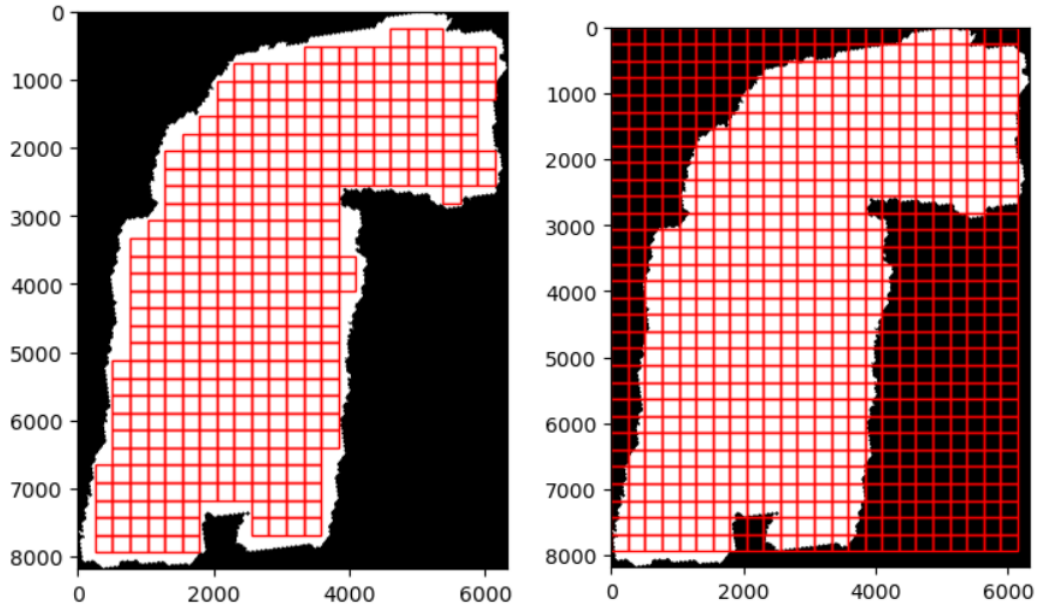
LeaderBoard:

Approach:

資料載入與前處理:

一開始照著教程的方式結果卷軸二會因為太大導致 CPU 超載，在尋找解決方法時剛好看到有一篇在分享如何利用 patchify 將影像事先分割好(如下圖)，好處是可以在一開始 loading 時就只取有資訊的地方，不用像教程那樣透過 subvolume 來分割，當時覺得這個方式比較好理解也可以順利解決我們的問題，所以就先使用這個模板，不過後來發現有兩個比較大的問題。



左圖:(train_dataset 只將中間的方格餵給 model 訓練)
右圖:(valid 跟 test 都是採取切到不能再切)

第一個問題是他需要安裝一個函式庫，然而這次競賽的要求是不能聯網，所以不能用安裝的方式。解決方法是去官往找 patchify 的檔案，載下來再丟回 input_dataset 中，用 import 的方式取得相關的函式。

```
import sys
sys.path.append('/kaggle/input/patchify/patchify-0.2.3')

import patchify
```

```
#pip install patchify
```

第二個問題是以為這樣節省了很多空間，沒想到其實這樣做反而會佔著很多記憶體，導致最後層數只能取六層，而且不能 padding，導致資料量很少，model 很快就 overfitting 了，本來想用 data_argument 讓資料變多一點，不過最後沒有成功用出來。

