

Individual Study in Computer Engineering Report  
PET/CT Brain Scan Imaging Diagnosis in Epilepsy Patient  
(การตรวจจับจุดกำเนิดชักในสมองจากภาพถ่าย PET/CT Scan)

จัดทำโดย อศิราภรณ์ สิทธิการิยะวัตร 6330563421

อาจารย์ผู้ดูแล อ.ดร.พรรณราย ศิริเจริญ

### ความเป็นมาและวัตถุประสงค์

การตรวจหาจุดกำเนิดชักในสมองของผู้ป่วยจากภาพถ่าย PET/CT Scan เป็นหนึ่งในงานที่ละเอียดอ่อน เนื่องจากต้องทำการเปรียบเทียบความแตกต่างของสีและลักษณะสมองทั้งส่วนของ gray matter และ white matter ระหว่างสมองทั้งสองซีกจากภาพถ่ายของผู้ป่วย ซึ่งสมองมนุษย์เราเองนั้นไม่มีความสมมาตร การเปรียบเทียบจึงเป็นค่อนข้างซับซ้อน นอกจากนั้นเองปริมาณของภาพถ่ายต่อสมองผู้ป่วยหนึ่งรายที่ต้องทำการตรวจสอบนั้นค่อนข้างมาก เนื่องจากเป็นภาพตัดตามแนวราบตั้งแต่บริเวณโคนสมองถึงส่วนบนสุดของสมองซึ่งจำเป็นต้องทำการตรวจสอบทั้งหมดเพราะจุดกำเนิดชักมีโอกาสเกิดได้ทุกที่ การนำศาสตร์ของ machine learning มาช่วยในการวินิจฉัยผลย่อมช่วยลดภาระงาน และเพิ่มประสิทธิภาพและความรวดเร็วในการทำงานในส่วนองแพทย์ได้ค่อนข้างมาก

ในการทำโครงงานครั้งนี้ ผมหวังว่าจะได้ศึกษาเรียนรู้การทำ machine learning ในการทำงานจริง ตั้งแต่การศึกษา model และวิธีการต่างๆ การเตรียมข้อมูล และการ train model ของ machine จริง รวมถึงการเพิ่มประสิทธิภาพของ model ที่ได้ลงมือทำอีกด้วย

### ขั้นตอนการทำงาน

1. ศึกษา machine learning ทั้งในเรื่องพื้นฐาน ML, object detection, convolutional network รวมถึง model ต่างๆ เนื่องจากยังไม่มีพื้นฐานในด้านนี้จึงจำเป็นต้องปูพื้นฐานก่อน โดยศึกษาจากหนังสือ Deep Learning with Python และ Coursera รวมถึงการลอง Implement จากตัวอย่าง
2. ศึกษาเรื่องสมองและจุดกำเนิดชักของโรคชักกระตุก โดยการบรรยายจากแพทย์ รวมถึงการอ่านภาพถ่าย PET/CT Brain Scan จริงของผู้ป่วย
3. จัดการเตรียม dataset สำหรับใช้ในการ train model ทั้งสำหรับ training และ validating
4. ทำการ train model ในการหาจุดกำเนิดชักในสมอง

## ผลลัพธ์จากการทำงาน

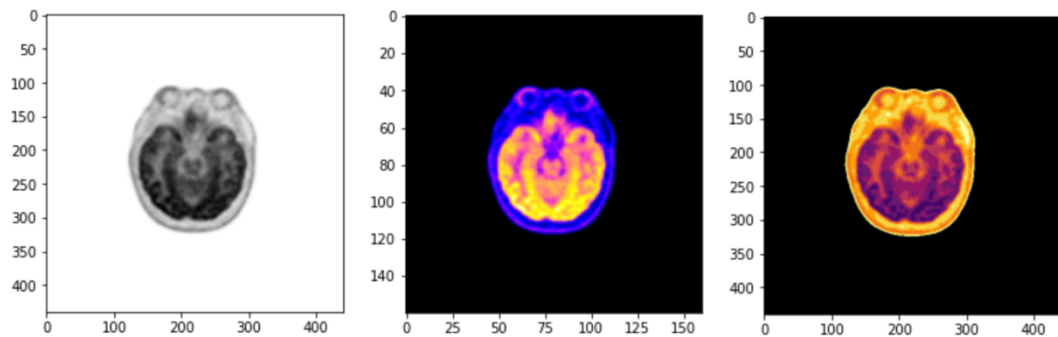
จากในขั้นตอนแรกที่ได้ศึกษาพื้นฐานของ ML, Object detection, Convolutional network รวมถึง model ต่างๆ ผมได้เริ่มต้นรู้จักการ train model ทั้งในรูปแบบของ supervised learning ซึ่งก็คือการนำ data ที่มีการ label คำตอบที่ถูกต้องไว้แล้วไปให้ machine เรียนรู้ ทั้งแบบ Regression และ Classification รวมถึงการ train อีกรูปแบบที่ทำการ train จาก unlabeled data หรือ data ที่ไม่ได้มีการบอกคำตอบที่ถูกต้องไว้ ซึ่งก็คือ unsupervised learning ได้แก่การทำ Clustering ซึ่งจากพื้นฐานต่างๆเหล่านี้สามารถนำไปต่อยอดในแขนงที่ลึกขึ้นได้ต่อไป

