บทที่ 1

บทนำ

ที่มาและความสำคัญ

กล้องจุลทรรศน์เป็นหนึ่งในเครื่องมือที่ใช้แพร่รายในหลากหลายวิชาสาขา เพื่อช่วยในการวิเคราห์และจำแนกประเภทของ Sample แต่หนึ่งในปัญหาหลักที่คณะผู้วิจัยพบคือ ในการใช้กล้องจุลทรรศน์อาจใช้เวลานาน ทั้งยังอาจทำให้เกิดความผิดพลาดในการตรวจจับของมนุษย์เอง คณะผู้วิจัยจึงเล็งเห็นว่าการนำ Ai ที่มีขีดความสามารถในการทำ Image Classification และ Real-Time Detection เข้ามาช่วยในการแก้ไขปัญหา

แต่หนึ่งในปัญหาที่คณะผู้วิจัยพบเจอคือ ปัญหาในการเลือกใช้ อัลกอรึทึม ในการทำ Image Classification และ Real-Time Detection นั้นไม่ว่าจะเป็น machine learning อัลกอรึทึม (ML) , deep learning methods (DL) และอื่นๆ ที่จะทำให้บรรลุปัญหาที่เกิดขึ้น

โดย machine learning ที่คณะผู้วิจัยสนใจในการทดสอบประกอบไปด้วย Support Vector Machines (SVMs) โดยใช้ร่วมกับ Kernel ต่างๆประกอบไปด้วย Linear, Radial Basis Function (RBF) และ Polynomial (Poly), Naive Bayes โดยใช้ Event Model ที่ประกอบไปด้วย Gaussian naive Baye, Multinomial naive Bayes และ Bernoulli naive Bayes , Decision tree, Random Forest, Ada Boosting, Gradient Boosting, Hist Gradient Boosting, eXtreme Gradient Boosting, light gradient-boosting และ Cat boost

อีกหนึ่งในปัญหาที่พบคือ จำนวนข้อมูลที่มีจำกัดและมีปริมาณน้อย เนื่องจากในการฝึกฝนและพัฒนา ML จำเป็นต้องใช้ข้อมูลจำนวนมาก การนำเทคนิคที่เรียกว่าการทำ Data augmentation เพื่อช่วยให้ปริมาณข้อมูลมากขึ้นโดยการทำเทคนิคดั้งกล่าวโดยใช้ การทำแบบ Geometric และ Color space transformations

ในการทำโครงงานนี้จึงเป็นการทำ Data Preparation ในการจัดเตรียมข้อมูลเพื่อใช้ในการฝึกฝน อัลกอรึทึมแต่ล่ะตัว ทำ Training Model ทำการฝึกฝน และ เมื่อฝึกฝนสำเร็จจึงทำ การ Evaluate เพื่อเพิ่มประสิทธิภาพของ อัลกอรึทึม โดยการทำ Hyperparameter tunning โดยใช้ Randomize Search เพื่อหา parameter ที่ดีทีสุดสำหรับ อัลกอรึทึมแต่ล่ะตัว จากนั้นจึงทำการเปรียบเทียบกับ ML อัลกอรึทึมทุกตัว เพื่อค้นหาอัลกอรึทึมประสิทธิภาพมากที่สุดในการทำ Image Classification เพื่อช่วยร่วมกับ อัลกอรึทึม Real-Time Detection และใช้งานร่วมกับการตรวจจับวัตถุใต้กล้องจุลทรรศน์

วัตถุประสงค์

1. เพื่อศึกษาและเปรียบเทียบประสิทธิภาพของ Machine learning อัลกอรึทึม ความสามารถในการ Image Classification ในการตรวจจับวัตถุใต้กล้องจุลทรรศน์
2. เพื่อศึกษาวิธีการ และ การทำ Data augmentation ในการเพิ่มข้อมูลที่มีอยู่อย่างจำกัด

ขอบเขตของการศึกษาค้นคว้า

1. ขอบเขตด้านเนื้อหา

1.1 โดยมีการนำ Model Classifier อัลกอรึทึม อย่าง Support Vector Machines (SVMs) โดยใช้ร่วมกับ Kernel ต่างๆประกอบไปด้วย Linear, Radial Basis Function (RBF) และ Polynomial (Poly), Naive Bayes โดยใช้ Event Model ที่ประกอบไปด้วย Gaussian naive Baye, Multinomial naive Bayes และ Bernoulli naive Bayes, Decision tree, Random Forest, Ada Boosting, Gradient Boosting, Hist Gradient Boosting, eXtreme Gradient Boosting, light gradient-boosting และ Cat boost

1.2 Dataset อย่าง BCCD Dataset ที่เป็น Dataset ของ เซลล์เม็ดเลือดใต้กล้องจุลทรรศน์ ของ โดยที่นำมาใช้ในการทำ Image Classification ประกอบไปด้วย Neutrophil, Eosinophil, Lymphocyte และ Monocyte

1.3 การทำ Data augmentation ที่ประกอบไปด้วย การทำ Geometric transformations และ Color space transformations

2. ขอบเขตด้านเวลา

2.1 เดือนมิถุนายน พ.ศ. 2565 - เดือนมกราคม พ.ศ. 2566

สมมุติฐาน

หลังจากการทำทดสอบประสิทธิภาพของ อัลกอรึทึม Classifier รูปแบบต่างๆแล้วจะสามารถหา Model ที่มีประสิทธิภาพสูงสุดในการตรวจจับ รูปแบบ ลักษณะของเซลล์ แบ่งแยกและจัดหมวดหมู่ของเซลล์ ได้ดีที่สุดได้

ตัวแปรที่ศึกษา

ตัวแปรต้น อัลกอรึทึม Classification

ตัวแปรตาม ประสิทธิภาพของ อัลกอรึทึม Classification ในการตรวจจับ รูปแบบ ลักษณะของเซลล์ แบ่งแยกและจัดหมวดหมู่ของเซลล์

ตัวแปรควบคุม Dataset ของข้อมูล, Software Environments

นิยามศัพท์เฉพาะ

ตัวอย่าง, เซลล์ตัวอย่าง หรือ เนื้อเยื่อตัวอย่าง คือ Data set ของข้อมูล

ML คือ Machine learning การเรียนรู้ของโปรแกรมด้วยตัวเอง

SVMs คือ Support vector machine

ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถทำ Image Classification กับ Dataset ของ เซล์เม็ดเลือดขาว

2. ค้นพบ อัลกอรึทึม ที่มีความสามารถ ประสิทธิภาพ และความถูกต้องในการทำ Image Classification ใน Data set ตัวนี้

3. สามารถนำข้อมูลที่ได้มาพัฒนาต่อโดยมีการ ควบคุม Software environments ผ่านคอมพิวเตอร์ส่วนตัว บน Google colabs

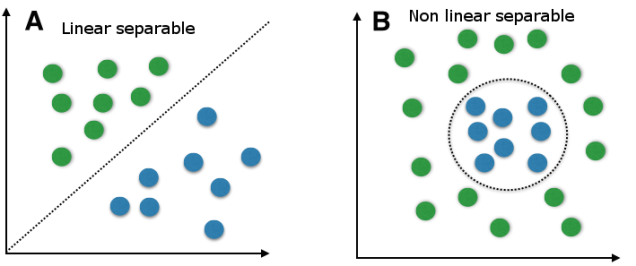
บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