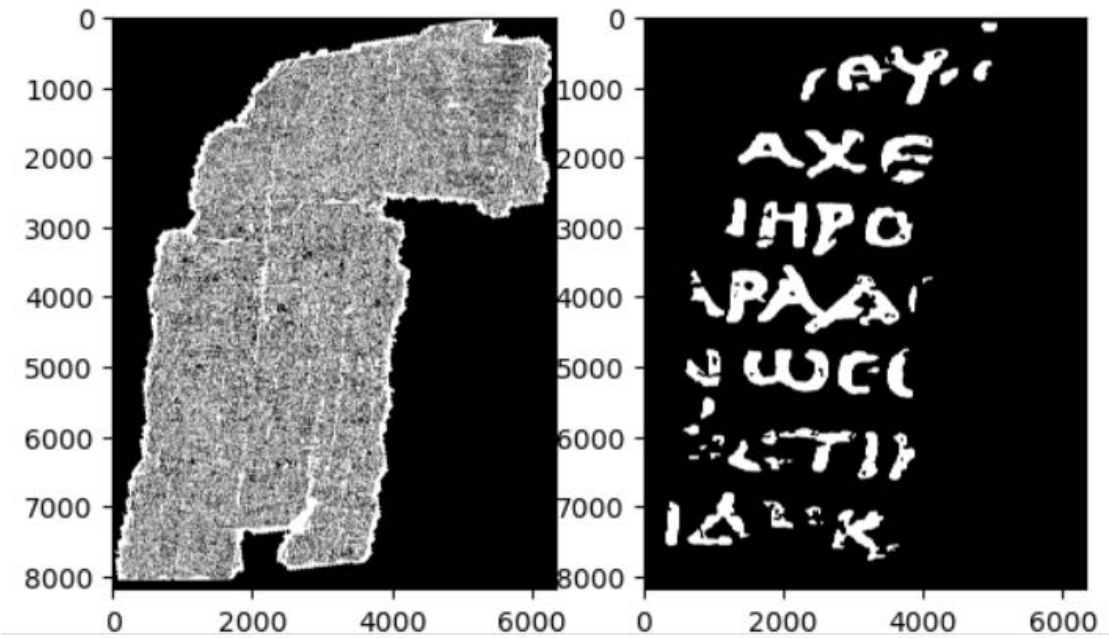
所以為了讓資料量更多一些，把原本拿來當 valid 的卷軸一也加入 train 裡面，但這樣就會需要另外找 valid_data，最後是取中間六層，兩層兩層取，一層當 train 的取層，一層當 valid 的取層，本來想說不知道取不同層會不會是造成 overfitting 的問題(層數的不同導致 train 跟 valid 本身差異太大，造成對 train_data 的貼合反而對 valid_data 沒有幫助)，結果實測出來也不是這個原因，單純資料量不足。

資料後處理:

由於我們的 model 是將[6,224,224]轉成[1,224,224]，而前面處理資料時是先切成相鄰的子圖，所以將 model 的 output 中每個 pixel 對應到其真實的位子，先用 valid 做測試，調整 threshold 使其跟他的 label 比較相像，取得比較適合的 threshold 拿來當這次的 test_data 的 threshold。

Model:

我們使用的模型是 U-net，一開始我們用圖像尺寸為 224*224、輸入通道為 6、輸出通道為[64, 128, 256, 512, 1024,1]，不過一開始的 train 在過不了多少就 overfiting，於是我們先增加了正規化同時把原本的激活函數從只有 Sigmoid 變成先做 ReLU 再做 Sigmoid(因為我們的輸出必須限制在 0 到 1 之間)，但做完這些之後發現還是 overfitting，之後我們在助教的建議下縮小模型的複雜度，最後我們完成的通道數是[32, 32, 64, 32, 32, 1]，降低模型的複雜度之後，我們的模型就能夠在預期的訓練次數跑完了(不會跑一兩次就 overfiting)，不過因為時間不足的因素，我們交出的成果是訓練相對最好的 model。



Confusion matrix:

不太確定這邊有沒有用對，我是將 valid_data 出來的結果，跟它本身的 label 用涵式做對比，由於涵式要求的是一維的資料，所以先把二維展開成一維

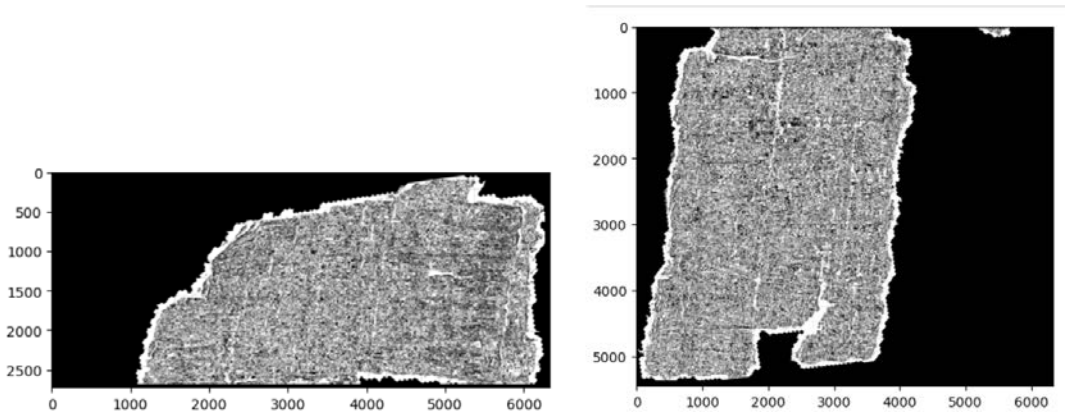
F1 score 不知道為何當 average=binary 的時候會出現問題，所以先用 average=micro , macro , weight 分別計算。

```
torch.Size([8181, 6330])
(8181, 6330)
f1_micro: 0.6039563987994376
f1_macro: 0.2612692534920918
f1_weighted: 0.7029933474788701
```

```
iou =  0.15875378551900599
sensitivity =  0.00239137479089721
recall =  0.17670398297708925
```

Test 輸出:

test 跟 valid 後半生出圖片的步驟很像，原本以為可以直接改變數名稱，後來發現前面的架構都是建立在有 true_label 的情況下，所以當資料型態少了 test 之後，需要重新修改前面的 code 以符合這個要求。



從最後結果的圖來說，雖然看不太到什麼文字，至少整個輪廓有出來。

Group info:

隊名:Yes No Team

成員:

謝宗翰(50%)

王柏弘(27%)

陳俊諺(10%)

李群晟(8%)

蘇柏瑜(5%)

Discussion or issue:

謝宗翰:

這次的 final project 我們做得並不好，除了期末時間緊迫外，主要是對於人工智慧的程式的不了解。pythorch(python)這種封包的編譯環境跟之前休的程式課程都很不一樣，想要使用某個功能要先去查詢對應的 function，並且還要搞清楚輸入與輸出的資料型態，然後跳出的錯誤訊息可能是那個 function 中某一行導致的，而在不了解 function 實際如何運作的狀況下我實在是難以 debug，將錯誤訊息 google 後還是看不懂，最後都是請 chatGPT 幫忙解釋或 debug，學到的第一件事是如何利用 chatGPT 幫助我更好的學習人工智慧。

由於隊友也跟我有一樣的問題，導致最後我幾乎參與了整份 code 的製作與修改，光是 debug 就很光時間了，所以沒針對哪些參數做實際的討論，最後能讓程式順利運行出結果對我而言已經很值得慶幸了。雖然成品很粗糙，不過這次我也更加了解人工智慧程式中整體的架構，包含參數的意義與設定、資料如何處理、pytorch 的語法等等……

儘管 final project 讓我感到痛苦與無力，感受到自己自學能力需要加強，像有很多功能都實做不出來、看不懂 error messege 等等……，不過我感謝這堂課讓我去多練習這些平時我沒有動力去做的事情，也很謝謝助教很認真的回答我

遇到的問題，相信不管是老師上課教的還是程式方面的內容，都加深了我對於人工智慧的認知與理解。

王柏弘:

這次的 final project 真的讓我受益匪淺。相較於前面的作業一和作業二，那只是單純地調整參數或建立一個模型而已。然而，這次的 final project 需要我們自己從頭到尾完全撰寫程式碼。一開始時，我完全不知道如何下手，只能參考 Kaggle 上其他人提供的範例程式碼，試著把各種零散的資料拼湊在一起。將不同作者的程式碼整合時，也遇到了許多問題。

起初我們嘗試使用 TIFF 檔案格式讀取資料，卻發現標籤(label)不是由 0 和 1 組成，而是一些奇怪的小數值。這導致我們在訓練模型時產生了一些奇怪的結果。此外，由於這次可用的訓練資料量不多、以及 GPU 大小有限，無法將太多的資料餵給模型，也使得我們難以訓練出好的模型。因此，我們將更多的時間投入在研究資料前處理，包括層數選擇、資料裁切等，以及評估不同模型對於這次競賽的優劣，也因如此，讓我對資料前處理以及不同的模型有更進一步的了解。

