ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

----------o0o----------

BÁO CÁO THỰC TẬP TỐT NGHIỆP

Đề tài: VIRTUAL PAINTER

GVHD: Ths. Nguyễn Khánh Lợi

(nkloi@hcmut.edu.vn)

Sinh viên thực hiện: Phan Nguyên Trung – 1814519

(trung.phantrung.phan@hcmut.edu.vn)

Hồ Chí Minh, 08/2021

Mục lục.

Phụ lục hình ảnh.

Tóm tắt nội dung

Trong những năm gần đây, các ứng dụng về trí tuệ nhân tạo ngày càng phát triển và được đánh giá cao. Một lĩnh vực đang được quan tâm của trí tuệ nhân tạo nhằm tạo ra các ứng dụng thông minh, mang tri thức của con người đó là nhận dạng. Trong báo cáo này, em xin trình bày một phương pháp học máy nhận diện cử chỉ tay trong thời gian thực (real time).

Từ khóa: MediaPipe, machine learning, computer vision.

1. Giới thiệu

Ngày nay, khả năng nhận biết hình dạng và chuyển động của bàn tay là một thành phần quan trọng trong việc cải thiện trải nghiệm người dùng trên nhiều lĩnh vực và nền tảng công nghệ khác nhau. Ví dụ: nó có thể tạo cơ sở cho việc hiểu ngôn ngữ ký hiệu và điều khiển cử chỉ tay. Khả năng nhận thức mạnh mẽ của bàn tay trong thời gian thực (real-time) là một nhiệm vụ khó khăn trong thị giác máy tính (computer vision) vì bàn tay thường tự đan hoặc chạm vào nhau. Một vấn đề khác là sự thiếu các tương phản cao.

Trong báo cáo cuối kì này, em xin trình bày một phương pháp nhận diện bàn tay được triển khai trong MediaPipe[1]. Phương pháp này cung cấp khả năng theo dõi ngón tay và bàn tay với độ trung thực cao bằng cách sử dụng máy học (Machine Learning) để tạo ra 21 điểm 3D của bàn tay chỉ từ một khung hình duy nhất. Đồng thời ứng dụng hand tracking thực hiện họa sĩ ảo (Vitural Painter).

2. Mô hình

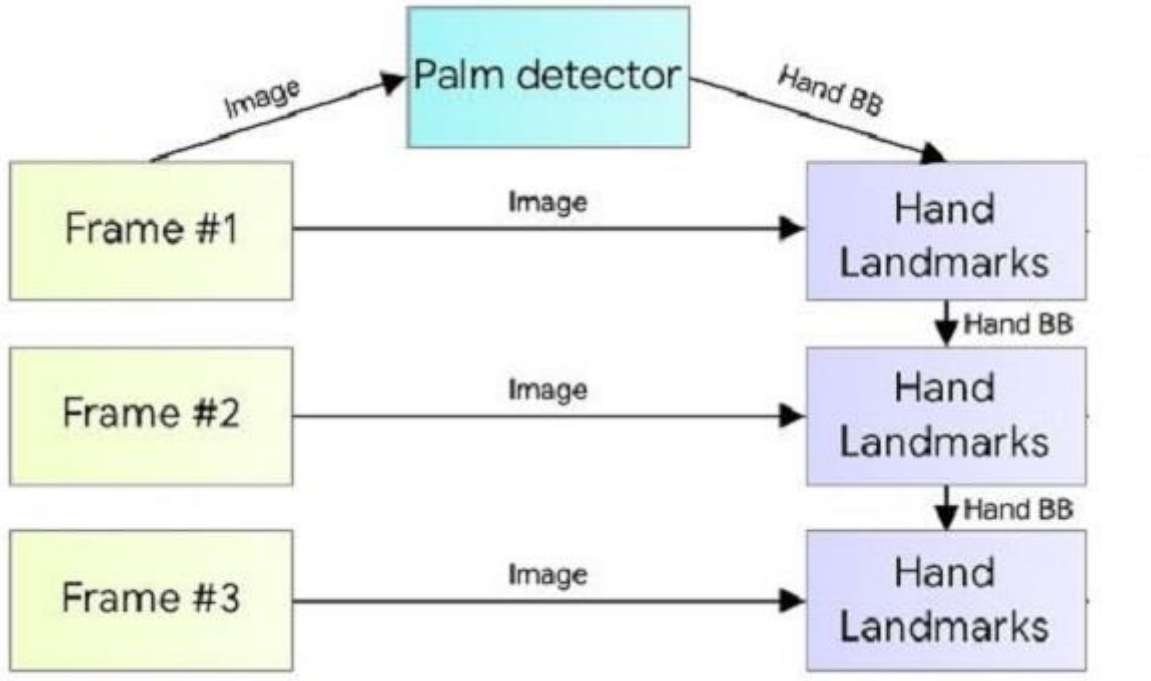
MediaPipe Hands sử dụng nhiều một pipeline bao gồm nhiều mô hình hoạt động cùng nhau để xác định vị trí bàn tay và 21 điểm mốc 3D của bàn tay. Sau các quá trình đó, việc xác định cử chỉ tay sẽ được tiến hành. Việc cung cấp hình ảnh lòng bàn tay chính xác cho mô hình mốc bàn tay giúp giảm đáng kể nhu cầu tăng dữ liệu. Điều đó cho phép mạng dành phần lớn dung lượng của mình cho độ chính xác của các tọa độ dự đoán.

Các mô hình để thực hiện những việc trên bao gồm :

- Mô hình xác định lòng bàn tay (BlazePalm) hoạt động trên hình ảnh đầy đủ và trả về một khung hình (bounding box) giới hạn bàn tay có định hướng,
- Mô hình xác định điểm mốc bàn tay hoạt động trên vùng hình ảnh đã cắt được xác định bởi mô hình xác định lòng bàn tay ở trên (BlazePalm) và nó trả về điểm mốc bàn tay 3D có độ trung thực cao.

[1] MediaPipe: Một khung nền tảng chéo mã nguồn mở được dùng trong việc xây dựng pipline xử lý dữ liệu cảm nhận của các phương thức khác nhau như là video, audio.

• Một trình nhận dạng cử chỉ phân loại các điểm mốc tay đã được tính toán và cấu hình từ tập hợp các cử chỉ rời rạc.



