

1. Data

1.1. Dataset

使用的資料集為 PH² 資料集 [1]。此資料集包含 200 張黑色素細胞病變（melanocytic lesions）的皮膚鏡影像，其中有 80 張為正常痣、80 張為異常痣、40 張為黑色素瘤。所有皮膚鏡影像皆為 RGB 彩色影像，並以二元遮罩（binary mask）標註病變區域。

1.2. Exploratory data analysis

首先觀察原始影像大小分布，可以看到所有影像的大小相近（如圖 1），高度平均為 766 ± 1.57 ，寬度平均為 576 ± 2.00 ，高度與寬度比約為 4:3。

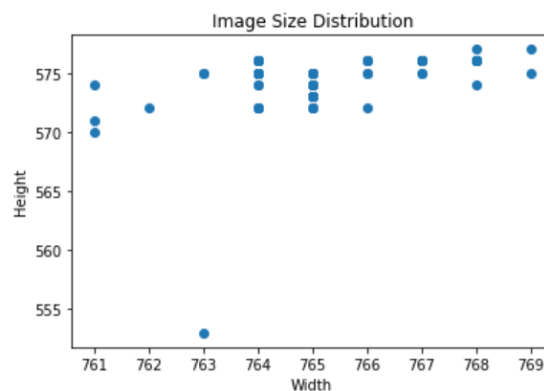


圖 1、影像大小分布圖

隨機取出 4 張原始影像觀察，如圖 2。可觀察到影像內包含皮膚鏡視野外的範圍（圓圈外黑色部分），且膚色也有差異，從較淺色到較深色皆有，此外病變處可能會有毛髮阻擋，以上這些都可能是影響模型分割結果的因素。

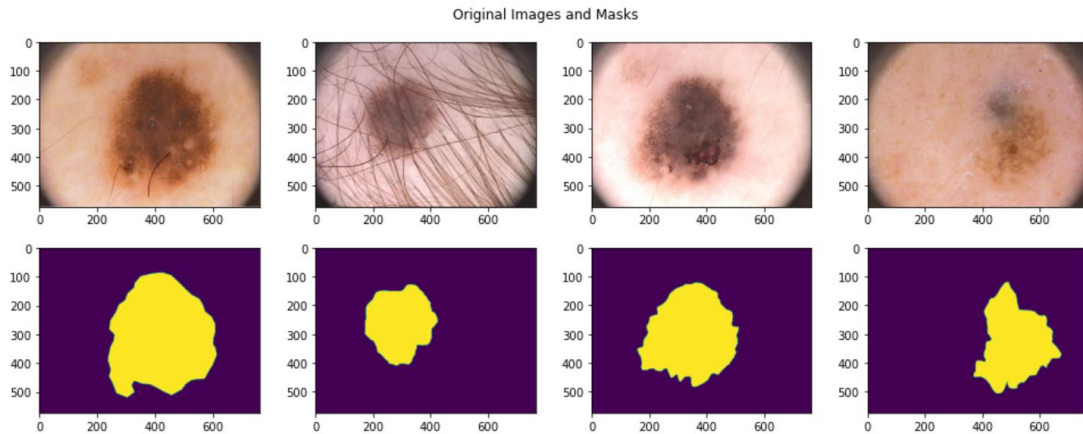


圖 2、原始皮膚鏡影像與其遮罩

由於原始影像高寬比約為 4:3，因此統一縮放為 192×256 ，並取出 4 張影像觀察，如圖 3。與原始影像（圖 2）比較確定影像經由重新縮放後沒有太大的變化，並且仍保留原始影像的特徵，因此後續會將所有原始影像縮放為同一比例 192×256 後再輸入深度學習模型分析。

