

# 智慧化企業整合

英語手語字母辨識實驗

以 MNIST 公開資料為例

指導教授：邱銘傳 教授

學生：109034537 邱靖中

中華民國 110 年 一 月 七 日

## 目錄

<b>1 研究背景</b> .....	<b>1</b>
<b>2 研究目的</b> .....	<b>1</b>
<b>3 公開資料集介紹</b> .....	<b>2</b>
<b>4 研究方法</b> .....	<b>3</b>
4.1 Python 視覺化工具 Seaborn .....	4
4.2 One-hot Encoding .....	4
4.3 資料標準化 .....	4
4.4 資料增強 .....	4
4.5 CNN 原理 .....	5
<b>5 研究流程</b> .....	<b>5</b>
<b>6.結論</b> .....	<b>12</b>
6.1 研究結果 .....	12
6.1.1 公開資料準確度高 .....	12
6.1.2 外來照片準確度問題 .....	12
6.2 未來發展 .....	12
6.2.1 蒐集不同資料集 .....	12
6.2.2 結合手掌捕捉技術 .....	12

## 圖目錄

圖 1 美國手語示意圖 .....	1
圖 2 手語字母辨識流程示意圖 .....	2
圖 3 訓練與測試資料示意圖 .....	2
圖 4 訓練與測試資料視覺化示意圖 .....	3
圖 5 研究架構圖 .....	3
圖 6 CNN 演算法流程示意圖 .....	5
圖 7 套件匯入程式碼示意圖 .....	5
圖 8 資料匯入與合併程式碼示意圖 .....	6
圖 9 資料視覺化程式碼示意圖 .....	6
圖 10 資料視覺化示意圖 .....	6
圖 11 資料分割程式碼與各資料集筆數示意圖 .....	7
圖 12 One-hot Encoding 程式碼示意圖 .....	7
圖 13 One-hot Encoding 結果示意圖 .....	7
圖 14 資料維度轉換程式碼 .....	8
圖 15 標準化程式碼示意圖 .....	8
圖 16 標準化結果示意圖 .....	8
圖 17 資料增強設定程式碼示意圖 .....	9
圖 18 CNN 預設模型架構 .....	9

## 表目錄

表 1	Activation Function 評估比較表 .....	10
表 2	Batch Size 評估比較表 .....	10
表 3	Learning Rate 評估比較表 .....	10
表 4	Loss Function 評估比較表 .....	11

## 1 研究背景

在社會中，聾啞人士往往淪為容易被忽略的一群。由於部分器官受損，幼時的他們無法如常人般，藉由反覆聆聽與模仿來訓練發音與咬字，所說的話語也因此經常難以被現有的語音辨識系統所辨識，雖然聾啞人士能夠藉由書寫進行溝通，但若是面對面交流，快速的反應與回饋的需求下使手語成為他們不可或缺的技能。近年來，人工智慧的興起，透過機器學習和深度學習，手語動作就可以和已知的手勢相比對，從字母和數字的手語符號對應到所指語言，而這項技術將為聾啞族群帶來極大的好處。



圖 1 美國手語示意圖

## 2 研究目的

本實驗欲以 MNIST 美國手語照片公開資料集建立手語字母分類模型，運用卷積神經網路 (CNN) 架構實作手語字母分類模型。冀以此模型之實作練習影像處理與 CNN 演算法之模型建立，對於人工智慧於聽障溝通與相關醫療運用之貢獻進行更進一步地瞭解。手語字母辨識流程示意圖如圖 2。

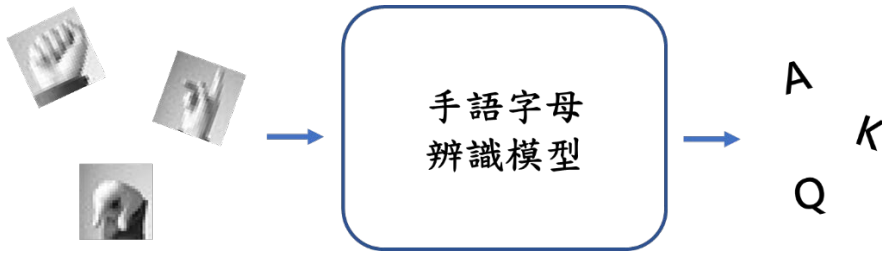


圖 2 手語字母辨識流程示意圖

### 3 公開資料集介紹

本實驗採取 MNIST 美國手語照片資料集進行手語辨識分類實驗，本資料集以 CSV 格式呈現，如圖 3 所示，左欄以 0~24 代表英文字母 A~Y，而 J 和 Z 由於手勢鑑別度不高並沒有收入致這項資料集中，因此若字母 I 所對應的數字是 9，而後續的 K 所對應的數據會是 10。資料集中特徵值的呈現方式為像素值，每張照片皆以 784 像素之單維陣列表現出來。預設的公開資料將訓練資料集與測試資料集分為兩個檔案，訓練資料集共 27,455 筆，而測試資料集共 7172 筆。圖三為 CSV 檔資料視覺化的示意圖，每張照片皆為灰階圖，並以解析度 28 x 28 像素值呈現，如圖 4 所示。

