三上專題

Implementation of Decoupling Convolutional Neural Network for Intelligent Compound Fault Diagnosis

目錄

1. 專題實作

在實作上，我嘗試解決的問題是：單一 label 的 CNN 分類模型不一定能夠反應機器的準確情況，如在機器上的各個部分都有可能出現故障，其單一和複合故障的程度不一，這會影響到機器系統的維護策略和預測。因此我的專題目標為實現可應對輸入輸出多 label 的 CNN 分類模型用於機器偵測訊號。

採用的輸入數據為實驗室學長在實驗室中提取出來的馬達震動訊號，下面是使用的馬達系統：

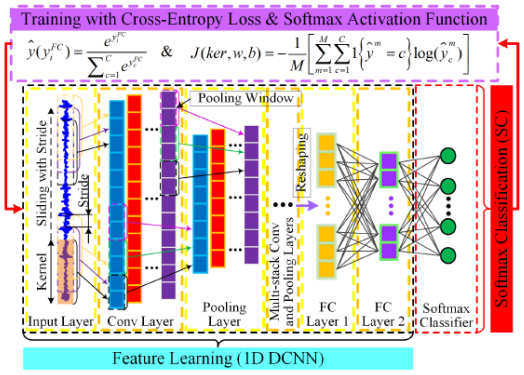


Fig. 1. Structure of 1D Deep-CNN

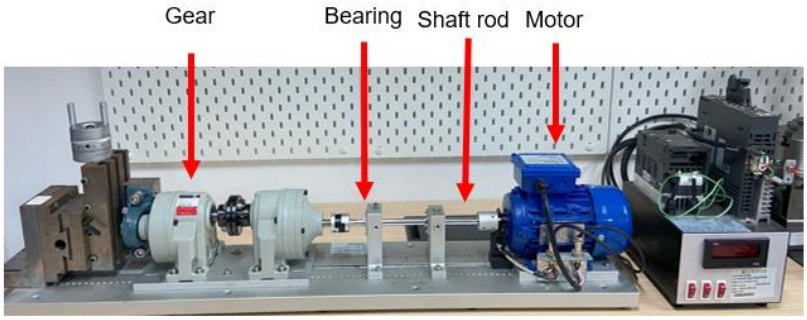


Fig. 2. Motor for getting data

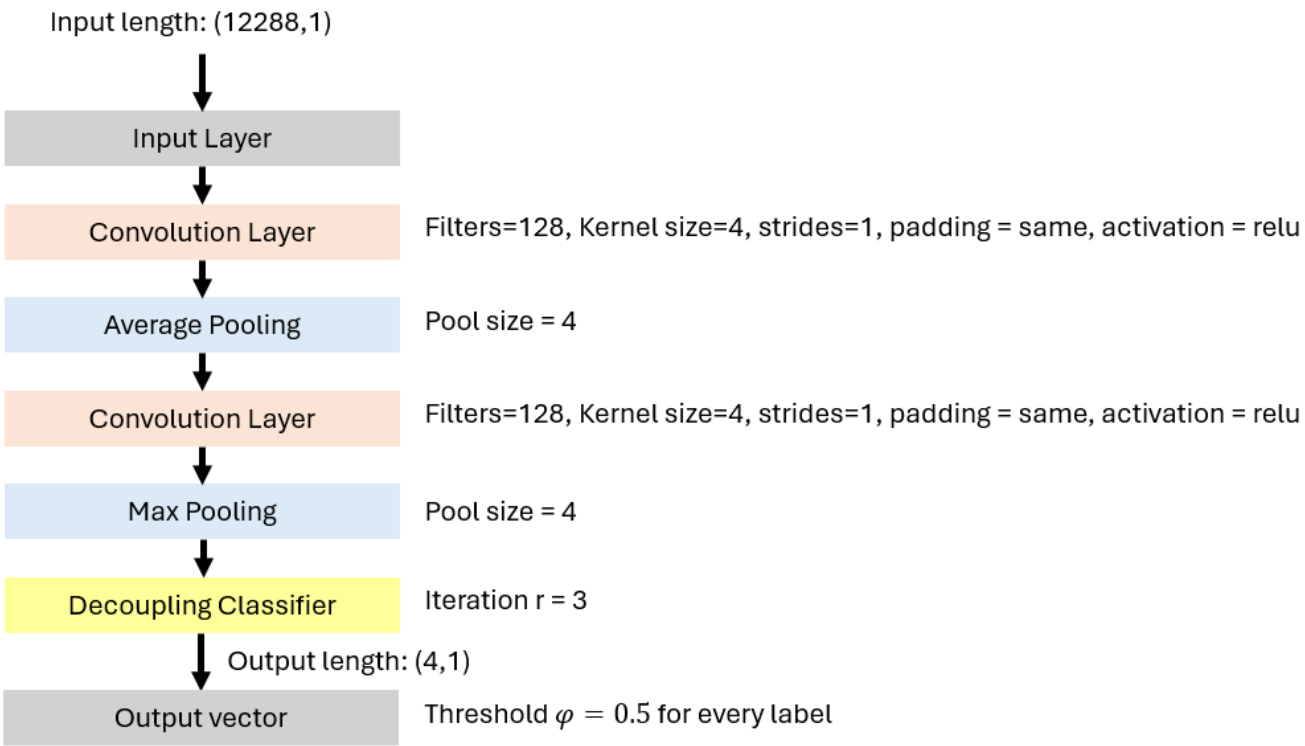
實驗上我們主要測試齒輪(Gear)和軸承(Bearing)在這個系統上對馬達產生訊號的影響。