เนื่องจากในงานวิจัยนี้มีการใช้งาน อัลกอรึทึม Classification ที่มีชื่อว่า Support Vector Machine

Support Vector Machine (SVM)

 Support Vector Machines (SVMs) เป็น Algorithm แบบ Supervised Learning ที่ใช้สำหรับแก้ปัญหาการจัดกลุ่มข้อมูล Classification และการวิเคราะห์การถดถอย Regression ซึ่งจะมีความคล้ายคลึงกับ Logistic Regression (LR)

Support Vector Machine จะเป็นการจัดกลุ่มข้อมูล Classification โดยการแบ่ง Class ของข้อมูลออกจากกัน ซึ่งสามารถใช้การแบ่งด้วยสมการเชิงเส้นได้ทั้ง Linear และ Non-Linear

การจัดกลุ่มข้อมูล Classification ไม่ใช่เรื่องยาก สามารถแบ่งข้อมูลได้หลาย Model โดยใช้สมการเส้นตรง Linear แต่ปัญหาคือแล้ว Model ไหนที่ดีที่สุด ซึ่ง Support Vector Machine จะใช้ในการหา Hyperplane ซึ่งก็คือ Model ที่ดีที่สุด จากระยะห่างสูงสุดของแต่ละ Class ข้อมูล Maximum Margin ในแต่ละมิติ N – Dimension โดย N เป็นจำนวนของ Feature

โดยต่อไป SVMs จะประกอบไปด้วยการทำ Kernel Functions

SVMs Kernel Functions คือสมการทางคณิตศาสตร์มาช่วยในการจำแนกข้อมูล จัดการข้อมูล โดย การนำข้อมูลเข้าไป Kernel Functions ส่งคืนผลคูณสเกลาร์ระหว่างสองจุดในพื้นที่

Linear Kernel

สมการของ Linear Kernel

Polynomial Kernel

สมการของ Polynomial Kernel

Gaussian Radial Basis Function (RBF)

โดยที่

สมการของ Gaussian Radial Basis Function

Sigmoid Kernel

สมการของ Sigmoid Kernel

https://data-flair.training/blogs/svm-kernel-functions

Native Bayes https://en.wikipedia.org/wiki/Naive\_Bayes\_classifier

Naive Bayes Classification เป็นการจัดหมวดหมู่โดยใช้หลักความน่าจะเป็นเข้ามาช่วยคำนวณ

สมการรูปทั่วไปของ Native Bayes

โดยที่

เมื่อ C คือ Class, x คือ Attribute และ P คือ Probability (ความน่าจะเป็น) โดย Naïve Bayes ประกอบไปด้วย Event model ทีมีการใช้ร่วมกับสมการคณิตศาสตร์ประกอบไปด้วย

Gaussian naive Bayes

สมการของ Gaussian naive Bayes

Multinomial naive Bayes

สมการของ Multinomial naive Bayes

*และ* Bernoulli naive Bayes

สมการของ Bernoulli naive Bayes

Decision Tree Classification www.borntodev.com/2022/09/15/รู้จักกับ-decision-tree/

Decision Tree เป็น model แบบ rule-based คือ สร้างกฎ if-else จากค่าของแต่ละ feature โดยไม่มีสมการมากำกับความสัมพันธ์ระหว่าง feature & target สิ่งที่สำคัญในการสร้าง Decision Tree คือ การเลือก split ค่า feature แต่ละครั้ง จะต้อง minimize ค่าของ cost function ให้น้อยที่สุด (regression — mse, classification- impurity, entropy)

Classification Tree คือ Decision Tree ที่ใช้สำหรับการทำ Classification โดยจะใช้ Gini Impurity หรือ Entropy เป็น Objective Function ในการหาจุดที่ดีที่สุดในการแบ่งข้อมูล (Split point)

Gini Impurity คือ การวัดค่า Impurity หรือ ค่าความไม่บริสุทธิ์ในการอธิบาย Target ของกลุ่มที่ถูกแบ่งออกมาจากตัวแปร นั่นหมายความว่าถ้าค่า Impurity ยิ่งต่ำก็ยิ่งแบ่งข้อมูลออกมาได้ดีนั่นเอง

การคำนวณ Gini Impurity คือ การนำเอาผลรวมของค่าความน่าจะเป็นของเหตุการณ์ที่เราสนใจมาคูณด้วย (1 ลบ ค่าความน่าจะเป็นของเหตุการณ์ที่เราสนใจ)

หลังจากได้ค่า Gini Impurity ของแต่ละกลุ่มในทุก ๆ ตัวแปรแล้ว จะทำการหาค่า Weighted Gini Impurity เพื่อเลือกตัวแปรที่มีค่า Weighted Gini Impurity ต่ำที่สุดมาใช้ในการตัดสินใจก่อน เพราะสามารถ Split ข้อมูลได้ดีที่สุด

การคำนวณ Weighted Gini Impurity คือ การนำเอาผลรวมของค่า Gini ในเหตุการณ์ที่เราสนใจ คูณกับจำนวนข้อมูลของ Class ที่ i ในตัวแปรที่เราสนใจ และหารด้วยจำนวนข้อมูลทั้งหมดในตัวแปรที่เราสนใจ

การคำนวณ Weighted Gini Impurity คือ การนำเอาผลรวมของค่า Gini ในเหตุการณ์ที่เราสนใจ คูณกับจำนวนข้อมูลของ Class ที่ i ในตัวแปรที่เราสนใจ และหารด้วยจำนวนข้อมูลทั้งหมดในตัวแปรที่เราสนใจ

Random Forest Classification medium.com/@pradyasin/random-forest-คืออะไร-74d2a0af3d7

เป็น อัลกอรึทึม ประเภทหนึ่งของ Machine Learning ถูกพัฒนาขึ้นจาก Decision Tree ต่างกันที่ Random Forest เป็นการเพิ่มจำนวน Tree เป็น Tree หลายๆ ต้น ทำให้ประสิทธิภาพในการทำงานสูงขึ้น แม่นยำมากขึ้น ซึ่งโมเดล Random Forest เป็นโมเดลที่ได้รับความนิยมไปอย่างมากในการใช้ Machine Learning โดยมีลักษณะคล้ายกับการ Bagging

Bagging จะมีการแบ่งข้อมูลออกเป็น Tree หลายๆ ต้นแต่การทำ Bagging จะมีปัญหาเรื่องความไม่เป็นอิสระของข้อมูลเนื่องจากต่อให้เราแยกออกไปหลายๆ Tree ก็จริงแต่มันก็คือข้อมูลเดียวกัน Random Forest จึงเข้ามาแก้ปัญหาตรงนี้ โดยการทำ Random Sample Feature

Random Sample Feature คือ นอกจากจะแบ่งเป็น Tree หลายๆ ต้นแล้ว ยังแบ่ง Feature ของ Tree แต่ละต้นจะมี Feature ที่ไม่เหมือนกันทั้งหมด เพื่อทำให้แต่ละ Tree มีความหลากหลายและมีความอิสระกันมากขึ้น