目標值	特徵值																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	pixel10	pixel11	pixel12	pixel13	pixel14	pixel15	pixel16	pixel17	pixel18	pixel19	pixel20	pixel21	pixel22	pixel23	pixel24	pixel25	pixel26	pixel27	pixel28	pixel29	pixel30	pixel31	pixel32	pixel33	pixel34	pixel35	pixel36	pixel37	pixel38	pixel39	pixel40	pixel41	pixel42	pixel43	pixel44	pixel45	pixel46	pixel47	pixel48	pixel49	pixel50	pixel51	pixel52	pixel53	pixel54	pixel55	pixel56	pixel57	pixel58	pixel59	pixel60	pixel61	pixel62	pixel63	pixel64	pixel65	pixel66	pixel67	pixel68	pixel69	pixel70	pixel71	pixel72	pixel73	pixel74	pixel75	pixel76	pixel77	pixel78	pixel79	pixel80	pixel81	pixel82	pixel83	pixel84	pixel85	pixel86	pixel87	pixel88	pixel89	pixel90	pixel91	pixel92	pixel93	pixel94	pixel95	pixel96	pixel97	pixel98	pixel99	pixel100	pixel101	pixel102	pixel103	pixel104	pixel105	pixel106	pixel107	pixel108	pixel109	pixel110	pixel111	pixel112	pixel113	pixel114	pixel115	pixel116	pixel117	pixel118	pixel119	pixel120	pixel121	pixel122	pixel123	pixel124	pixel125	pixel126	pixel127	pixel128	pixel129	pixel130	pixel131	pixel132	pixel133	pixel134	pixel135	pixel136	pixel137	pixel138	pixel139	pixel140	pixel141	pixel142	pixel143	pixel144	pixel145	pixel146	pixel147	pixel148	pixel149	pixel150	pixel151	pixel152	pixel153	pixel154	pixel155	pixel156	pixel157	pixel158	pixel159	pixel160	pixel161	pixel162	pixel163	pixel164	pixel165	pixel166	pixel167	pixel168	pixel169	pixel170	pixel171	pixel172	pixel173	pixel174	pixel175	pixel176	pixel177	pixel178	pixel179	pixel180	pixel181	pixel182	pixel183	pixel184	pixel185	pixel186	pixel187	pixel188	pixel189	pixel190	pixel191	pixel192	pixel193	pixel194	pixel195	pixel196	pixel197	pixel198	pixel199	pixel200	pixel201	pixel202	pixel203	pixel204	pixel205	pixel206	pixel207	pixel208	pixel209	pixel210	pixel211	pixel212	pixel213	pixel214	pixel215	pixel216	pixel217	pixel218	pixel219	pixel220	pixel221	pixel222	pixel223	pixel224	pixel225	pixel226	pixel227	pixel228	pixel229	pixel230	pixel231	pixel232	pixel233	pixel234	pixel235	pixel236	pixel237	pixel238	pixel239	pixel240	pixel241	pixel242	pixel243	pixel244	pixel245	pixel246	pixel247	pixel248	pixel249	pixel250	pixel251	pixel252	pixel253	pixel254	pixel255	pixel256	pixel257	pixel258	pixel259	pixel260	pixel261	pixel262	pixel263	pixel264	pixel265	pixel266	pixel267	pixel268	pixel269	pixel270	pixel271	pixel272	pixel273	pixel274	pixel275	pixel276	pixel277	pixel278	pixel279	pixel280	pixel281	pixel282	pixel283	pixel284	pixel285	pixel286	pixel287	pixel288	pixel289	pixel290	pixel291	pixel292	pixel293	pixel294	pixel295	pixel296	pixel297	pixel298	pixel299	pixel300	pixel301	pixel302	pixel303	pixel304	pixel305	pixel306	pixel307	pixel308	pixel309	pixel310	pixel311	pixel312	pixel313	pixel314	pixel315	pixel316	pixel317	pixel318	pixel319	pixel320	pixel321	pixel322	pixel323	pixel324	pixel325	pixel326	pixel327	pixel328	pixel329	pixel330	pixel331	pixel332	pixel333	pixel334	pixel335	pixel336	pixel337	pixel338	pixel339	pixel340	pixel341	pixel342	pixel343	pixel344	pixel345	pixel346	pixel347	pixel348	pixel349	pixel350	pixel351	pixel352	pixel353	pixel354	pixel355	pixel356	pixel357	pixel358	pixel359	pixel360	pixel361	pixel362	pixel363	pixel364	pixel365	pixel366	pixel367	pixel368	pixel369	pixel370	pixel371	pixel372	pixel373	pixel374	pixel375	pixel376	pixel377	pixel378	pixel379	pixel380	pixel381	pixel382	pixel383	pixel384	pixel385	pixel386	pixel387	pixel388	pixel389	pixel390	pixel391	pixel392	pixel393	pixel394	pixel395	pixel396	pixel397	pixel398	pixel399	pixel400	pixel401	pixel402	pixel403	pixel404	pixel405	pixel406	pixel407	pixel408	pixel409	pixel410	pixel411	pixel412	pixel413	pixel414	pixel415	pixel416	pixel417	pixel418	pixel419	pixel420	pixel421	pixel422	pixel423	pixel424	pixel425	pixel426	pixel427	pixel428	pixel429	pixel430	pixel431	pixel432	pixel433	pixel434	pixel435	pixel436	pixel437	pixel438	pixel439	pixel440	pixel441	pixel442	pixel443	pixel444	pixel445	pixel446	pixel447	pixel448	pixel449	pixel450	pixel451	