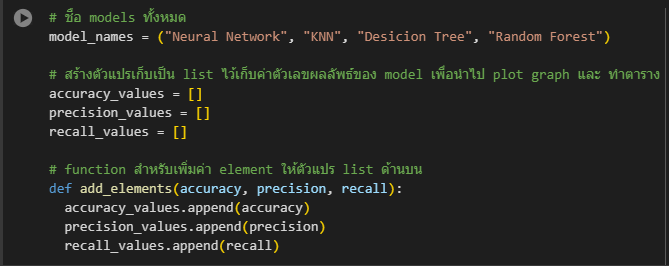
6.ตัวแปร classes เก็บเป็น tuple ไว้เก็บคำตอบ



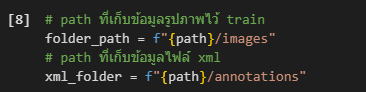
7. ชื่อ models ทั้งหมด

สร้างตัวแปรเก็บเป็น list ไว้เก็บค่าตัวเลขผลลัพธ์ของ model เพื่อนำไป plot graph และ ทำตาราง

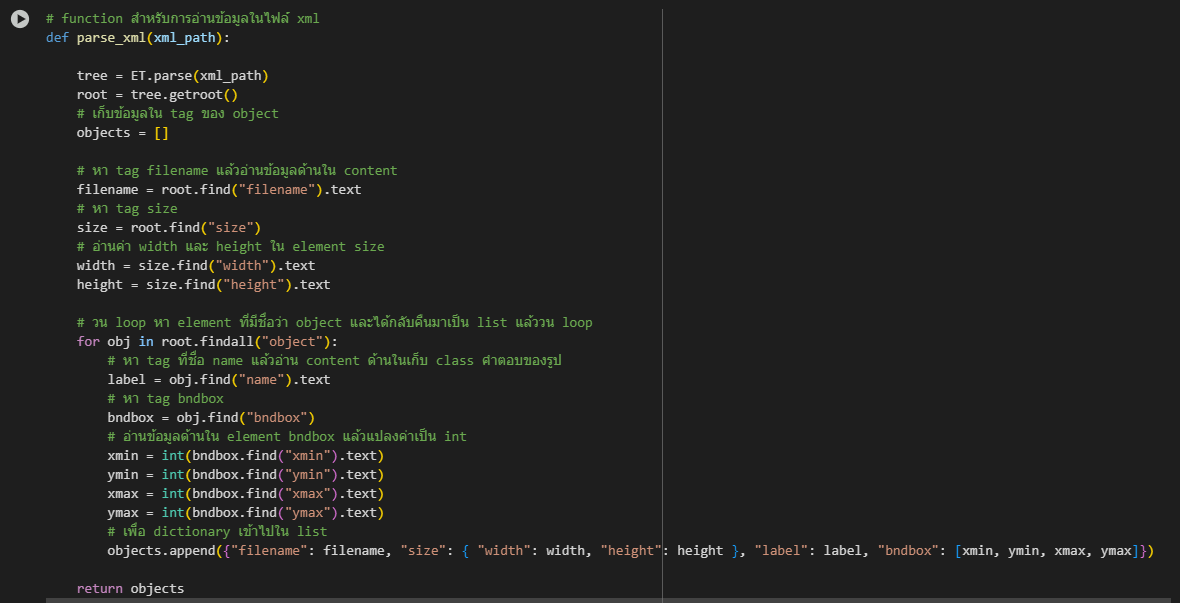
function สำหรับเพิ่มค่า element ให้ตัวแปร list ด้านบน



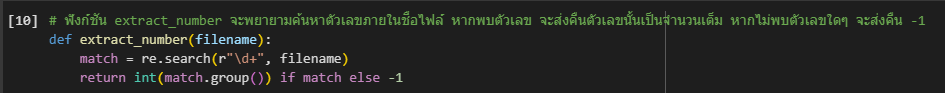
8.path ที่เก็บข้อมูลรูปภาพไว้ train



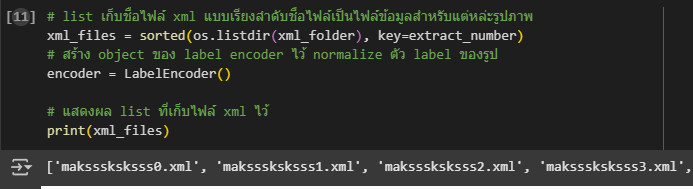
9. path ที่เก็บข้อมูลไฟล์ xml



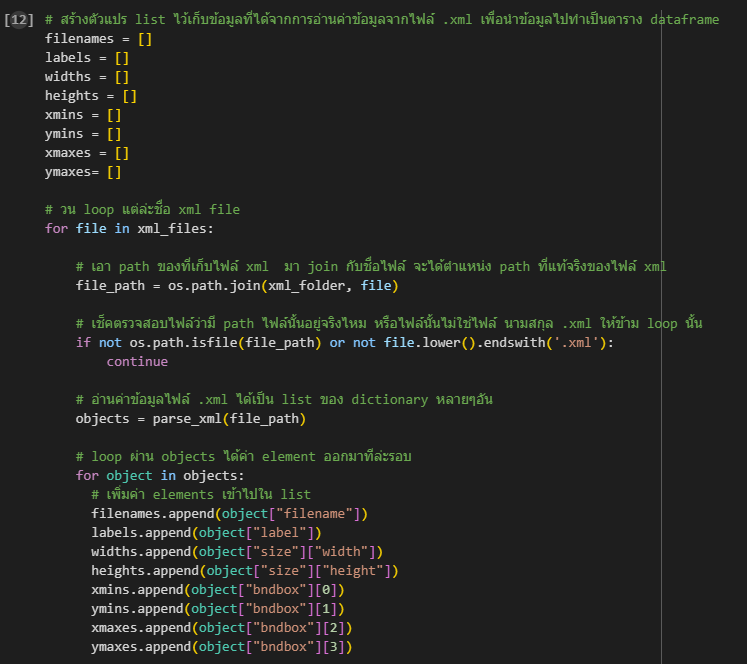
10. function สำหรับการอ่านข้อมูลในไฟล์ xml