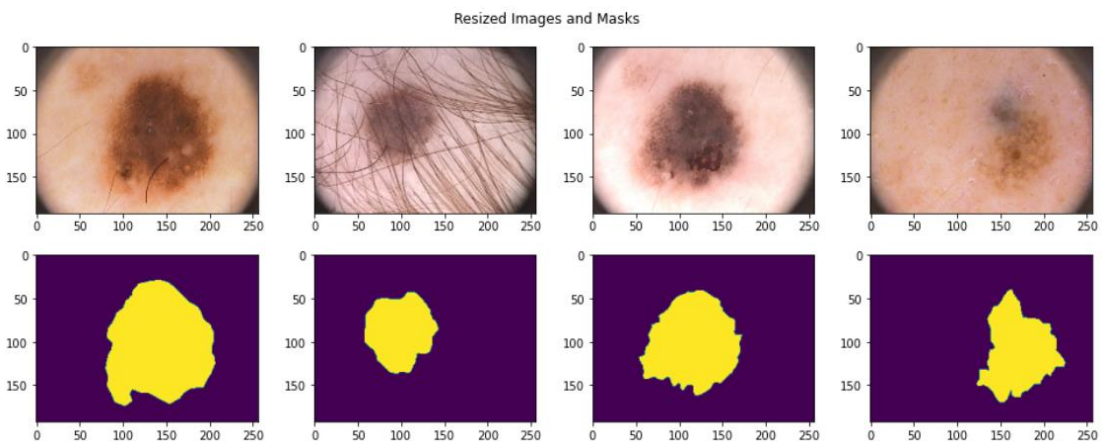


圖 3、縮放後皮膚鏡影像與其遮罩

2. Project objective

給予黑色素細胞病變皮膚鏡影像，利用深度學習模型將病變區域分割出來。

3. Methodology

3.1. 訓練、驗證與測試資料

將資料集隨機以 0.8-0.1-0.1 比例分為訓練、驗證與測試資料集，不過控制三種病變情形（正常痣、異常痣、黑色素瘤）在三個資料集所佔比例相同，實際資料量如表 1。驗證資料集用於模型訓練時評估模型表現以調整參數，測試資料集則用於確定模型後的最終表現評估。

表 1、訓練、驗證與測試資料個數

	Train	Valid	Test
Normal Nevus	64	8	8
Atypical Nevus	64	8	8
Melanoma	32	4	4
Total	160	20	20
Percentage	0.8	0.1	0.1

在此測試 4 種方法，一為預訓練的語意分割模型；二為參考同樣做皮膚病變影像分割的論文提出的 Full Resolution Convolutional Network (FrCN) [4]；三為基本的 U-Net；四為 U-Net 的改良。

3.2. 方法一：預訓練的語意分割模型

使用已在 COCO 資料集預訓練的語意分割模型，包括：fully convolutional network (FCN) [2] 與 DeepLab V3 [3]。FCN 將傳統 CNN 中 fully connected layer 改為 convolutional layer，並在最後進行 upsampling 輸出與原圖大小相同的特徵圖；DeepLab V3 則引入 atrous convolution 的概念，利用填入空洞，在不增加模型參數的情況下使 receptive field 變大，以避免 pooling 造成特徵訊息損失。在此測試修改兩個模型的 backbone，且由於使用的資料集數量不大，因此嘗試在資料預處理方面做 data augmentation。

3.3. 方法二：FrCN

參考 [4]，此論文目標同樣是要分割黑色素細胞病變之皮膚鏡影像，作者修改已在 ImageNet 資料集上預訓練的 VGG-16，移除其中的 pooling layer，並且將原本最後的 fully connected layer 改為 convolutional layer。

此架構大致與 FCN 相似，只是將 pooling layer 去除，作者表示此目的是為了避免 pooling 造成特徵訊息損失。由於其使用的資料集與本作業略有不同，因此嘗試將其方法套用於此作業使用的資料集。

3.4. 方法三：U-Net

利用 U-Net [5]，架構包括 encoder 和 decoder，與傳統 autoencoder 相似，但引入 skip connection 的概念，利用 channel-wise concatenation 連接 encoder 與 decoder 在 subsampling 與 upsampling 過程中對應位置的特徵圖，使模型可以保留更多高層特徵圖的細節特徵，進而提高影像分割的精確度。在此直接以 PH² 資料集訓練，測試不同層數與 filter 數目的影響。同時，也會在資料預處理的部分做 data augmentation。