pixel452	pixel453	pixel454	pixel455	pixel456	pixel457	pixel458	pixel459	pixel460	pixel461	pixel462	pixel463	pixel464	pixel465	pixel466	pixel467	pixel468	pixel469	pixel470	pixel471	pixel472	pixel473	pixel474	pixel475	pixel476	pixel477	pixel478	pixel479	pixel480	pixel481	pixel482	pixel483	pixel484	pixel485	pixel486	pixel487	pixel488	pixel489	pixel490	pixel491	pixel492	pixel493	pixel494	pixel495	pixel496	pixel497	pixel498	pixel499	pixel500	pixel501	pixel502	pixel503	pixel504	pixel505	pixel506	pixel507	pixel508	pixel509	pixel510	pixel511	pixel512	pixel513	pixel514	pixel515	pixel516	pixel517	pixel518	pixel519	pixel520	pixel521	pixel522	pixel523	pixel524	pixel525	pixel526	pixel527	pixel528	pixel529	pixel530	pixel531	pixel532	pixel533	pixel534	pixel535	pixel536	pixel537	pixel538	pixel539	pixel540	pixel541	pixel542	pixel543	pixel544	pixel545	pixel546	pixel547	pixel548	pixel549	pixel550	pixel551	pixel552	pixel553	pixel554	pixel555	pixel556	pixel557	pixel558	pixel559	pixel560	pixel561	pixel562	pixel563	pixel564	pixel565	pixel566	pixel567	pixel568	pixel569	pixel570	pixel571	pixel572	pixel573	pixel574	pixel575	pixel576	pixel577	pixel578	pixel579	pixel580	pixel581	pixel582	pixel583	pixel584	pixel585	pixel586	pixel587	pixel588	pixel589	pixel590	pixel591	pixel592	pixel593	pixel594	pixel595	pixel596	pixel597	pixel598	pixel599	pixel600	pixel601	pixel602	pixel603	pixel604	pixel605	pixel606	pixel607	pixel608	pixel609	pixel610	pixel611	pixel612	pixel613	pixel614	pixel615	pixel616	pixel617	pixel618	pixel619	pixel620	pixel621	pixel622	pixel623	pixel624	pixel625	pixel626	pixel627	pixel628	pixel629	pixel630	pixel631	pixel632	pixel633	pixel634	pixel635	pixel636	pixel637	pixel638	pixel639	pixel640	pixel641	pixel642	pixel643	pixel644	pixel645	pixel646	pixel647	pixel648	pixel649	pixel650	pixel651	pixel652	pixel653	pixel654	pixel655	pixel656	pixel657	pixel658	pixel659	pixel660	pixel661	pixel662	pixel663	pixel664	pixel665	pixel666	pixel667	pixel668	pixel669	pixel670	pixel671	pixel672	pixel673	pixel674	pixel675	pixel676	pixel677	pixel678	pixel679	pixel680	pixel681	pixel682	pixel683	pixel684	pixel685	pixel686	pixel687	pixel688	pixel689	pixel690	pixel691	pixel692	pixel693	pixel694	pixel695	pixel696	pixel697	pixel698	pixel699	pixel700	pixel701	pixel702	pixel703	pixel704	pixel705	pixel706	pixel707	pixel708	pixel709	pixel710	pixel711	pixel712	pixel713	pixel714	pixel715	pixel716	pixel717	pixel718	pixel719	pixel720	pixel721	pixel722	pixel723	pixel724	pixel725	pixel726	pixel727	pixel728	pixel729	pixel730	pixel731	pixel732	pixel733	pixel734	pixel735	pixel736	pixel737	pixel738	pixel739	pixel740	pixel741	pixel742	pixel743	pixel744	pixel745	pixel746	pixel747	pixel748	pixel749	pixel750	pixel751	pixel752	pixel753	pixel754	pixel755	pixel756	pixel757	pixel758	pixel759	pixel760	pixel761	pixel762	pixel763	pixel764	pixel765	pixel766	pixel767	pixel768	pixel769	pixel770	pixel771	pixel772	pixel773	pixel774	pixel775	pixel776	pixel777	pixel778	pixel779	pixel780	pixel781	pixel782	pixel783	pixel784	pixel785	pixel786	pixel787	pixel788	pixel789	pixel790	pixel791	pixel792	pixel793	pixel794	pixel795	pixel796	pixel797	pixel798	pixel799	pixel800	pixel801	pixel802	pixel803	pixel804	pixel805	pixel806	pixel807	pixel808	pixel809	pixel810	pixel811	pixel812	pixel813	pixel814	pixel815	pixel816	pixel817	pixel818	pixel819	pixel820	pixel821	pixel822	pixel823	pixel824	pixel825	pixel826	pixel827	pixel828	pixel829	pixel830	pixel831	pixel832	pixel833	pixel834	pixel835	pixel836	pixel837	pixel838	pixel839	pixel840	pixel841	pixel842	pixel843	pixel844	pixel845	pixel846	pixel847	pixel848	pixel849	pixel850	pixel851	pixel852	pixel853	pixel854	pixel855	pixel856	pixel857	pixel858	pixel859	pixel860	pixel861	pixel862	pixel863	pixel864	pixel865	pixel866	pixel867	pixel868	pixel869	pixel870	pixel871	pixel872	pixel873	pixel874	pixel875	pixel876	pixel877	pixel878	pixel879	pixel880	pixel881	pixel882	pixel883	pixel884	pixel885	pixel886	pixel887	pixel888	pixel889	pixel890	pixel891	