在收集數據中所使用的取樣頻率為$f_s = 10kHz$，每筆資料為 2048*6=12288 個取樣點。下面是採集到的每筆資料數量：

"齒輪正常、軸承正常"	171 筆
"齒輪磨耗、軸承正常"	181 筆
"齒輪不對中、軸承正常"	178 筆
"齒輪正常、軸承內斷"	176 筆
"齒輪不對中、軸承內斷"	174 筆
"齒輪磨耗、軸承內斷"	169 筆

由上面的分類能夠得到，6 類資料可被分為系統正常(label 0)，以及三種錯誤(label 1-3)對應齒輪磨耗、齒輪不對中以及軸承內斷。

對比參考論文的架構，因為資料集數量較少，對比參考論文每種訊號都有(訓練+測試) = 715+715 筆資料，所以只能考慮縮小模型。在實作我根據參考論文[1]所搭建的 CNN 模型如下：



[1] Huang, R., Liao, Y., Zhang, S., & Li, W. (2019). Deep decoupling convolutional neural network for intelligent compound fault diagnosis. IEEE Access, 7, 1848-1858.

TABLE 3. The parameters of DDCNN.

Layer Type	Activation Function	Parameter Name	Parameter Size	Output Size
Input	/	/	/	(8192,1)
Conv1d_1	ReLU	Kernels	256×64×16	(256,512,1)
Conv1d_2	ReLU	Kernels	256×3×1	(256,512,1)
Average Pooling	/	Pooling size	2	(256,256,1)
Conv1d_3	ReLU	Kernels	128×3×1	(128,256,1)
Conv1d_4	ReLU	Kernels	128×3×1	(128,256,1)
Max Pooling	/	Pooling size	2	(128,128,1)
Reshape	/	/	/	(128,128)
DC_1	squash	Vectors	32×128	(32,128)
DC_2	squash	Vectors	3×32	(3,32)
L2 norm	/	/	/	(3,1)

Fig. 3. CNN model comparison

在論文所使用的 Decoupling classifier，其演算法如下：

TABLE 1. The algorithm of decoupling classifier.

Algorithm: The decoupling classifier

Input: The reshaped feature $\boldsymbol{y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_{R_l}]^T\in\Re^{K_l\times R_l}$ obtained from the last pooling layer, the number of classes C, the number of iteration r, the threshold φ, the weight tensor $\boldsymbol{W}\in\Re^{C\times K_l\times R_d\times R_l}$ which can be optimized by the backpropagation algorithm, $i=1,2,\cdots,K_l$, $j=1,2,\cdots,C$.

