

Rescue the Acne Face

*Deep Learning Final Project

藍翊庭
資工所

wublu0117@gmail.com

黃亮臻
數據所

liangjenh@gmail.com

楊祐昇
資工所

P76121186@gs.ncku.edu.tw

黃書堯
醫資所

david070889@gmail.com

Abstract—青春痘為常見的臉部病症，造成了許多生理與心理問題。此研究旨在提供一套便捷、快速的方法，讓飽受此種病症困擾的人們，可以不需要花費大量精力在尋找、使用各類修圖、相機軟體，一鍵即可獲得一張不再被痘痘所害的清晰臉部圖片。文內比較了許多的方法，像是-Inpainting, GAN, Pix2Pix，並將結果輔以問卷調查，客觀分析提出方法的優缺。程式碼提供於：<https://github.com/Sherry2580/Rescue-the-Acne-Face.git>

I. INTRODUCTION

青春痘是一種常見的皮膚問題，不僅對外貌造成影響，還可能引發一系列生理與心理問題。傳統的痘痘處理方法，包括使用修圖軟體或相機軟體，不僅耗時且效果不一，並無法滿足大多數人的需求。近年來，深度學習技術的快速發展為圖像處理帶來了新的可能性。本研究旨在開發一種便捷、快速且高效的自動化去痘方法，使得使用者能夠輕鬆獲得無痘痘的清晰臉部圖像，而不需要耗費大量時間在繁瑣的修圖操作上。我們探討並比較了多種方法，包括-CycleGAN [1] [2]、YOLO v8 [3] 結合濾波及圖像修復技術 (Inpainting)，以及 Pix2Pix [2] [4]，並通過實驗和問卷調查來評估這些方法的效果與實用性。

II. THE PROPOSED METHOD

在此研究中，我們主要使用四種方法評估痘痘消除的成效。首先，我們使用 CycleGAN 將不同人的痘痘照片和無痘照片進行轉換（不成對資料集），這是一種 One Stage 的方法。此外，我們也嘗試了兩種 Two Stage 的方法：一種是先進行物件偵測，接著使用濾波器將痘痘位置平滑化處理；另一種是物件偵測後，利用 Inpainting 技術對痘痘位置進行圖像修復。最後，我們在痘痘照片上添加假痘痘，以及使用 Inpainting 技術得到去痘照片，獲得同一人的有痘及無痘照片後（成對的資料集），使用 Pix2Pix 進行轉換，這也是一種 One Stage 的方法。具體的流程圖可參見圖1。

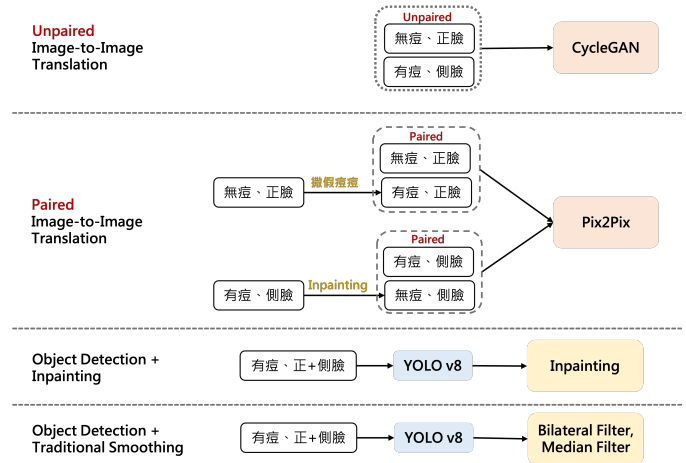


Fig. 1. 方法流程圖

A. CycleGAN

CycleGAN 是一種用於圖像到圖像轉換的生成對抗網絡 (GAN)，其核心技術包括兩個生成器 (Generators) 和兩個判別器 (Discriminators)，分別負責將圖像從一個域轉換到另一個域，再將其轉回來。實現 CycleGAN 的關鍵在循環一致性損失 (Cycle-Consistency Loss)，它能確保轉換後的圖像跟原始圖像的一致性。在一開始，因為我們沒有成對的痘痘和無痘痘的臉部圖像，CycleGAN 又可以在沒有成對數據的情況下學習兩種圖像域之間的映射關係，所以選擇 CycleGAN 作為主要模型。