Hình 2.1 Hands Pipeline

Để hiểu rõ hơn về cấu trúc và cách thực hiện của mô hình, chúng ta sẽ đi tìm hiểu từng thành phần .

2.1. Mô hình xác định lòng bàn tay trong thời gian thực BlazePalm.

Mô hình BlazePalm sẽ giúp chúng ta trong việc phát hiện vị trí ban đầu của bàn tay. Phát hiện tay là một nhiệm vụ phức tạp: mô hình phải làm việc được trên nhiều kích cỡ bàn tay khác nhau với khoảng tỷ lệ lớn so với khung ảnh. Trong khi khuôn mặt có các các kiểu tưởng phản cao, ví dụ như ở vùng mắt và miệng thì việc thiếu các đặc điểm ở tay khiến cho việc phát hiện chúng chính xác từ các điểm thị giác là tương đối khó khăn. Thay vào đó, việc cung cấp ngữ cảnh bổ sung, chẳng hạn như các đặc điểm cánh tay, cơ thể hoặc người, hỗ trợ xác định bàn tay chính xác hơn.

Giải pháp định vị được sử dụng ở trên cũng là một công việc khá khó khăn. Đầu tiên, chúng ta sẽ train một mô hình xác định lòng bàn tay thay vì bàn tay bởi vì việc ước tính các bounding box của lòng bàn tay đơn giản hơn đáng kể so với phát hiện bàn tay. Ngoài ra vì

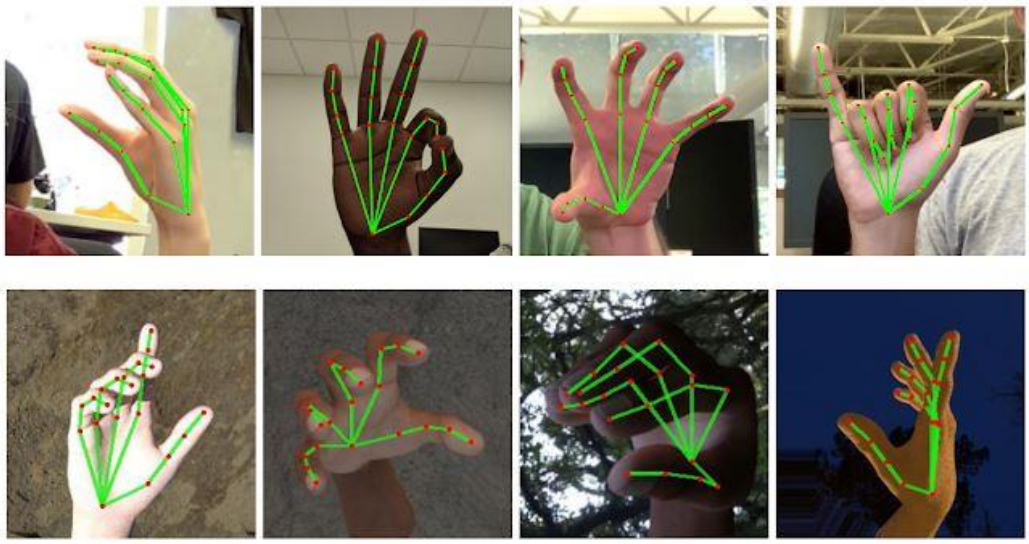
lòng bàn tay là vật nhỏ hơn, thuật toán có thể hoạt động tốt ngay cả khi các bàn tay tương tác với nhau (ví dụ như bắt tay). Hơn nữa, bàn tay có thể được mô hình hóa sử dụng bounding box hình vuông. Sau khi xác định lòng bàn tay, bộ mã hóa- giải mã trích xuất các đặc trưng được sử dụng để nhận biết ngữ cảnh lớn hơn. Cuối cùng là việc tối thiểu các focal loss trong việc train để hỗ trợ một lượng lớn anchors do phương sai tỷ lệ cao.

Với kỹ thuật trên, mô hình đạt được precision trung bình khoảng 95.7% trong việc phát hiện bàn tay.

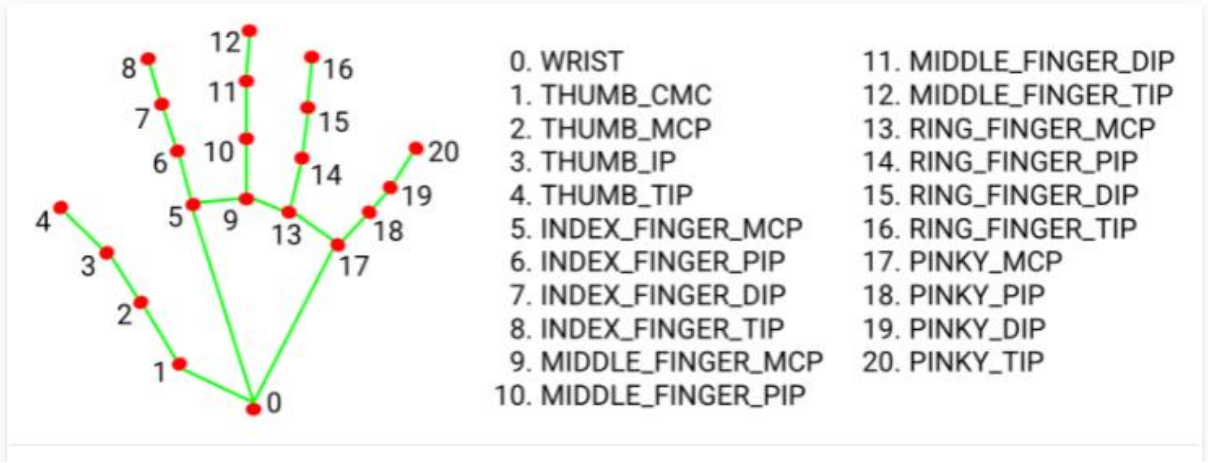
2.2. Mô hình Hand Landmark

Sau khi phát hiện lòng bàn tay trên toàn bộ hình ảnh, mô hình mốc bàn tay (Hand Landmark) sẽ thực hiện định vị chính 21 điểm tọa độ đốt ngón tay 3D bên trong các vùng bàn tay được phát hiện thông qua hồi quy. Mô hình học cách thể hiện tư thế bàn tay nhất quán và hiệu quả ngay cả khi chỉ nhìn thấy một phần bàn tay hoặc khi bàn tay co lại.

