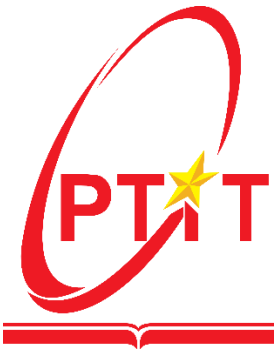
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO NGHIÊN CỨU

ĐỀ TÀI:

NGHIÊN CỨU ƯỚC LƯỢNG KHOẢNG CÁCH BẰNG CAMERA 2D

Người thực hiện:

Trần Hữu Phúc

HÀ NỘI 2024

LỜI CAM ĐOAN

Tôi xin cam đoan rằng bài báo cáo nghiên cứu với đề tài "**Nghiên cứu ước lượng khoảng cách bằng camera 2D**" là công trình nghiên cứu độc lập của riêng tôi. Toàn bộ nội dung và kết quả nghiên cứu trong bài báo cáo này đều trung thực và chưa từng xuất hiện trong bất kỳ công trình khoa học nào trước đây.

Các tài liệu tham khảo đều được trích dẫn một cách rõ ràng, minh bạch và có tính kế thừa, phát triển từ các công trình, bài báo, và nghiên cứu khoa học của nhiều tác giả uy tín đã được công bố.

Trong trường hợp phát hiện có bất kỳ hành vi gian lận nào, tôi xin chịu hoàn toàn trách nhiệm về nội dung của bài báo cáo này.

MỤC LỤC

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH ẢNH

DANH MỤC KÍ HIỆU, KÍ TỰ VIẾT TẮT

AI	Artificial Intelligence
AP	Average Precision
CNN	Convolutional neural network
DL	Deep Learning
HMI	Human-Machine Interface
HQPTT	Mô hình hồi quy phi tuyến tính
HQTT	Mô hình hồi quy tuyến tính
ML	Machine Learning
R-CNN	Region-Based CNN
RoI	Region of Interest
SGD	Stochastic Gradient Descent
SSD	Single Shot Multibox Detector
SVM	Support Vector Machine

A. MỞ ĐẦU

1. Lý do chọn đề tài

Trong kỷ nguyên số hóa hiện nay, sự phát triển mạnh mẽ của công nghệ thông tin đã tạo ra một nhu cầu cấp thiết về việc tối ưu hóa tương tác giữa con người và máy móc, đặc biệt trong các lĩnh vực cần xác định khoảng cách chính xác trong không gian 3D. Hiện tại, việc đo lường khoảng cách chủ yếu dựa vào các cảm biến chuyên dụng như cảm biến siêu âm, laser hay cảm biến độ sâu, khiến chi phí triển khai cao và hạn chế khả năng ứng dụng rộng rãi.

Một trong những xu hướng đang được quan tâm nghiên cứu là sử dụng camera 2D thông thường để ước lượng khoảng cách trong không gian 3D. Phương pháp này có nhiều ưu điểm nổi bật như chi phí thấp, dễ dàng triển khai và có thể tận dụng được hệ thống camera sẵn có. Đặc biệt, việc ước lượng khoảng cách từ camera 2D đến bàn tay có nhiều ứng dụng thực tiễn trong các hệ thống tương tác người - máy, thực tế ảo và điều khiển thông minh.

Xu hướng phát triển các hệ thống thông minh ngày càng đòi hỏi khả năng xác định chính xác vị trí và khoảng cách của đối tượng trong không gian 3D. Trong đó, việc sử dụng một camera 2D đơn giản để thực hiện nhiệm vụ này đang nhận được sự quan tâm đặc biệt từ cộng đồng nghiên cứu, bởi nó mở ra khả năng phát triển các ứng dụng với chi phí tối ưu.

Việc ước lượng khoảng cách bằng camera 2D có tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực như: hệ thống an ninh thông minh, robot tự hành, các ứng dụng thực tế ảo, và đặc biệt trong các hệ thống tương tác người - máy yêu cầu độ chính xác cao về khoảng cách. Tuy nhiên, việc chuyển đổi từ thông tin 2D thu được qua camera sang ước lượng khoảng cách trong không gian 3D vẫn còn nhiều thách thức.

Qua quá trình tìm hiểu, tôi nhận thấy các công trình nghiên cứu đã công bố còn một số hạn chế đáng kể. Một số thách thức được đưa ra như: độ chính xác bị ảnh hưởng bởi điều kiện ánh sáng, góc nhìn camera, sự chuyển động của đối tượng gây mờ ảnh và đặc biệt là việc chuyển đổi từ thông tin 2D sang không gian 3D một cách chính xác. Từ những thách thức này, tôi quyết định chọn đề tài "Ước lượng khoảng cách bằng camera 2D nhằm ước lượng không gian 3D" để nghiên cứu sâu hơn, với mong muốn đóng góp vào việc phát triển một phương pháp ước lượng khoảng cách hiệu quả, chính xác, tiết kiệm chi phí và tài nguyên, góp phần thúc đẩy sự phát triển của các ứng dụng tương tác thông minh trong tương lai.