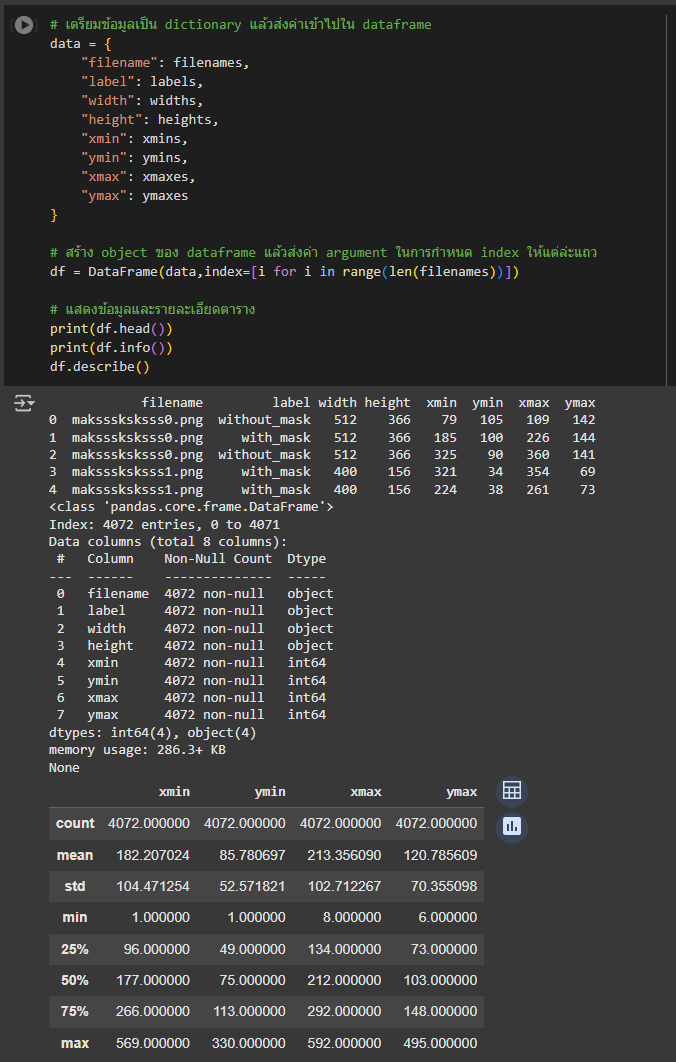
11. เก็บข้อมูลใน tag ของ object



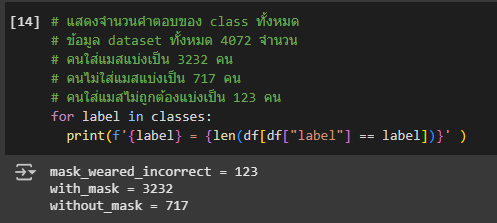
12. อ่านไฟล์ .xml ดึงข้อมูลที่ต้องการ แล้วเก็บไว้ใน list เพื่อนำไปสร้าง DataFrame โดยตรวจสอบไฟล์, แปลง XML เป็น dictionary, และเพิ่มค่าลงใน list ตามหมวดหมู่ เช่น ชื่อไฟล์, ป้ายกำกับ, ขนาดภาพ และพิกัดกรอบ



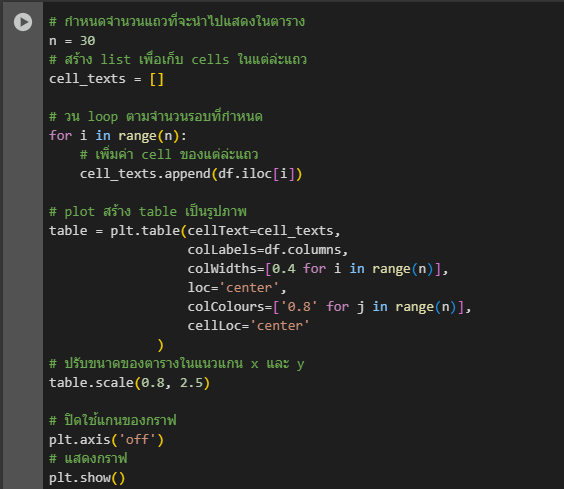
13. สร้าง DataFrame จาก dictionary ของข้อมูล แล้วแสดงตัวอย่าง head(), โครงสร้าง info(), และค่าสถิติ describe()



14. แสดงจำนวนคำตอบของ class ทั้งหมด

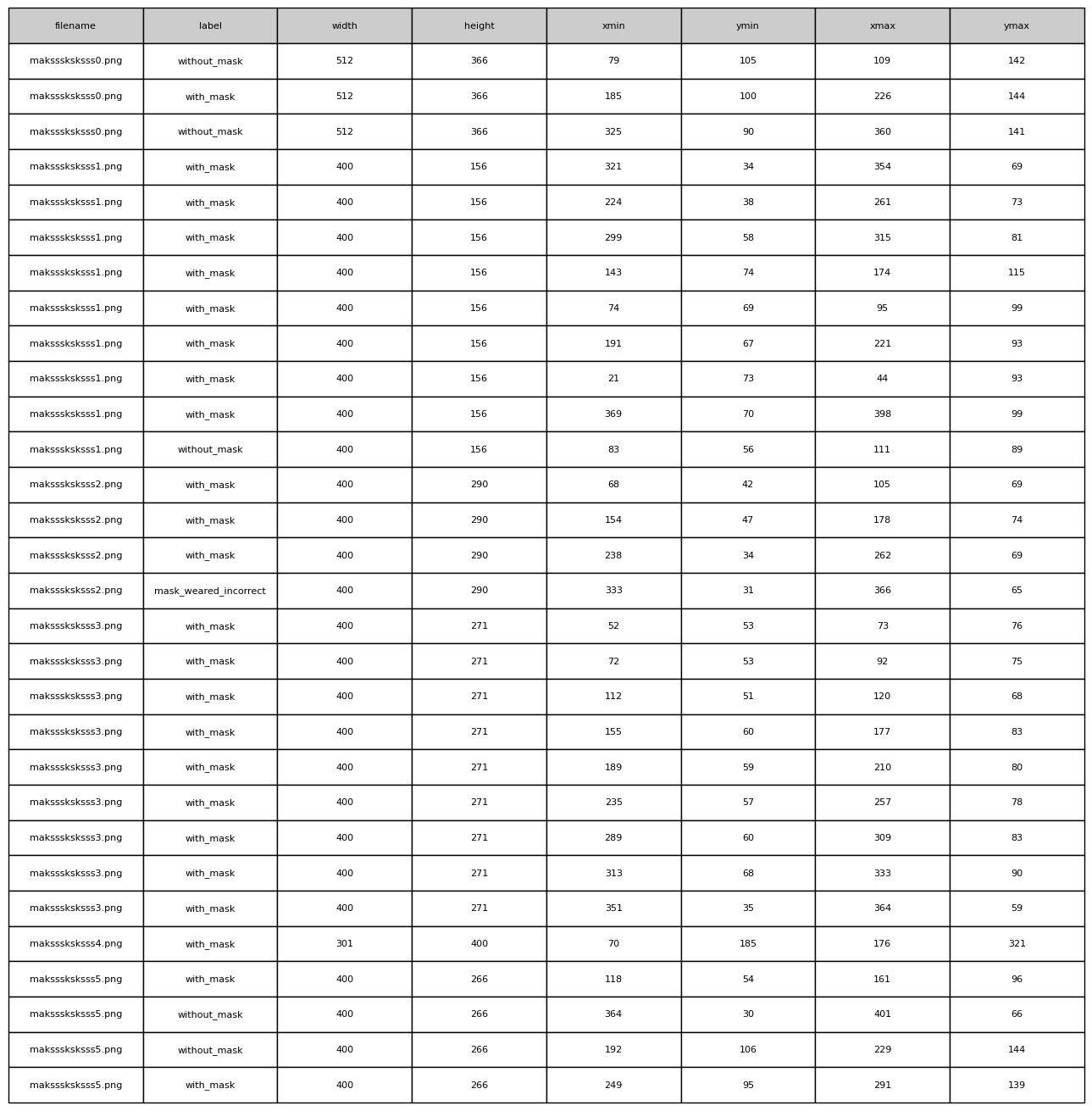


15. โค้ดนี้ใช้ Matplotlib สร้างตารางจาก DataFrame โดยดึง 30 แถวแรก แล้วใช้ plt.table() สร้างตาราง, ปรับขนาด scale(), ปิดแกน axis('off'), และแสดงผล show()