Để có dữ liệu sát với thực tế, mô hình đã tự chú thích thủ công 21 tọa độ 3D trong xấp xĩ 30 000 hình ảnh. Để bao quát tốt hơn các tư thế bàn tay có thể có và cung cấp bổ sung các hình dạng tự nhiên của bàn tay, mô hình được cung cấp thêm các mô hình tay tổng hợp trên nhiều nền khác nhau và ánh xạ nó tới các tọa độ 3D tương ứng.

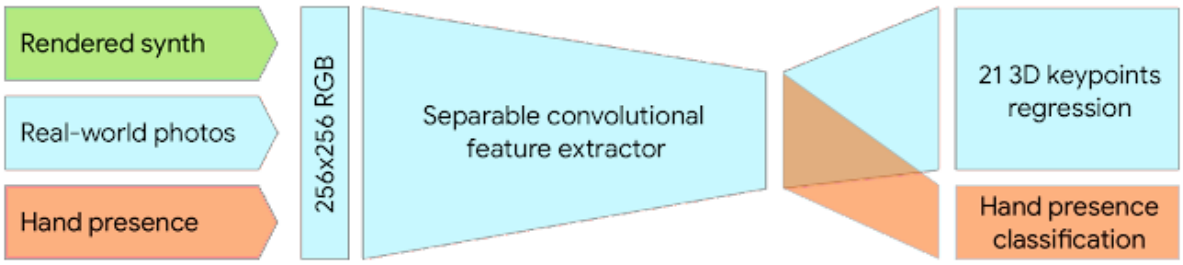


Hình 2.2 Một số mô hình 3D các tư thế tay.



Hình 2.3 Sơ đồ các điểm mốc của bàn tay.

Tuy nhiên, dữ liệu tổng hợp có tính khái quát thấp đối với các miền hoang dã. Để cải thiện vấn đề này, các nhà phát triển sử dụng một lược đồ đào tạo hỗn hợp. Sơ đồ training mô hình được trình bày trong các hình sau.



Hình 2.4 Lược đồ training cho mạng xác định cử chỉ tay.

Bảng dưới đây tóm tắt regression accuracy dựa vào bản chất dữ liệu training. Có thể thấy sử dụng kết hợp dữ liệu của hình thái tay và hình ảnh tay sẽ cho kết quả tốt nhất.

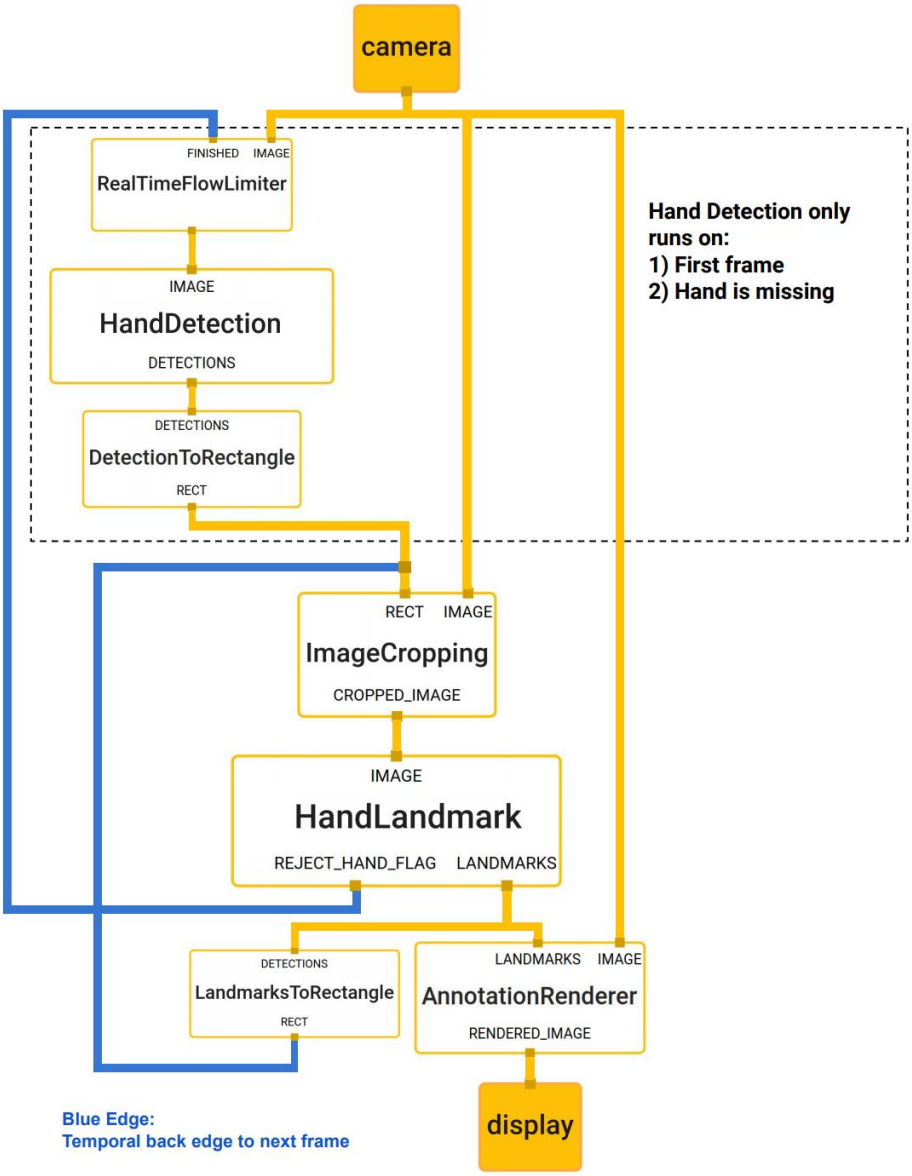
Dataset	Mean regression error normalized by palm size
Only real-world	16.1 %
Only rendered synthetic	25.7 %
Mixed real-world + synthetic	**13.4 %**

Hình 2.5 Regression Accuracy.

2.3. Thực hiện mô hình với MediaPipe

MediaPipe được xây dựng dưới dạng đồ thị có hướng của các thành phần mô đun đu kèm với một bộ máy tính có thể mở rộng để giải quyết các tác vụ như suy luận mô hình, thuật toán xử lý phương tiện và chuyển đổi dữ liệu trên nhiều loại thiết bị và nền tảng.