Boosting https://guopai.github.io/ml-blog11.html

Boosting เป็นอีกเทคนิคใน Ensemble learning ที่ใช้ Classifier หลายๆ Instance มาช่วยกันสร้างโมเดลและพยากรณ์ Boosting คล้ายกับ Random Forest แต่เป็นการนำ Classifier หลายตัวมาทำงานเป็นโซ่ต่อกัน โดยแต่ละตัวจะแก้ไขจุดด้อยของ Classifier ตัวก่อนหน้า พอฝึกฝนสำเร็จแล้ว Classifier ทุกตัวจะพยากรณ์ร่วมกัน

AdaBoost

AdaBoost เป็น Algorithm พื้นฐานของ Boosting method จึงควรลองทำความเข้าใจเพื่อเป็นฐานสำหรับ Algorithm อื่น

หลักการทำงานของ AdaBoost คือการใช้ Classifier ที่ไม่ซับซ้อน เช่น Decision tree ที่มีชั้นเดียว (เรียกว่า Decision stump) หลายๆ Instance มาเทรนต่อกันเป็นลูกโซ่ โดยในการเทรนแต่ละรอบจะมีการกำหนดค่าน้ำหนักของข้อมูลแต่ละรายการ Classifier instance จะเรียนรู้จากค่าน้ำหนักเหล่านั้น โดยถ้าพยากรณ์ผิด ค่าน้ำหนักของรายการนั้นจะมีค่ามากขึ้น ส่งผลให้ Classifier instance นั้นได้รับ "คะแนน" ต่ำ แต่ในทางกลับกัน ถ้า Instance ไหนพยากรณ์ถูกเป็นสัดส่วนที่มาก ก็จะได้คะแนนมาก

การพยากรณ์ของ AdaBoost คือการถ่วงคะแนนเข้ากับคำตอบของแต่ละ Instance แล้วเลือกคำตอบจาก Class ที่ได้รวมแล้วได้ค่าน้ำหนักมากที่สุด

Gradient boosting

Gradient boosting เลือกวิธีการในการ Optimize อีกวิธี โดยการพยายามให้ Classifier instance ที่มาใหม่แต่ละตัว มีความแม่นยำขึ้นเรื่อยๆ โดยเรียนรู้จากค่าความคลาดเคลื่อนสะสมที่เกิดจากการทำนายขอ Instance ก่อนหน้า

Hist Gradient boosting https://www.analyticsvidhya.com/blog/2022/01/histogram-boosting-gradient-classifier/

Histogram คือ กราฟแท่งที่แสดงความผันแปรของข้อมูลด้วยการเรียงลำดับและจัดกลุ่ม การทำ Hist Gradient boosting จึงเหมือนการจัดกลุ่มโดนมีรูปแบบของการทำ Gradient boosting ที่สนใจความถี่ของข้อมูลเป็นหลัก

XGBoost หรือ eXtreme Gradient Boosting medium.com/@nutorbitx/ทำไมใครๆก็ใช้-xgboost-กันจัง-a775b53cc1eb

XGBoost เป็น ensemble learning method ซึ่งเป็นการเรียนรู้แบบ multiple-learners สำหรับ Ensemble learning โดยมีรูปแบบเดียวกับ Gradient boosting โดยความสามารถที่เพิ่มขึ้นมาคือ Regularization, Sparse Aware, Parallelization และการ Cache Optimization

Light Gradient Boosting Machine (LGBM) finnomena.com/finnomena-ic/light-gradient-boosting-machine-model/

เป็นโมเดลทางคณิตศาสตร์ที่มีโครงสร้างเป็นแบบต้นไม้หลาย ๆ ต้น (trees) โดยต้นไม้เหล่านี้จะถูกสร้างขึ้นจากข้อมูลที่ใช้สอนโมเดล โดยโมเดลจะใช้ข้อมูลที่ทางทีมส่งให้โมเดลเรียนรู้ โดยในการสร้างต้นไม้แต่ละครั้ง จะมีขั้นตอนดังนี้

* โมเดลจะทำการค้นหาตัวแปรต้นที่ส่งผลอย่างมีนัยสำคัญต่อค่าตัวแปรตามที่สนใจ (ในกรณีนี้คือผลตอบแทน 6 เดือนถัดไปของกองทุน) โดยการเปรียบเทียบความสามารถในการแบ่งกลุ่มค่าผลลัพธ์ของตัวแปรตาม ตามค่าตัวแปรต้นที่เปลี่ยนแปลงไป จากนั้นจะทำการสร้างเงื่อนไขว่า ถ้าหากตัวแปรต้นดังกล่าวมีค่าอยู่ในช่วงแต่ละช่วง ค่าตัวแปรตามควรจะมีค่าเท่าไหร่ โดยใช้ค่าเฉลี่ยของตัวแปรตามที่ถูกจัดกลุ่มอยู่ในกลุ่มเดียวกัน
* จากนั้นโมเดลจะตรวจสอบว่าการแบ่งกลุ่มดังกล่าวสามารถแยกค่าตัวแปรตามที่มีค่าแตกต่างกันออกจากกัน และจับกลุ่มค่าตัวแปรตามที่มีค่าใกล้เคียงกันไว้ด้วยกัน ดีกว่าการไม่แบ่งกลุ่มหรือไม่ ซึ่งถ้าหากไม่ ก็จะไม่ทำการแบ่งกลุ่มนั้น
* เมื่อได้ตัวแปรต้นที่ใช้ในการแบ่งกลุ่มแล้ว โมเดลจะเริ่มวาดต้นไม้โดยใช้ตัวแปรต้นดังกล่าวแตกกิ่งค่าความเป็นไปได้ของตัวแปรตามออกมา
* จากนั้นโมเดลจะทำซ้ำสามขั้นตอนดังกล่าว กับตัวแปรต้นอื่น ๆ ที่ยังไม่ถูกเลือก จนได้ต้นไม้ที่ประกอบด้วยกิ่งจำนวนมาก ซึ่งมีความละเอียดในการทำนายค่าตัวแปรตามที่เพียงพอ

โมเดลจะทำการสร้างต้นไม้ด้วยวิธีดังกล่าวหลาย ๆ ต้น ซึ่งในการสร้างต้นไม้แต่ละต้น จะมีการสุ่มเลือกตัวแปรต้น และชุดข้อมูล ทำให้การสร้างแต่ละครั้ง ได้ผลลัพธ์ที่แตกต่างกัน จากนั้นใช้ค่าเฉลี่ยของการทำนายจากต้นไม้หลาย ๆ ต้น เป็นค่าทำนายตัวแปรตามสุดท้าย

Cat Boosting

ในขั้นตอนต่อไปเป็นการเลือกชุดข้อมูล คณะผู้วิจัยต้องการข้อมูลที่มีความสับสอน และ เป็นข้อมูลรูปภาพใต้กล้องจุลทรรศน์ จึงเลือกที่จะใช้ เซลล์เม็ดเลือดขาว เซลล์เม็ดเลือดขาวอันประกอบไปด้วย นิวโตรฟิล (Neutrophil) ลิมไฟไซท์ (Lymphocyte) โมโนไซท์ (Monocyte) อิโอซิโนฟิลด์ (eosinophil)