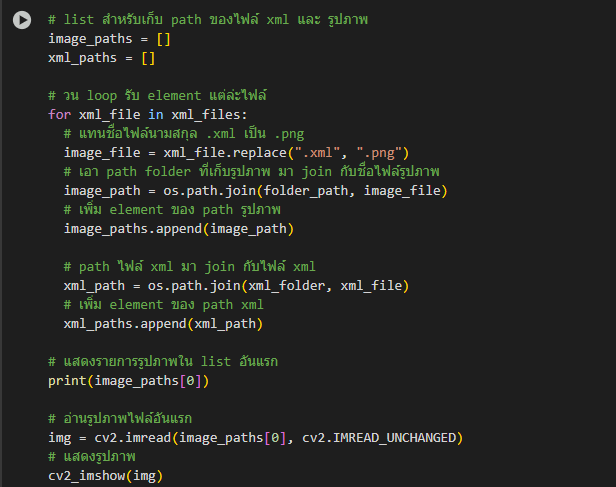


16. แสดงตารางรูปภาพโดยใช้ matplotlib.table ในการสร้างตาราง โดยแสดงข้อมูล 30 แถวได้แก้

ชื่อไฟล์รูปภาพ, คำตอบ (Label), ความกว้าง, ความสูง, xmin, ymin, xmax และ ymax

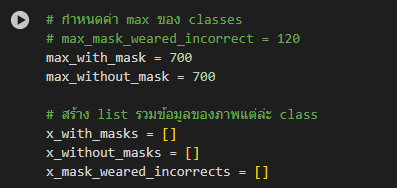


17. สร้างตัวแปรเก็บชื่อ path ของรูปภาพ และ path ของไฟล์ xml จากนั้นวน loop แต่ล่ะชื่อไฟล์ ทำการแปลงไฟล์นามสกุล .xml เป็น .png (ตัวอย่าง example.xml เป็น example.png) เพิ่ม path รูปภาพ และ เพิ่ม path ไฟล์ xml จากนั้นทำการแสดงรูปภาพอันแรกออกมาเป็นตัวอย่าง

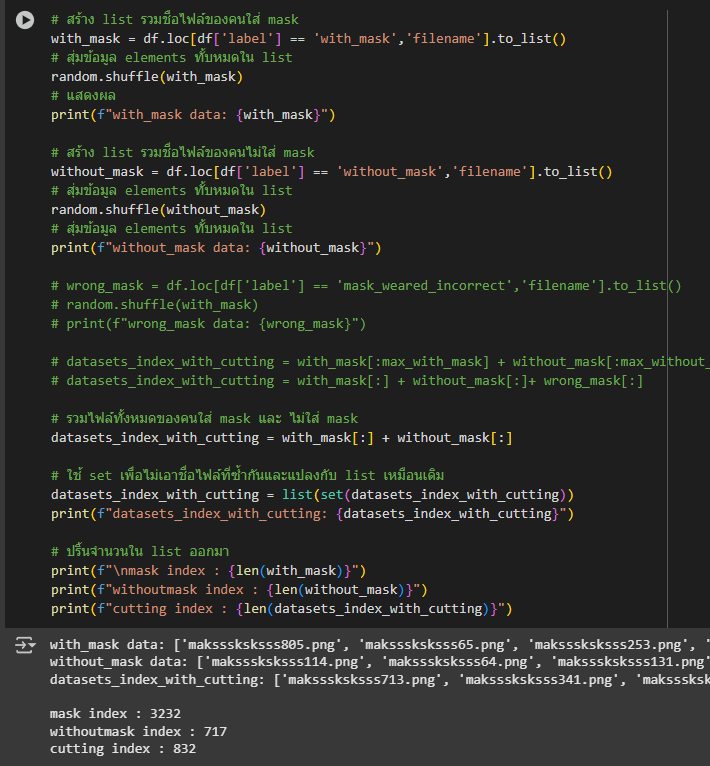




18. กำหนดตัวแปรเก็บค่าเลขสูงสุดสำหรับคนใส่ mask และคนไม่ใส่ mask 700 จำนวน และสร้างตัวแปร list สำหรับเก็บรูปภาพข้อมูลที่แบ่งเป็นข้อมูลแต่ล่ะ class



19. แยกชื่อไฟล์ของภาพตามประเภท with\_mask / without\_mask จาก DataFrame, สุ่มลำดับ shuffle(), รวมรายการโดยลบค่าซ้ำ set(), และแสดงจำนวนไฟล์แต่ละประเภท len()



20. สร้างรูปภาพโดยการแปลไฟล์ png เป็น xml แล้วเก็บไว้ใน ตัวแปล file สร้าง file\_path มาเก็บ ตำแหน่ง path ของไฟล์ xml โดยใช้ os.path.join แล้วจะทำการตรวจสอบว่า ไฟล์นั้นมีจริงหรือไม่ แล้วทำการแปลง xml เป็น png .ใหม่อีกรอบ แล้วเอา path ที่เก็บไว้มา Join อีกครั้ง เพื่อให้ได่ path รูปภาพที่แท้จริง

ถ้า path นั้นไม่มีอยู่ จะ return ค่ากลับมา

ใช้ตัวแปล objects ในการอ่านไฟล์ xml ทำการ ลุปหา label แล้วเช็ค เงื่อนไขว่าคนนี้ใส่แมสไหม หรือ ไม่ใส่ ถ้าเป็นจริงเก็บไว้ใน List

