2D HAND POSE ESTIMATION

Lê Nhật Kha - 20520208, Lê Viết Lâm Quang – 20520290, Nguyễn Hoàng Tuấn – 20520344,
Mai Duy Ngọc – 20520654, Lê Nguyễn Minh Huy - 20521394

***Tóm tắt nội dung*—Với sự phát triển chóng mặt của Trí tuệ nhân tạo hiện nay, yêu cầu về việc tự động hóa các tác vụ càng trở nên cần thiết hơn. Trước tình hình đó, các mô hình Deep Learning phục vụ việc thay thế các thao tác của con người thành các hành động tương tác ảo đang dần trở nên phổ biến. Bài toàn Hand Pose Estimation được sinh ra để giải quyết triệt để vấn đề này. Từ ảnh chụp các tư thế tay khác nhau với nhiều góc nhìn khác nhau được thu thập, chúng tôi tiến hành sử dụng phương pháp học sâu để đưa ra dự đoán vị trí các khớp xương của tay.**

***Index Terms*—hand pose estimation, machine learning, deep learning, hand detection, stacked hourglass networks, SSD.**

I. Introduction

DESCRIBE: Những năm vừa qua, khi mà công nghệ kỹ thuật đã phát triển vượt bậc, không khó để thấy những ngôi nhà thông minh với những tiện ích chỉ xuất hiện trong trí tưởng tượng của con người như vỗ tay để tắt đèn, phẩy tay để mở nhạc. Bên cạnh đó, việc nhận biết được tư thế của bàn tay không chỉ phục vụ cho đời sống vật chất mà còn đáp ứng nhu cầu về đời sống tinh thần. Cụ thể là các tựa game thực tế ảo như Pokemon Go hay chơi nhạc ảo.

Hơn hết, định vị được tư thế tay một cách chính xác còn giúp ích rất nhiều vào lĩnh vực y tế, với các phòng phẫu thuật xuyên quốc gia. Khi mà trên thế giới vẫn còn xảy ra những cuộc chiến tranh ác liệt, không thể mạo hiểm để đưa đội ngũ y tế chuyên môn cao vào tâm điểm chiến trường, khi đó việc điều trị từ xa rất cần thiết, bài toán ước tính cử chỉ tay là lời giải tối ưu cho vấn đề này.

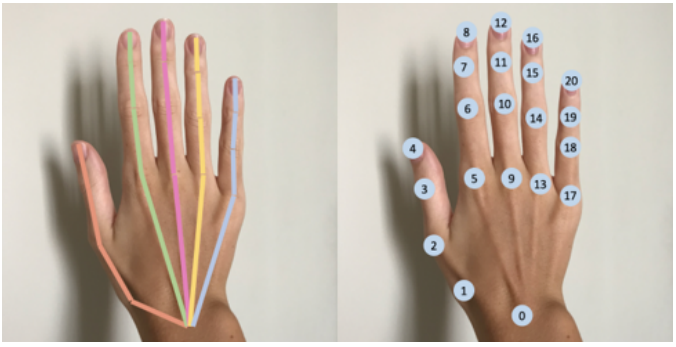
Thêm vào đó, bài toán ước tính cử chỉ tay còn là tiền đề cho bài toán đọc hiểu cử chỉ con người, là bước đệm cho máy tính không chỉ có thể hiểu hình ảnh, âm thanh từ con người mà còn hiểu cử chỉ, hành động của họ.

Mô hình hóa bài toán:

- **Input:** Hình ảnh/ video (webcam) có vật thể chính là bàn tay.
- **Output:** Một mảng(list) chứa tạo độ của các keypoint. Trong đó, keypoint là các đốt xương chính của bàn tay người, gồm 21 keypoints: 1 cổ tay + 5 ngón tay * (3 khớp ngón tay + 1 đầu ngón tay) = 21. Các tọa độ của keypoint ứng với từng pixel trong ma trận pixel của ảnh input. Sau đó dựa vào các keypoints mô hình sẽ nối chúng lại thành các khớp tay, tạo thành bàn tay hoàn chỉnh.