3.5. 方法四：U-Net 改良

參考 [6]，以 U-Net 為基礎，針對 encoder 與 decoder 的架構進一步實驗其他的調整。[6] 的作者提出 DSNet，encoder 使用在 ImageNet 資料集上預訓練過的 DenseNet-121，decoder 則使用 depth-wise separable convolution，且 loss function 同時參考 binary cross entropy 與 IoU (Intersection over Union)，計算式如下：

$$L(y, \hat{y}) = 1 - \frac{\sum_{i=1}^N y_i \times \hat{y}_i}{\sum_{i=1}^N y_i + \sum_{i=1}^N \hat{y}_i - \sum_{i=1}^N y_i \times \hat{y}_i} - \frac{1}{N} \sum_{i=1}^N [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)]$$

其中 y 、 \hat{y} 與 N 分別為真實 label、預測 label 與影像 pixel 數目。

這邊會測試修改 encoder 為不同的預訓練網絡，同時也會觀察使用不同 loss function 的影響。另外，資料預處理的部分會嘗試做 data augmentation。

3.6. 模型表現評估

利用平均 IoU (mIoU, 又稱 Jaccard Index) 作為模型最終評估標準, IoU 表示真實病變區域與預測病變區域的重疊比例, 計算式如下:

$$\text{IoU} = \frac{(\text{Predicted Lesion Area}) \cup (\text{Real Lesion Area})}{(\text{Predicted Lesion Area}) \cap (\text{Real Lesion Area})}$$

4. Experiments

使用 PyTorch 框架, 預訓練模型使用 torchvision 套件 [7], data augmentation 方法使用 albumentations 套件 [8]。以下方法皆訓練 40 個 epoch, 並且選擇在驗證資料集具有最低 loss 的參數模型用於測試。若未特別說明, 訓練模型時的 loss function 為 binary cross entropy, learning rate 為 0.0001, 使用 Adam optimizer。

4.1. 方法一：預訓練的語意分割模型

實驗不同的 backbone, FCN 測試 ResNet-50 與 ResNet-101; DeepLab V3 測試 ResNet-50、ResNet-101 與 MobileNet V3, 並且取兩者在測試資料集上表現最好的條件, 嘗試 ensemble learning。此外, 也測試有無進行 data augmentation 的差異, data augmentation 的方法為 horizontal/vertical flipping 與 rotation。

4.2. 方法二：FrCN

以 VGG-16 為基礎, 移除其中的 pooling layer, 並且原本由 fully connected layer 組成的最後一個 block 改為三層 convolutional layer, 層與層之間通過 ReLU 函數與一層 dropout layer。在此測試使用不同的 optimizer 與 learning rate。

4.3. 方法三：U-Net

測試不同層數與 filter 數目的 U-Net。第一個測試模型的 encoder 由三層 convolutional layer 組成, filter 個數分別為 16、32 與 64; 第二個模型的 encoder 由四層 convolutional layer 組成, filter 個數分別為 64、128、256、512 與 1024;

decoder 則與 encoder 對稱。另外，參考 [6]，嘗試使用不同的 loss function，同時也測試有無進行 data augmentation 的差異，data augmentation 的方法為 horizontal/vertical flipping 與 rotation。

4.4. 方法四：U-Net 改良

架構為 encoder 與 decoder。Encoder 測試使用不同的預訓練網路，包括：DenseNet-121、ResNet-18 與 ResNet-101；decoder 則參考 [6] 使用 depth-wise separable convolution。另外，也測試使用不同的 loss function 以及有無進行 data augmentation 的差異，data augmentation 的方法為 horizontal/vertical flipping 與 rotation。

5. Results & Analysis

5.1. 方法一

由表 2 可以觀察到，使用預訓練的語意分割模型，不論是 FCN 或 DeepLab V3 若有做 data augmentation，在測試資料集上平均 IoU 皆較高。同時，可以看到在同一個架構下，若資料未做 augmentation，以 ResNet-101 為 backbone 的模型表現最佳；而若資料有做 augmentation，則以 ResNet-50 為 backbone 的模型表現最好，這可能是因為在未做 data augmentation 且訓練資料量不多的情況下，較複雜的網絡容易發生 overfitting。