pixel892	pixel893	pixel894	pixel895	pixel896	pixel897	pixel898	pixel899	pixel900	pixel901	pixel902	pixel903	pixel904	pixel905	pixel906	pixel907	pixel908	pixel909	pixel910	pixel911	pixel912	pixel913	pixel914	pixel915	pixel916	pixel917	pixel918	pixel919	pixel920	pixel921	pixel922	pixel923	pixel924	pixel925	pixel926	pixel927	pixel928	pixel929	pixel930	pixel931	pixel932	pixel933	pixel934	pixel935	pixel936	pixel937	pixel938	pixel939	pixel940	pixel941	pixel942	pixel943	pixel944	pixel945	pixel946	pixel947	pixel948	pixel949	pixel950	pixel951	pixel952	pixel953	pixel954	pixel955	pixel956	pixel957	pixel958	pixel959	pixel960	pixel961	pixel962	pixel963	pixel964	pixel965	pixel966	pixel967	pixel968	pixel969	pixel970	pixel971	pixel972	pixel973	pixel974	pixel975	pixel976	pixel977	pixel978	pixel979	pixel980	pixel981	pixel982	pixel983	pixel984	pixel985	pixel986	pixel987	pixel988	pixel989	pixel990	pixel991	pixel992	pixel993	pixel994	pixel995	pixel996	pixel997	pixel998	pixel999	pixel1000	pixel1001	pixel1002	pixel1003	pixel1004	pixel1005	pixel1006	pixel1007	pixel1008	pixel1009	pixel1010	pixel1011	pixel1012	pixel1013	pixel1014	pixel1015	pixel1016	pixel1017	pixel1018	pixel1019	pixel1020	pixel1021	pixel1022	pixel1023	pixel1024	pixel1025	pixel1026	pixel1027	pixel1028	pixel1029	pixel1030	pixel1031	pixel1032	pixel1033	pixel1034	pixel1035	pixel1036	pixel1037	pixel1038	pixel1039	pixel1040	pixel1041	pixel1042	pixel1043	pixel1044	pixel1045	pixel1046	pixel1047	pixel1048	pixel1049	pixel1050	pixel1051	pixel1052	pixel1053	pixel1054	pixel1055	pixel1056	pixel1057	pixel1058	pixel1059	pixel1060	pixel1061	pixel1062	pixel1063	pixel1064	pixel1065	pixel1066	pixel1067	pixel1068	pixel1069	pixel1070	pixel1071	pixel1072	pixel1073	pixel1074	pixel1075	pixel1076	pixel1077	pixel1078	pixel1079	pixel1080	pixel1081	pixel1082	pixel1083	pixel1084	pixel1085	pixel1086	pixel1087	pixel1088	pixel1089	pixel1090	pixel1091	pixel1092	pixel1093	pixel1094	pixel1095	pixel1096	pixel1097	pixel1098	pixel1099	pixel1100	pixel1101	pixel1102	pixel1103	pixel1104	pixel1105	pixel1106	pixel1107	pixel1108	pixel1109	pixel1110	pixel1111	pixel1112	pixel1113	pixel1114	pixel1115	pixel1116	pixel1117	pixel1118	pixel1119	pixel1120	pixel1121	pixel1122	pixel1123	pixel1124	pixel1125	pixel1126	pixel1127	pixel1128	pixel1129	pixel1130	pixel1131	pixel1132	pixel1133	pixel1134	pixel1135	pixel1136	pixel1137	pixel1138	pixel1139	pixel1140	pixel1141	pixel1142	pixel1143	pixel1144	pixel1145	pixel1146	pixel1147	pixel1148	pixel1149	pixel1150	pixel1151	pixel1152	pixel1153	pixel1154	pixel1155	pixel1156	pixel1157	pixel1158	pixel1159	pixel1160	pixel1161	pixel1162	pixel1163	pixel1164	pixel1165	pixel1166	pixel1167	pixel1168	pixel1169	pixel1170	pixel1171	pixel1172	pixel1173	pixel1174	pixel1175	pixel1176	pixel1177	pixel1178	pixel1179	pixel1180	pixel1181	pixel1182	pixel1183	pixel1184	pixel1185	pixel1186	pixel1187	pixel1188	pixel1189	pixel1190	pixel1191	pixel1192	pixel1193	pixel1194	pixel1195	pixel1196	pixel1197	pixel1198	pixel1199	pixel1200	pixel1201	pixel1202	pixel1203	pixel1204	pixel1205	pixel1206	pixel1207	pixel1208	pixel1209	pixel1210	pixel1211	pixel1212	pixel1213	pixel1214	pixel1215	pixel1216	pixel1217	pixel1218	pixel1219	pixel1220	pixel1221	pixel1222	pixel1223	pixel1224	pixel1225	pixel1226	pixel1227	pixel1228	pixel1229	pixel1230	pixel1231	pixel1232	pixel1233	pixel1234	pixel1235	pixel1236	pixel1237	pixel1238	pixel1239	pixel1240	pixel1241	pixel1242	pixel1243	pixel1244	pixel1245	pixel1246	pixel1247	pixel1248	pixel1249	pixel1250	pixel1251	pixel1252	pixel1253	pixel1254	pixel1255	pixel1256	pixel1257	pixel1258	pixel1259	pixel1260	pixel1261	pixel1262	pixel1263	pixel1264	pixel1265	pixel1266	pixel1267	pixel1268	pixel1269	pixel1270	pixel1271	pixel1272	pixel1273	pixel1274	pixel1275	pixel1276	pixel1277	pixel1278	pixel1279	pixel1280	pixel1281	pixel1282	pixel1283	pixel1284	pixel1285	pixel1286	pixel1287	pixel1288	pixel1289	pixel1290	pixel1291	pixel1292	pixel1293	pixel1294	pixel1295	pixel1296	pixel1297	pixel1