ในส่วนของโครงงานนี้ ซึ่งเป็นการตรวจหาจุดกำเนิดชักในสมองของผู้ป่วยจากภาพถ่าย PET/CT Scan เป็นลักษณะของ supervised learning ในรูปแบบของการทำ Classification เพื่อแยกสมองส่วนที่เป็นจุดกำเนิดชักออกจากรูปถ่าย ในส่วนนี้ผมจึงได้ทำการศึกษาเพิ่มเติมตั้งแต่การเตรียม data, เทคนิคการแบ่งรูปหรือการปรับแต่งสีของรูปเพื่อใช้สำหรับการเตรียม data ของทั้ง training set และ validating set ได้รู้จักปัญหาของการ train model ต่างๆทั้ง overfit และ underfit รวมถึงวิธีการแก้ไข ซึ่งทั้งหมดนี้จะได้ใช้ถัดไปในขั้นตอนการ train model จริง

ในขั้นถัดไปผมจึงได้เริ่มศึกษาในเรื่องของ Edge detection, Deep convolution model และ object detection ซึ่งในท้ายที่สุด ผมได้เลือก model U-Net มา เพื่อใช้ในตรวจหาจุดกำเนิดชักในสมองของผู้ป่วยจากภาพถ่าย PET/CT Scan โดยที่ U-Net เองมีความโดดเด่นในการทำ biomedical image segmentation ซึ่งมีทั้งการใช้เทคนิค transpose convolution ในการ resize output ให้กลับมามีขนาดเท่าเดิม และใช้ skip connection เพื่อส่งต่อข้อมูลไปยังส่วนลึกของ network ซึ่งหลังจากนั้น ผมจึงได้ลองศึกษา demo ของ U-net liked architecture ในการทำ segmentation รูปหมาและแมว เพื่อศึกษาเพิ่มความเข้าใจและนำไปเป็นแนวทางในการ train model จริงต่อไป

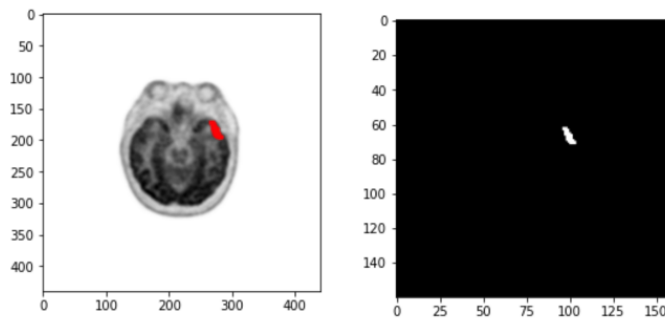
ในส่วนต่อไปซึ่งเป็นการลงมือ train model จริงมีดังต่อไปนี้

1. เตรียม data สำหรับ training ในส่วนของ data นั้น file ภาพเป็น file แบบ dicom จึงต้องมีการใช้ library ข้างนอกเข้ามาช่วยอ่าน file นั้นๆ รวมถึงการแปลงรูปจากขาวดำเป็นทั้งแบบ 10 color mapping และแบบโทนสี `gnuplot2_r` ของ `matplotlib` เพื่อใช้ในการตรวจจับความต่างของระดับสีในรูป อันเป็นสัญญาณของจุดกำเนิดชักในสมอง