Output: The predicted labels

Initialization: for all $b_{ij}\leftarrow 0$

For r iterations **do**

for i in $[1,2,\cdots,K_l]$: $\boldsymbol{c}_i\leftarrow softmax(\boldsymbol{b}_i)$ (10)

for i in $[1,2,\cdots,K_l]$ and j in $[1,2,\cdots,C]$: $\hat{\boldsymbol{y}}_{j|i}\leftarrow\boldsymbol{W}_{ij}\boldsymbol{y}_i$ (9)

for i in $[1,2,\cdots,K_l]$ and j in $[1,2,\cdots,C]$: $\boldsymbol{d}_j\leftarrow\sum_i^{K_l}c_{ij}\hat{\boldsymbol{y}}_{j|i}$ (8)

for j in $[1,2,\cdots,C]$: $\boldsymbol{v}_j\leftarrow squash(\boldsymbol{d}_j)$ (11)

for i in $[1,2,\cdots,K_l]$ and j in $[1,2,\cdots,C]$: $b_{ij}\leftarrow b_{ij}+<\boldsymbol{v}_j,\ \hat{\boldsymbol{y}}_{j|i}>$

Return v_j

L2 norm: $\boldsymbol{y}_{pred}\leftarrow\|\boldsymbol{v}\|_2$

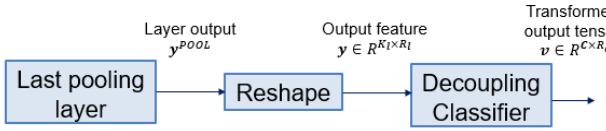
If $y^i_{pred}\geq\varphi$

return i-th class label as 1

else

return i-th class label as 0

Return The predicted labels



$$v_j=squash(\boldsymbol{d}_j)=\frac{\|\boldsymbol{d}_j\|^2}{1+\|\boldsymbol{d}_j\|^2}\frac{\boldsymbol{d}_j}{\|\boldsymbol{d}_j\|}$$

Fig. 4. CNN Decoupling classifier algorithm

其作用是取代單一 label 中的 output classifier，透過$\boldsymbol{r}$次 iteration 迭代更新 $\boldsymbol{W}$和$\boldsymbol{b}$，並透過 squash function 修正過強特徵，實現多 label 的特徵判斷和輸出。

另外新的 Classifier 需要搭配新的 Margin loss function，其中$m^+=0.9$, $m^-=0.1$, $\lambda=0.25$：

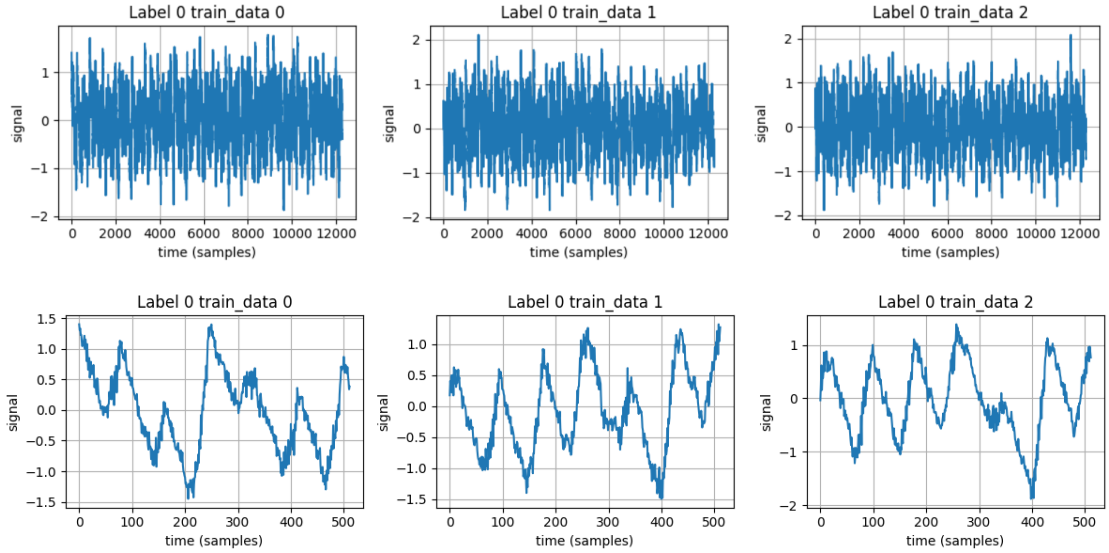
$$J=\sum_{c=1}^{C}L_c=\sum_{c=1}^{C}\{T_c\max(0,m^+-\hat{y}_c)^2+\lambda(1-T_c)\max(0,\hat{y}_c-m^-)^2\}$$