接下來，取兩種模型架構在測試資料集上表現最好的條件做 ensemble，可以觀察到若以 ResNet-50 為 backbone 並且資料預處理做 flipping 和 rotation 在這個方法中在測試資料達到最高的平均 IoU，其值為 0.9425，圖 4 為三張測試影像的分割結果；反觀，若改以 ResNet-101 為 backbone 並且不做 data augmentation，卻得到相比所有其他條件最低的平均 IoU，其值為 0.9081，推測這可能即是因為 overfitting 的問題。

表 2、方法一之測試條件與結果

	Model	Backbone	Data Aug.	mIoU
1	FCN	ResNet-50	No	0.9305
2	FCN	ResNet-101	No	0.9377
3	FCN	ResNet-50	Flip/Rotate	0.9388
4	FCN	ResNet-101	Flip/Rotate	0.9364
5	DeepLab V3	ResNet-50	No	0.9363
6	DeepLab V3	ResNet-101	No	0.9409
7	DeepLab V3	MobileNet V3	No	0.8958
8	DeepLab V3	ResNet-50	Flip/Rotate	0.9365
9	DeepLab V3	ResNet-101	Flip/Rotate	0.9326
10	DeepLab V3	MobileNet V3	Flip/Rotate	0.9219
11	FCN + DeepLab V3	ResNet-101	No	0.9081
12	FCN + DeepLab V3	ResNet-50	Flip/Rotate	0.9425

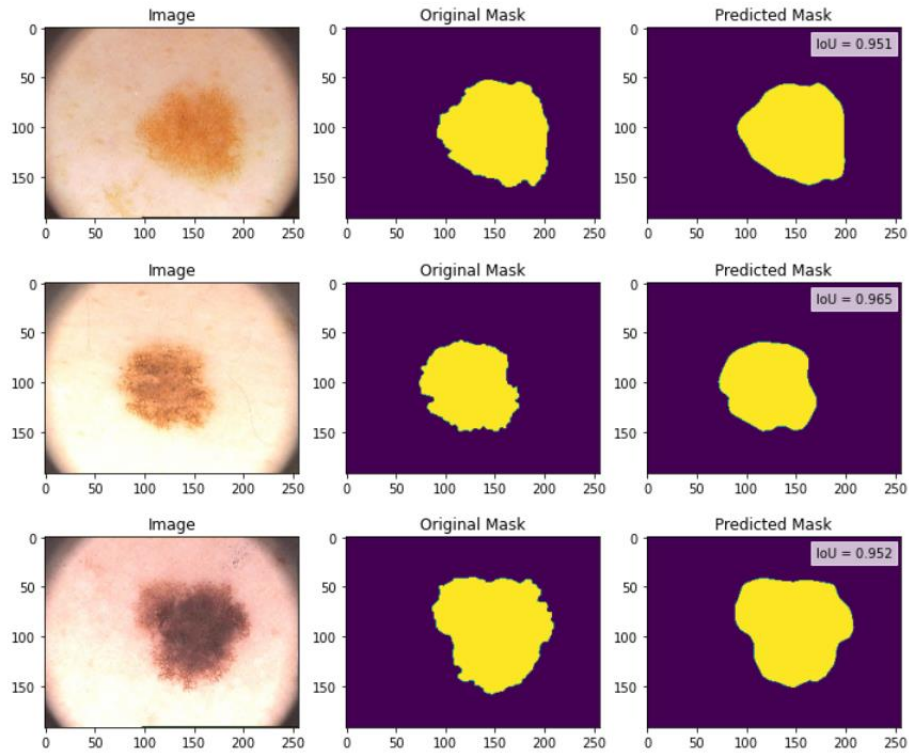


圖 4、方法一條件 12 在測試資料集上的分割結果

5.2. 方法二

參考論文 [4]，嘗試應用 FrCN，然而調整多種 optimizer 與 learning rate 模型都無法訓練起來，測試條件與結果如表 3。觀察訓練過程（如圖 5），loss 一直無法下降，推測可能是因為訓練的資料量不夠多。由於調整多種參數皆無法達到預期的結果，因此決定嘗試其他架構，試著使用 U-Net（方法三）。