2.1.2 เม็ดเลือดขาว

เม็ดเลือดขาวมีขนาดใหญ่กว่าเซลล์เม็ดเลือดแดง จำนวนเม็ดเลือดขาวปกติประมาณ 4,000 – 10,000 เซลล์ต่อ 1 ไมโครลิตร

เม็ดเลือดขาวแต่ละชนิดมีลักษณะและคุณสมบัติแตกต่างกัน ทำให้ปริมาณที่เพิ่มขึ้น เป็นผลต่อการตอบสนองของร่างกายต่อปัจจัยการกระตุ้นที่แตกต่างกัน

|  |  |
| --- | --- |
|  | นิวโตรฟิล (Neutrophil) |
|  | ลิมไฟไซท์ (Lymphocyte) |
|  | โมโนไซท์ (Monocyte) |
|  | อิโอซิโนฟิลด์ (eosinophil) |

2.2 งานวิจัยที่เกี่ยวข้อง

เนื่องจากในขั้นตอนในการทำ Machine learning จำเป็นต้องเลือก อัลกอรึทึม Classifier ในการตรวจจับ และจัดหมวดหมู่ (Image Classification) โดยคณะผู้วิจัยเลือกใช้ Traditional Machines learning

2.2.1 Machine Learning อัลกอรึทึม และ Deep learning อัลกอรึทึม

1 การศึกษาและเปรียบเทียบประสิทธิภาพของ Support Vector Machine (SVM) และ Deep learning อย่าง Sparse Autoencoder (SAE) auto-encoder (SAE)

1.1 การใช้งาน Parameter

จากการทดสอบระหว่าง SVM และ SAE พบว่า SAE จำเป็นต้องมีการตั้ง Parameter ที่มากกว่าเพื่อให้ได้ประสิทธิภาพที่ดีกว่า

1.2 ปริมาณข้อมูลที่ใช้ในการฝึกฝน

จากการทดสอบพบว่า SVM นั้นใช้ปริมาณของข้อมูลที่มีกลาง ถึงน้อย อย่างไรก็ตามเนื่องจาก SAE นั้นต้องการ Parameter มากกว่าทำให้ปริมาณของข้อมูลจึงมีขนาดมากตาม ทั้งนี้ในการทดสอบแสดงให้เห็นว่า หากเป็นการนำใช้ในจำนวนข้อมูลที่มากกว่าทำให้เห็นว่า SAE นั้นมีประสิทธิภาพมากกว่า

1.3 ระยะเวลาในการฝึกฝน

จากการทดสอบพบว่า SVM ใช้เวลาที่น้อยกว่าในการประมวลผลการฝึกฝนทั้งหมดในทุกรูปแบบของการทดลอง เนื่องจากปริมาณของ Parameter ที่จำเป็นต้องมีการตั้งเพื่อให้ได้ประสิทธิภาพสูงสุดน้อยกว่า

1.4 ประสิทธิภาพในการตรวจจับ และจัดหมวดหมู่ Image Classification

จากการเปรียบเทียบประสิทธิภาพของ Classification พบว่า SVM มีประสิทธิภาพในการตรวจจับ และจัดหมวดหมู่ ได้ดีที่สุดในทุกๆการทดลอง เนื่องจาก SAE มีความสามารถมากกว่าในการประมวลผลข้อมูลที่มีขนาดใหญ่

1.5 ประสิทธิภาพของการทำ Active learning

จากการทดสอบทำให้เห็นว่าทั้ง SVM และ SAE มีความสามารถในการคัดเลือกตัวอย่างที่มีการทำ การระบุตัวทดสอบและตัวฝึกฝน (label) อย่างไรก็ตามพบว่า SVM จะสามารถพัฒนาประสิทธิภาพในการตรวจจับ และจัดหมวดหมู่ได้รวดเร็วกว่า SAE

1.6 ความ ละเอียดอ่อนของการประมวลผล

จากการทดสอบในเรื่องของ ความละเอียดอ่อนของการประมวลผลพบว่า การประมวลผลผ่านข้อมูลทื่มี noise พบว่า SVE ไม่มีการเพิ่มประสิทธิภาพในข้อมูลที่มีความละเอียดอ่อนเท่า SVM

2.2.2 Hyperparameter tuning

1. การเปรียบเทียบ Hyperparameter Optimization สำหรับ model classification

จากการทดสอบการทำ neural architecture search approach (NAS) โดยมีการทำ Hyperparameter Optimization มีการใช้อย่างแพร่หลายทั้งหมด 3 ตัวคือ Grid Search, Random Search และ Genetic Algorithm พบว่าในการทำทั้ง 3 ในการทำ Grid Search และ Genetic Algorithm ที่มีรูปแบบของ อัลกอรึทึม ที่เป็นแบบ Brute Force ค่อนข้างใช้ระยะเวลานานในการประมวลผล ในส่วนของ Randomize Search มีเวลาที่รวดเร็วมากกว่า แต่ ในส่วนของ Randomize Search เป็นลักษณะของการ Random ทำให้ค่าที่ออกมาในการทำแต่ล่ะครั้งไม่สามารถรับประกันได้ว่า Randomize Search จะสามารถหา parameter ที่ดีที่สุดได้เสมอ

2.2.3 Data augmentation

1 แบบสำรวจการทำ Data augmentation ในการทำ Image Classification สำหรับการทำ Deep learning

Data Augmentations based on basic image manipulations

Data augmentation เป็นรูปแบบการสร้างความแปรปวน เพื่อช่วยเพิ่มประสิทธิภาพ เช่น geometric transformations, color space transformations, kernel filters, mixing images, random erasing, feature space augmentation, adversarial training, GAN-based augmentation, neural style transfer, and meta-learning schemes

1. Geometric transformations

Geometric transformations เป็นวิธีพื้นฐานของการทำ Data augmentations โดยมีรูปแบบจากการปรับเปลี่ยนรูปภาพโดยวิธีต่างๆ โดยใช้ร่วมกับฟังก์ชั่นเพื่อเพิ่มปริมาณของข้อมูลโดยยังคงเหลือรูปแบบเดิมจากข้อมูลทำให้การทำ Data augmentation โดยพื้นฐานของการทำประกอบไปด้วย

1.1 Flipping การพลิกรูปภาพในแกน X และ แกน Y

1.2 Rotation การหมุนรูปภาพในองศาต่างๆ

1.3 Cropping การตัดหรือปิดภาพบางส่วน

1.4 Translation การเอียงรูปภาพที่ทางซ้าย หรือ ทางขวา

2. Color space transformations

Color space transformations หรือ photometric transformations เป็นรูปแบบของการทำ Data augmentations โดยมีรูปแบบจากการปรับเปลี่ยน pixel ของรูปภาพโดยปรับเปลี่ยน รูปแบบของสี RGB โดยการเพิ่มลด ปรับ และ เปลี่ยนรูปแบบของ สีใน pixel นั้นๆ โดยวิธีพื้นฐานของ ประกอบไปด้วย

2.1 Contrast

2.2 White balance

2.3 Sharpen

บทที่ 3

วิธีการดำเนินการทดลอง

ในลำดับถัดไปจะเป็น โปรแกรม, ไลบรารี่ที่ใช้ในการทดสอบ, ชุดข้อมูล, อัลกอรึทึมที่ใช้ทดสอบ และ parameter ที่ใช้สำหรับอัลกอรึทึมแต่ล่ะตัว สำหรับโครงงานนี้