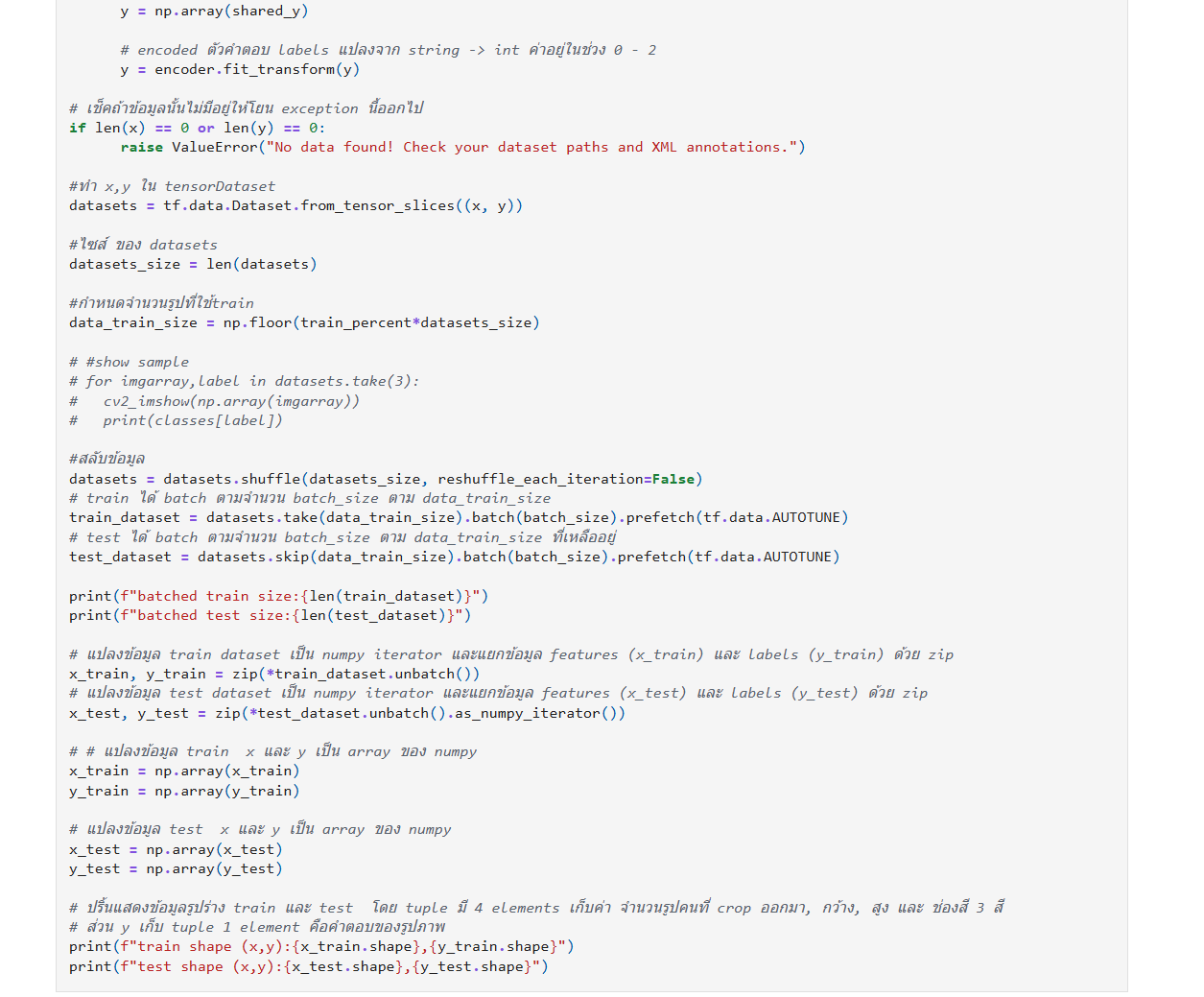


กำหนดการใช้งาน multiprocess เพื่อการสร้างdataที่เร็วขึ้น กำหนด manager ไว้เก็บตัวเเปรที่ใช้งานในหลายprocess

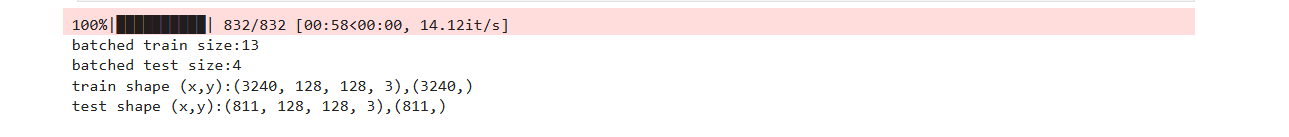
shared\_x,shared\_y เเละ เรียกใช้ฟังก์ชัน fetch\_dataset เเบบ parallel



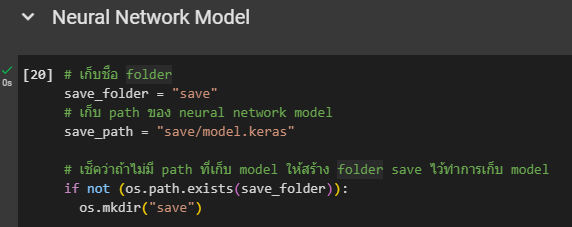
ตัวเเปร x,y เก็บค่า shared\_x,shared\_y เป็น numpy array ทำการ fit\_transform y ให้อยู่ในช่วง 0-2 เเละสร้าง Tensorflow Dataset Pipline จาก x,y , shuffle Dataset เเละเเยก train\_dataset,test\_dataset ที่กำหนดจากbatch size ,จากข้อมูล train\_dataset เเยก dataset ออกมา เป็น x\_train , y\_train เเละ test\_dataset เเยก dataset ออกมา เป็น x\_test, y\_test



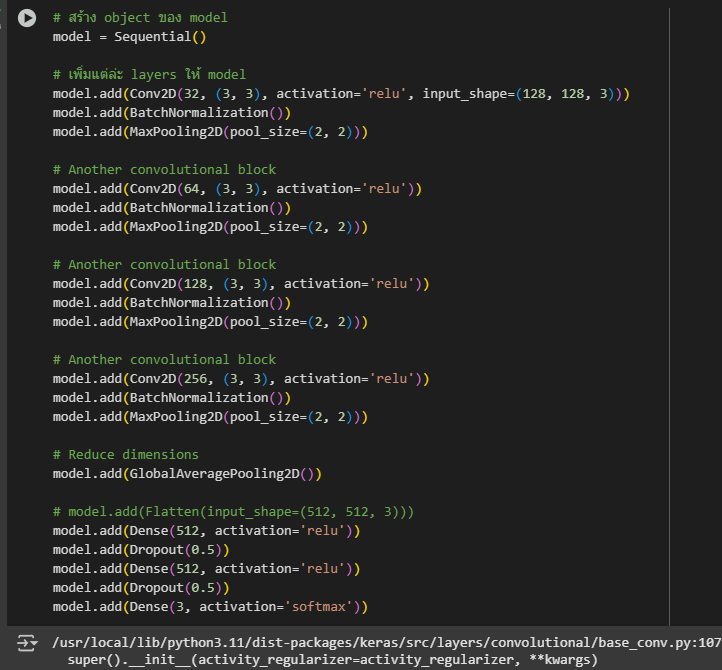
รูปเเสดงการโหลดข้อมูล สร้าง dataset



21. กำหนดชื่อ folder ชื่อ save สำหรับเก็บตัว model ที่ training เสร็จ แล้วเขียนเงื่อนไขเช็คว่าถ้าไม่มี path folder save ให้ทำการสร้างโฟลเดอร์ตัวนั้น