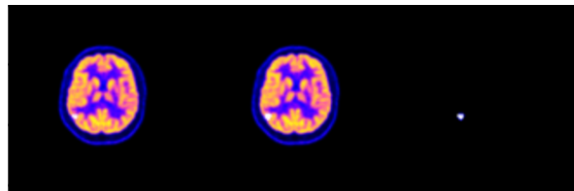
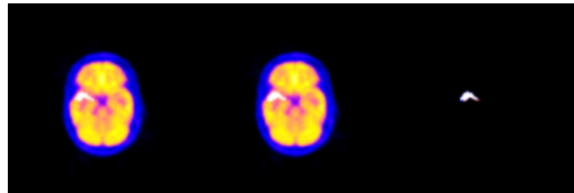
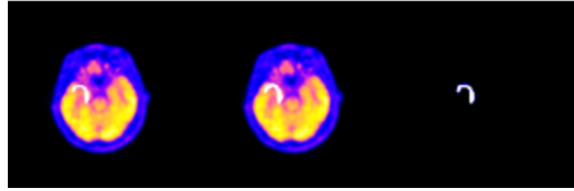
รูปจริง (ซ้าย), รูปที่แปลงแบบ `gnuplot2_r` (กลาง), รูปที่แปลงแบบ 10 color mapping (ขวา)

2. เตรียม label รูป PET/CT Scan ซึ่งมีการ label มาเป็นสีแดงในเบื้องต้นให้แล้ว แต่ยังเป็นการ label บนรูปขาวดำ (3-channel 3D array) จึงต้องแปลงข้อมูลดังกล่าวเป็น 1-channel 3D array ที่ประกอบไปด้วยค่า 0 (ส่วนที่ไม่เป็นจุดกำเนิดชัก) และ 1 (ส่วนที่เป็นจุดกำเนิดชัก) เท่านั้น

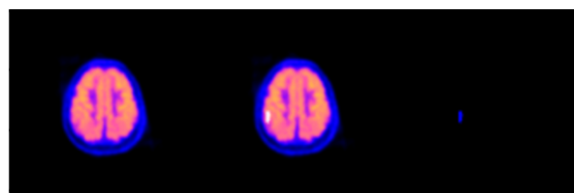
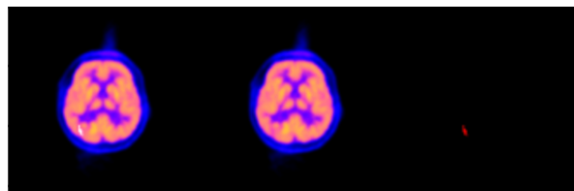


รูปจริงที่มี Label (ซ้าย), รูปที่แปลงให้มีแค่ค่า 0 และ 1 สำหรับเป็น Label จริงในการทำ segmentation (ขวา)

3. ในขั้นถัดไปเป็นการ train model โดยใช้ model U-Net ซึ่งกำหนดให้แบบแรกมี input size ขนาด (160,160,3) ที่ผ่านการแปลงสีโดย gnuplot2\_r โดยทำการ train ทั้งหมด 50 epochs ซึ่งได้ค่า training accuracy และ validation accuracy เป็น 0.99 ทั้งคู่