2. Mục tiêu nghiên cứu

Nghiên cứu và ứng dụng hồi quy phi tuyến tính vào bài toán ước lượng khoảng cách từ bàn tay đến camera 2D.

Xây dựng được mô hình có khả năng ước lượng khoảng cách bàn tay với sai số thấp. Từ đó đánh giá và so sánh độ chính xác của các phương pháp ước lượng khoảng cách khác nhau trong các điều kiện môi trường, ánh sáng khác nhau.

3. Nội dung nghiên cứu

Nội dung 1: Nghiên cứu các kỹ thuật phát hiện bàn tay và các kỹ thuật ước lượng khoảng các từ ảnh RGB.

Nội dung 2: Nghiên cứu phát triển mô hình ước lượng khoảng cách bàn tay dựa trên khái niệm hồi quy phi tuyến tính.

4. Đối tượng, phạm vi, phương pháp nghiên cứu.

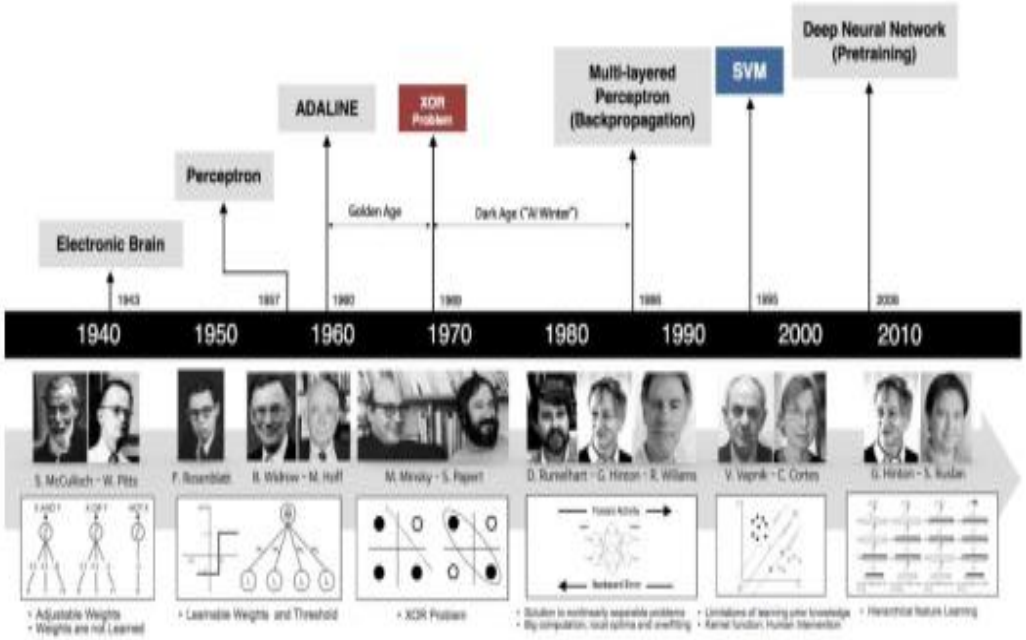
- Đối tượng nghiên cứu:
  + Bàn tay người.
  + Khung xương 3D của bàn tay.
  + Nonlinear regression (hồi quy phi tuyến tính).
- Phạm vi nghiên cứu:
  + Phát hiện bàn tay trên ảnh.
  + Ước lượng khoảng cách bàn tay tới camera 2D với thuật toán đề xuất.
- Những phương pháp nghiên cứu:
  + Phương pháp nghiên cứu mô hình hóa: Nghiên cứu và xây dựng mô hình hồi quy phi tuyến tính cho bài toán ước lượng khoảng cách bàn tay từ ảnh 2D. Thiết kế thuật toán chuyển đổi từ tọa độ 2D sang ước lượng khoảng cách 3D.
  + Phương pháp nghiên cứu thực nghiệm: Cài đặt thực tế mô hình, thuật toán đề xuất sử dụng cho bài toán ước lượng khoảng cách bàn tay từ video.
  + Phương pháp nghiên cứu tham khảo ý kiến chuyên gia: Đánh giá tính khả thi của các phương pháp ước lượng khoảng cách tay bằng hồi quy phi tuyến tính. Triển khai ý tưởng, cài đặt thực nghiệm, phân tích, đánh giá kết quả.
  + Phương pháp nghiên cứu điều tra, khảo sát: Tổng hợp các nghiên cứu liên quan đến ước lượng khoảng cách 3D từ ảnh 2D. Đánh giá ưu nhược điểm của các phương pháp hiện có.