Fig. 5. New Loss function for decoupling classifier

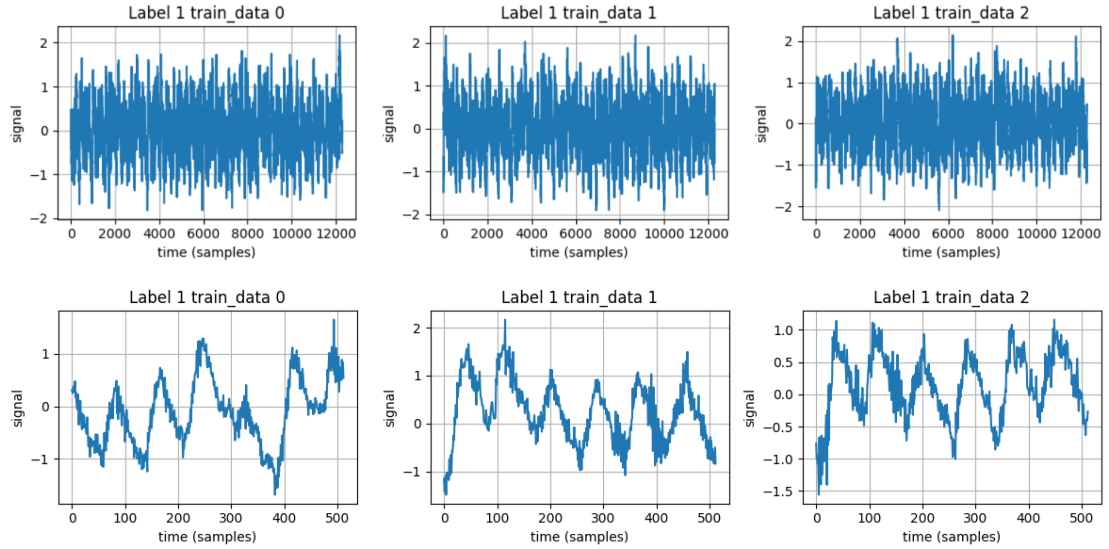
2. 實驗結果與分析

在實作上，我在 Colab 環境下使用 python 搭建 CNN 模型以及 Decoupling classifier，訓練的 epoch=20 / 30 (當複合故障資料用作訓練時)，batch size=32，使用 Adam optimizer 和其默認參數；首先對每種馬達訊號隨機挑 3 筆做視覺化：