Với vấn đề được đặt ra, chúng tôi đã áp dụng kiến trúc mạng thắt nút cổ chai (Stacked Hourglass Networks) kết hợp với module nhận diện bàn tay (Hand Detection).

Chúng tôi tập trung xây dựng mạng Stacked Hourglass Networks, còn phần module Hand Detection liên quan đến bài toán Object Detection hiện tại đã rất tối ưu rồi nên chúng tôi sẽ sử dụng lại thành quả của những nghiên cứu trước đó.

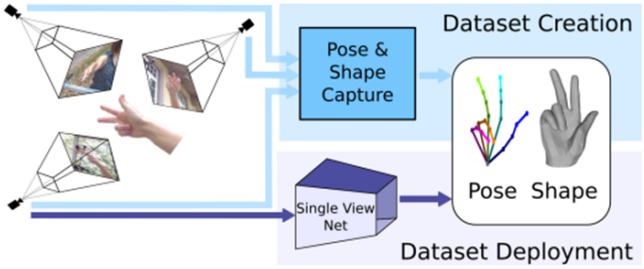


Hình 1: *Mô hình hóa bài toán Hand Pose Estimation*

II. Dataset

Trong đồ án này, nhóm chọn sử dụng tập dataset FreiHand nổi tiếng thường được sử dụng cho bài toán Hand Pose Estimation. Dataset này được public trên Kaggle. (Đường dẫn cụ thể được trích dẫn trong phần phụ lục)

FreiHand là 1 dataset 3D lớn, gồm các ảnh chụp bàn tay thực. Bàn tay trong tấm ảnh nằm ở góc nhìn thứ 3 (do camera chụp), bàn tay ở trung tâm bức ảnh và chỉ có duy nhất 1 bàn tay phải.



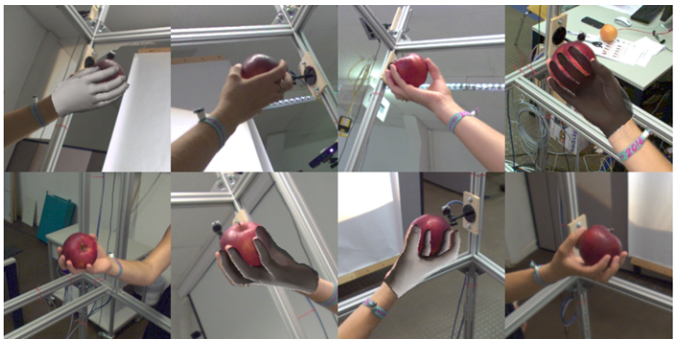
Hình 2: *Với phương pháp tận dụng nhiều góc nhìn và sparse annotation, FreiHand là một dataset lớn gồm ảnh các bàn tay thực, với nhãn được đánh dấu gồm cả bàn tay (hand shape) và các điểm keypoint trên bàn tay (hand pose)*

Dataset gồm đa dạng các kiểu dáng của bàn tay, bao gồm cả có và không có tương tác với vật thể. Các cử chỉ tay không tương tác với vật thể bao gồm: ngôn ngữ kí hiệu của Mỹ, thao tác đếm, cử chỉ thường dùng... Các vật thể tương tác với bàn tay gồm các dụng cụ xây dựng như: búa, kiềm, tua vít... và các vật dụng trong nhà bếp như: đũa, muỗng, chai... Các bức ảnh vừa gồm vật thể được đặt sẵn trong bàn tay vừa có thể là quá trình cầm lấy vật thể.

Có 32 người khác nhau thực hiện chụp các bàn tay, trong tập training thì mỗi kiểu tay được chụp ở 8 góc độ khác nhau.