Biểu đồ MediaPipe để theo dõi bàn tay của được hiển thị bên dưới. Biểu đồ bao gồm hai đồ thị con. Một đồ thị để phát hiện bàn tay và một đồ thị để tính toán các điểm mốc . Một tối ưu hóa quan trọng mà MediaPipe cung cấp là máy dò lòng bàn tay chỉ được chạy khi cần thiết (khá hiếm khi xảy ra), tiết kiệm đáng kể thời gian tính toán. Chúng em đạt được điều này bằng cách suy ra vị trí bàn tay trong các khung video tiếp theo từ các điểm chính của bàn tay được tính toán trong khung hình hiện tại, loại bỏ nhu cầu chạy máy dò lòng bàn tay trên mỗi khung hình. Để mạnh mẽ, mô hình theo dõi bàn tay xuất ra một đại lượng vô hướng bổ sung để đảm bảo độ tin cậy rằng một bàn tay có mặt và được căn chỉnh hợp lý trong phần cắt đầu vào. Chỉ khi độ tin cậy giảm xuống dưới một ngưỡng nhất định thì mô hình phát hiện bàn tay mới được áp dụng lại cho toàn bộ khung.



Hình 2.6 Biểu đồ MediaPipe.

3. Vitural Painter sử dụng Hand Tracking

Như đã giới thiệu ở trên, chúng em sẽ ứng dụng Hand Tracking để thực hiện Vitural Painter. Để dễ hình dung, công việc sẽ được mô tả khái quát như sau:

Đầu tiên chúng ta sẽ theo dõi bàn tay của mình và lấy các điểm mốc của nó. Sau đó sử dụng các điểm để vẽ trên màn hình. Chúng em sử dụng 2 ngón tay để thực hiện chọn các chế độ như chọn mực vẽ, tẩy. Sử dụng một ngón tay trỏ để vẽ hình.

Tất cả những hoạt động này đều được thực hiện trong thời gian thực

Hình 3.1 Virtual Painter

Các bước thực hiện.

Bước 1: Trích xuất hình ảnh từ webcam.

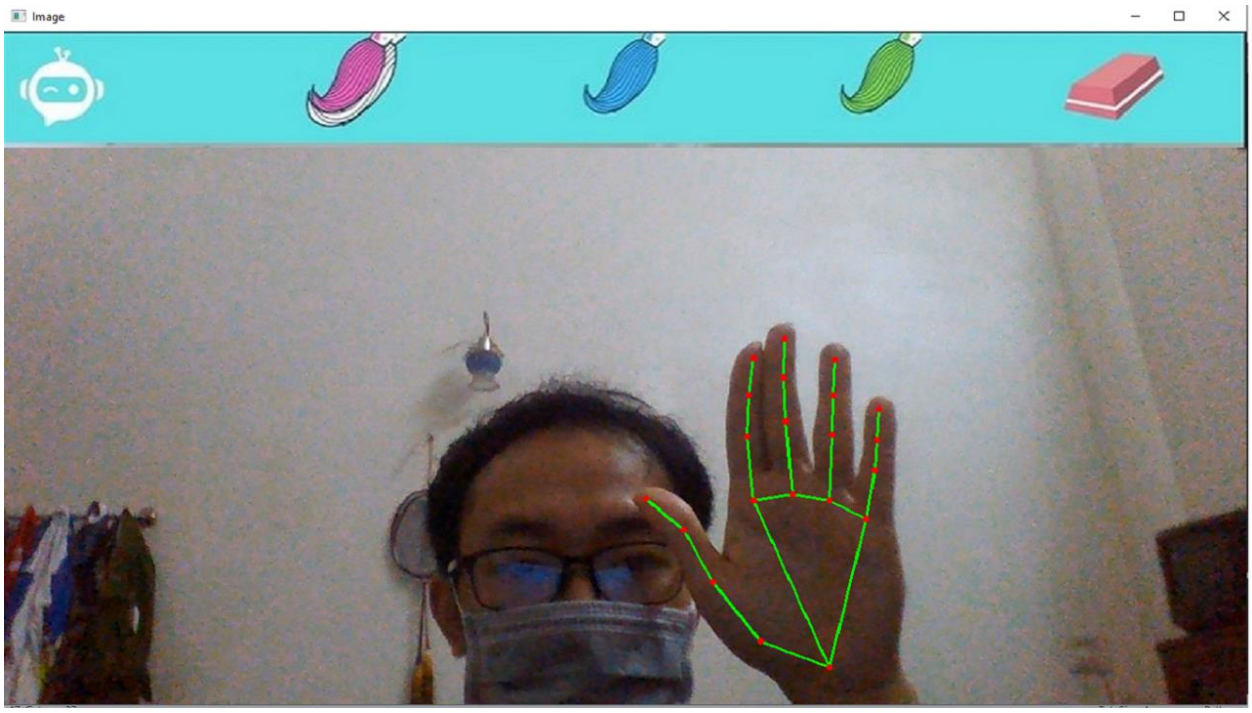
Khung hình được trích xuất từ webcam có kích thước là 3x1280x720. Vì webcam sẽ cho hình ngược với thực tế nên để người dùng dễ thao tác trong thời gian thực thì hình sẽ được Filp trước khi đưa vào xử lý.

Hình 3.2 Hình trước khi được Flip.

Hình 3.3 Hình sau khi được Flip.