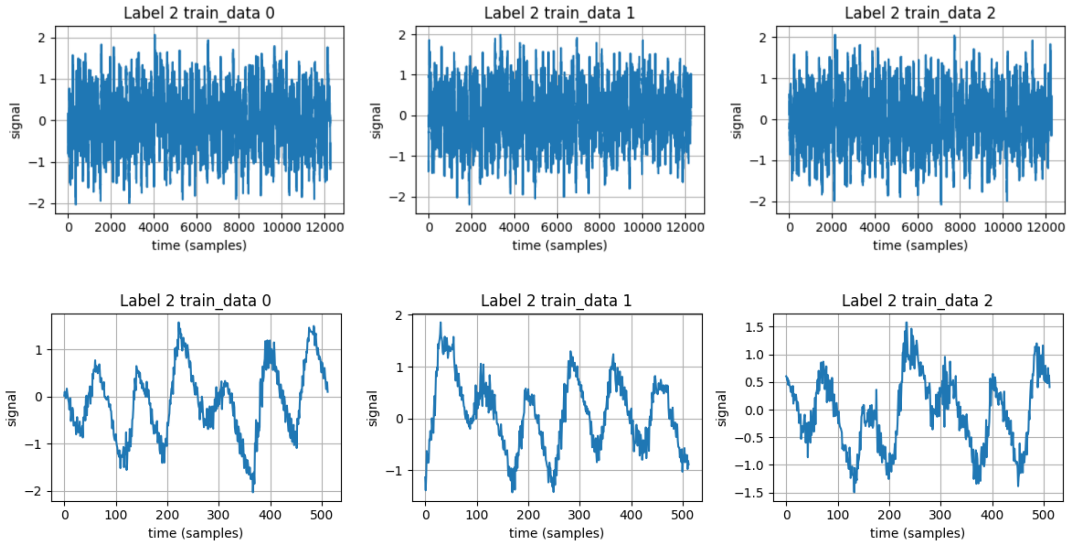
"齒輪正常、軸承正常"：



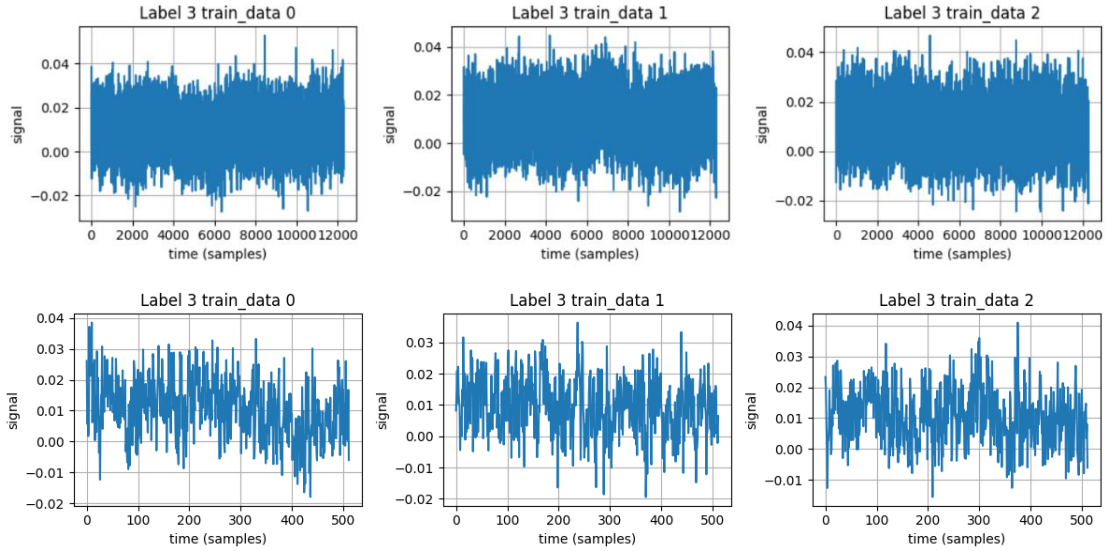
"齒輪磨耗、軸承正常"：



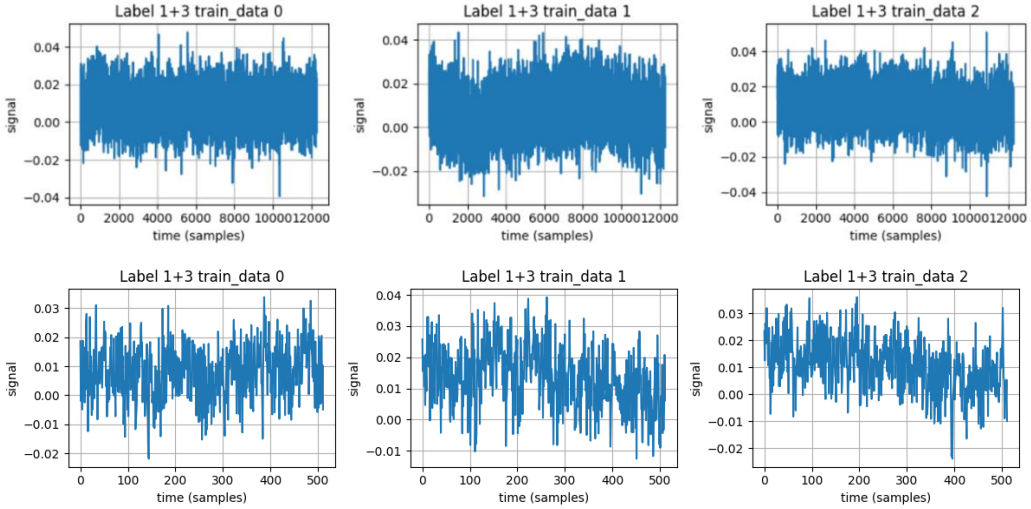
"齒輪不對中、軸承正常"：



"齒輪正常、軸承內斷"：



"齒輪不對中、軸承內斷"：



"齒輪磨耗、軸承內斷"：

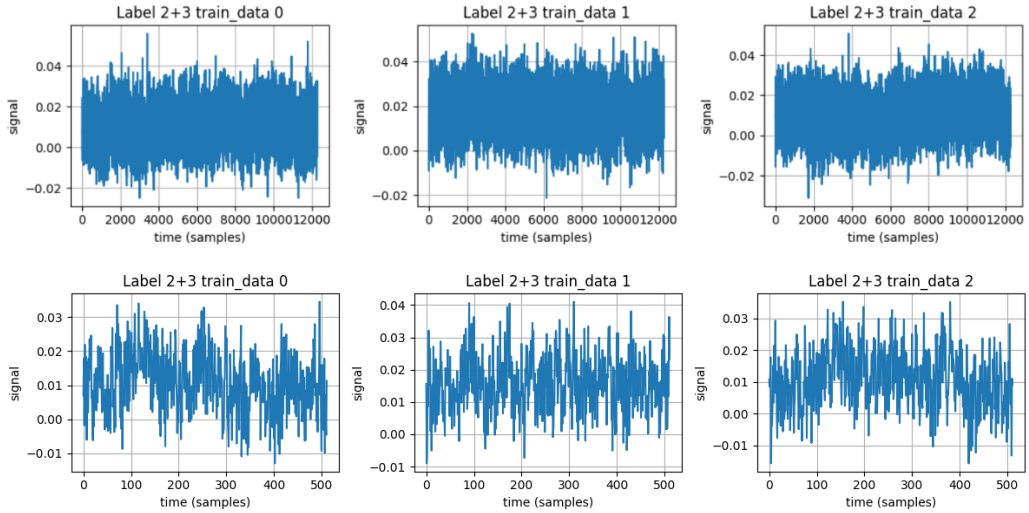


Fig. 6. Different types of input signal graph

可以看到在齒輪故障時，訊號的較正常馬達訊號會變「粗」，而軸承內斷時，則訊號波形會變成不規則的雜訊。因此也可以看到軸承內斷對整個系統的影響比齒輪故障大得多。

下面是模型訓練後的結果，資料分配方面若參加訓練則七成資料用作訓練，三成用作測試，參考論文對於複合資料，全部用於測試而沒有提前訓練。

具體分配方式如下，故不同訓練集下每種類別數量有所不同：

1) 所有類型都用作訓練和測試：

```
# 分割數據集
X_train, X_test, y_train, y_test = train_test_split(data_df, multi_label_matrix, test_size=0.3, random_state=42)
```

2) 參考論文方法 - 複合資料全部用於測試而沒有提前訓練：

```
# 單一標籤資料進行 73 分割
X_train, X_test_single, y_train, y_test_single = train_test_split(
    single_label_data, single_label_labels, test_size=0.3, random_state=42
