Experiments

Pre-processing

|  |  |
| --- | --- |
| Data split | Test ตามที่กำหนด แบ่ง Train เป็น Train/Validation แบบ random seed 80:20 |
| Sampling rate | 16 kHz |
| Clipping | เลือกเฉพาะ 70 วินาทีแรก |
| Padding | ไฟล์ที่สั้นกว่า 70 วินาทีนั้นเติม 0 |
| Diarization | เลือกตัดเสียงเฉพาะ participant โดยใช้ speaker diarization ของ pyannote |
| Feature extraction | Mel-frequency cepstral coefficients (MFCCs) 13 coefficients  Audio Augmentation (noise,pitch,specaugment) |

Algorithms

|  |  |
| --- | --- |
| Algorithm | Convolutional neural network 2d |
| Parameter set | **Conv2d layer** – filter=4, kernel size=(3,3), activation function = relu , l1 regularization (alpha = 0.02)  **BatchNormalization layer**  **MaxPooling2D layer** – size 2,2  **Conv2d layer** – filter=32, kernel\_size=(3,3), activation function = relu , elastic net regularization (alpha = 0.01)  **BatchNormalization layer**  **MaxPooling2D layer** – size 2,2  **Flatten layer**  **Dropout layer** – rate = 0.3  **Dense layer** – units=32, activation function = relu, l2 regularization (alpha = 0.02)  **Dense layer** – units=1, activation function = sigmoid  **Optimizer** = sgd  **Loss function** = binary\_crossentropy |
| Other settings | Incremental learning batch size=64 จำนวน 100 epoch  Learning rate decay – patience = 3, factor = 0.5 based on val\_accuracy  Early stopping – mode = auto, patience = 30, restore best weight based on val\_accuracy  (Train on x\_train, y\_train)  (Validation data=x\_test, y\_test) |
| Result | |  |  |  | | --- | --- | --- | |  | Actual Control | Actual AD | | Predicted Control | 18 | 2 | | Predicted AD | 3 | 17 |   Accuracy = 87.5% |

Cross validation result

ใช้ทั้ง dataset ทำ 5-fold cross validation แต่ใช้ parameter set เดียวกับอันก่อนหน้า แล้วรายงาน accuracy เฉลี่ยของทุก fold

**Best Average Accuracy**

|  |  |
| --- | --- |
| Algorithm | Convolutional neural network 2d |
| Parameter set | **Conv2d layer** – filter=4, kernel size=(3,3), activation function = relu , l1 regularization (alpha = 0.02)  **BatchNormalization layer**  **MaxPooling2D layer** – size 2,2  **Conv2d layer** – filter=32, kernel\_size=(3,3), activation function = relu , elastic net regularization (alpha = 0.01)  **BatchNormalization layer**  **MaxPooling2D layer** – size 2,2  **Flatten layer**  **Dropout layer** – rate = 0.3  **Dense layer** – units=32, activation function = relu, l2 regularization (alpha = 0.02)  **Dense layer** – units=1, activation function = sigmoid  **Optimizer** = sgd  **Loss function** = binary\_crossentropy |
| Other settings | Incremental learning batch size=64 จำนวน 100 epoch  Learning rate decay – patience = 3, factor = 0.5 based on val\_accuracy  Early stopping – mode = auto, patience = 30, restore best weight based on val\_accuracy  (Train on x\_train[train], y\_train[train])  (Validation data=x\_train[test], y\_train[test]) |
| Result | Accuracy เฉลี่ยของทุก fold = 75.0%, 82.5%, 90%, 80%, 82.5% ~ 82% |

**Top Accuracy**

|  |  |
| --- | --- |
| Algorithm | Convolutional neural network 2d |
| Parameter set | **Conv2d layer** – filter=16, kernel size=(3,3), activation function = relu , l1 regularization (alpha = 0.02)  **BatchNormalization layer**  **MaxPooling2D layer** – size 2,2  **Conv2d layer** – filter=8, kernel\_size=(3,3), activation function = relu , elastic net regularization (alpha = 0.02)  **BatchNormalization layer**  **MaxPooling2D layer** – size 2,2  **Flatten layer**  **Dropout layer** – rate = 0.3  **Dense layer** – units=8, activation function = relu, l2 regularization (alpha = 0.02)  **Dense layer** – units=1, activation function = sigmoid  **Optimizer** = sgd  **Loss function** = binary\_crossentropy |
| Other settings | Same as the above |
| Result | Accuracy เฉลี่ยของทุก fold = 95%, 70%, 72.5%, 67.5%, 72.5% ~ 75.5% |

Note/Reference

<https://www.kaggle.com/puntwitty/adress-ml-mfcconly-tawatwee-indiv> (For training one model at a time)

<https://www.kaggle.com/code/twittypun/runningml> (For doing cross-validation and grid-search)

<https://github.com/PunnyOz2/MachineLearning_SummerChula/blob/main/gridSearch_result.json>

(grid-search result)