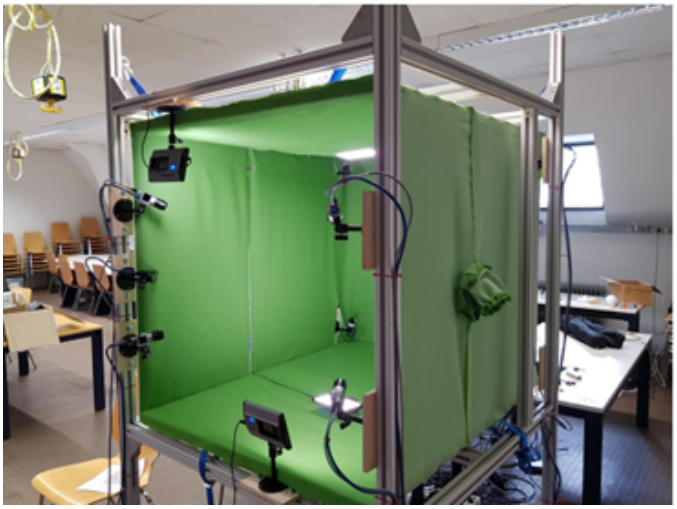
Tập dữ liệu chứa 130,240 trainset và 3960 testset.

Hình 3: *Minh họa một kiểu tay không có vật thể (trái) và một kiểu tay có cầm vật thể (phải) trong dataset FreiHand*



Hình 4: *Một kiểu tay được chụp ở 8 góc độ khác nhau*

Trong tập ảnh training, có 32560 bức ảnh ở background xanh, số ảnh này sau đó được xử lý qua các bước như Image Harmonization và Deep Image Colorization, rồi ghép 3 background khác nhau (2 background indoor và 1 background outdoor). Tính thêm cả 32560 ảnh gốc thì con số này được gấp 4 lần lên thành số ảnh trong tập training (nghĩa là ¾ số ảnh training là lặp lại của ¼ số ảnh đầu, chỉ khác background). Số ảnh trong tập evaluation đều là ảnh có background.



Hình 5: *Bối cảnh chụp các bức ảnh có phông nền xanh, với tổng cộng 8 camera chụp cùng lúc*

Mỗi training sample đều là ảnh RGB có kích thước 224x224px. Nhãn của sample sẽ bao gồm hand segmentation mask, intrinsic camera matrix, hand scale, 3D shape annotation, và 3D keypoint annotation cho 21 hand joints, tương tự nhãn cũng được cung cấp cho testset

1) Ưu điểm:
   - Một trong những lý do để dataset này được tạo nên là về vấn đề cross-dataset generalization (tổng quát hóa trên nhiều dataset). Việc 1 mô hình được train trên 1 dataset, khi được thử nghiệm trên chính dataset đó thì cho kết quả khá tốt nhưng khi dùng chính mô hình đó để thử nghiệm trên các dataset khác thì kết quả lại rất tệ. Nghĩa là mô hình được train trên 1 dataset thì không thể tổng quát hóa trên các dữ liệu mới (dataset khác). FreiHand đã giải quyết được vấn đề này, khi kiểm tra kết quả model được train trên FreiHand với các dataset khác thì kết quả tốt hơn nhiều so với các model được train trên các dataset đó, nghĩa là model train trên FreiHand có tính tổng quát hóa tốt hơn.

train \ eval	STB	RHD	GAN	PAN	LSMV	FPA	HO-3D	**Ours**	Average Rank
STB [35]	**0.783**	0.179	0.067	0.141	0.072	0.061	0.138	0.138	6.0
RHD [38]	0.362	**0.767**	0.184	0.463	0.544	0.101	0.450	0.508	2.9
GAN [23]	0.110	0.103	**0.765**	0.092	0.206	0.180	0.087	0.183	5.4
PAN [15]	0.459	0.316	0.136	**0.870**	0.320	0.184	0.351	0.407	3.0
LSMV [8]	0.086	0.209	0.152	0.189	**0.717**	0.129	0.251	0.276	4.1
FPA [6]	0.119	0.095	0.084	0.120	0.118	**0.777**	0.106	0.163	6.0
HO-3D [9]	0.154	0.130	0.091	0.111	0.149	0.073	-	0.169	6.1
Ours	0.473	0.518	0.217	0.562	0.537	0.128	**0.557**	**0.678**	2.2