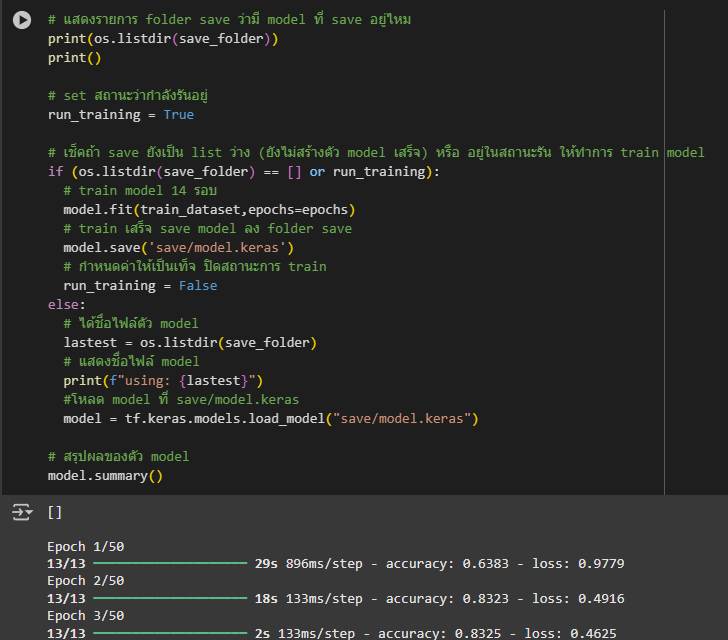
22. สร้าง Sequential Model ที่มีConvolution Layer 4 layers เเละ Dense 512 X 512 เเละ OutputLayer = 3

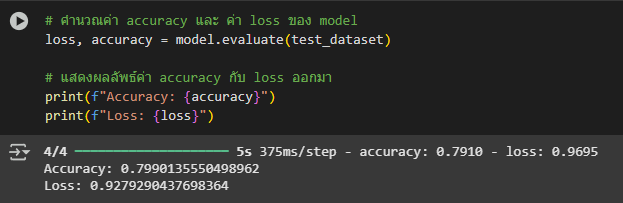


23. compile โมเดลที่สร้างมากำหนดค่า arguments ใน method ตามรูปภาพ

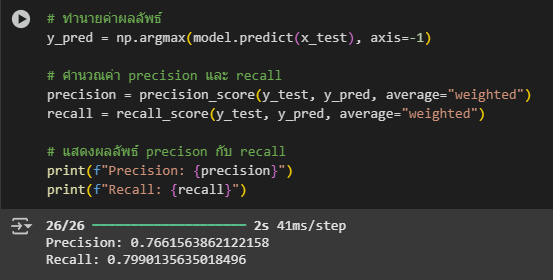


24. แสดง folder ว่ามีไฟล์โมเดลไหม model.keras จากนั้นกำหนดค่า สถานะ training เป็น True เช็คเงื่อนไขว่าถ้า folder เป็น folder ว่างไม่เก็บตัว model.keras ที่สร้างมา หรือ ค่าสถานะนั้น traing เป็น True ให้ทำตามเงื่อยไข statements ด้านในของ if คือให้ model ทำการ traing ผึกสนตัว model ตามจำนวน epochs ที่ 50 รอบ หลังจาก train เสร็จให้ save ตัว model ลง folder save แล้วเปลี่ยนค่าสถานะ traing เป็น False เพื่อหยุดการ train ถ้าเงื่อนไข if เป็นเท็จจะทำเงื่อนไข else คือให้โหลดตัว model.keras ที่เราทำการบันทึกไว้ใน folder save หลังจากทำเงื่อนไข if หรือ else เสร็จให้เรียกใช้ model.summary() แสดงผลสรุปตัว model Neural Network

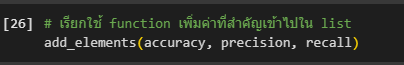


25. แสดงค่าผลลัพธ์ของ accuracy และ ค่า loss ของตัว model

26. ให้ model ทำการทำนายผลลัพธ์คนใส่ mask หรือ คนไม่ใส่ mask แล้วเก็บลงตัวแปร y\_pred นำ y\_pred ส่งเข้าไปใน function precsion และ recall เพื่อคำนวณค่าดังกล่าวแล้วแสดงผลออกมา

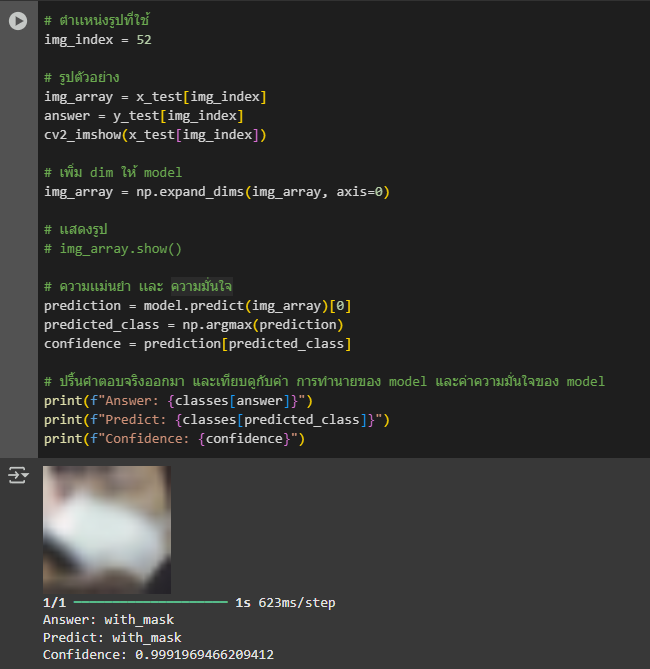


27. เพิ่มค่า accuaracy, precision และ recall เก็บไว้ใน list

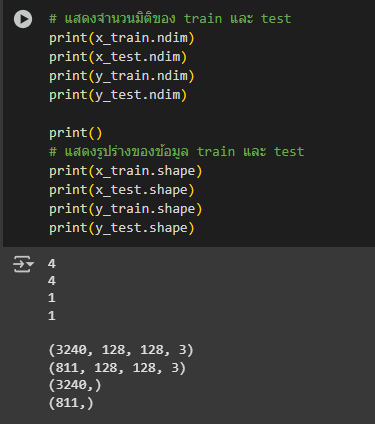


28. ทำการ predict test ตัว model Neural Network โดยเลือก index 52 ของรูปภาพ test ทำการอ่านรูปภาพและคำตอบของภาพ จากนั้นทำการ predict รูปภาพ และ ค่าความมั่นใจของ model จากนั้นแสดงผลปริ้น

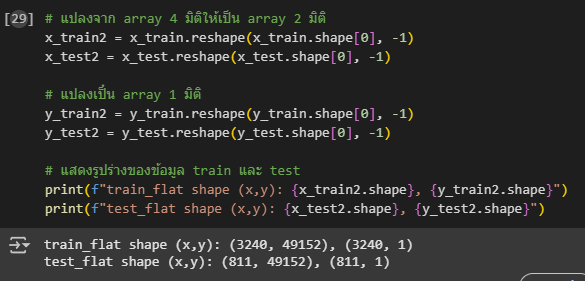
ออกมา



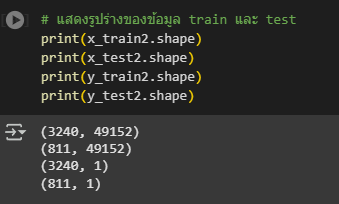
29. แสดงจำนวนมิติของข้อมูล train และ test และ แสดงรูปร่างของข้อมูล train และ test