圖 4 訓練與測試資料視覺化示意圖

#### 4 研究方法

圖 5 為本次實驗的研究架構圖，首先會將資料及進行視覺化的分析，接著進行資料分割、One-hot Encoding、資料標準化與資料增強等前處理作業。清理後的資料，藉由 CNN 模型框架完成手語字母辨識模型。

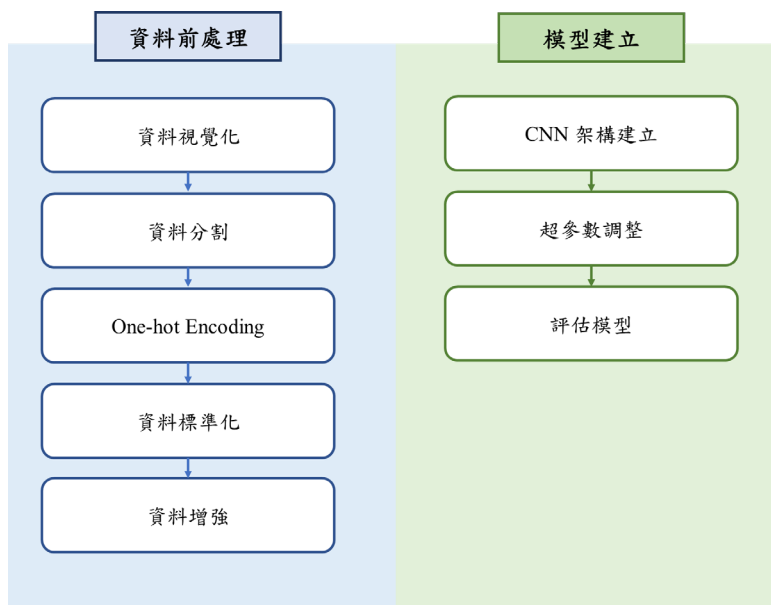


圖 5 研究架構圖

## 4.1 Python 視覺化工具 Seaborn

Seaborn 為 python 繪圖函式庫，以 matplotlib 為基礎封裝了許多實用的統計圖表，相較於 matplotlib，seaborn 的預設圖形更為好看，語法更為簡潔，並且良好的支援了 pandas 函式庫，讓使用者更加輕鬆地建立圖表，可以將其視為 matplotlib 的補強。本實驗運用此套件進行手語照片數量的視覺化呈現，並判斷是否有資料類別不平衡的問題。

## 4.2 One-hot Encoding

One Hot encoding 的編碼邏輯為將類別拆成多個行 (column)，每個列中的數值由 1、0 替代，當某一系列的資料存在的該行的類別則顯示 1，反則顯示 0。目的是為了將類別 (categorical) 或是文字 (text) 的資料轉換成數字，而讓程式能夠更好的去理解及運算。

## 4.3 資料標準化

由於此資料集以像素值呈現，而色碼定義之像素值皆為 0~255，因此將一張照片中的每個像素除上 255 以利縮小數據的跨度。

## 4.4 資料增強

一張圖片經過旋轉、調整大小、比例尺寸，或者改變亮度色溫、翻轉等處理後，人眼仍能辨識出來是相同的相片，但對於機器來說那可是完全不同的新圖像，而資料增強即是將 dataset 中既有的圖片予以修改變形，以創造出更多的圖片來讓機器學習，彌補資料量不足的困擾。本實驗運用 Keras 之 ImageDataGenerator 套件進行資料增強的作業。



## 4.5 CNN 原理

深度學習中的 CNN 較傳統的 DNN 多了 Convolutional (卷積) 及池化 (Pooling) 兩層 Layer，用以維持形狀資訊並且避免參數大幅增加。在加入此兩層後，我們所看到的架構就如下圖分別有兩層的卷積和池化層，以及一個全連結層 (即傳統的 DNN)，最後再使用 Softmax activation function 來輸出分類結果。