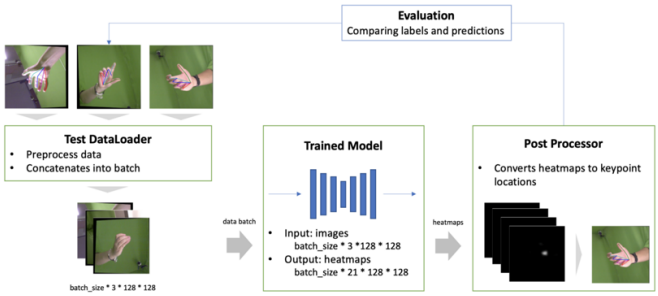
Hình 6: *Bảng này thể hiện tính tổng quát hóa giữa nhiều dataset khác nhau (cross-dataset generalization), tính theo độ đo AUC (area under the curve) dựa theo phần trăm số keypoints dự đoán đúng của từng mô hình được train trên từng dataset. Cột cuối cùng cho thấy xếp hạng trung bình mỗi tập training đạt được dựa trên nhiều tập evaluation khác nhau. Nguồn (paper-reference)*

2) Nhược điểm:
   - Các bàn tay đều nằm ở trung tâm bức ảnh, với kích thước không quá khác nhau, đồng thời phông nền cũng không có nhiều nhiễu nên model train trên dataset này có thể cho kết quả không tốt khi chạy trên môi trường real-time, khi có nhiều nhiễu và kích thước bàn tay thay đổi liên tục -> đây cũng chính là lý do nhóm thêm 1 module detect hand vào để cắt ra khung ảnh có tay ở vị trí trung tâm và ít bị nhiễu bởi môi trường.
3) Lý do chọn dataset:
   - FreiHand phù hợp với bài toán của nhóm đề ra lúc đầu, đó là bài toán hand pose estimation trên ảnh đầu vào RGB 2D. Nhãn đầu ra của dataset cũng gồm 21 keypoint trên bàn tay, dù là kết quả 3D nhưng trong github của FreiHand cũng có đoạn code giúp chuyển tọa độ các keypoint về 2D.
   - FreiHand chứa các ảnh tay thật, kích cỡ dữ liệu cũng hợp lý khi không quá lớn, không quá nhỏ (140k tấm ảnh) khi so với các bộ dữ liệu real hand trước nó như STB (18k) và sau nó như InterHand (2.6m). Đây cũng là một trong các bộ dữ liệu mới nhất (2019) trong lĩnh vực này.

III. Related Work

Có 2 hướng tiếp cận chính:

1) Bottom-up: Xác định các keypoint, sau đó vẽ các đường thẳng nối keypoint lại tạo thành khung xương tay. Tiêu biểu: SRHandNet, NSRM Hand, InterHand
   - Ưu điểm: Mô hình gọn, nhẹ, xử lý nhận diện tay nhanh
   - Nhược điểm: Dễ bị nhiễu bởi môi trường xung quanh -> false positives -> low recall (độ chính xác thấp). Vì sử dụng toàn bộ khung ảnh để dự đoán nên nếu không có vật thể tay mô hình vẫn dự đoán -> không hiệu quả.



Hình 7: *Một phương pháp theo hướng tiếp cận bottom-up*

2) Top-down: Đầu tiên mô hình sẽ xác định vật thể tay trước (bài toán Object Detection), sau đó dựa vào bounding box chứa bàn tay để bắt đầu dự đoán. Tiêu biểu: FastHand, Medipipe
   - Ưu điểm: Ít chịu ảnh hưởng bởi môi trường, độ chính xác cao.
   - Nhược điểm: Vì có thêm module hand detect nên mô hình cồng kềnh hơn, chạy lâu hơn, tốn nhiều tài nguyên hơn khi train và ứng dụng.