我們最初的目標是利用 CycleGAN 將有痘痘的臉部圖像轉換為無痘痘的臉部圖像。我們嘗試使用將痘痘的位置作為第四個通道，並將輸入圖像表示為 RGB+A 格式，其中“A”表示痘痘特徵通道。然而，由於正臉和側臉圖像的不對稱性以及數據集的限制，這個方法未能取得理想的效果。正臉圖像和側臉圖像在特徵分佈上存在顯著差異，導致 CycleGAN 難以有效學習到有意義的轉換關係。

B. YOLO v8

在當前物件偵測技術領域，方法主要分為一階段 (one-stage) 和二階段 (two-stage)。YOLO (You Only Look Once) 是一種著名的一階段模型。YOLO v8 [3] 使用自適應錨點盒 (anchor box) 設計，根據物體的形狀和尺寸動態調整錨點盒，提高了邊界預測的精確性。此外，它也採用了深度特徵金字塔融合策略，增強小物體的檢測能力。在本次實驗中，我們使用 YOLO v8 模型偵測出的痘痘位置，並將這些位置數據提供給 Inpainting 和 Filter 進行後續處理。

C. Filter

對於較傳統的影像處理中，我們選擇了 median filter [5] 和 bilateral filter [6] 嘗試消除經由 YOLO v8 所偵測的類似噪點的痘痘。第一個方法 median filter 的做法為 apply 一個 mask 到圖片中，mask 中獲得的數值將其從小排到大，取其中位數放入 mask 最中間的 pixel，持續將範圍內的 pixel 都使用 apply 過一次，即完成一張圖片的 median filter 處理。第二個 bilateral filter 則是類似 Gaussian filter 的改良版，在 spatial 的基礎上，加入像素差的權重並作出正規化，所以結果會比 Gaussian filter 更加平滑。

D. Image Inpainting

參考 Hugging Face [7] 中的使用方法，首先我們將圖片送進訓練好的 YOLO v8 模型中，接著將 YOLO v8 輸出結果轉化成座標位置，根據產生的座標位置與 conf 限制生成 Mask Image，將原始圖片與 Mask Image 與希望 inpaint 的結果文字敘述送入已經預先訓練好的 Inpainting 模型中並收集其輸出之結果圖，我們一共使用了兩種 Inpainting 模型，分別為 kandinsky2.2 與 stable-diffusion。

E. Pix2Pix

由於在實驗初期，CycleGAN 的結果不盡理想，於是在考量到可行性之後，我們決定轉為使用 Pix2Pix 來實現去除痘痘的方法。與 CycleGAN 不同的是，Pix2Pix 是一種條件生成對抗網絡 (Conditional GAN)，適用於成對的圖像到圖像轉換任務。Pix2Pix 需要成對的訓練數據，意即輸入圖像和目標圖像之間存在明確的對應關係，整個模型包括一個生成器和一個判別器，其中生成器負責將輸入圖像轉換為目標圖像，而判別器則負責區分生成的圖像和真實的目標圖像。在訓練過程中，對抗損失 (Adversarial Loss) 用於提高生成圖像的真實性，而 L1 損失 (L1 Loss) 則用於保證生成圖像與目標圖像在像素級的相似性，這種結合使得 Pix2Pix 能夠在保持圖像細節的同時生成較高質量

TABLE I
DATASETS

Type	Dataset	Use For	Num	Split Ratio
痘痘正臉	acne-new Dataset	Train YOLO v8 Inference on all models (Choose #acne > 30)	3239	85/10/5
痘痘側臉	ACEN04 dataset	Train Pix2Pix Inference on all models	330 out of 3199	97/x/3
無痘正臉	Dataset6	Train Pix2Pix	541 out of 1841	100/x/x

的圖像轉換。由於我們的需求是想要在微小的地方做圖像轉換 (去痘)，於是我們發想了能夠生成成對的痘痘和無痘痘的臉部圖像的方法，並在解決了沒有成對數據集的問題之後，選擇改用 Pix2Pix 作為替代 CycleGAN 的模型。