เครื่องมือที่ใช้ในการพัฒนา

1. ฮาร์ดแวร์

1.1. คอมพิวเตอร์ ระบบปฏิบัติการ Window

1.2. Virtual Computer อย่าง Google Colabs

2. ซอฟต์แวร์

2.1. ภาษา Python เวอร์ชั่น 3.11.1

2.1.1. ไลบรารี่ pathlib

2.1.2. ไลบรารี่ matplotlib.pyplot

2.1.3. ไลบรารี่ numpy

2.1.4. ไลบรารี่ sklearn

2.1.5. ไลบรารี่ pandas

2.1.6. ไลบรารี่ skimage

2.1.7. ไลบรารี่ XGBoost

2.1.8. ไลบรารี่ LightGBM

2.1.9. ไลบรารี่ CatBoost

2.2. Jupyter Notebook

2.3. Visual Studio Codes

3. ชุดข้อมูล

BCCD Dataset ชุดข้อมูลของ เซลล์เม็ดเลือดที่มีการ Label ข้อมูลภาพทั้งหมด โดยประกอบไปด้วย

|  |  |
| --- | --- |
| ชื่อ | จำนวนภาพ |
| นิวโตรฟิล (Neutrophil) | 211 |
| ลิมไฟไซท์ (Lymphocyte) | 37 |
| โมโนไซท์ (Monocyte) | 22 |
| อิโอซิโนฟิลด์ (eosinophil) | 93 |

4. อัลกอรึทึมและ parameter ของอัลกอรึทึม

|  |  |  |
| --- | --- | --- |
| อัลกอรึทึม | จำนวนของ parameter | parameter |
| SVM linear kernel | 1 | C |
| SVM radial basis function kernel | 2 | C, Gamma |
| SVM Polynomial kernel | 3 | Gamma, Degree, Coef0 |
| Gaussian naive Bayes | 1 | var\_smoothing |
| Multinomial naive Bayes | 2 | alpha, fit\_prior |
| Bernoulli naive Bayes | 2 | alpha, fit\_prior |
| Decision Tree | 5 | criterion, max\_feature, max\_depth, min\_samples\_split, min\_samples\_leaf |
| Random Forest | 4 | criterion, max\_feature, max\_depth, n\_estimators |
| K-Nearest Neighbors | 2 | n\_neighbors, metric |
| Ada Boosting | 2 | n\_estimators, learning\_rate |
| Gradient Boosting | 3 | n\_estimators, n\_estimators, learning\_rate |
| Hist Gradient Boosting | 2 | n\_estimators, learning\_rate |
| eXtreme Gradient Boosting | 3 | n\_estimators, n\_estimators, learning\_rate |
| light gradient-boosting machine | 12 | subsample\_for\_bin, subsample, reg\_lambda, reg\_alpha, num\_leaves, n\_estimators, min\_child\_samples, learning\_rate, is\_unbalance, colsample\_bytree, boosting\_type |
| Cat Boost | 4 | learning\_rate, l2\_leaf\_reg, iterations, depth |

ขั้นตอนการดำเนินการ

ในการทดสอบการเปรียบเทียบประสิทธิภาพของ Machine learning ในการทำ Image Classification รูปภาพของเซลล์เม็ดเลือดจากฐานข้อมูล BCCD ทั้งหมด 4 ขั้นตอน ประกอบไปด้วย Data preparation, Data Augmentation, Training, Hyperparameter tunning

ขั้นตอนที่ 1 Data preparation

1. ทำการดาว์โหลดข้อมูลโดยการ Git Clone จาก ฐานข้อมูล BCCD ใน GitHub Repository ที่มีชื่อว่า Shenggan / BCCD\_Dataset

2. ทำการแบ่งข้อมูลที่ได้มาจาก BCCD\_Dataset โดยใช้ข้อมูล Label จาก Annotations เป็นทั้งหมด 4 คลาสตามข้อมูลที่ต้องการจะทำ Image Classification

3. ทำการตรวจสอบข้อมูลภาพภายในโฟลเดอร์ของฐานข้อมูลว่าข้อมูลภาพนั้นๆเป็นข้อมูลที่สามารถใช้งานได้ (ไม่ใช้สีดำ หรือ ขาวล้วน, ไม่ใช้ภาพเสีย และ มีข้อมูลที่ต้องการจะจำแนก) โดยตรวจสอบและจำแนกข้อมูลดั้งกล่าวอีกครั้ง

5. ทำการปรับขนาดภาพของ ข้อมูลทั้งหมดให้อยู่ในขนาด 64x64 pixel และ ย้ายข้อมูลทั้งหมดที่ได้ไปยัง โฟลเดอร์ ที่มีชื่อคลาสของ ภาพนั้นๆโดยทำการตั้งชื่อภาพนั้นๆ โดย \_(label ที่ n ของ ภาพเดิม)

4. ทำการสร้าง Histogram เพื่อตรวจสอบข้อมูลที่ได้จากกระบวนการทั้งหมด

ขั้นตอนที่ 2 Data Augmentation

1. นำข้อมูลภายในโฟลเดอร์ทั้ง 4 คลาสทีได้มาจากขั้นตอนที่แล้วทำการ Data Augmentation โดยการทำ Augmentation ทั้งการ Flipping, Rotation, Cropping, Translation, Contrast, White Balance และ Sharpen

2. ทำกระบวนการทั้งหมด โดยใช้การสุ่มวิธีการในภาพแต่ล่ะภาพโดยนำมาเหล่านั้นมาใช้งานในการปรับเปลี่ยนและเพิ่มข้อมูลไปเรื่อยๆ โดยใช้ภาพและเปลี่ยนวิธีการในการทำทั้งหมด ประมาณภาพล่ะ 34 ครั้ง โดยแต่ล่ะครั้งวิธีการไม่ซ่ำกัน

3. นำภาพแต่ล่ะครั้งทีได้มาทำการย้ายไปยัง โฟลเดอร์ที่เป็น คลาสของตัวเอง และ ทำการตั้งชื่อ \_(label ที่ n ของ ภาพเดิม)\_(เป็นการทำ Data augmentation ครั้งที่ n)

4. ทำการตรวจสอบข้อมูลภาพภายในโฟลเดอร์ของฐานข้อมูลว่าข้อมูลภาพนั้นๆเป็นข้อมูลที่สามารถใช้งานได้อีกครั้ง และ ทำการตรวจสอบความถูกต้องของคลาสของ ภาพนั้นๆอีกครั้ง

5. ทำการสร้าง Histogram เพื่อตรวจสอบข้อมูลที่ได้จากกระบวนการทั้งหมด

ขั้นตอนที่ 3 Training

1. ทำการอัพโหลดข้อมูลทั้งหมดจากคอมพิวเตอร์ที่ได้จากกระบวนการขั้นต้นขึ้นบน Google Colabs โดยแบ่งแยกโฟลเดอร์ทั้งหมดตามชื่อ คลาส

2. ทำการตรวจสอบข้อมูลภาพภายในโฟลเดอร์ของฐานข้อมูลว่าข้อมูลภาพนั้นๆเป็นข้อมูลที่สามารถใช้งานได้อีกครั้ง และ ทำการตรวจสอบความถูกต้องของคลาสของ ภาพนั้นๆอีกครั้ง