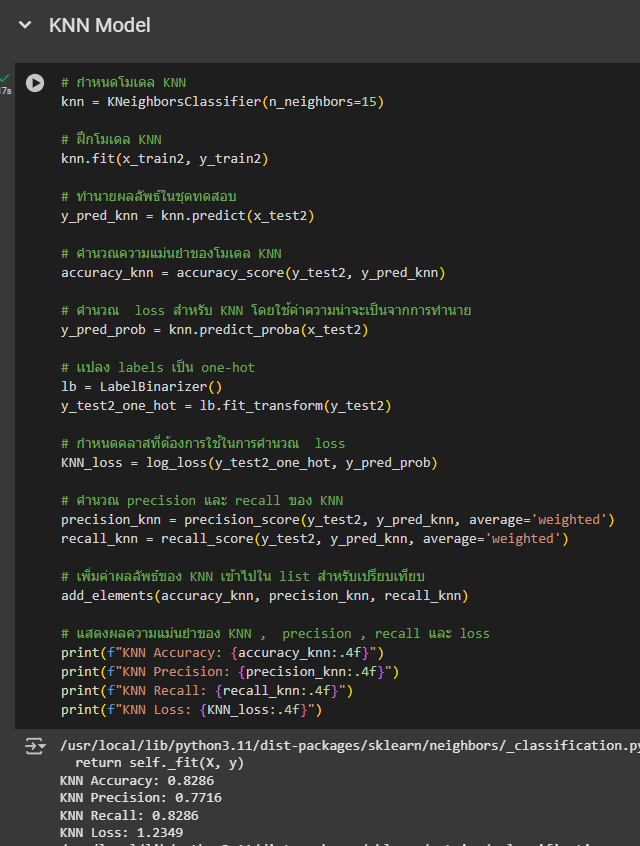
30. ทำการแปลงรูปร่างข้อมูลอันเก่าของ Neural Network จาก 4 มิติเป็น 2มิติ และคำตอบของรูปภาพเป็น 1มิติ เพื่อใช้สำหรับการ train model Machine Learning จากนั้นแสดงรูปร่างของข้อมูล train และ test อันใหม่



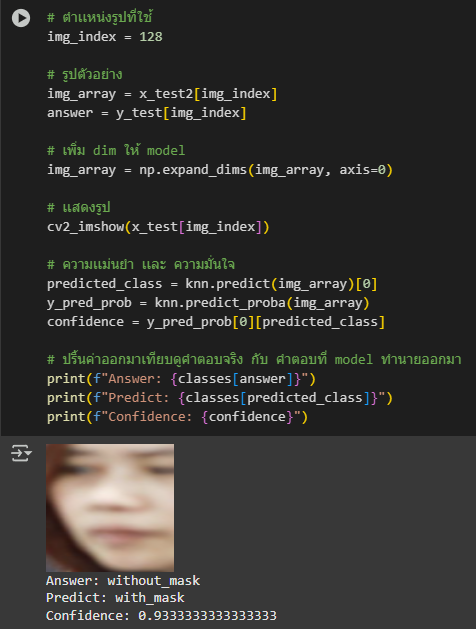
31. แสดงรูปร่างข้อมูล train และ test อันใหม่



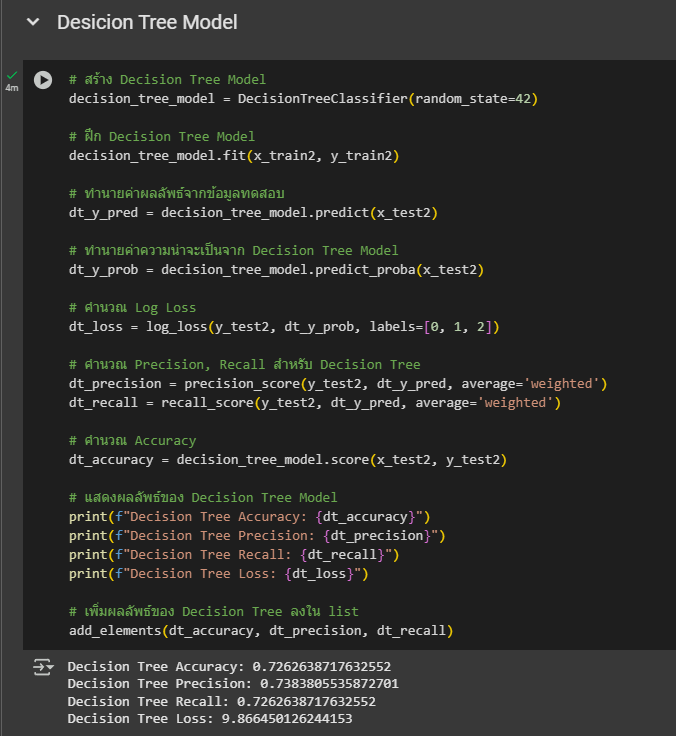
32. สร้างและประเมินโมเดล KNN โดยใช้ 15 neighbors ในการจำแนกประเภทข้อมูล และคำนวณ Accuracy, Precision, Recall และ Loss เพื่อวัดประสิทธิภาพของโมเดล



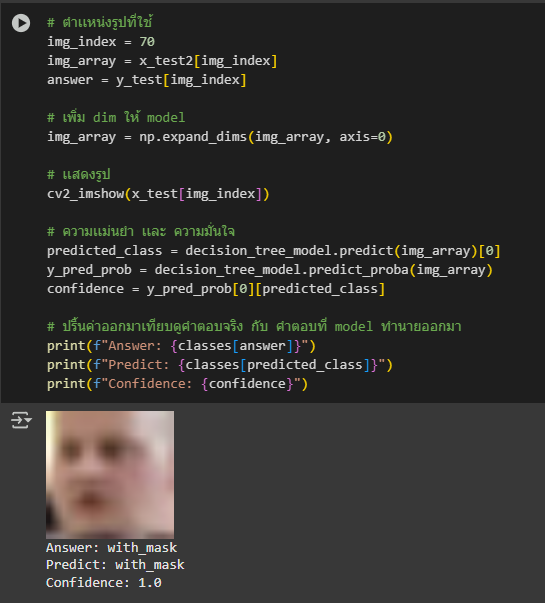
33. ใช้ โมเดล KNN ทำนายคลาสของภาพตัวอย่าง จากชุดทดสอบ พร้อมแสดงผลเปรียบเทียบระหว่าง ค่าจริงกับค่าที่โมเดลทำนาย รวมถึงระดับความมั่นใจในการพยากรณ์