Bước 2: Tìm các mốc điểm tay

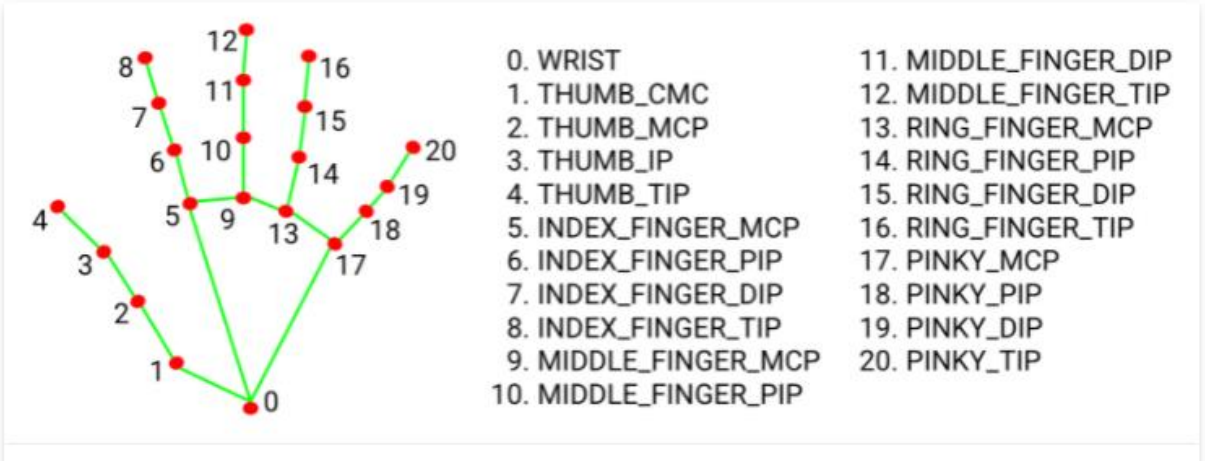
Như đã giới thiệu ở trên, ta sẽ áp dụng MediaPipe trong việc xác định các điểm mốc tay. Quá trình này gồm hai giai đoạn. Đầu tiên sẽ tìm vị trí của bàn tay trong khung hình được trích xuất ở bước 1. Tiếp theo, từ bàn tay đã tìm, 21 điểm mốc tay sẽ được xác định với mỗi điểm tay là một tọa độ được đánh dấu màu đỏ trên khung hình.



Hình 3.4 Xác định các điểm mốc của bàn tay.

Bước 3: Xác định ngón tay dơ lên.

Việc xác định ngón tay dơ lên thông qua một thuật toán rất đơn giản. Sau khi bước 2 được hoàn thành thì ta sẽ có biểu đồ đồ thị ngón tay trên hình như sau



Hình 3.5 Sơ đồ các điểm mốc bàn tay.

Ta sẽ lấy ví dụ ở ngón tay trỏ để minh họa thuật toán xác định ngón tay dơ lên như sau. Có thể thấy ngón tay trỏ có các điểm mốc tay được đánh số là 8, 7, 6, 5. Mỗi điểm mốc tay có 1 tọa độ (x_i, y_i) xác định trên khung hình. Dễ dàng nhận thấy nếu ngón tay trỏ giơ lên thì tọa độ của điểm số 8 (x_8, y_8) sẽ cao hơn tọa độ điểm 6 (x_6, y_6). Hay nói cách khác thì $y_8<y_6$. Áp dụng tương tự phương pháp này cho 3 ngón là ngón giữa, áp út và ngón út ta cũng xác định được các ngón này có được dơ lên hay không. Ngón tay cái là một trường hợp đặc biệt hơn nên ta sẽ không thể áp dụng phương pháp này. Trong phạm vi báo cáo này, chúng em chỉ xác định hai ngón được dơ lên là ngón trỏ và ngón giữa nên xin phép sẽ không giới thiệu phương pháp cho ngón cái.

Bước 4: Nếu 2 ngón tay dơ lên thì đang ở chế độ chọn. Nếu một ngón tay dơ lên thì đang ở chế độ vẽ.

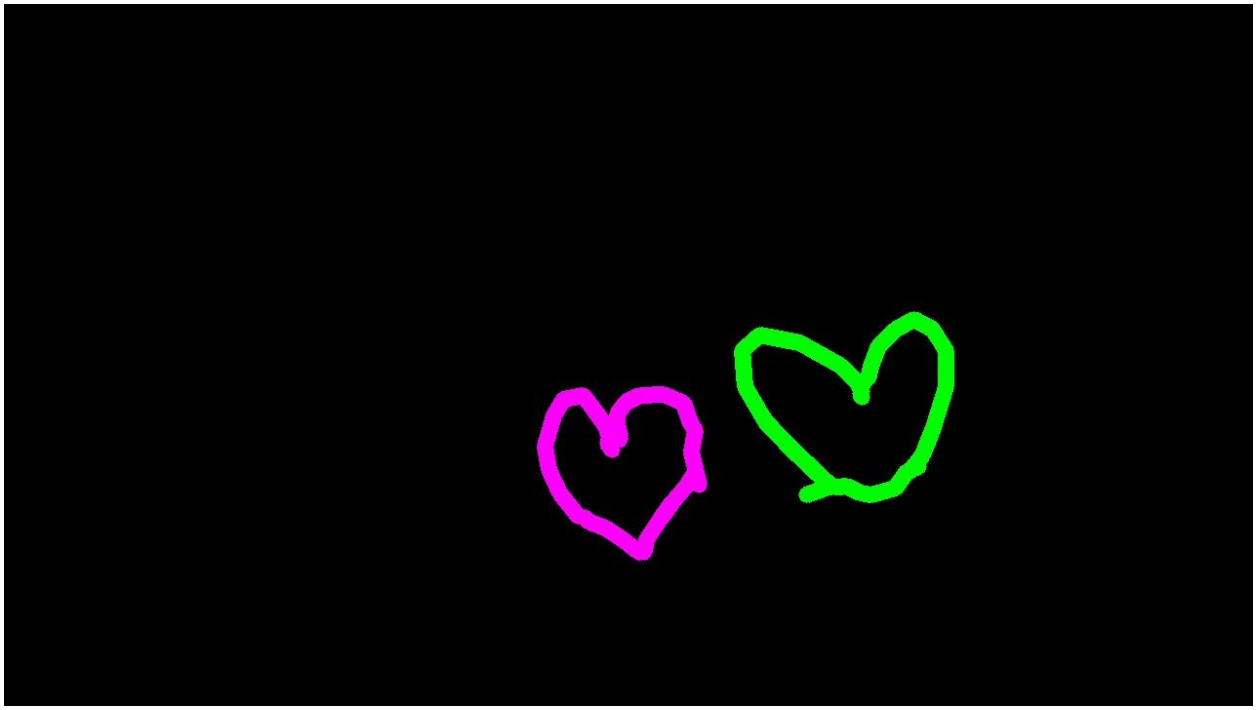
Trường hợp 1: Nếu hai ngón tay (ngón giữa và ngón trỏ dơ lên).