ตัวอย่างผลลัพธ์ของ Training set



ตัวอย่างผลลัพธ์ของ Validating set

ในแต่ละรูปประกอบไปด้วย 3 รูปย่อย ได้แก่

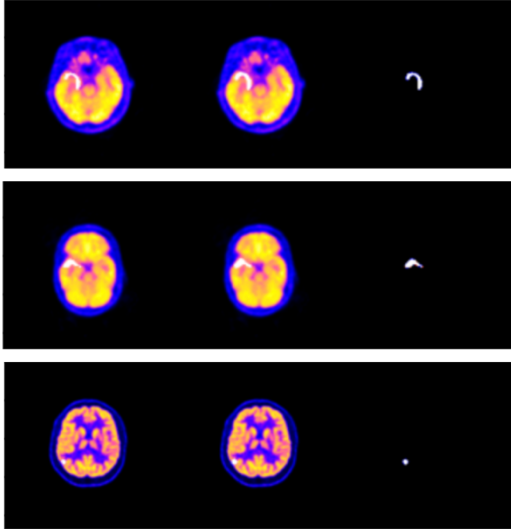
(ซ้าย) รูปสมองพร้อมกับ Label จริง

(กลาง) รูปสมองพร้อมกับ Label จาก prediction ของ model

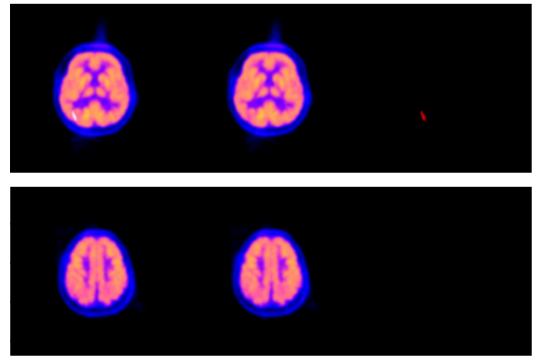
(ขวา) ความแตกต่างระหว่าง Label จริง กับ Label จาก machine โดยที่บริเวณสีขาวคือ Label ตรงกัน สีแดงคือ Label ขาดจากความเป็นจริง และสีน้ำเงินคือ Label เกินความเป็นจริง

จากผลลัพธ์ สังเกตได้ว่าถึงแม้จะมี validation accuracy ที่สูงมาก แต่เป็นเพราะ accuracy ได้มาจาก dataset ส่วนใหญ่ที่ไม่มีจุดกำเนิดชักในรูป (segmentation เป็น 0) รวมถึงรูปที่จุดกำเนิดชักเองก็มีบริเวณของจุดกำเนิดชักที่เล็กมาก สัดส่วนของบริเวณที่ mark ถูกผิดจึงต่างกันมาก และหากพิจารณาเพียงแค่ validation set พบว่าการ label ในบริเวณที่เป็นจุดกำเนิดชัก (segmentation เป็น 1) ยังไม่สามารถ label ได้อย่างถูกต้อง

4. ลองเปลี่ยน input size เป็นขนาด (224,224,3) ที่ผ่านการแปลงสีโดย `gnuplot2_r` และจำนวน epochs เป็น 50 เหมือนเดิม ได้ค่า training accuracy และ validating accuracy เป็น 0.99 เช่นเดิม



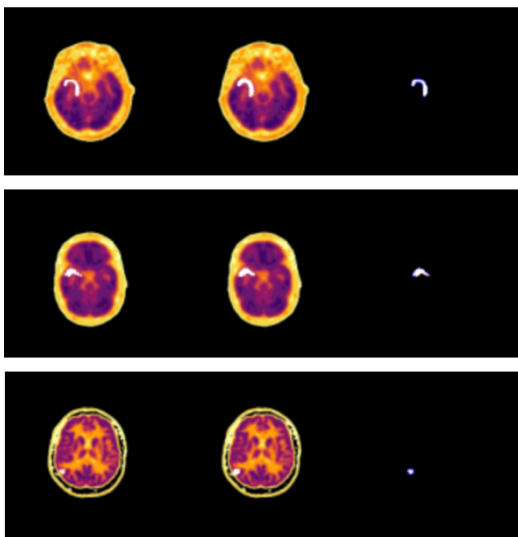
ตัวอย่างผลลัพธ์ของ Training set



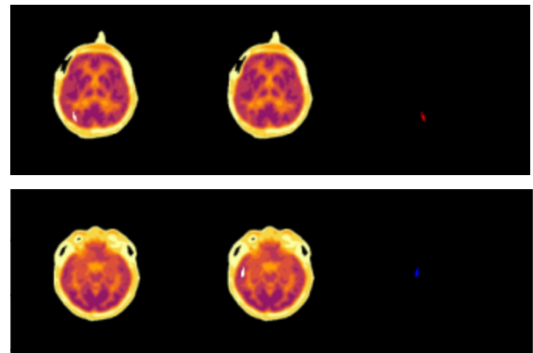
ตัวอย่างผลลัพธ์ของ Validating set

ผลลัพธ์ที่ได้เหมือนกับผลในข้อ 3 มาก โดยที่ validating set ยังไม่สามารถ label ได้อย่างถูกต้อง แต่ไม่มีการ mark เกินเหมือนในข้อ 3 ในรูปเดียวกัน รวมถึง mark ได้ตรงจริงกว่า

5. ลองเปลี่ยนเป็นใช้ input size ขนาด (160,160,3) ที่ผ่านการแปลงสีโดย 10 color mapping โดยให้จำนวน epochs เป็น 50 เช่นเดิม ได้ค่า training accuracy และ validating accuracy เป็น 0.99 เช่นกัน



