A # Step 3: 資料增強

datagen = ImageDataGenerator(rotation\_range=10, width\_shift\_range=0.1, height\_shift\_range=0.1, zoom\_range=0.1) datagen.fit(x\_train)

# Step 4: 設計 CNN 模型

model = Sequential([ Conv2D(32, (3, 3), activation='relu', input\_shape=(64, 64, 1)), MaxPooling2D((2, 2)), Conv2D(64, (3, 3), activation='relu'), MaxPooling2D((2, 2)), Flatten(), Dense(128, activation='relu'), Dense(num\_classes, activation='softmax') ]) model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

B # Step 3: 資料增強

datagen = ImageDataGenerator( rotation\_range=10, width\_shift\_range=0.1, height\_shift\_range=0.1, zoom\_range=0.1, shear\_range=0.1, horizontal\_flip=True, fill\_mode='nearest' ) datagen.fit(x\_train)

# Step 4: 設計 CNN 模型

model = Sequential([ Conv2D(32, (3, 3), activation='relu', input\_shape=(64, 64, 1)), MaxPooling2D((2, 2)), Conv2D(64, (3, 3), activation='relu'), MaxPooling2D((2, 2)), Conv2D(128, (3, 3), activation='relu'), MaxPooling2D((2, 2)), Flatten(), Dense(128, activation='relu'), Dropout(0.5), Dense(num\_classes, activation='softmax') ]) model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

**差異**

**片段 A：**

* 使用基本的增強技術：旋轉、水平位移、垂直位移和縮放。

python

複製程式碼

datagen = ImageDataGenerator(rotation\_range=10, width\_shift\_range=0.1, height\_shift\_range=0.1, zoom\_range=0.1)

**片段 B：**

* 除了基本的增強技術外，還包括剪切變換、水平翻轉和設定填充模式為‘最近’。

python

複製程式碼

datagen = ImageDataGenerator(

rotation\_range=10,

width\_shift\_range=0.1,

height\_shift\_range=0.1,

zoom\_range=0.1,

shear\_range=0.1,

horizontal\_flip=True,

fill\_mode='nearest'

)

**CNN 模型設計**

**片段 A：**

* 模型架構較簡單，包含兩個卷積層，後接池化層。
* 在輸出層前有一個全連接層。

python

複製程式碼

model = Sequential([

Conv2D(32, (3, 3), activation='relu', input\_shape=(64, 64, 1)),

MaxPooling2D((2, 2)),

Conv2D(64, (3, 3), activation='relu'),

MaxPooling2D((2, 2)),

Flatten(),

Dense(128, activation='relu'),

Dense(num\_classes, activation='softmax')

])

**片段 B：**

* 模型架構較為複雜，包含額外的一個卷積層。
* 包含一個 dropout 層來減少過擬合，訓練過程中隨機將部分輸入單元設為 0。

python

複製程式碼

model = Sequential([

Conv2D(32, (3, 3), activation='relu', input\_shape=(64, 64, 1)),

MaxPooling2D((2, 2)),

Conv2D(64, (3, 3), activation='relu'),

MaxPooling2D((2, 2)),

Conv2D(128, (3, 3), activation='relu'),

MaxPooling2D((2, 2)),

Flatten(),

Dense(128, activation='relu'),

Dropout(0.5),

Dense(num\_classes, activation='softmax')

])

**編譯**

兩個片段使用相同的編譯設置：

python

複製程式碼

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

**差異總結**

1. **資料增強：**
   * 片段 A 使用基本的增強技術。
   * 片段 B 包括額外的增強技術，如剪切變換、水平翻轉和填充模式。
2. **模型架構：**
   * 片段 A 的架構較簡單，有兩個卷積層和一個全連接層。
   * 片段 B 的架構較複雜，有三個卷積層、一個 dropout 層和一個全連接層。
3. **過擬合預防：**
   * 片段 B 包含 dropout 層來幫助預防過擬合。

總之，片段 B 在資料增強和模型複雜度上更為健全，可能提高模型的性能和泛化能力，但也可能增加訓練時間和計算需求。