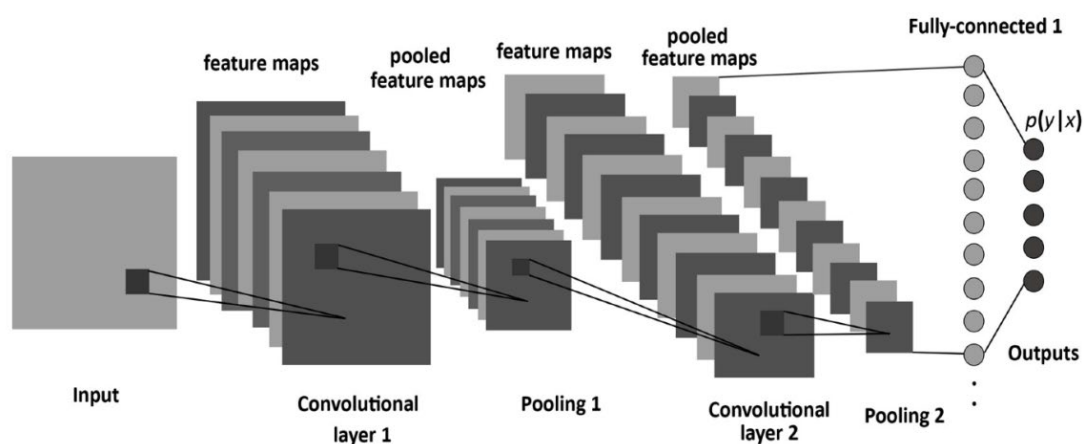


圖 6 CNN 演算法流程示意圖

## 5 研究流程

本實驗以 Python 進行實作，先將欲使用之套件匯入其中，如圖 7 所示。

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import keras
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Conv2D, MaxPool2D, Flatten, Dropout, BatchNormalization
from keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from keras.callbacks import ReduceLROnPlateau
```

圖 7 套件匯入程式碼示意圖

接著將公開資料集給予之訓練與測試資料集匯入進來，並先將其合併。圖 8 為資料匯入與合併程式碼示意圖。

```
train_df = pd.read_csv("D:/IIE/Project3/sign_mnist_train.csv")
test_df = pd.read_csv("D:/IIE/Project3/sign_mnist_test.csv")

raw_data = pd.concat([train_df, test_df], axis=0)
```

圖 8 資料匯入與合併程式碼示意圖

以 Seaborn 套件進行每種手勢出現的次數，可以發現每個類別的資料筆數範圍皆在 1400~1500 筆之間，也可以說明資料沒有不平衡的問題。圖 9 為資料視覺化程式碼示意圖，圖 10 為資料視覺化示意圖。

```
plt.figure(figsize = (15,15)) # Label Count
sns.set_style("darkgrid")
sns.countplot(raw_data['label'])
```

圖 9 資料視覺化程式碼示意圖

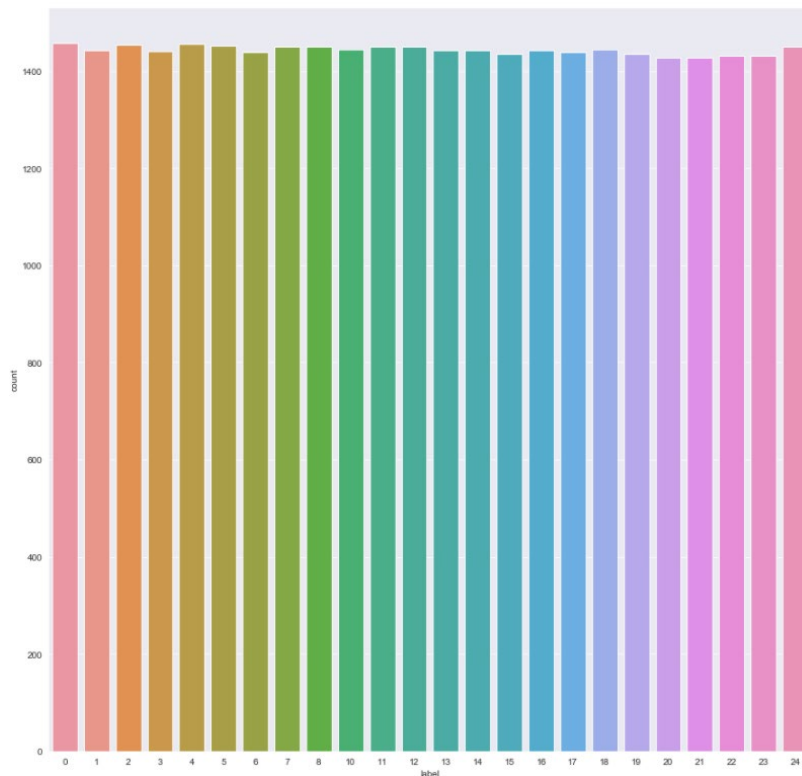


圖 10 資料視覺化示意圖

接著進行資料分割的作業，將資料分成訓練資料 70%、驗證資料集 20%與測試資料集 10%。資料分割程式碼與各資料集筆數示意圖如圖 11。

```
def datasplit(dataframe,ratio1,ratio2):
    return dataframe[0:(round(len(dataframe)*(ratio1)))] ,dataframe[(round(len(dataframe)*(ratio1))):(round(len(dataframe)*(ratio1+ratio2)))]
train_all, validation_all, test_all =datasplit(raw_data,0.7,0.2)

#特徵值
train_x = train_all.drop(columns=['label']).values
validation_x = validation_all.drop(columns=['label']).values
test_x = test_all.drop(columns=['label']).values