)

# 複合標籤資料直接作為測試集
X_test_multi = multi_label_data
y_test_multi = multi_label_labels
```

Fig. 7. Data Preprocessing

先測試模型在單一 label 下的效能：

```
# 定義資料夾和標籤（多標籤）
folders = ["齒輪不對中、軸承內斷", "齒輪不對中、軸承正常", "齒輪正常、軸承內斷", "齒輪正常、軸承正常", "齒輪磨耗、軸承內斷", "齒輪磨耗、軸承正常"]
labels_map = {
    # 都正常 - 0 齒輪磨耗 - 1  齒輪不對中 - 2  軸承內斷 - 3
    "齒輪正常、軸承正常": [0],
    "齒輪磨耗、軸承正常": [1],
    "齒輪不對中、軸承正常": [2],
    "齒輪正常、軸承內斷": [3],
    #"齒輪磨耗、軸承內斷": [1, 3],
    #"齒輪不對中、軸承內斷": [2, 3]
}
```

```
Test Data Accuracy by Folder:
  齒輪不對中、軸承正常: 44 test samples, 100.00% accuracy
  齒輪正常、軸承正常: 61 test samples, 100.00% accuracy
  齒輪磨耗、軸承正常: 58 test samples, 100.00% accuracy
  齒輪正常、軸承內斷: 49 test samples, 100.00% accuracy
Number of successful predictions: 212 out of 212
Overall Accuracy: 100.00%
```