Hình 8: Các phương pháp theo hướng tiếp cận top-down

Nhìn chung mỗi hướng tiếp cận đều có ưu và nhược điểm riêng. Chúng tôi lựa chọn hướng tiếp cận top-down để xây dựng mô hình của mình.

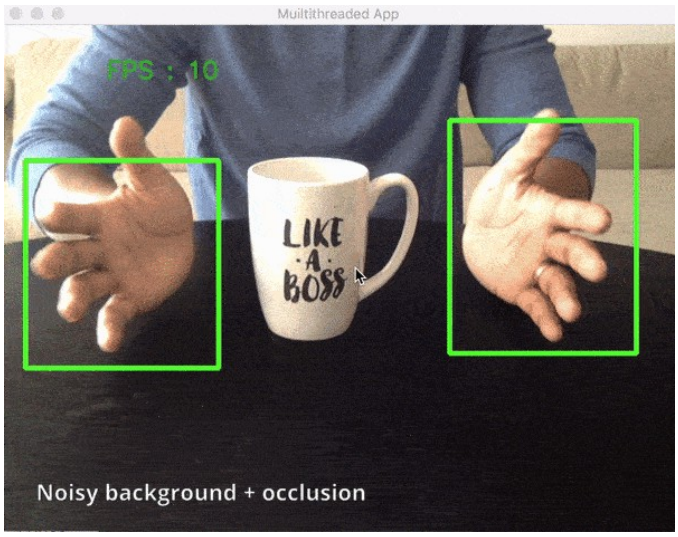
IV. Methods

Trong phần này, chúng tôi sẽ miêu tả kiến trúc mạng đề xuất mà chúng tôi đã cài đặt và các thành phần chính để ước lượng tọa độ 2D của bàn tay từ một bức hình RGB. Ban đầu, hướng tiếp cận chính của chúng tôi là Bottom-up nhưng vì giới hạn tài nguyên nên sẽ cần một mô hình gọn nhẹ. Sau khi đã nghiên cứu kiến trúc mạng Stacked Hourglass Networks được giới thiệu lần đầu trong bài báo "Stacked Hourglass Networks for Human Pose Estimation" cua nhóm tác giả đến từ University of Michigan công bố năm 2016, nhận thấy tiềm năng to lớn của mô hình khi nó chiến thắng cuộc thi dự doán tư thế xe của Peking University và được nhận định là mô hình mạnh mẽ trong lĩnh vực Pose Estimation. Đồng thời nhận ra sự tương đồng giữa 2 bài toán Human Pose Estimation và Hand Pose Estimation, chúng tôi quyết định lựa chọn mô hình này để phát triển bài toán của mình.

Tuy nhiên với những nhược điểm của hướng tiếp cận Bottom-up như đã trình bày ở phần trên, chúng tôi đã khắc phục bằng cách thêm một module Detect Hand phía trước để tăng độ chính xác của mô hình, vì thế, hướng tiếp cận của chúng tôi đã thay đổi từ Bottom-up thành Top-down để phù hợp với giải pháp của nhóm nghiên cứu. Do đó, tổng quan lại, kiến trúc của chúng tôi sẽ gồm hai phần: Detect Hand và Stacked Hourglass Networks.