ตัวอย่างผลลัพธ์ของ Training set



ตัวอย่างผลลัพธ์ของ Validating set

ผลลัพธ์ที่ได้เหมือนกับผลในข้อ 3 มาก โดยที่ validating set ยังไม่สามารถ label ได้อย่างถูกต้อง รวมถึง training set เองมีการ label เกินมากกว่าการใช้ `gnuplot2_r` และมีการ mark เกินในรูปอื่นๆ มากกว่าในข้อ 3 และ 4

จากการลอง input ทั้งแบบ `gnuplot2_r` และแบบ 10 color mapping ในแบบ `gnuplot2_r` สามารถทำได้ดีกว่า เนื่องจากผมมองว่าเป็นการ map ค่าสีขาวดำตั้งแต่ 0-255 ไปยังแถบสีซึ่งเป็นค่าต่อเนื่อง ทำให้ค่าสีทุกค่าไม่ถูกทิ้งไป ในขณะที่แบบ 10 color mapping เป็นการแบ่งค่าสีขาวดำเป็น 10 ช่วง เพื่อ map ไปยังสีที่กำหนดไว้ ทำให้ค่าสีที่มีความใกล้เคียงกันถูกมองเป็นเพียงสีเดียว ทำให้ข้อมูลตรงนี้ถูกทิ้งไปได้ ทำให้ model ทำนายผลได้แยกว่า นอกจากนี้ จากขนาด input image จริงๆที่มีขนาด (440,440,3) การ resize ลงมาย่อมทำให้ข้อมูลค่าสีบางค่าถูกทิ้งไปเช่นกัน โดยสาเหตุที่เลือกใช้ (160,160,3) และ (224,224,3) แทนที่จะ scale รูปขึ้นไปให้เป็น (448,448,3) เนื่องจากมีข้อจำกัดทางด้าน memory ของ Google Colab นอกจากนี้ยังสามารถมองเห็นถึงปัญหา overfit ได้อย่างชัดเจนเนื่องจาก model สามารถ predict ได้จากแค่ training set เท่านั้น ซึ่งการลดจำนวน epochs ลงอาจช่วยในส่วนนี้ได้

สรุปในภาพรวม ในการทดลองทำ machine learning ของผมเพื่อใช้ในการตรวจหาจุดกำเนิดชักในสมองของผู้ป่วยโดยใช้ Model U-Net ยังไม่สามารถทำได้ อาจเป็นเพราะลักษณะของจุดกำเนิดชักนั้นมีรูปแบบที่ค่อนข้างไม่ตายตัว และยังต้องมีการเปรียบเทียบระดับสีของภาพทั้ง 2 ผัง ซึ่งทำให้ factor ในการ detect จุดกำเนิดชักเหล่านี้มีเยอะมากๆ Model U-Net นี้ที่เด่นในเรื่องการทำ biomedical image segmentation จึงอาจยังไม่ตอบโจทย์มาก การใช้ Model อื่นๆ เช่น Model Siamese ที่เป็นการเปรียบเทียบรูป 2 รูป (ใช้เป็นการเปรียบเทียบสมองฝั่งซ้ายและขวา เพื่อหาจุดที่ระดับสีมีความต่าง) จึงอาจเหมาะสมกว่าในกรณีนี้

## สิ่งที่ได้เรียนรู้

จากการได้ทำ model ในการตรวจหาจุดกำเนิดชักในสมองของผู้ป่วยจากภาพถ่าย PET/CT Scan ผมได้เรียนรู้ตั้งแต่พื้นฐานของ machine learning รวมถึงแขนงย่อยต่างๆของ ML จากตั้งแต่ว่าไม่มีความรู้ทางด้านนี้เลย จนกระทั่งได้ทำความรู้จักและลองเล่นกับ model ต่างๆ จนเริ่มมีความเข้าใจในด้านนี้ นอกจากนี้เองจากการที่ได้ลงมือ train machine จริง จึงได้เรียนรู้ตั้งแต่ขั้นตอนในการเตรียม data สำหรับการ training ต่างๆ รวมถึงได้พบเจอปัญหาจริงๆระหว่างการ train model จริง ทั้งในแง่ของการ design การเลือกใช้ model และ input ของข้อมูล รวมถึงผลลัพธ์จากการ train model ซึ่ง overfit มากเกินไปจนยังไม่สามารถนำไปใช้งานได้ ซึ่งนับเป็นการเรียนรู้และประสบการณ์ที่ดีเพื่อนำไปใช้ประโยชน์ในอนาคตในภายภาคหน้าได้