Chúng ta sẽ ở chế độ chọn. Tại đây chúng ta có chọn màu bút (hồng, lam, lục), chọn bút tẩy bằng việc dơ 2 ngón tay đến các vùng đã được cài đặt trước.

Hình 3.6 Vùng chọn các chế độ.

Trường hợp 2: Nếu chỉ duy nhất một ngón tay trỏ được dơ lên.

Chúng ta sẽ ở chế độ vẽ. Tại đây chúng ta có thể vẽ các hình tùy ý theo hướng di chuyển điểm mốc số 8 của ngón tay trỏ. Cơ chế để vẽ rất đơn giản. Ta sẽ vẽ đường thẳng từ điểm có tọa độ (x_a, y_a) đến điểm có tọa độ (x_b, y_b). Với (x_a, y_a) là tọa độ điểm mốc số 8 của ngón tay trỏ trong khung hình trước, (x_b, y_b) là tọa độ điểm mốc số 8 của ngón tay trỏ trong khung hình sau. Hình vẽ thực chất là tập hợp của các điểm liên tiếp nhau. Các hình vẽ sẽ được lưu trong một mảng có tên là Canvas.Mảng Canvas có kích thước 3x1280x720 Giá trị các đại lượng trong mảng được quy định như sau: vị trí các hình được vẽ ở trên có giá trị bằng giá trị màu của màu được sử dụng trong chế độ chọn. Các phần tử còn lại có giá trị bằng 0.

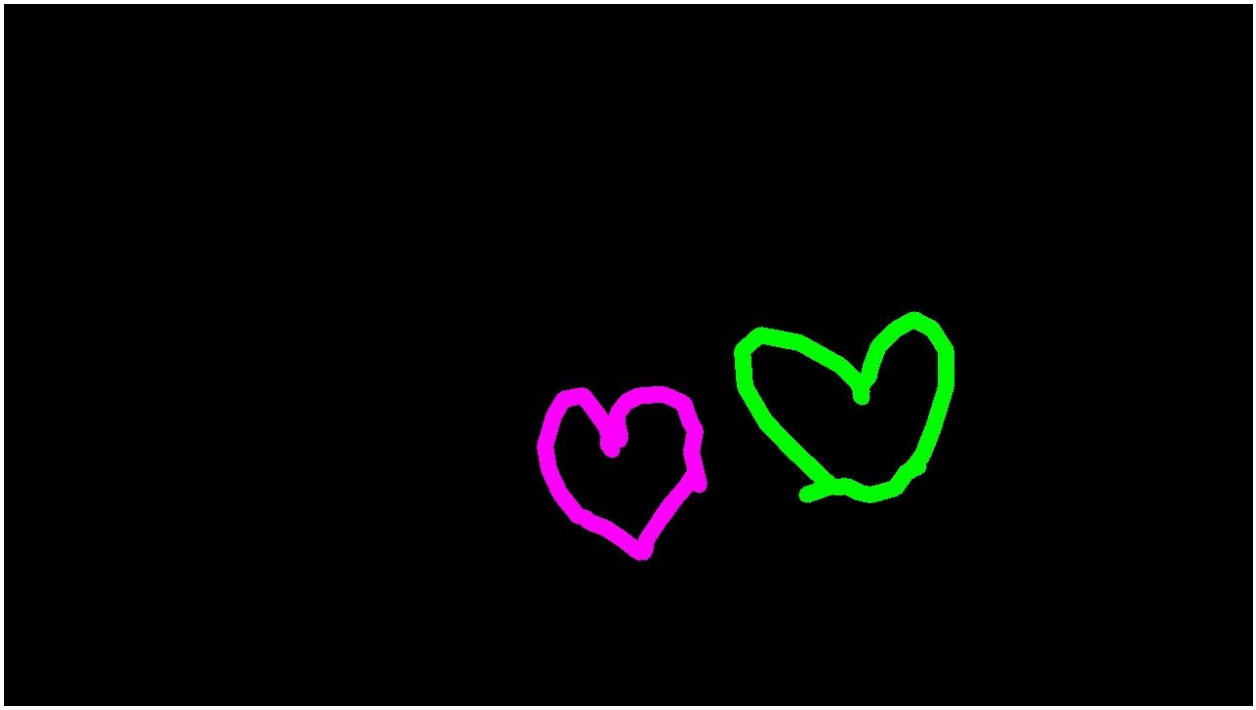


Hình 3.7 Ảnh Canvas.

Bước 5: Xuất khung hình sau khi thực hiện các bước trên và quay lại bước 1.

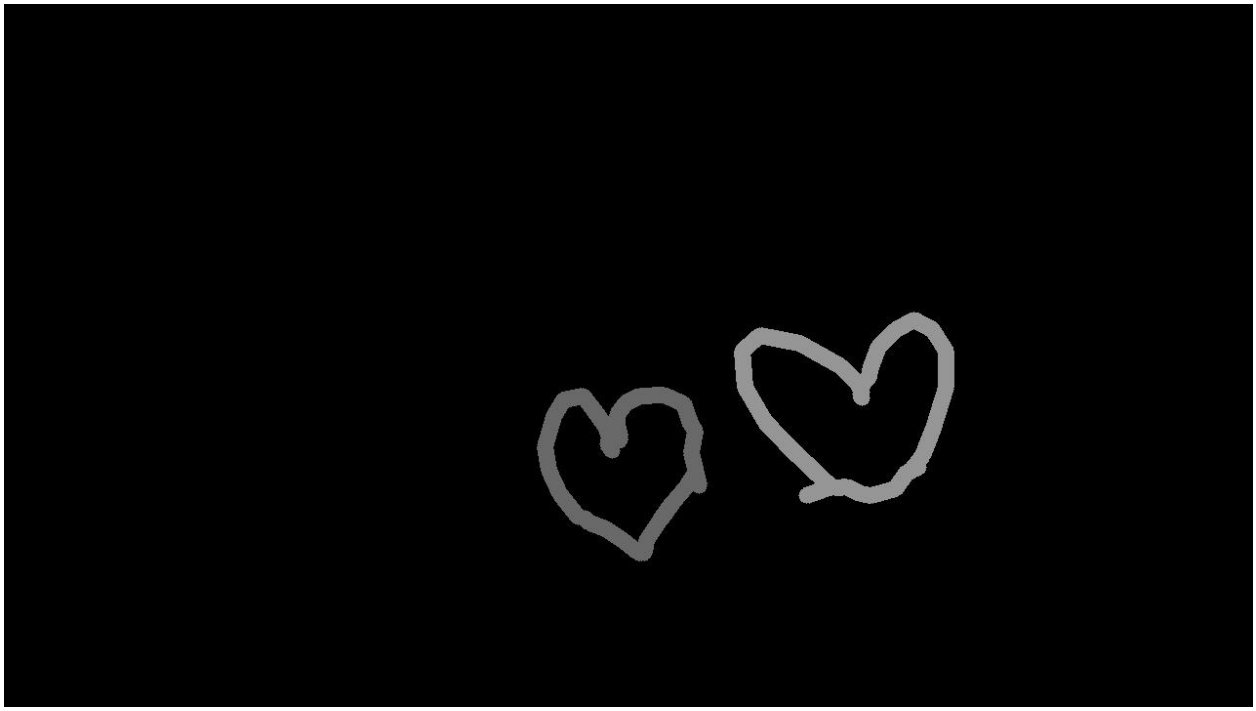
Để có được hình vẽ xuất hiện trên mà hình ta cần thông qua một số thao tác sau.