3. ทำการติดตั้ง ไลบรารี่ที่จำเป็นสำหรับการ Training

4. ทำการสร้างตัวแปร โดยใช้ตัวแปรรูปแบบ Data Frame และดึงข้อมูลเวกเตอร์ภาพของแต่ละภาพออกมาและเก็บไว้ที่ตัวแปรนั้นๆ และ ใส่ข้อมูล label ของคลาสของรูปนั้นๆกำกับ

5. ทำการตั้งชื่อ Data Frame โดยในส่วนที่เป็นเวกเตอร์ภาพให้เป็น X และ ในส่วนของ label ให้เป็น y

6. ทำการ แบ่งข้อมูลโดยนำข้อมูลจำนวน 70% ไปใช้ในการฝึกฝน และ ข้อมูลอีก 30% ไปใช้ในการทดสอบ อัลกอรึทึม และเก็บไว้ใน X\_train, X\_test, y\_train และ y\_test ตามลำดับ

7. ทำการนำข้อมูลที่ได้ทดสอบกับ อัลกอรึทึม โดยนำข้อมูลจากข้อข้างต้นมาทดสอบทั้งหมด 15 อัลกอรึทึม

ขั้นตอนที่ 4 Hyperparameter tunning

1. นำอัลกอรึทึมที่ทดสอบทั้งหมดมาใช้ร่วมกับ RandomizedSearchCV โดยนำข้อมูล parameter ที่โมเดลแต่ล่ะตัวต้องใช้ตามตาราง และ ใช้ข้อมูลเวกเตอร์ภาพมาใช้

2. ทำการทดสอบโดยมีการนำ K-fold cross validation ทั้งหมด 5 ครั้งในแต่ล่ะรอบ และ มีการลิมิตการค้นหา parameter ทั้งหมด (n\_iter) ที่ 10\*จำนวนของ parameter รูปแบบ

3. เมื่อข้อที่ 2 เสร็จสิ้นทำการเรียกค่า parameter ที่ดีที่สุดในแต่ล่ะรอบมาเก็บไว้

ขั้นตอนที่ 5 Evaluation

1. นำ อัลกอรึทึมที่ได้จากการนำ parameter ในขั้นตอนที่ 4 มาทำการ predict และ predict possibility ทั้งหมด

2. นำข้อมูลที่ได้จากข้อที่ 1. แต่ล่ะตัวมาทดสอบผ่านการทำ Metrics and scoring โดยเลือกใช้ข้อมูลอย่าง accuracy, precision, recall, f1\_score ทั้งแบบ Macro และ Weighted

3. ข้อมูลการ predict และ predict possibility ของอัลกอรึทึมที่ได้มาทำการ พร็อตกราฟ Receiver operating characteristic (ROC) แบบ One Vs Rest (OvR) โดยทำการระบุ

3.1 เส้นภายในกราฟ 4 เส้นแรกเป็นความสามารถในการ predict ของคลาสแต่ล่ะคลาสว่า อัลกอรึทึมมีความสามารถขนาดไหน

3.2 เส้น Medium ของกราฟ

3.3 เส้นสุดท้ายเป็นคะแนน Overall ของกราฟ Macro และ Micro และ หาพื้นที่ใต้กราฟ

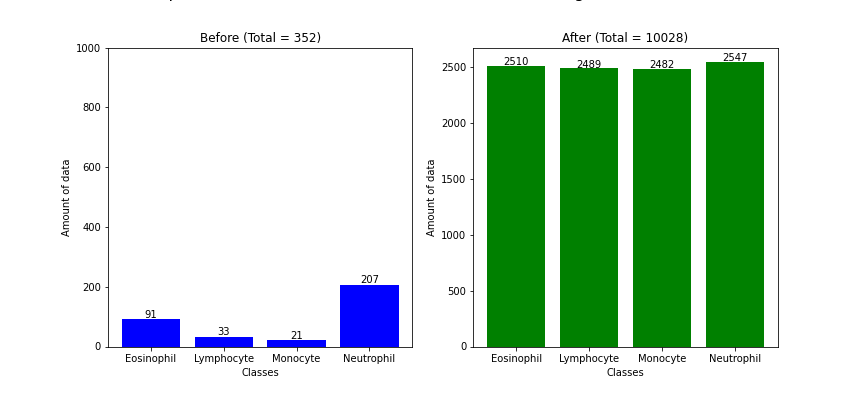
4. ทำซ้ำกับอัลกอรึทึมทั้งหมด 15 ตัวและเปรียบเทียบข้อมูล

บทที่ 4

เอกสารและงานวิจัยที่เกี่ยวข้อง

ขั้นตอนที่ 1 และ ขั้นตอนที่ 2 การทำ Data preparation และ Data Augmentation

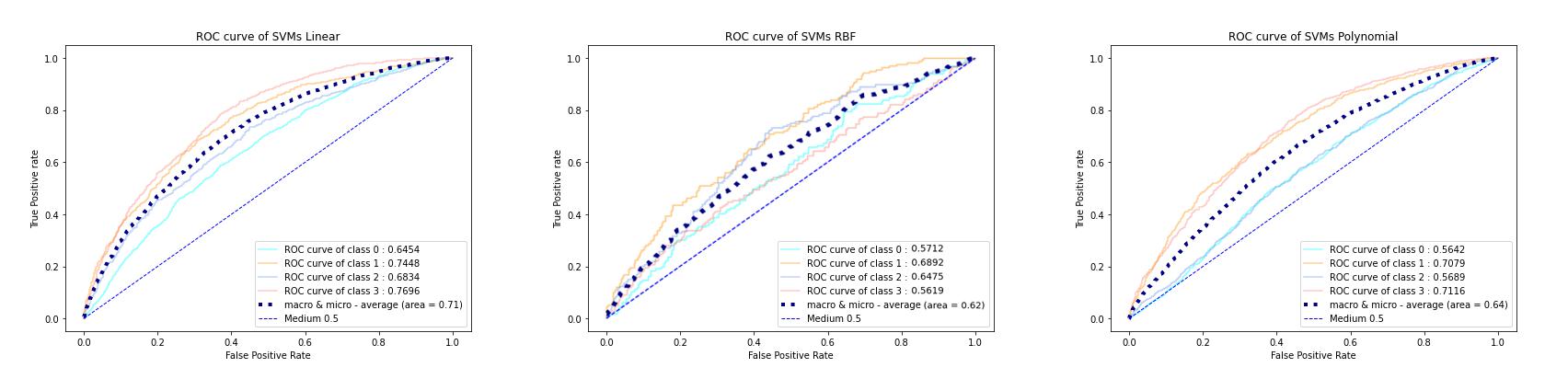
ในขั้นตอนทั้งสองเป็นการทดสอบกับการปริมาณและรูปแบบของข้อมูลภาพ ในช่วงก่อนและหลังมีการทำ Histogram เพื่อช่วยในการตรวจสอบความ balance และปริมาณของข้อมูลโดยได้ผลการเปรียบทั้ง 2 กราฟ