#目標值
train_y = train_all['label'].values
validation_y = validation_all['label'].values
test_y = test_all['label'].values
print(len(train_x), len(validation_x), len(test_x))
print(len(train_y), len(validation_y), len(test_y))
```

24239 6925 3463  
24239 6925 3463

圖 11 資料分割程式碼與各資料集筆數示意圖

由於原先目標值為 0~24 之整數類別資料，因此本實驗將進行 One-hot Encoding 程序將目標值轉換成 0,1，以利電腦作業。圖 12 為 One-hot Encoding 程式碼示意圖，圖 13 為 One-hot Encoding 結果示意圖。

```
from sklearn.preprocessing import LabelBinarizer
label_binarizer = LabelBinarizer()
train_y = label_binarizer.fit_transform(train_y)
validation_y = label_binarizer.fit_transform(validation_y)
test_y = label_binarizer.fit_transform(test_y)
```

圖 12 One-hot Encoding 程式碼示意圖

```
train_y
array([[0, 0, 0, ..., 1, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 1],
       [0, 1, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

圖 13 One-hot Encoding 結果示意圖

模型建立之前，必須將輸入之特徵值轉換為 CNN 可以支持的格式，

圖 14 為資料維度轉換程式碼示意圖。

```
train_x_cnn = train_x.reshape(-1,28,28,1)
validation_x_cnn = validation_x.reshape(-1,28,28,1)
test_x_cnn = test_x.reshape(-1,28,28,1)
```

圖 14 資料維度轉換程式碼

由於色碼各個像素值範圍介於 0~255，因此將所有特徵值除上 255，標準化以縮減數據間的跨度，標準化程式碼示意圖如圖 15，標準化結果示意圖如圖 16。

```
# Normalize the data
train_x = train_x / 255
validation_x = validation_x / 255
test_x = test_x / 255
```

圖 15 標準化程式碼示意圖



```
train_x
array([[107, 118, 127, ..., 204, 203, 202],
       [155, 157, 156, ..., 103, 135, 149],
       [187, 188, 188, ..., 195, 194, 195],
       ...,
       [155, 159, 163, ..., 222, 221, 221],
       [145, 147, 148, ..., 181, 180, 180],
       [ 50,  14,  52, ..., 169, 169, 169]], dtype=int64)
```

```
train_x
array([[0.41960784, 0.4627451, 0.49803922, ..., 0.8, 0.79607843,
        0.79215686],
       [0.60784314, 0.61568627, 0.61176471, ..., 0.40392157, 0.52941176,
        0.58431373],
       [0.73333333, 0.7372549, 0.7372549, ..., 0.76470588, 0.76078431,
        0.76470588],
       ...,
       [0.60784314, 0.62352941, 0.63921569, ..., 0.87058824, 0.86666667,
        0.86666667],
       [0.56862745, 0.57647059, 0.58039216, ..., 0.70980392, 0.70588235,
        0.70588235],
       [0.19607843, 0.05490196, 0.20392157, ..., 0.6627451, 0.6627451,
        0.6627451 ]])
```

圖 16 標準化結果示意圖

為了避免訓練資料與測試資料過擬合，以 Keras 之 ImageDataGenerator 套件對於訓練資料集進行資料增強的作業。調整的範圍有照片旋轉的角度、隨機鏡像調整、隨機垂直與水平移動圖像等設定。圖 17 為資料增強設定程式碼示意圖。

```

datagen = ImageDataGenerator(
    rotation_range=10,
    zoom_range = 0.1,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    vertical_flip=False)

datagen.fit(train_x_cnn)

```

圖 17 資料增強設定程式碼示意圖

設定完資料增強架構後，接著將建立 CNN 模型架構，本實驗之 CNN 模型流程如下：進行一次神經元數為 25 之卷積層、進行一次池化層、經過神經元數為 75 之卷基層、進行一次池化層、經過比率為 0.5 之 Dropout、進行一次池化層、經過神經元數為 100 之卷基層、進行一次池化層、經過比率為 0.5 之 Dropout、進行一次池化層。然後進行扁平層將神經元拉直，接著進行一次神經元數為 128 之全連接層、0.5 之 Dropout 比率與一次神經元數為 24 之全連接層，最後輸出 24 單位之輸出項。CNN 預設模型架構如圖 18。

```

model = Sequential()
model.add(Conv2D(25, (3,3), strides = 1, padding = 'same', activation = 'relu', input_shape = (28,28,1)))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2), strides = 2, padding = 'same'))

model.add(Conv2D(75, (3,3), strides = 1, padding = 'same', activation = 'relu'))
model.add(Dropout(0.5))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2), strides = 2, padding = 'same'))
model.add(Conv2D(100, (3,3), strides = 1, padding = 'same', activation = 'relu'))
model.add(Dropout(0.5))
model.add(BatchNormalization())
model.add(MaxPool2D((2,2), strides = 2, padding = 'same'))

model.add(Flatten())
model.add(Dense(units = 128, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(units = 24, activation = 'softmax'))
model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
model.summary()

```

圖 18 CNN 預設模型架構

接著進行超參數的調整，首先調整 Activation Function，本實驗比較三

種 Activation Function，如 softmax、ReLU、Sigmoid。可以得知 Activation Function 為 ReLU 的時候模型表現最好，因此保留其參數設定。表 1 為 Activation Function 評估比較表。

表 1 Activation Function 評估比較表

Activation Function	Training	Validation	Testing
softmax	58.99%	78.12%	74.62%
ReLU	90.89%	96.82%	96.15%
Sigmoid	86.54%	67.15%	67.13%