A. Kiến trúc mạng:

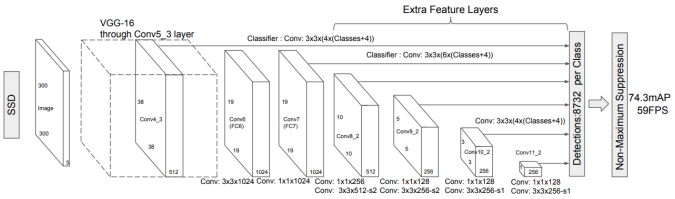
1) Detect Hand: Vì giới hạn về tài nguyên, thời gian, đồng thời bài toán Detect Hand không phải là bài toán trọng tâm của chúng tôi, do đó chúng tôi sử dụng trực tiếp mô hình đã được pre-trained với kiến trúc mạng DNN. Sau khi đã detect được bàn tay trong bức hình RGB, chúng tôi sẽ thực hiện cắt bounding box chứa bàn tay đã được nhận dạng và đưa qua mô hình Stacked Hourglass Networks.



Hình 9: *Module Hand Detector sử dụng kiến trúc SSD*

Chúng tôi quyết định sử dụng mô hình SSD (Single Shot MultiBox Detector) được pre-trained trên tập dữ liệu Egohands Dataset, thực hiện bởi Victor Dibia. Mô hình SSD được Wei Liu giới thiệu vào năm 2016. Tác giả đã công bố một phương

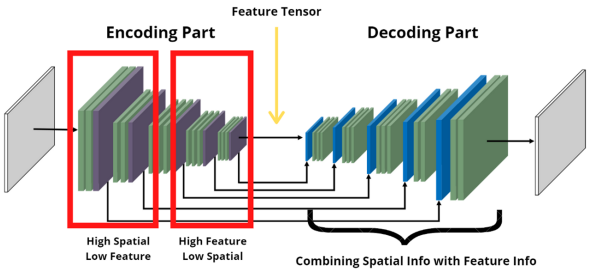
pháp có thể xác định các đối tượng bằng cách sử dụng mạng tích chập chuyển tiếp bằng cách sử dụng một lần chuyển tiếp đơn lẻ. Giải thích về tên gọi SSD (Single Shot MultiBox Detector), nó có ý nghĩa là mạng có khả năng phát hiện đối tượng được thực hiện trên 1 phase duy nhất dựa trên kĩ thuật MultiBox.



Hình 10: *Sơ đồ kiến trúc của mạng SSD*

Kiến trúc của mô hình SSD được xây dựng trên mạng thần kinh tích chập VGG-16 được loại bỏ đi tầng fully-connected, theo sau là một số lớp tích chập bổ sung, làm giảm kích thước của đầu vào ở mỗi lớp.

2) Stacked Hourglass Networks: Thiết kế của Hourglass được lấy cảm hứng từ sự cần thiết trong việc phải nắm bắt tất cả thông tin trên tất cả các tỉ lệ của bức ảnh. Dịch ra Tiếng Việt Hourglass có nghĩa là đồng hồ cát, đúng với tên gọi đó, mô hình này có câu trúc đối xứng nhau như một chiếc đồng hồ cát với một bên thực hiện việc giảm độ phân giải của bức hình và một bên tái tạo lại để tập hợp các đặc điểm của bức ảnh trên nhiều tỉ lệ khác nhau.

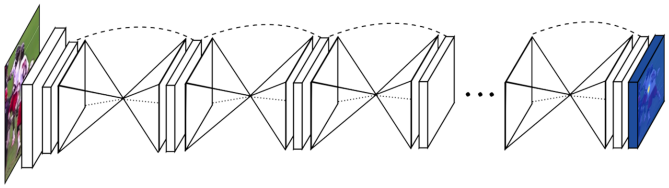


Hình 11: *Cấu trúc của một Hourglass*

Hourglass là một dạng của convolutional encoder-decoder network (có nghĩa là nó sử dụng các lớp tích chập để chia nhỏ và tái tạo lại input)