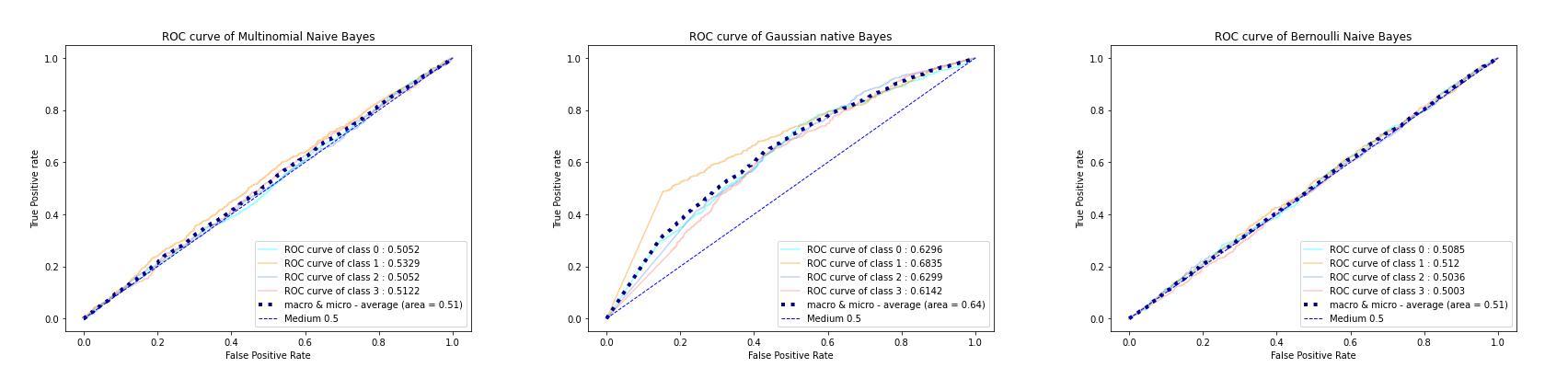


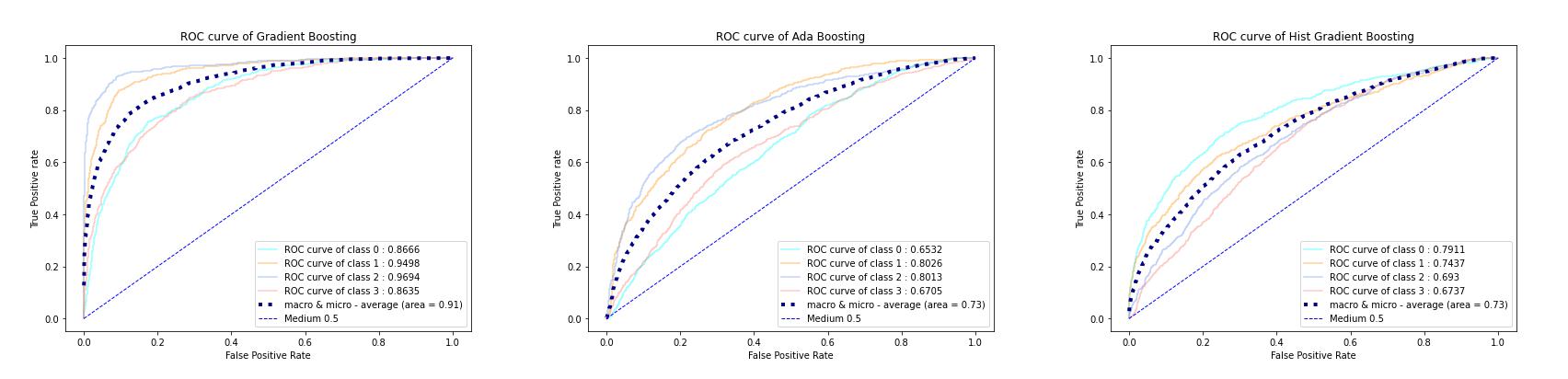
ขั้นตอนที่ 5 Evaluation

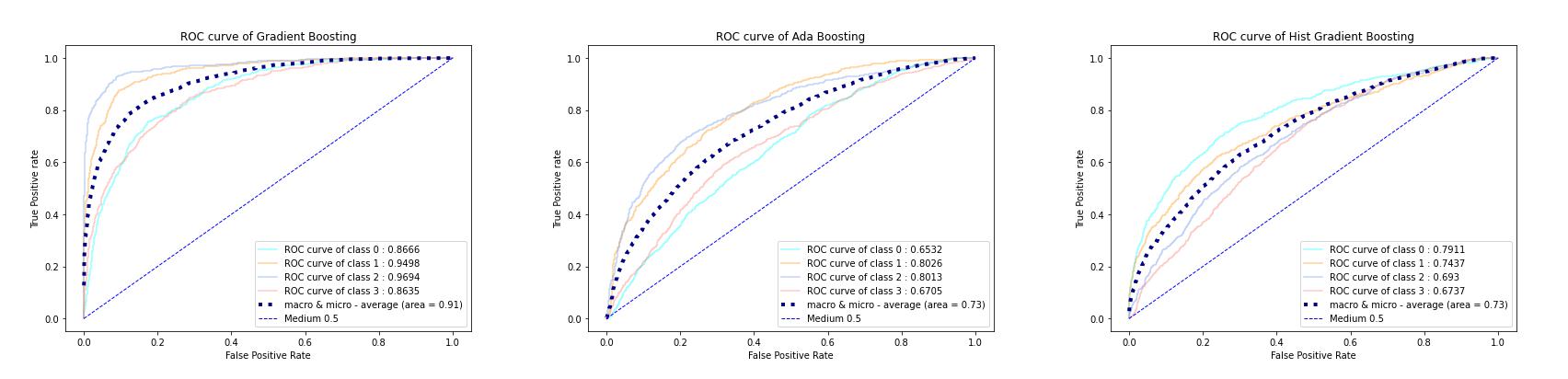
ในขั้นตอนนี้เป็นการนำ อัลกอรึทึมที่ผ่านการทำ Hyperparameter มาทำการ predict และ predict possibility และนำนำข้อมูลที่ได้มาทำการ Metrics and scoring โดยเลือกใช้ข้อมูลอย่าง accuracy, precision, recall, f1\_score ทั้งแบบ Macro และ Weighted และมีการทำ ROC เพื่อหาพื้นที่ใต้กราฟ