表 3、方法二之測試條件與結果

	Optimizer	Learning Rate	mIoU
1	Adam	0.001	0.1670
2	Adam	0.01	0.1670
3	Adadelata	1.0	0.1670
4	SGD	0.1	0.1670

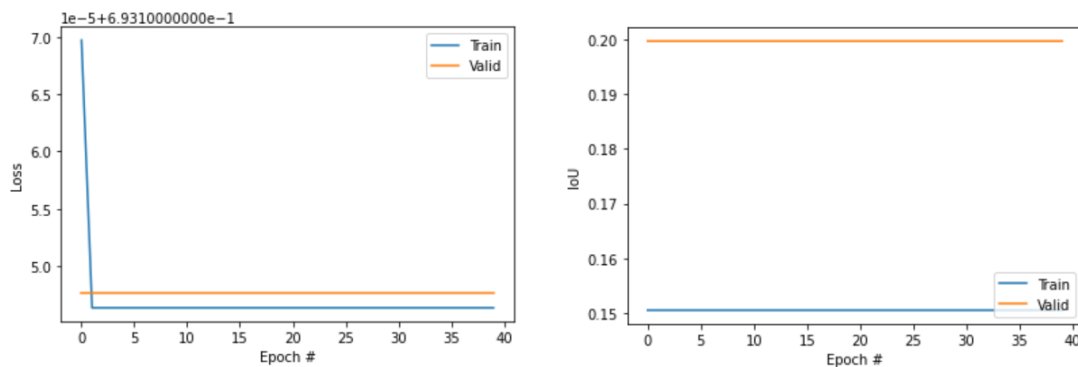


圖 5、方法二條件 2 訓練過程的 loss 與 IoU 變化

5.3. 方法三

由表 4 可觀察到，當 encoder 和 decoder 的 filter 個數和層數增加，在測試資料上平均 IoU 皆降低，可能是 overfitting 的問題。另外，參考 [6]，訓練時改用不同的 loss function，可以觀察到若以 IoU loss 做為 loss function，模型都無法訓練起來，而相較於 binary cross entropy (BCE)，若再加上 IoU，在測試資料的表現上也會變好。針對條件 1、2 與 3 若在資料預處理多加 data augmentation 後，在測試資料上的平均 IoU 都有提升。

綜觀而言，若使用 U-Net，其條件為：encoder 由三層 convolutional layer 組成，filter 個數分別為 16、32 與 64，而 decoder 與 encoder 對稱，訓練資料先以 flipping 和 rotation 前處理，訓練時使用 BCE 與 IoU 之和作為 loss function 時，有最佳的表現，在訓練資料的平均 IoU 為 0.7951，圖 6 為三張測試影像的分割結果。

表 4、方法三之測試條件與結果

	Encoder	Decoder	Data Aug.	Loss Function	mIoU
1	CNN (16,32,64)	CNN (64,32,16)	No	BCE	0.7695
2	CNN (16,32,64)	CNN (64,32,16)	No	IoU	0.1670
3	CNN (16,32,64)	CNN (64,32,16)	No	BCE + IoU	0.7925
4	CNN (64,128,256,512)	CNN (512,256,128,64)	No	BCE	0.6619
5	CNN (64,128,256,512)	CNN (512,256,128,64)	No	IoU	0.1670
6	CNN (64,128,256,512)	CNN (512,256,128,64)	No	BCE + IoU	0.7340
7	CNN (16,32,64)	CNN (64,32,16)	Flip/Rotate	BCE	0.7856
8	CNN (16,32,64)	CNN (64,32,16)	Flip/Rotate	IoU	0.1713
9	CNN (16,32,64)	CNN (64,32,16)	Flip/Rotate	BCE + IoU	0.7951