III. EXPERIMENTS

A. Dataset

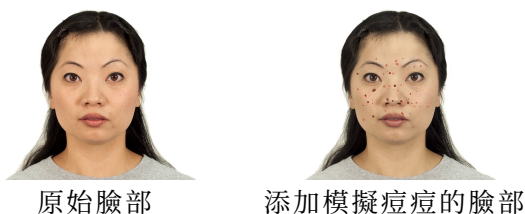
表 I 為我們使用的 3 個資料集。acne-new Dataset [8] 主要用於訓練物件偵測模型: YOLO v8，其資料集包含五個類別: Acne, Mole, Pimple, Scar, Spot，內容多為痘痘數沒有很多的正臉生活照。在這裡我們將偵測到的所有類別，都視為要做清除的痘痘區域，並另外取偵測大於 30 顆痘痘的正臉照片作為所有模型的測試資料。

ACEN04 dataset [9] 與 Dataset6 [10] 主要用於訓練 Pix2Pix 模型。ACEN04 dataset 為側臉的痘痘特寫照，而 Dataset6 則為無痘痘的正臉照。我們特別選取了痘痘分級為 1 級與 2 級 (較嚴重) 的照片，並涵蓋了不同種族的人像，如印度人和亞洲人等，以增加模型的適應性。此外，我們也隨機選取了一些側臉的痘痘照作為測試資料，以評估模型的表現。

B. Preprocessing

為了使我們採用的數據集符合 Pix2Pix 模型的格式，我們對數據集進行了預處理，包括挑選適合的照片、剔除不合適的照片，並生成帶有人工痘痘的圖像以進行訓練等等。經過逐步的實驗，發現對於 Pix2Pix 模型來說，數據集的質量將會直接影響到模型的訓練效果和最終的生成結果。我們在過程中實現了以下幾個步驟：

1. **添加痘痘到臉部：**我們使用隨機噪聲來模擬人類真實痘痘，這樣可以生成大量有痘痘的臉部圖像，這些圖像與無痘痘的圖像成對，為 Pix2Pix 模型提供訓練數據。這一步驟通過在無痘痘的臉部圖像上添加紅色和其他相似於痘痘顏色的斑點來實現，並且避開眼部周圍，這些斑點的大小、形狀和位置 (除了避開眼周) 都是隨機的。



原始臉部

添加模擬痘痘的臉部

2. **裁剪和調整圖像大小**：由於原始圖像的尺寸和構圖各不相同，我們需要對其進行統一處理。這包括按照設置的裁剪範圍將圖像裁剪至統一大小，以便於後續的模型訓練。

3. **更改背景顏色**：我們注意到，數據集中的大部分圖像背景都是白色的，這會在訓練過程中導致生成的圖像中出現不自然的白斑。為了解決這個問題，我們隨機生成了一些低飽和度的顏色作為背景，以提高生成圖像的質量和自然度。

通過這些預處理步驟，我們確保了數據集的質量和多樣性，讓後續的模型訓練能順利進行。

C. Experimental Setup

1) *YOLO v8*: 在物件偵測任務中，我們使用 YOLO v8 作為預訓練模型，並以表 I 中的痘痘正臉照片作為訓練資料。為了優化訓練過程，我們設定了最多 250 個訓練週期 (epochs)，並實施了早停 (early stopping) 機制，設定條件為如果在連續 100 個訓練週期中模型性能沒有進一步改善，則提前終止訓練。實際訓練中，在第 232 個訓練週期時，由於沒有觀察到顯著性能提升，訓練便根據早停條件提前結束。

2) *Inpainting*: 我們使用共 800 張圖片，將 YOLO v8 conf threshold 設置為 0，為了再之後可以實驗 Inpainting 在不同 conf 值下的表現，並針對 kandinsky2.2 與 stable-diffusion 在不同 conf 值下的結果進行比較分析，其中 conf 值設定分別為 0、0.05、0.1、0.3，以肉眼形式觀察其效果。