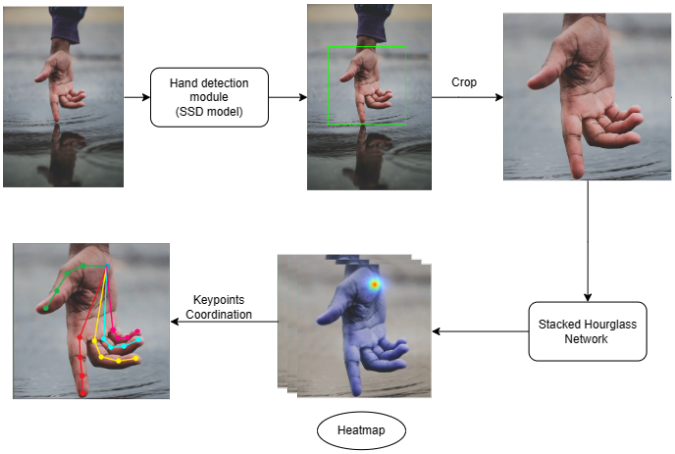
Encoding: Trích xuất các đặc điểm thông qua việc chia nhỏ đầu vào thành một ma trận đặc điểm.

Decoding: Kết hợp thông tin về không gian với thông tin về đặc điểm.



Hình 12: *Cấu trúc Stacked Hourglass Networks*

Stacked Hourglass có nghĩa là chồng các Hourglass lại với nhau, output của Hourglass này sẽ là input của Hourglass kia. Cuối cùng, kết quả dự đoán sẽ được tạo ra sau khi đi qua hết tất cả các Hourglass nơi mà mạng đã có cơ hội xử lý các đặc điểm của bức hình trên cả một ngữ cảnh cục bộ và toàn cục.



Hình 13: *Tổng quan về mô hình hệ thống*

V. Experiment

Tôi cài đặt hệ thống và công khai source code của mình trên Github. [1]

A. Training:

Trong suất quá trình training, chúng tôi đã sử dụng RMSprop optimizer với batchsize là 64 và quá trình đào tạo này được thực hiện trên GPU được cung cấp bởi Google Colab Pro. Source code được cài đặt và trình bày trên Github (đường dẫn cụ thể được mô tả trong phần References)

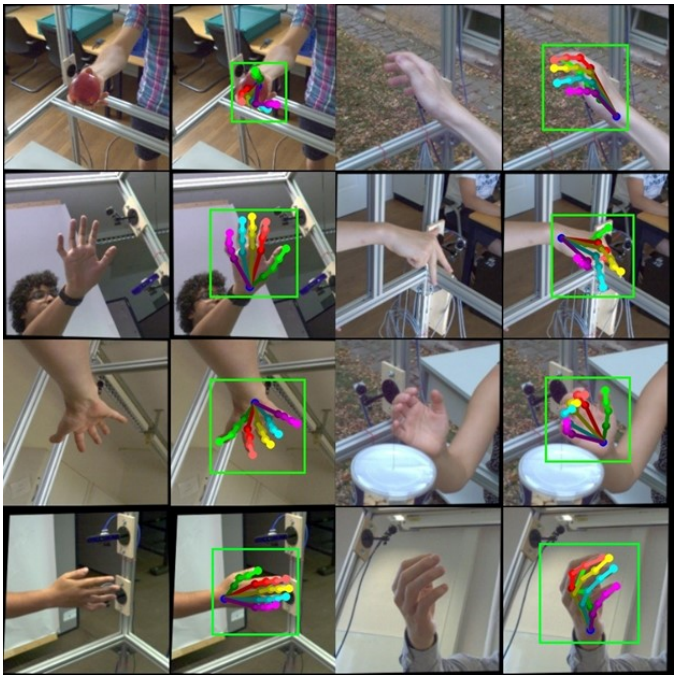
Chúng tôi thực hiện train Stacked Hourglass Networks trên tập FreiHand. Tuy nhiên vì Freihand ban đầu được dùng để huấn luyện mô hình dự đoán 3D từ ảnh 2D RGB, nên nó chỉ có các label 3D. Do đó chúng tôi thực hiện thêm thao tác chuyển label từ 3D sang 2D, bao gồm ma trận tọa độ 21 điểm keypoints. Sau đó tổng hợp nó thành heatmap, sử dụng hàm loss là MSE (tính đổ lỗi giữa các pixel với nhau).