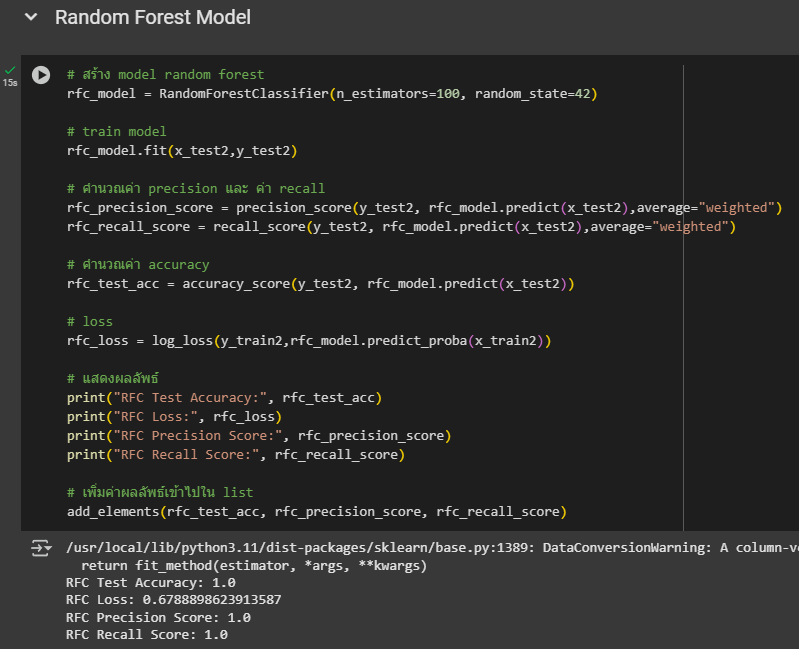
34. โค้ดนี้ใช้ Decision Tree Classifier เพื่อสร้างโมเดลสำหรับการจำแนกประเภทข้อมูล จากนั้นทำการ ฝึกโมเดล, ทำนายผลลัพธ์, และวัดประสิทธิภาพของโมเดล โดยใช้ตัวชี้วัดหลายค่า เช่น Accuracy, Precision, Recall และ Log Loss ซึ่งช่วยให้สามารถเปรียบเทียบโมเดลนี้กับโมเดลอื่นได้อย่างชัดเจน



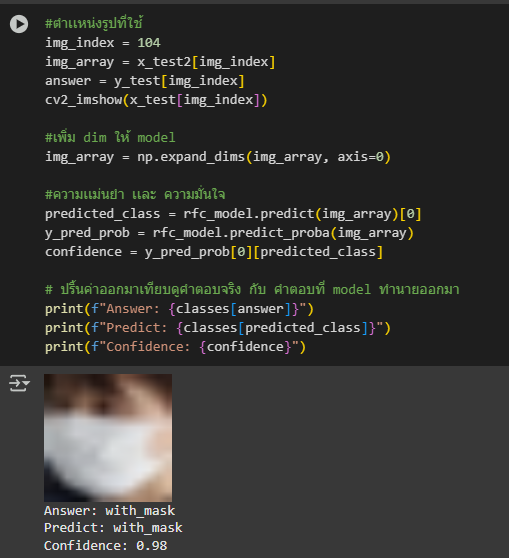
35. โค้ดนี้ใช้ Decision Tree Model เพื่อทำนายคลาสของรูปภาพจากชุดข้อมูลทดสอบ (x\_test2) โดยเลือกภาพที่มี ดัชนี (index) เท่ากับ 70 จากนั้นทำการแสดงภาพ ทำนายคลาส และคำนวณค่าความมั่นใจของโมเดล ก่อนพิมพ์ผลลัพธ์ออกมาเพื่อเปรียบเทียบกับค่าจริง



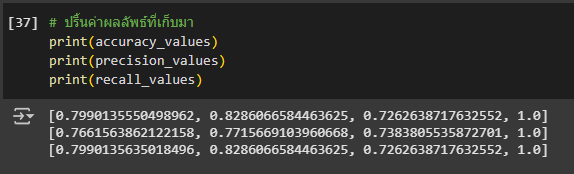
36.ทำการสร้าง model random forest คำนวณค่า precision และ ค่า recall คำนวณค่า accuracy เพิ่มค่าผลลัพธ์เข้าไปใน list และแสดงผลลัพธ์



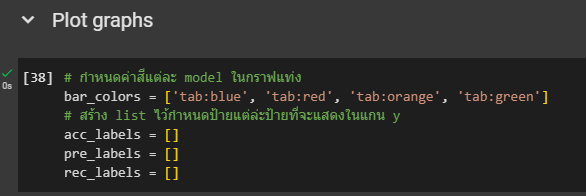
37. โค้ดนี้ใช้ Random Forest Model เพื่อทำนายคลาสของรูปภาพจากชุดข้อมูลทดสอบ (x\_test2) โดยเลือกภาพที่มี ดัชนี (index) เท่ากับ 104 จากนั้นทำการแสดงภาพ ทำนายคลาส และคำนวณค่าความมั่นใจของโมเดล ก่อนพิมพ์ผลลัพธ์ออกมาเพื่อเปรียบเทียบกับค่าจริง



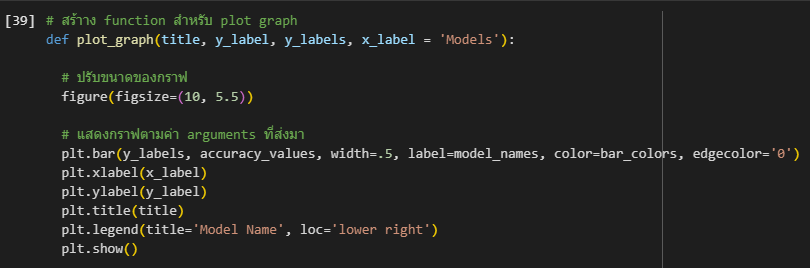
38. ปริ้นค่าผลลัพท์ที่เก็บมาจากทั้ง 4 model



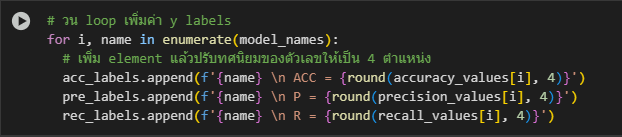
39. โค้ดนี้กำหนดค่าและเตรียมข้อมูลสำหรับการสร้างกราฟแท่ง (bar chart) ที่จะใช้แสดงผลลัพธ์ของ ค่าสถิติ ต่างๆ ที่ได้จากการทดสอบโมเดลหลายๆ ตัว (เช่น ความแม่นยำ (accuracy), Precision, Recall) โดยมีการใช้ สีที่กำหนดไว้เพื่อแยกแต่ละโมเดลในกราฟให้ชัดเจน



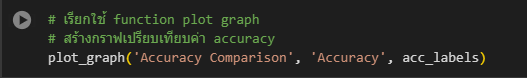
40. ฟังก์ชัน plot\_graph สร้างกราฟแท่งโดยรับพารามิเตอร์ชื่อกราฟ, ป้ายแกน y, ค่าของแกน y, และป้ายแกน x (ค่าเริ่มต้นเป็น 'Models'). ใช้ plt.bar() เพื่อสร้างกราฟ, ปรับขนาดกราฟ, แสดงชื่อแกน, ชื่อกราฟ, และคำอธิบายของกราฟ ก่อนแสดงผลด้วย plt.show()