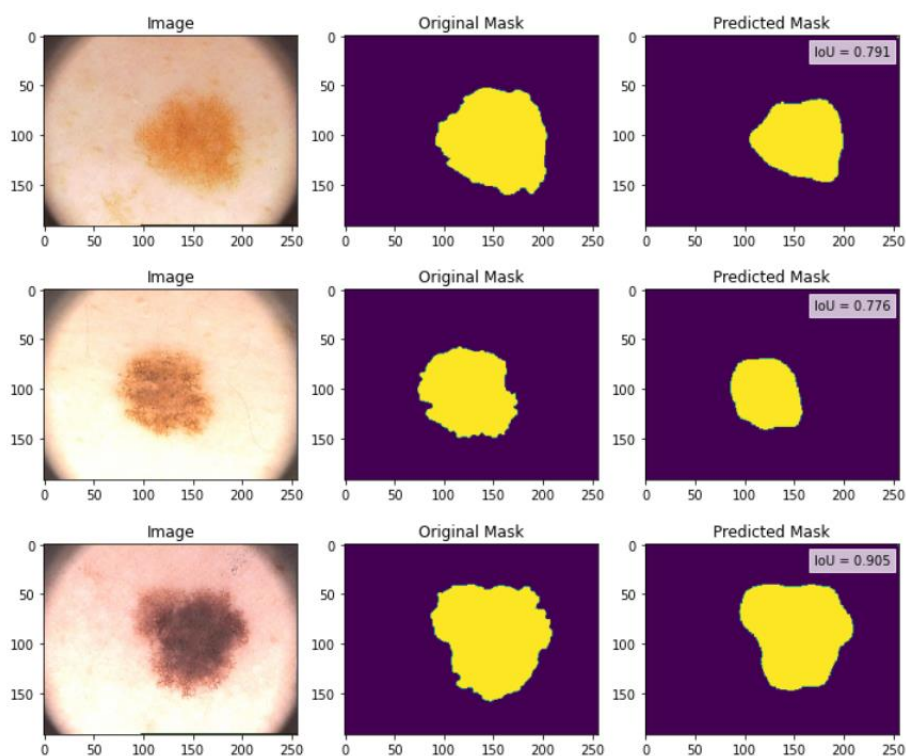


圖 6、方法三條件 9 在測試資料集上的分割結果

5.4. 方法四

參考論文 [6]，首先測試使用 Adadelta optimizer，比較表 5 中條件 1、2 與 5、8，若改用 Adam optimizer，在測試資料可以獲得較高的平均 IoU，因此後續測試皆使用 Adam optimizer。此外，比較在不同 loss function，其他條件都固定的情況，使用 IoU 的模型表現最差，而使用 BCE 或 BCE 與 IoU 之和的模型表現則相當。在此方法中，不進行 data augmentation 而直接取原始資料訓練，大多結果較有做 data augmentation 預處理後佳。

原本論文 [6] 中使用 DenseNet-121 做為 encoder，但由表 5 可以看到，若改為 ResNet-101，在各種條件下，在測試資料上的平均 IoU 皆最高。其中又以不做 data augmentation，並且訓練時使用 BCE 與 IoU 之和作為 loss function 為最好，在測試資料的平均 IoU 為 0.9229，圖 7 為三張測試影像的分割結果。

表 5、方法四之測試條件與結果

	Encoder	Optimizer	Data Aug.	Loss Function	mIoU
1	DenseNet-121	Adadelta	No	BCE + IoU	0.8237
2	DenseNet-121	Adadelta	Flip/Rotate	BCE + IoU	0.8416
3	DenseNet-121	Adam	No	BCE	0.9092
4	DenseNet-121	Adam	No	IoU	0.8838
5	DenseNet-121	Adam	No	BCE + IoU	0.9100
6	DenseNet-121	Adam	Flip/Rotate	BCE	0.8866
7	DenseNet-121	Adam	Flip/Rotate	IoU	0.8478
8	DenseNet-121	Adam	Flip/Rotate	BCE + IoU	0.8799
9	ResNet-101	Adam	No	BCE	0.9159
10	ResNet-101	Adam	No	BCE + IoU	0.9229
11	ResNet-101	Adam	Flip/Rotate	BCE	0.9051
12	ResNet-101	Adam	Flip/Rotate	BCE + IoU	0.9024
13	ResNet-18	Adam	No	BCE	0.8423
14	ResNet-18	Adam	No	BCE + IoU	0.8818
15	ResNet-18	Adam	Flip/Rotate	BCE	0.8704
16	ResNet-18	Adam	Flip/Rotate	BCE + IoU	0.8645