Fig. 8. Testing model with only single label

CNN 模型有效地識別 4 個單一 label 的馬達訊號。

如果在原本模型基礎上，用 multi-label 作測試：

```
# 定義資料夾和標籤（多標籤）
folders = ["齒輪不對中、軸承內斷", "齒輪不對中、軸承正常", "齒輪正常、軸承內斷", "齒輪正常、軸承正常", "齒輪磨耗、軸承內斷", "齒輪磨耗、軸承正常"]
labels_map = {
    # 都正常 - 0 齒輪磨耗 - 1  齒輪不對中 - 2  軸承內斷 - 3
    "齒輪正常、軸承正常": [0],
    "齒輪磨耗、軸承正常": [1],
    "齒輪不對中、軸承正常": [2],
    "齒輪正常、軸承內斷": [3],
    "齒輪磨耗、軸承內斷": [1, 3],
    "齒輪不對中、軸承內斷": [2, 3]
}
```

```
Test Data Accuracy by Folder:
  齒輪正常、軸承正常: 42 test samples, 100.00% accuracy
  齒輪不對中、軸承正常: 63 test samples, 95.24% accuracy
  齒輪正常、軸承內斷: 56 test samples, 100.00% accuracy
  齒輪磨耗、軸承正常: 51 test samples, 100.00% accuracy
  齒輪不對中、軸承內斷: 169 test samples, 0.00% accuracy
  齒輪磨耗、軸承內斷: 174 test samples, 0.00% accuracy
Number of successful predictions: 209 out of 555
Overall Accuracy: 37.66%
```

```
Test Data 10:
  Original Labels: [0 1 0 1]
  Predicted Labels: [0 0 0 1]
  Result: Incorrect

Test Data 28:
  Original Labels: [0 0 1 1]
  Predicted Labels: [0 0 0 1]
  Result: Incorrect
Test Data 29:
  Original Labels: [0 0 1 0]
  Predicted Labels: [0 0 1 0]
  Result: Correct
Test Data 30:
  Original Labels: [0 0 1 0]
  Predicted Labels: [0 0 1 0]
  Result: Correct
```

Fig. 9. Testing model with single label training + multi label testing

模型的效能並不太理想，但可以發現實際上是因為軸承內斷對馬達訊號影響太大，在沒有給複合資料作訓練的情況下，模型難以做多 label 的判斷。

(下面的訓練 epoch 為 30)

接著測試如果把複合資料加入訓練時的情況，但假設訊號之間沒有關聯：

```
# 定義資料夾和標籤（多標籤）
folders = ["齒輪不對中、軸承內斷", "齒輪不對中、軸承正常", "齒輪正常、軸承內斷", "齒輪正常、軸承正常", "齒輪磨耗、軸承內斷", "齒輪磨耗、軸承正常"]
labels_map = {
        # 都正常 - 0 齒輪磨耗 - 1   齒輪不對中 - 2   軸承內斷 - 3
        "齒輪正常、軸承正常": [0],
        "齒輪磨耗、軸承正常": [1],
        "齒輪不對中、軸承正常": [2],
        "齒輪正常、軸承內斷": [3],
        "齒輪磨耗、軸承內斷": [4],
        "齒輪不對中、軸承內斷": [5]
}
```

```
Test Data 303:
  Original Labels: [0 1 0 1]
  Predicted Labels: [0 1 0 1]
  Result: Correct
Test Data 304:
  Original Labels: [0 0 1 1]
  Predicted Labels: [0 0 1 1]
  Result: Correct
Test Data 305:
  Original Labels: [0 0 1 0]
  Predicted Labels: [0 0 1 0]
  Result: Correct
Test Data 306:
  Original Labels: [0 0 0 1]
  Predicted Labels: [0 1 0 1]
  Result: Incorrect
Test Data 307:
  Original Labels: [0 1 0 0]
  Predicted Labels: [0 1 0 0]
  Result: Correct
```

```
Test Data 304:
  Original Labels: [0 1 0 0 0 0]
  Predicted Labels: [0 1 0 0 0 0]
  Result: Correct
Test Data 305:
  Original Labels: [0 0 0 1 0 0]
  Predicted Labels: [0 0 0 1 1 1]
  Result: Incorrect
Test Data 306:
  Original Labels: [0 0 0 0 0 1]
  Predicted Labels: [0 0 0 1 0 1]
  Result: Incorrect
Test Data 307:
  Original Labels: [0 0 1 0 0 0]
  Predicted Labels: [0 0 1 0 0 0]
  Result: Correct
```

```
Test Data Accuracy by Folder:
  齒輪正常、軸承內斷: 53 test samples, 0.00% accuracy
  齒輪磨耗、軸承正常: 54 test samples, 100.00% accuracy
  齒輪不對中、軸承內斷: 53 test samples, 66.04% accuracy
  齒輪不對中、軸承正常: 61 test samples, 95.08% accuracy
  齒輪磨耗、軸承內斷: 40 test samples, 0.00% accuracy
  齒輪正常、軸承正常: 54 test samples, 100.00% accuracy
Number of successful predictions: 201 out of 315
Overall Accuracy: 63.81%
```