3) *Filter*: 將 YOLO v8 中獲得的座標取出，並使用 opencv [11] 中提供的 cv2.bilateralFilter() 與 cv2.medianBlur()，並觀察兩種方法的平滑度與去噪效果。

4) *Pix2Pix*: 在實驗中，我們進行了多次嘗試和調整，以找到最佳訓練方法和參數。

初期全部都使用 Dataset6 數據集的正臉照片（調整大小為 256x256）進行訓練，但生成的圖像因背景為白色而出現白斑。改為隨機背景色後，效果仍不佳，我們認為是因為資料集過於單一。為提升模型泛化能力，我們加入經過 inpainting 處理後的 330 張側臉痘痘照片，與正臉照片合併，形成 871 張成對的圖像。

訓練 700 個 epochs 後，還有約 30% 的痘痘未能去除，且有雜訊問題。

而後我們嘗試用不同尺寸的圖像進行訓練。1024 x 1024 大小的圖像雖然可以讓畫質提高，但去除痘痘的效果較差。此外，為了提升訓練速度，還嘗試將 Batch size 上調至 64，但效果極差，因此維持 batch size = 1。最終，我們比較了兩種方法：

1. 使用 871 張 512x512 大小的圖像，訓練 900 個 epochs。

2. 使用 871 張 256x256 大小的圖像，訓練 900 個 epochs。

結果顯示，case 1 呈現出的圖像效果最佳，去痘效果達到 80% 至 90%，且維持了一定的清晰度。所以最終我們選擇了第一個 case 的結果，並與其他方法進行問卷調查和比較。

5) 問卷設計：從測試集中隨機選取 10 張正臉生活痘痘照及 10 張側臉痘痘照，共計 20 張照片。根據這些照片，我們設計了一份包含 5 張照片的問卷，分成四份不同的版本進行調查。從每份問卷中，我們分別收集到了如下數量的回答：A 卷 57 份、B 卷 43 份、C 卷 29 份、D 卷 27 份，共 156 份問卷。在這個過程中，我們詢問了以下三個問題：

1. 去除效果

請評價痘痘和痘疤去除的完成度。是否還可以看到瑕疵？

2. 自然度

處理後皮膚與原皮膚相比，是否顯得和諧？還是存在不自然的過渡、紋理變化？

3. 真實度

整張圖片與原照片是否保持一致？有無非預期的效果，如雜點、異物、陰影、意外的鬍子或紅唇，或模糊不清？

D. Results

1) *CycleGAN*: 圖 2 展示了一些使用 CycleGAN 失敗的例子。我們認為可能的原因是由於收集到的數據不夠多樣，且 CycleGAN 本身就比較適合大範圍的風格轉換，而不是針對像痘痘這種小地方，以至於實驗出來的結果不理想，最後也決定放棄此方法。

2) *YOLO v8*: 圖 3 展示了使用 YOLOv8 模型對痘痘進行物件偵測的結果，並根據不同的置信度 (confidence) 閾值進行標註。從左到右依次顯示了原始圖像以及置信度閾值分別為 0.01、0.05 和 0.1 時的偵測結果。隨著置信度閾值的降低，更多的痘痘被偵測出來，但也可能有一些非痘痘的區塊被誤判為痘痘。反之，當閾值提高時，偵測的準確性增加，但一些痘痘可能會被遺漏。