Như đã nói, do giới hạn tài nguyên hạn chế, nên chúng tôi chia ra train từng đợt cho model, mỗi đợt 40 epoch sau đó lưu checkpoint lại. Tổng cộng khoảng 400 epoch và tốn hơn 1 tuần để train xong. Tuy nhiên nếu nhìn vào biểu đồ cuối, ta thấy hàm loss vẫn đang tiếp tục giảm, chứng tỏ model chưa hội tụ tại điểm tốt nhất. Trong tương lai, chúng tôi sẽ tiếp tục huấn luyện mô hình và làm cho nó tốt hơn.

B. Result:

Mô hình dự đoán khá tốt trên tập FreiHand test. Khi test real-time cũng cho kết quả khá ổn định, đôi lúc hơn giật lag vì mô hình nặng.

[1] https://github.com/KhaLee2307/hand-pose-estimation.git



Hình 14: *Kết quả thử nghiệm trên tập test của Freihand*

C. Metric:

Để đánh giá mô hình chúng tôi sử dụng ba thang đo là:

- Thông thường để đánh giá các mô hình Pose Estimation, chúng ta có 2 độ đo chính: MSE hoặc PCK. MSE thì khá là quen thuộc rồi, chúng ta sẽ lấy trung bình đổ lỗi bình phương của 21 điểm keypoint dự đoán được tương ứng với 21 keypoints của groundtruth. Cách đánh giá này khá hiệu quả nhưng lại có xu hướng thiên vị các phương pháp bottom-up hơn top-down. Vì có một số ảnh, tay bị khuất quá nhiều hoặc không rõ ràng khiến cho module Detector không nhận diện được tay, do đó phần module sau sẽ không dự đoán ra được các keypoints của tay. Khi đó, chúng ta sẽ không có ma trận keypoints dự đoán để tính đổ lỗi. Còn với bottom-up mặc dù không chính xác nhưng mô hình vẫn dự đoán các điểm keypoints.
- Thay vì dùng MSE chúng ta sẽ chuyển qua sử dụng 2D PCK score. Điểm khác biệt duy nhất là chúng ta thêm vào một thresh hold o. Nếu MSE của chúng ta < thresh hold thì sẽ được tính là 1. Cộng tất cả 21 trường hợp của keypoints lại sau đó lấy trung bình cộng chúng ta sẽ được 2D PCK score. Với trường hợp không predict ra được ma trận keypoints thì score mặc định bằng 0. Score càng cao thì mô hình càng dự đoán được chính xác.

VI. Kết luận

Chúng tôi đã xây dựng được một mô hình Ước lượng tư thế tay (2D Hand Pose Estimation) lấy cảm hứng từ kiến trúc Stacked Hourglass sau đó phát triển thêm bằng cách thêm một module Hand Detector nhằm cải thiện chất lượng của mô hình. Hệ thống dự đoán mà chúng tôi xây dựng chạy thử nghiệm rất tốt trên tập dataset FreiHand lẫn trong môi trường real-time (webcam). Trong tương lai, chúng tôi sẽ tiếp tục cải thiện bằng cách huấn luyện thêm mô hình trên nhiều bộ dữ liệu lớn hơn.

References

[1] https://lmb.informatik.uni-freiburg.de/projects/freihand/

[2] Newell, A., Yang, K. and Deng, J., 2016. Stacked hourglass networks for human pose estimation. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14 (pp. 483-499). Springer International Publishing.

[3] An, S., Zhang, X., Wei, D., Zhu, H., Yang, J. and Tsintotas, K.A., 2021. Fast Monocular Hand Pose Estimation on Embedded Systems. arXiv preprint arXiv:2102.07067.

[4] Chen, Y., Ma, H., Kong, D., Yan, X., Wu, J., Fan, W. and Xie, X., 2020. Nonparametric structure regularization machine for 2d hand pose estimation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 381-390).