Như đã biết ở bước 4, hình vẽ của ta lưu vào một mảng có tên Canvas. Mảng này có hình như ở dưới.



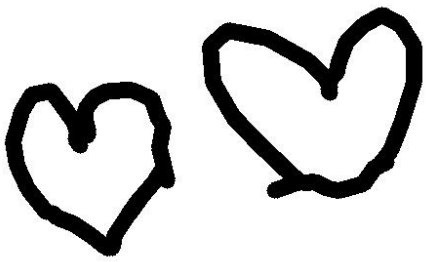
Hình 3.8 Ảnh Canvas.

Sau đó, ảnh Canvas sẽ được biến đổi sang ảnh xám. Ta đặt tên là imgGray.



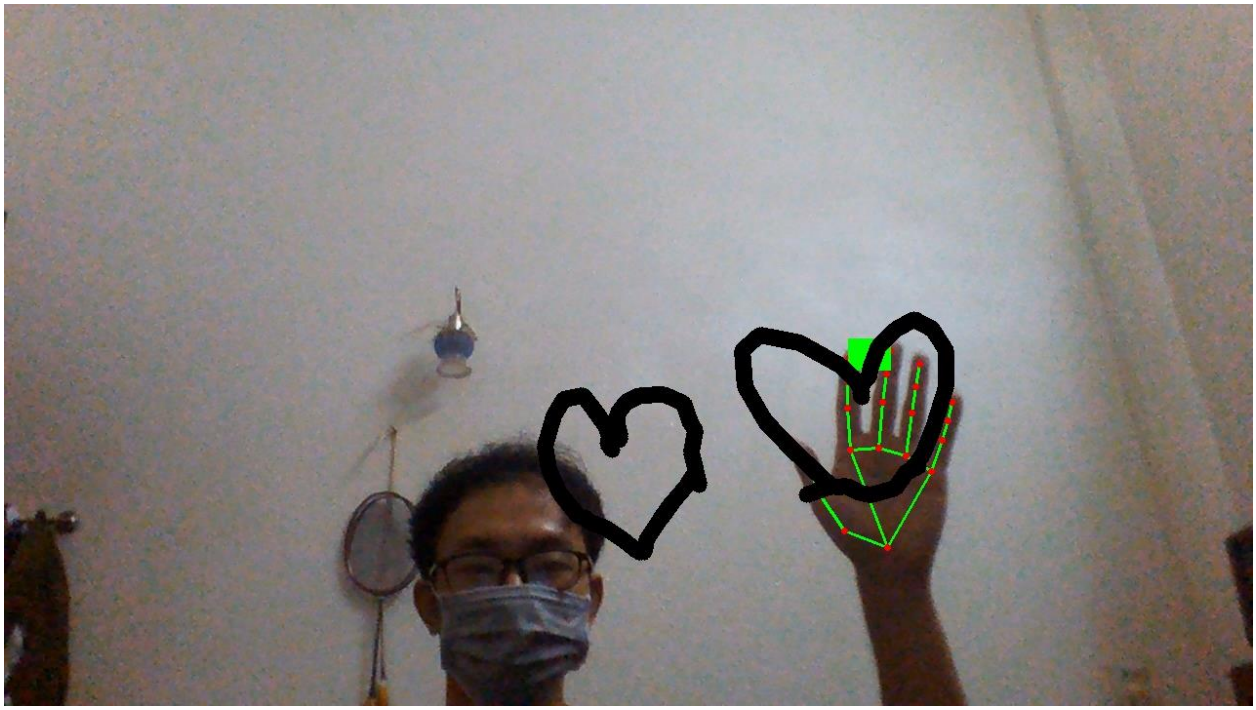
Hình 3.9 Ảnh xám imgGray.

Tiếp theo, ảnh xám imgGray sẽ được chuyển đổi thành imgThresholding. Dễ dàng nhận thấy imgThresholding có đặc điểm với các hình vẽ có màu đen và các phần không phải hình vẽ có màu trắng.