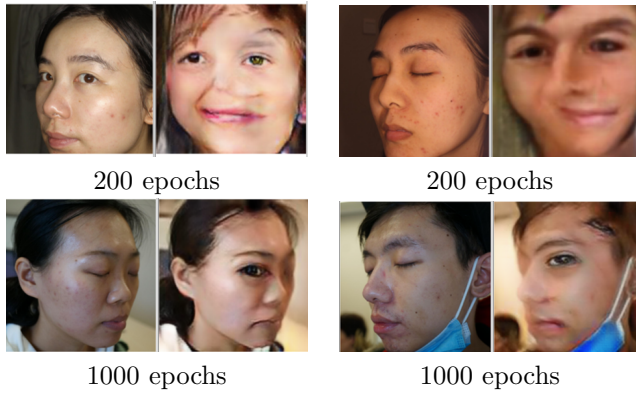


Fig. 2. CycleGAN 實驗結果

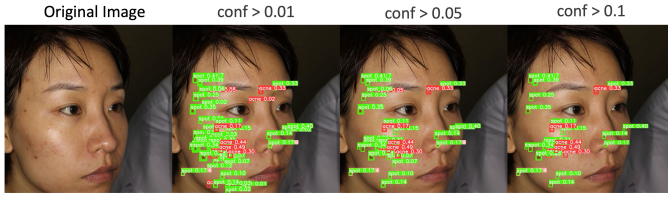


Fig. 3. Object Detection 實驗結果

3) *Inpainting*: 由圖4與圖5可以觀察到，在進行各項 conf 值實驗之後，在模型的比較上 stable-diffusion 所生成的圖片效果遠大於 kandinsky2.2。在 kandinsky2.2 中很容易產生不相關的色塊或者是不自然的光線腮紅，導致人臉看起來非常不自然，而在 stable-diffusion 中不會產生這個問題，整體效果看起來也相當自然，至於在不同 conf 值上的表現我們發現當 conf 值設定小於 0.05 時其痘痘的去除效果最好並且與大於 0.05 的 conf 值效果具有視覺上的差異，最終我們使用 stable-diffusion、conf 值為 0，作為我們的 Inpainting model 並將結果放置在問卷中進行測試。



Fig. 4. Inpainting 實驗結果 (好)

4) 痘痘消除方法的效果比較: 圖6展示了不同痘痘消除方法的效果對比，包括原始圖像、Median Filter、Inpainting 和 Pix2Pix 的結果。從比較中可以看出，Median Filter 使痘痘的可見性有所減少，但存在著不自然的馬賽克效果。Inpainting 使皮膚看起來更加光滑且有色澤，但有時會給人美肌開過強的感覺。

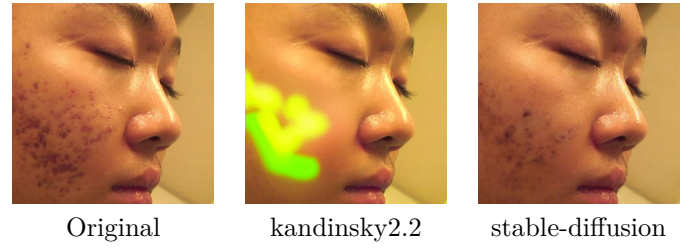


Fig. 5. Inpainting 實驗結果 (不穩定)