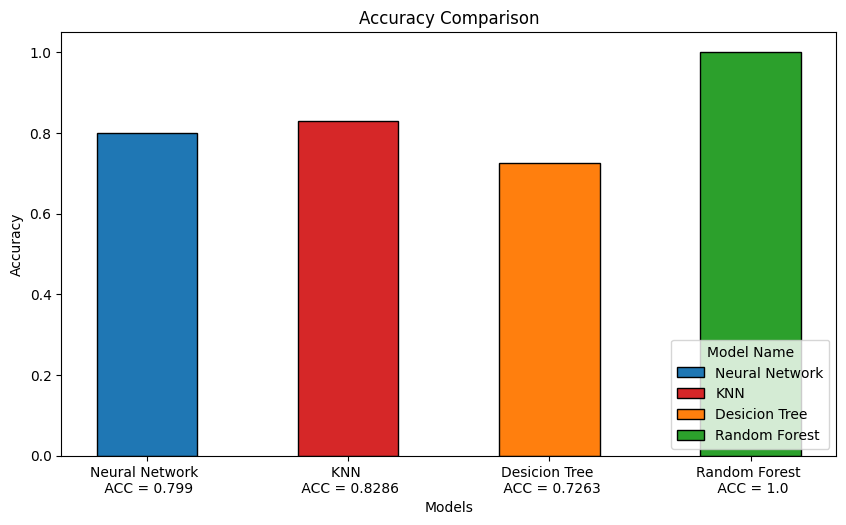


41.โค้ดนี้จะเพิ่มชื่อโมเดลและค่าประสิทธิภาพ (accuracy, precision, recall) ลงใน acc\_labels, pre\_labels, และ rec\_labels แล้วปัดค่าทศนิยม 4 ตำแหน่งสำหรับแต่ละค่า



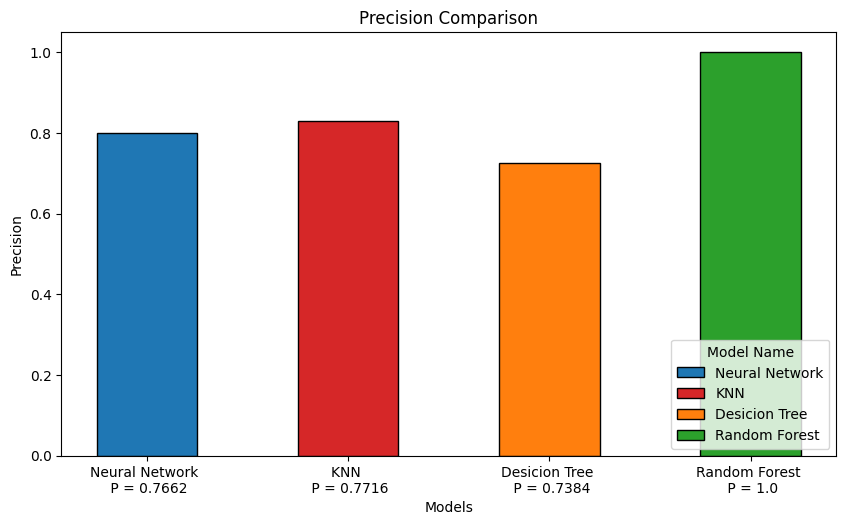
42.โค้ดนี้เป็นการสร้างกราฟเพื่อเปรียบเทียบค่า accuracy ของโมเดลต่าง ๆ โดยที่ชื่อกราฟคือ "Accuracy Comparison" และชื่อของแกน Y คือ "Accuracy" ซึ่งข้อมูลที่ใช้แสดงในกราฟจะมาจากตัวแปร acc\_labels ที่ถูกส่งผ่านไปในฟังก์ชัน plot\_graph()





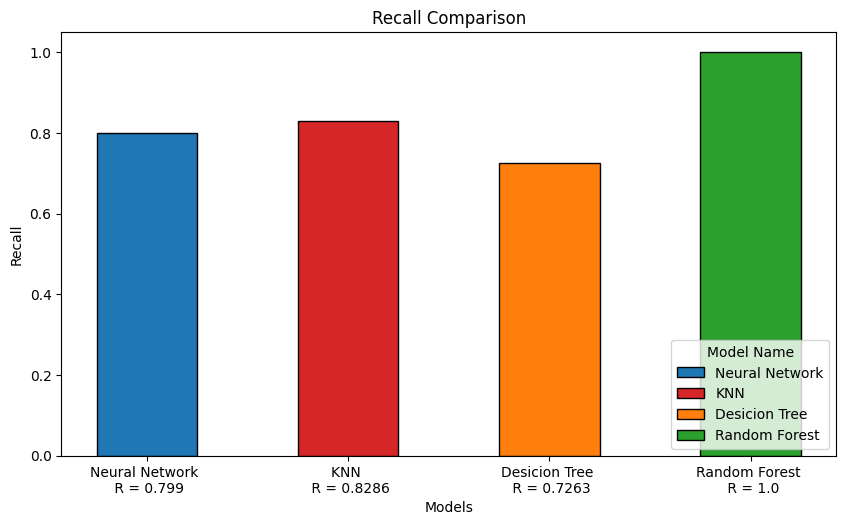
43.โค้ดนี้จะสร้างกราฟเปรียบเทียบค่า precision ของโมเดลต่าง ๆ โดยที่ชื่อกราฟคือ Precision Comparison และชื่อของแกน Y คือ Precision โดยใช้ข้อมูลจากตัวแปร pre\_labels ซึ่งเป็นค่าของ precision ที่ได้จากการประเมินผลของโมเดลต่าง ๆ





44.โค้ดนี้จะสร้างกราฟเปรียบเทียบค่า recall ของโมเดลต่าง ๆ โดยที่ชื่อกราฟคือ Recall Comparison และชื่อของแกน Y คือ Recall โดยใช้ข้อมูลจากตัวแปร rec\_labels ซึ่เป็นค่าของ recall ที่ได้จากการประเมินผลของโมเดลต่าง ๆ





**หน้าที่ความรับผิดชอบของสมาชิกแต่ละคน**

นาย ปัณณวัฒน์ นิ่งเจริญ รหัสนิสิต 6630250231(ทำ Neural Network ทำรายงาน)

นาย พันธุ์ธัช สุวรรณวัฒนะ รหัสนิสิต 6630250281 (ทำ Decision Tree Model และทำเอกสาร)

นาย วรินทร์ สายปัญญา รหัสนิสิต 6630250435 (เขียนตารางรูปภาพ, ทำ DataFrame, เขียนตารางวาดกราฟแท่งเปรียบเทียบแต่ล่ะ models, เขียน comments โดยรวมของโค้ด, เตรียมข้อมูล datasets จาก kagglehub, ช่วยจัดทำรายงานเอกสาร)

นางสาว อัมพุชินี บุญรักษ์ รหัสนิสิต 6630250532 (เขียนKNN Model,Desicion Tree Model และทำเอกสาร)

นาย ปุณณภพ มีฤทธิ์ รหัสนิสิต 6630250591 (แก้โค้ด Neural Network,สร้างDataset,random forest classifier )

**ลิ้งค์ Google Colab:**

<https://colab.research.google.com/drive/16cFUCI3QIDHvrgYe-p5nSR_Q2eJsXIKF?usp=sharing>