儘管在這次的 final project 中，我在程式碼的撰寫方面所做的貢獻不多，大部分的時間都是幫忙查詢資料，以及在除錯過程中提供協助。然而，在查詢資料和解決問題的過程中，我對人工智慧有了更深入的了解。我要感謝老師和助教們的教導，讓我能在這次的 final project 中獲得這樣的學習機會。

陳俊諺:

這次的 final project 是一個寶貴的學習經驗，即使我們開始得晚，但我們還是能夠在時間壓力下完成任務並取得不錯的成績。這次的項目讓我們不僅學習到了深度學習知識和編程技巧，還讓我認識到了 Kaggle 這個網站，並瞭解到它對於機器學習和數據科學領域的重要性。

這一次我負責的是 model 的部分，一開始我們是以 resnet 當 model 不過我們在 output 的部分和 inklabel 的 tensor size 不一樣而且不容易去改正，所以最後我們使用 U-net，而 tensor size 的部分也修正完，不過因為修正後的 train loss 會在一開始的時候就 overfit，所以我們在反覆修正我們的通道數和卷積層數，最後順利結束，真的很有成就感。

另一個我從中獲益良多的收穫是學習到了 Kaggle 這個網站。在我之前並不知道這個平台，但在這次項目中的同學介紹下，我進一步了解到了 Kaggle 提供的各種資源和功能。Kaggle 不僅提供了豐富的數據集和競賽，還有許多教育資源和社群交流的機會。透過參與 Kaggle 競賽，我能夠運用所學的知識，與來自世界各地的數據科學家和機器學習專家進行競爭和合作。這為我提供了

一個實踐和學習的平台，讓我能夠不斷提升自己的能力。

李群晟:

一開始我看到這次的 final project 的時候，我心中充滿了問號，因為它與作業 1 與作業 2 只需改一些參數與模型相比難了許多，同時也因為是第一次接觸到 kaggle 這個網站，所以對其所有的一切都極為的不熟悉，但在查詢很多網站與詢問組員後，才慢慢了解其要求與如何運作，而這次我負責的是 model 的部分，我了解到這項功課是墨水檢測，也就是輸入的資料會是圖片後，就去尋找之前作業寫過的 cnn 架構與其他有關的模型，然後我們就查到 resnet 與 unet，比較它們的優缺點之後我們決定使用主要應用在醫療方面的 unet，然後我們根據資料輸入的要求使用 chatgpt 來協助生成程式碼，但往往會出現 error message，使我們在做的過程中遇到蠻多困難的，例如: 資料量的不足、網站中執行 gpu 不夠、在 training 的途中發現 loss 值太大，但也不確定是哪個部分出現問題等等，然後又遭遇到期末各科的壓力，使我產生了深深地無力感，但也因為這次的 final project 使我更了解人工智慧的實作方面，同時也多虧了組員與助教的大力協助才能讓我們能跑出結果，謝謝組員與助教。

蘇柏瑜:

首先，我想先感謝其他組員如此地凱瑞，儘管我幫不上任何忙也能在期末做出這麼好的結果，也謝謝助教在課後幫我們找出程式中的問題。在前兩次的作業我就感受到這門課與先前修過的課完全不同，儘管我上網查了許多資料，還是無法看懂程式碼在敘述什麼，所以在前兩次作業就只是不斷嘗試網路上提供的方法不斷嘗試，完全沒搞懂背後運行的邏輯。而要討論 final project 時，我們已經大致想好要用哪個方法來處理資料，但由於前面基礎沒打好，因此我在討論區中無法得知哪個方法可以優化我們的資料處理，而在 debug 的時候更是幫不上忙。總之，謝謝其他組員們能頂著期末的壓力以及少一個人的狀況將這份作業做完，雖然這份作業有讓我稍微了解人工智慧的實作，但我相信在助教和教授這麼認真的指導下我不應該只有學習到這些東西，我感到很抱歉。