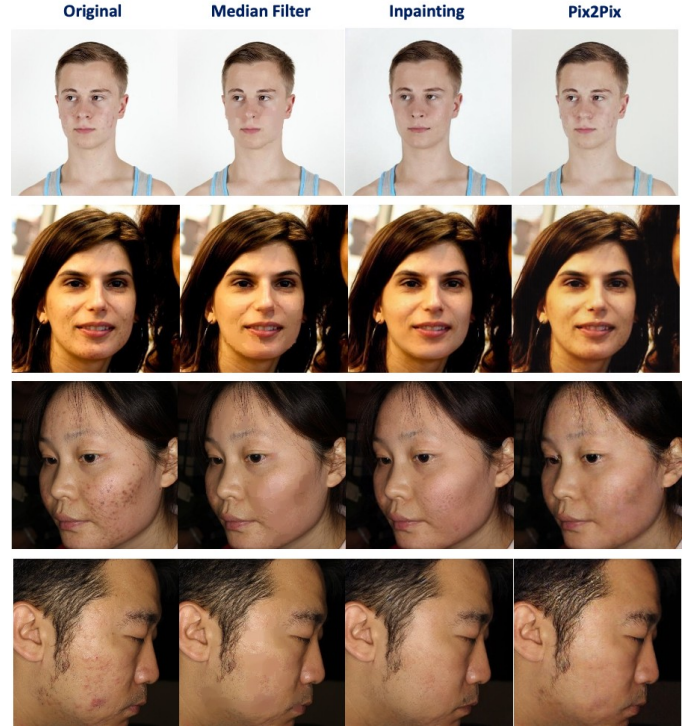


Fig. 6. 痘痘消除方法的效果比較

Pix2Pix 方法雖然沒有將痘痘清理得非常乾淨，但效果相對自然，突兀的區塊較少。

TABLE II
RESULTS: 正臉痘痘生活照

	Median Filter	Inpainting	Pix2Pix
去除效果	3.55 ± 0.98	4.10 ± 0.66	2.83 ± 0.98
自然度	2.53 ± 1.13	3.73 ± 0.90	3.22 ± 0.96
真實度	2.58 ± 1.11	3.63 ± 0.95	3.21 ± 1.00

5) 問卷結果: 由表 II, III, IV 可得知，整體來看，Inpainting 方法在去除效果、自然度、真實度各個方面都有比較好的表現，另外在 Pix2Pix 與 Median Filter 的比較中雖然 Median Filter 在側臉與整體的去除效果稍微比 Pix2Pix 來的好但是在自然度與真實度上卻遠低於 Pix2Pix，綜合三項指標來看效果排序分別為 Inpainting、Pix2Pix、Median Filter，另外從執

TABLE III
RESULTS: 側臉痘痘特寫照

	Median Filter	Inpainting	Pix2Pix
去除效果	3.19 ± 1.07	3.80 ± 0.87	3.40 ± 0.90
自然度	2.16 ± 0.97	3.73 ± 0.81	3.27 ± 0.97
真實度	2.23 ± 1.00	3.67 ± 0.87	3.27 ± 0.98

TABLE IV
RESULTS: 全部痘痘照

	Median Filter	Inpainting	Pix2Pix
去除效果	3.38 ± 1.04	3.96 ± 0.78	3.10 ± 0.99
自然度	2.36 ± 1.07	3.73 ± 0.86	3.24 ± 0.97
真實度	2.42 ± 1.07	3.65 ± 0.92	3.24 ± 0.99

行層面考量，Inpainting、Median Filter 會需要經過兩層模型且無法平行化處理，而 Pix2Pix 僅須一層，用戶可根據自身需求來決定適合的方法。

IV. CONCLUSIONS

本研究探討了多種深度學習技術在痘痘去除上的應用，包括 CycleGAN、YOLO v8 結合 filter 與 inpainting，以及 Pix2Pix。通過實驗結果與問卷調查，我們發現 Inpainting 方法在去除效果、自然度和真實度方面表現最佳，Pix2Pix 方法則在簡便性與可實現性方面具有優勢，而 Median Filter 雖然在去除效果上稍微領先，但其自然度與真實度相對較低。綜合考慮各方法的效果與執行層面的便捷性，使用者可以根據自身需求選擇最適合的方法。未來的研究可以進一步改進數據集的多樣性和模型的泛化能力，以提高去痘效果和應用範圍。

REFERENCES

- [1] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.
- [2] J.-Y. Zhu and T. Park, "pytorch-cycleGAN-and-pix2pix," <https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>, 2020.
- [3] Ultralytics, "Yolov8 documentation," <https://docs.ultralytics.com/models/yolov8/>, 2023.
- [4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.
- [5] T. Huang, G. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 1, pp. 13–18, 1979.
- [6] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 1998, pp. 839–846.

- [7] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Huggingface's transformers: State-of-the-art natural language processing," 2020.
- [8] BuyumeDatasets, "acne-new dataset," <https://universe.roboflow.com/buyumedatasets/acne-new>, nov 2023, visited on 2024-06-15. [Online]. Available: <https://universe.roboflow.com/buyumedatasets/acne-new>
- [9] M. Hettich, "acne04 dataset," https://www.kaggle.com/datasets/manuelhettich/acne04?select=acne_1024, feb 2024. [Online]. Available: https://www.kaggle.com/datasets/manuelhettich/acne04?select=acne_1024
- [10] Wedyan78, "Dataset6," <https://www.kaggle.com/datasets/wedyan78/dataset6>, mar 2024. [Online]. Available: <https://www.kaggle.com/datasets/wedyan78/dataset6>
- [11] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.