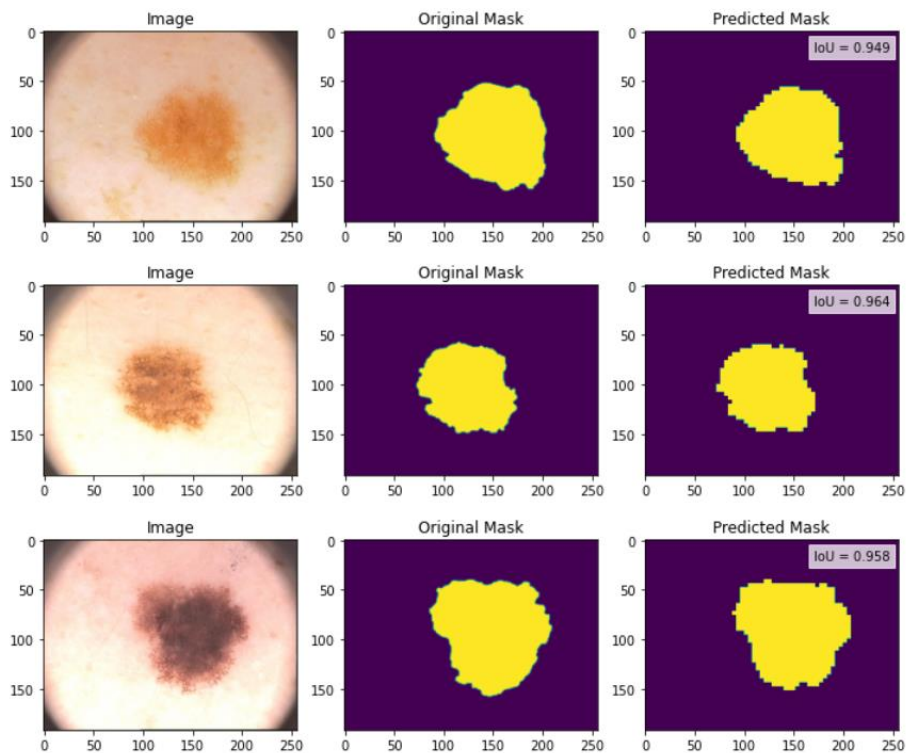


圖 7、方法四條件 10 在測試資料集上的分割結果

6. Reference

- [1] T. Mendonca, P. M. Ferreira, J. S. Marques, A. R. Marcal, and J. Rozeira, "PH² - A Dermoscopic Image Database for Research and Benchmarking," 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2013.
- [2] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
- [3] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," arXiv.org, 05-Dec-2017. [Online]. Available: <https://arxiv.org/abs/1706.05587>.
- [4] M. A. Al-masni, M. A. Al-antari, M.-T. Choi, S.-M. Han, and T.-S. Kim, "Skin Lesion Segmentation in Dermoscopy Images via Deep Full Resolution Convolutional Networks," Computer Methods and Programs in Biomedicine, vol. 162, pp. 221–231, 2018.

- [5] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *Lecture Notes in Computer Science*, pp. 234–241, 2015.
- [6] M. K. Hasan, L. Dahal, P. N. Samarakoon, F. I. Tushar, and R. Martí, “DSNet: Automatic Dermoscopic Skin Lesion Segmentation,” *Computers in Biology and Medicine*, vol. 120, p. 103738, 2020.
- [7] “Torchvision.” [Online]. Available: <https://pytorch.org/vision/0.12/>.
- [8] “Mask Augmentation for Segmentation.” [Online]. Available: https://albumentations.ai/docs/getting_started/mask_augmentation/.