Fig. 10. Testing model with (single + multi) label training

可以看到模型這時出現了多 label 的預測。說明在這個場景中，加入複合資料作訓練令模型有分類多 label 的能力。

最後根據上面的觀察，我把多類別資料也加入到訓練當中，並採用複合類別的方式 label 訓練資料：

```
labels_map = {
        # 都正常 - 0 齒輪磨耗 - 1   齒輪不對中 - 2   軸承內斷 - 3
        "齒輪正常、軸承正常": [0],
        "齒輪磨耗、軸承正常": [1],
        "齒輪不對中、軸承正常": [2],
        "齒輪正常、軸承內斷": [3],
        "齒輪磨耗、軸承內斷": [1, 3],
        "齒輪不對中、軸承內斷": [2, 3]
}
```

```
Test Data Accuracy by Folder:
  齒輪不對中、軸承正常: 56 test samples, 100.00% accuracy
  齒輪不對中、軸承內斷: 51 test samples, 74.51% accuracy
  齒輪磨耗、軸承內斷: 37 test samples, 100.00% accuracy
  齒輪正常、軸承正常: 53 test samples, 100.00% accuracy
  齒輪磨耗、軸承正常: 63 test samples, 100.00% accuracy
  齒輪正常、軸承內斷: 55 test samples, 30.91% accuracy
Number of successful predictions: 264 out of 315
Overall Accuracy: 83.81%
```

Fig. 11. Muti-label for training and testing

這時模型的總預測率達到 83.81%，且在面對複合故障的訊號大部分能正確辨別，說明模型有效地實現多 label 輸入、輸出的效果。

另外注意到在實際應用下，如果單一故障訊號被辨別為，包含該類別的複合故障訊號，這並不是絕對的錯誤。我們應該保守地認為這個機器系統有潛在的其他問題，於是將這個考量代入到最終的模型準確率計算：

```
----- 準確率評估報告 -----
嚴格成功預測數量 (完全匹配): 264 / 315
嚴格整體準確率 (Strict Accuracy): 83.81%
--------------------------------
可接受的成功預測數量 (包含部分正確): 302 / 315
可接受的整體準確率 (Acceptable Accuracy): 95.87%

----- 各資料夾可接受準確率 -----
  齒輪正常、軸承正常: 53 test samples, 100.00% accuracy
  齒輪磨耗、軸承正常: 63 test samples, 100.00% accuracy
  齒輪不對中、軸承正常: 56 test samples, 100.00% accuracy
  齒輪正常、軸承內斷: 55 test samples, 100.00% accuracy
  齒輪磨耗、軸承內斷: 37 test samples, 100.00% accuracy
  齒輪不對中、軸承內斷: 51 test samples, 74.51% accuracy
```

Fig. 12. More suitable accuracy

總結來說，參考論文的架構並不完全適用於本次實驗場景，不過根據論文的架構，在實驗室的馬達系統場景下，我**成功復現**與參考論文等效的 CNN 架構和多類別輸入輸出分類任務，且能兼顧單一 label 分類時的情景。