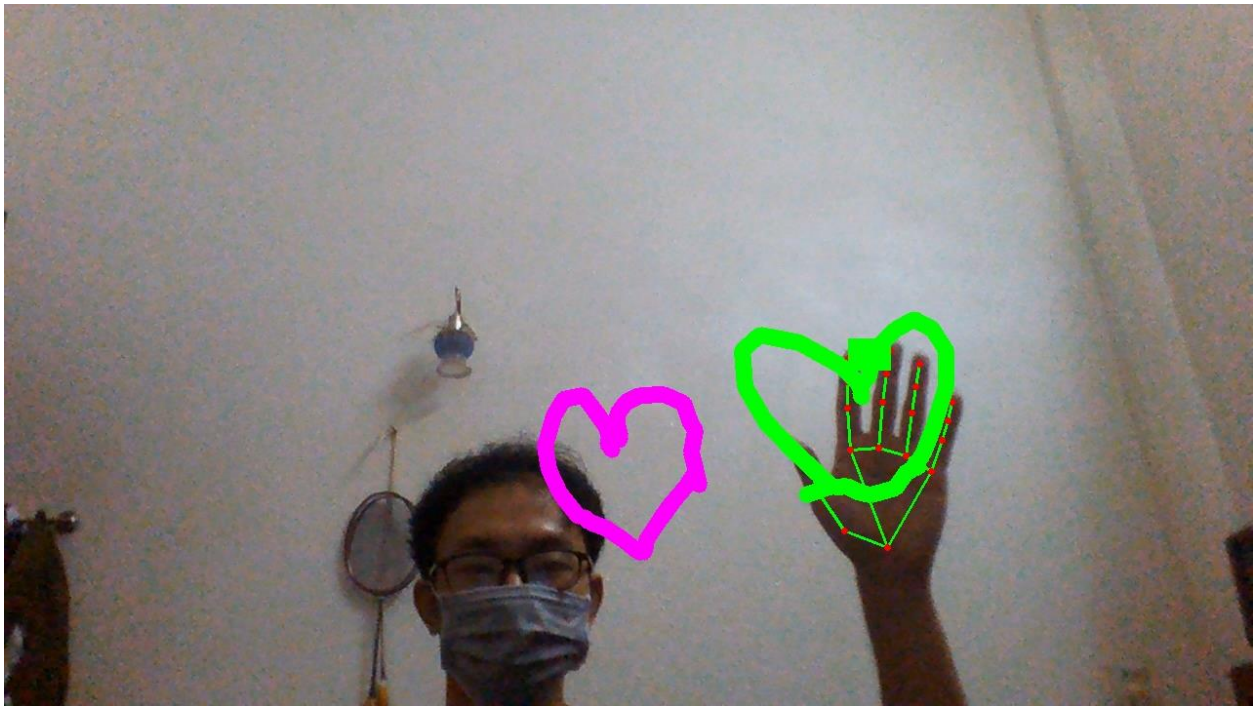
Hình 3.10 Ảnh imgThresholding.

Sau đó chúng ta đem ảnh imgThresholding trên thực hiện phép toán and với ảnh ở khung hình gốc. Ta được ảnh sau imgAnd như sau.



Hình 3.11 Ảnh imgAnd.

Ảnh cuối cùng được đưa ra màn hình là kết quả của ảnh imgAnd thực hiện phép toán or với ảnh Canvas.



Hình 3.12 Kết quả được đưa ra màn hình.

4. Ứng dụng Vitural Painter xây dựng trò chơi vẽ hình theo mẫu.

Nguyên lý trò chơi có thể miêu tả như sau; có 3 hình mẫu cho người chơi vẽ theo, mỗi lần vẽ xong ta sẽ lấy ảnh vẽ được đưa vào cho một mô hình học máy đánh giá. Chúng ta sẽ chỉ quan tâm xác suất của lớp mà chúng ta đang mong chờ. Nếu điểm lớp đó dự đoán không dưới 50%, thì coi như người chơi vẽ tốt và cộng điểm bằng với xác suất nhân với 100 (thang điểm 100). Nếu xác xuất không may dưới 50%, số điểm sẽ bị trừ, và trừ với một lượng 100 – 100*xác suất đầu ra.

Hình 4.1 Giao diện trò chơi.

Mô hình học máy

Tạo dữ liệu

Em tự tạo dữ liệu những hình viết tay bằng phần mềm Microsoft Paint.Với 140 dữ liệu mỗi lớp sẽ được chia ra thành 2 tập. Tập training với 120 tấm ảnh , tập test gồm 20 tấm ảnh. Tổng cộng số lượng dữ liệu tập training là 360 và dữ liệu tập test là 60.

Hình 4.2 Dữ liệu thu thập.

Ngoài ra, em cũng tiền làm giàu dữ liệu bằng phương pháp bồi đắp.

Hình 4.3 Dữ liệu sau khi được bồi đắp lần 1.

Hình 4.4 Dữ liệu sau khi được bồi đắp lần 2.

Kích thước ảnh crop ra đem phân loại là 300x300, tuy nhiên kích thước này có số chiều rất lớn, làm mô hình phải phức tạp mới có thể học tốt, nhưng việc này lại khiến quá trình xử

lý trong trò chơi tốn thời gian. Vì thế, em quyết định điều chỉnh kích thước tập dữ liệu xuống còn 60x60 (số chiều đã giảm đi 25 lần). Mô hình lúc này cũng không cần quá phức tạp.

Mô hình em triển khai không sử dụng các tầng tích chập vì mong muốn mô hình không quá tốt nên chỉ sử dụng các tầng dày đặc cổ điển như một mạng nơ-ron thông thường. Thuật toán tối ưu được áp dụng là Adam với tốc độ học là 0.01. Độ đo tối ưu là độ chính xác.

Kết quả huấn luyện sau 18 epoch cho được mất mát 1.33 và độ chính xác kiểm tra là 0.73.

5. Mô phỏng

Video mô phỏng trò chơi được chúng tối đính kèm cùng với bài báo cáo

6. Tổng kết

6.1. Kết luận.

Về kết quả đề tài, em đã thành công trong việc sử dụng MediaPipe thực hiện HandTracking. Ngoài ra, em cũng đã ứng dụng Vitural xây dựng trò chơi thị giác máy đơn giản.

6.2. Hướng phát triển.

Có thể phát triển trò chơi bằng cách tăng số lượng mẫu hình với sự đa dạng hình mẫu và độ khó.

Ứng dụng Vitural Painter phát triển các trò chơi khác như chém trái cây, xếp hình,…

Ứng dụng Vitura Painter trong việc giảng dạy online,…

Tài liệu tham khảo

(n.d.).

[1] ***Valentin Bazarevsky and Fan Zhang*****, "On-Device, Real-Time Hand Tracking with MediaPipe",** **https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html.**

[2] ***Valentin Bazarevsky*****, "MediaPipe Hands: On-device Real-time Hand Tracking",** **https://arxiv.org/abs/2006.10214.**

[3] ***Ann Yuan and Andrey Vakunov, "Face and hand tracking in the browser with MediaPipe and TensorFlow.js", https://blog.tensorflow.org/2020/03/face-and-hand-tracking-in-browser-with-mediapipe-and-tensorflowjs.html.***

[4] Murtaza's Workshop - Robotics and AI, "AI Virtual Painter | OpenCV Python | Computer Vision", **https://www.youtube.com/watch?v=ZiwZaAVbXQo.**