接著調整 Batch Size，本實驗比較三種 Batch Size，如 10, 50, 100。可以得知 Batch Size 為 50 的時候模型表現最好，因此保留其參數設定。表 2 為 Batch Size 評估比較表。

表 2 Batch Size 評估比較表

Batch Size	Training	Validation	Testing
10	90.89%	96.82%	96.15%
50	96.52%	98.71%	98.46%
100	97.53%	98.17%	97.31%

接著調整 Learning Rate，本實驗比較三種 Learning Rate，0.1、0.01、0.001。可以得知 Learning Rate 為 0.001 的時候模型表現最好，因此保留其參數設定。表 3 為 Learning Rate 評估比較表。

表 3 Learning Rate 評估比較表

Learning Rate	Training	Validation	Testing
0.1	96.52%	98.71%	98.46%
0.01	93.59%	97.39%	96.62%
0.001	94.74%	99.49%	99.24%

最後調整 Loss Function，本實驗比較三種 Loss Function，如 0.1、0.01、

0.001。可以得知 Loss Function 為 0.001 的時候模型表現最好，因此保留其參數設定。表 4 為 Loss Function 評估比較表。

表 4 Loss Function 評估比較表

Loss Function	Training	Validation	Testing
categorical_crossentropy	94.74%	99.49%	99.24%
binary_crossentropy	99.49%	99.98%	99.98%

從超參數調整的過程中，可以發現模型的訓練資料、驗證資料與測試資料準確度已經從原始設定的 90.89%、96.82%、96.15% 提升至 99.49%、99.98%、99.98%。另外以混淆矩陣的形式也可以看出模型預測的大致上為正確。圖 19 為模型預測結果混淆矩陣。

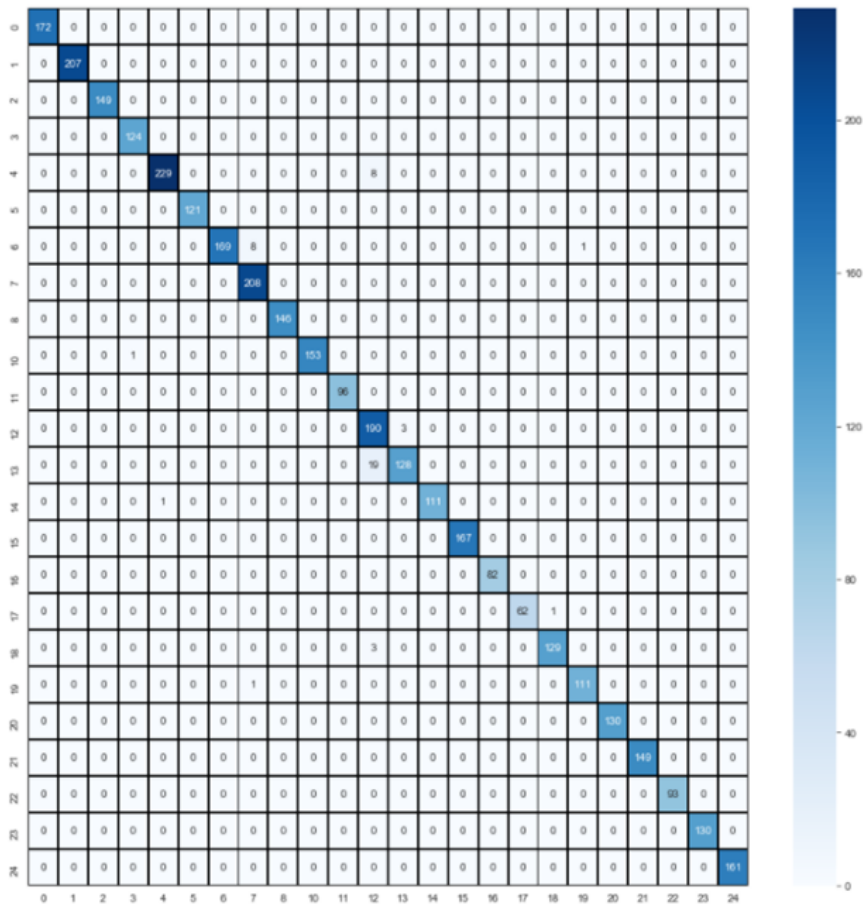


圖 19 模型預測結果混淆矩陣

## **6.結論**

### **6.1 研究結果**

#### **6.1.1 公開資料準確度高**

本實驗運用不同資料前處理方式與 CNN 深度學習架構建立 MNIST 手語字母辨識模型可以獲得高達 99.98% 之準確度。

#### **6.1.2 外來照片準確度問題**

由於本實驗僅根據公開資料進行訓練與評估模型，若以外來照片進行測試會產生預測不準的問題。

### **6.2 未來發展**

#### **6.2.1 蒐集不同資料集**

在未來有多餘的時間，期許蒐集來自不同來源的手語靜態圖建立模型，以避免環境過於單純而產生模型過擬合的問題。

#### **6.2.2 結合手掌捕捉技術**

未來有機會欲結合手掌動態捕捉的技術，以 3D 抓取手部關鍵點之方式建立手語辨識預測模型，以更真實的圖像與情境訓練與評估模型。