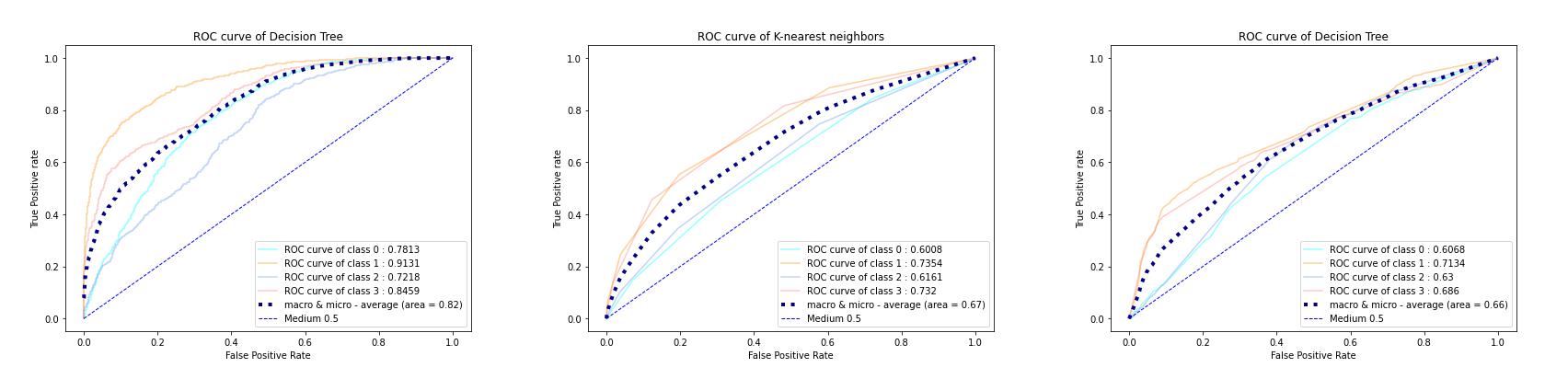
Metric and Scoring

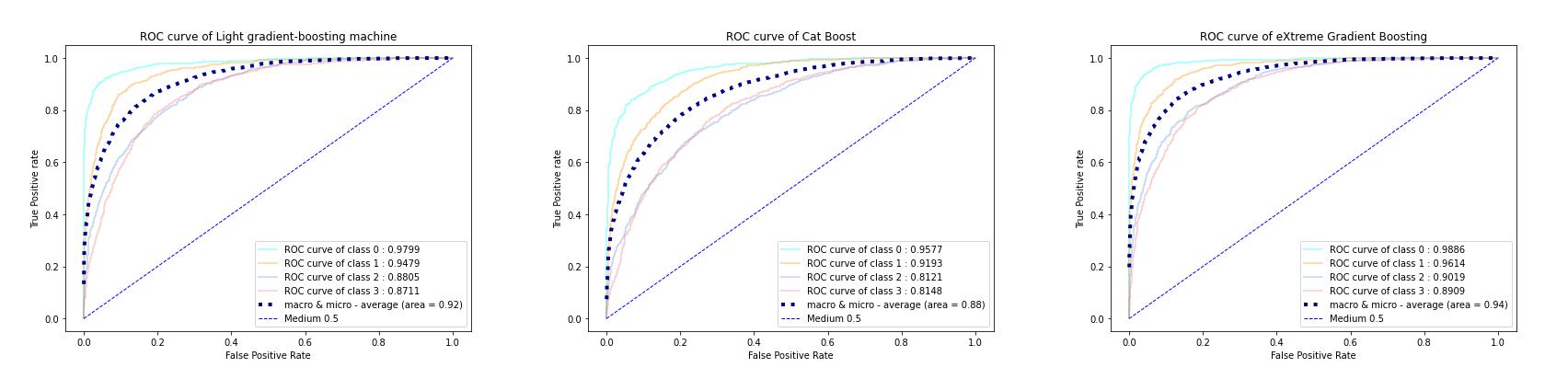
Receiver operating characteristic Curve ROC











บทที่ 5

สรุป อภิปรายผล และข้อเสนอแนะ

จากการศึกษาและเปรียบเทียบประสิทธิภาพของอัลกอรึทึมของ Machine learning ในการตรวจจับภาพขนาดเล็กภายใต้กล้องจุลทรรศน์ โดยข้อมูลจำนวนจำกัด สรุปผลการศึกษาได้ดั้งนี้

สรุปผล

1. การทำ Data Augmentation ด้วยการใช้เทคนิค Geometric และ Color Space transformation ทำให้สามารถเพิ่มประสิทธิภาพจาก ข้อมูลเดิมที่จำนวน 352 รูป เป็น 10,028 และ ข้อมูลมีความ Balance

2. อัลกอรึทึมที่มีประสิทธิภาพมากที่สุดคือ eXtreme Gradient Boosting โดยมี การใช้ parameter ที่ 3 ตัวและคะแนน Metric and Scoring โดยที่ Accuracy ที่ 0.77, Precision เฉลี่ยที่ 0.77, Recall เฉลี่ยที่ 0.77, F-measure ที่ 77% และ ROC AUC ที่ 93%

อธิปายผล

การศึกษาและเปรียบเทียบประสิทธิภาพของอัลกอรึทึมของ Machine learning ในการตรวจจับภาพขนาดเล็กภายใต้กล้องจุลทรรศน์ โดยข้อมูลจำนวนจำกัด ทำให้เห็นว่า การทำ Data Augmentation ด้วยเทคนิคอย่าง Geometric และ Color Space transformation สามารถเพิ่มปริมาณข้อมูลได้จากจำนวนเดิมคือ 352 รูปเป็น 10,028 คิดเป็น 2484% ทั้งนี้ จากกราฟฮิสโตรแกรมการเปรียบเทียบของข้อมูลก่อนและหลัง จากกราฟและตารางในส่วนของคะแนน Metric and Scoring แสดงให้เห็นคะแนน ของ Weighted และ Macro มีคะแนนที่เท่ากัน แสดงให้เห็นว่า ข้อมูลที่นำมาใช้ในการฝึกฝนหลังจากกระบวนการทำ Data Augmentation ทำให้เห็นว่าข้อมูลมีการ Balance ของปริมาณของทั้ง 4 คลาส และในส่วนของการ Training และ Evaluate ทำให้เห็นว่า อัลกอรึทึมที่มีประสิทธิภาพมากที่สุดคือ eXtreme Gradient Boosting โดยมี การใช้ parameter ที่ 3 ตัวและคะแนน Metric and Scoring โดยที่ Accuracy ที่ 0.77, Precision เฉลี่ยที่ 0.77, Recall เฉลี่ยที่ 0.77, F-measure ที่ 77% และ ROC AUC ที่ 93% โดยที่มีประสิทธิภาพมากกว่า อัลกอรึทึม Cat Boost ที่ใช้ parameter ที่ 11 ตัว

From the experimental results, it was possible to discuss that after using Data Augmentation with Geometric and Color space transformation data can increase data from 352 to 10028 with ratio of 2484% also from Histogram that compare between before and after the data augmentation and graph and table of Metric and Scoring that show Weighted and Macro of each score and each algorithm show with nearly score, this show that dataset is now balance. In Training and the process of Training and Evaluating showed that an algorithm for image classification had the most performance was the eXtreme gradient boosting at 0.93 with Accuracy in 0.77, Precision average is 0.77, Recall average is 0.77, F-measure at 77%, Which is better than Cat Boost and other algorithms that require parameter more than 3 and show that boosting base algorithms do better performance than other algorithms group

According to the experimental results, using Data Augmentation with Geometric and Color space transformation data can increase data from 352 to 10028 with a ratio of 2484%. Histograms that compare before and after data augmentation and graphs and tables of Metric and Scoring that show Weighted and Macro of each score and each algorithm show with nearly score, indicating that the dataset is now balanced. In the process of Training and Evaluating showed that an algorithm for image classification had the most performance was the eXtreme gradient boosting at 0.93 with Accuracy in 0.77, Precision average is 0.77, Recall average is 0.77, F-measure at 77%, Which performs better than Cat Boost and other algorithms that need more than three parameters and demonstrate that boosting base methods perform better than other algorithms group