B. NỘI DUNG

CHƯƠNG 1: NGHIÊN CỨU CÁC KỸ THUẬT PHÁT HIỆN BÀN TAY VÀ CÁC KỸ THUẬT ƯỚC LƯỢNG KHOẢNG CÁCH TỪ ẢNH RGB

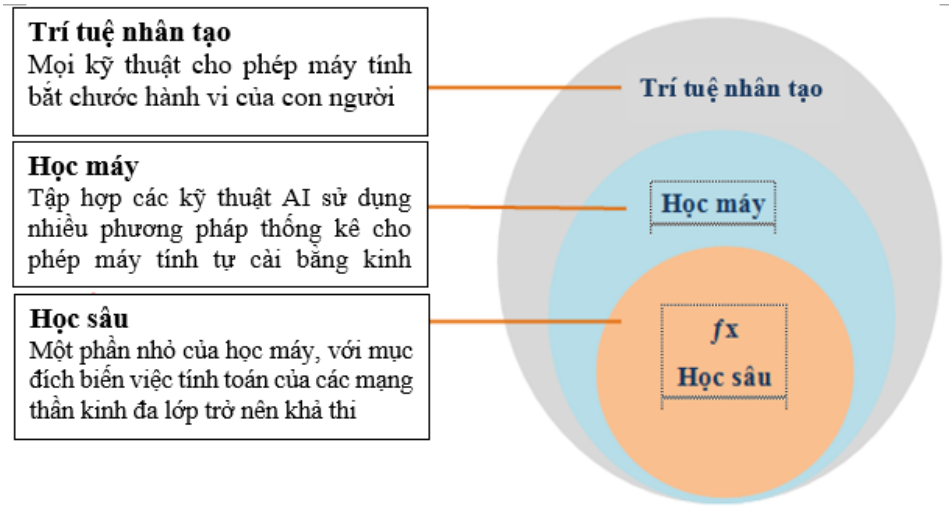
1.1. Khái niệm cơ bản về học sâu

Học sâu (Deep Learning – DL) là một trong những phương pháp học máy có sử dụng nhiều lớp biến đổi phi tuyến trên dữ liệu đầu vào từ đó trích xuất được các đặc trưng của dữ liệu [1]. Trong khi học, dữ liệu được xử lý qua nhiều lớp với các mức độ khác nhau. Dữ liệu có gán nhãn và đủ lớn thường được sử dụng để huấn luyện trong DL. Quá trình phát triển của DL được thể hiện qua hình:



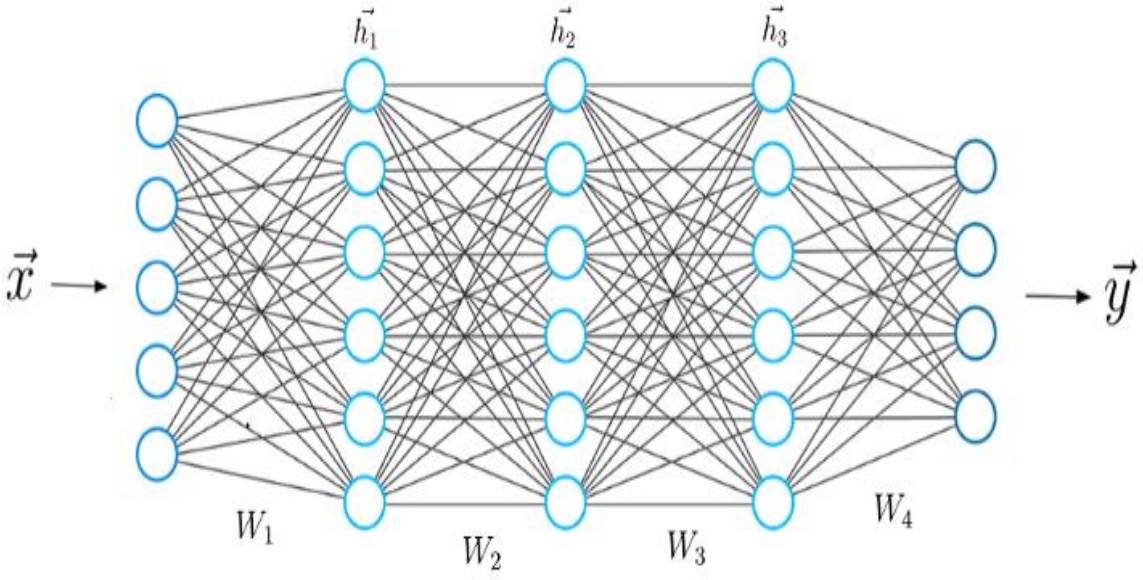
Hình 1: Quá trình phát triển của học sâu (Deep Learning).

Deep Learning là một tập con của Học máy (Machine Learning - ML), mặt khác là một chức năng của trí tuệ nhân tạo (Artificial Intelligence - AI). Thuật ngữ AI thường được dùng để đề cập đến các kỹ thuật sử dụng máy tính "học" các hành vi của con người trong đó ML lại là ứng cử viên sử dụng một tập các thuật toán dùng dữ liệu đầu vào để huấn luyện và làm cho những yêu cầu trở thành khả thi.



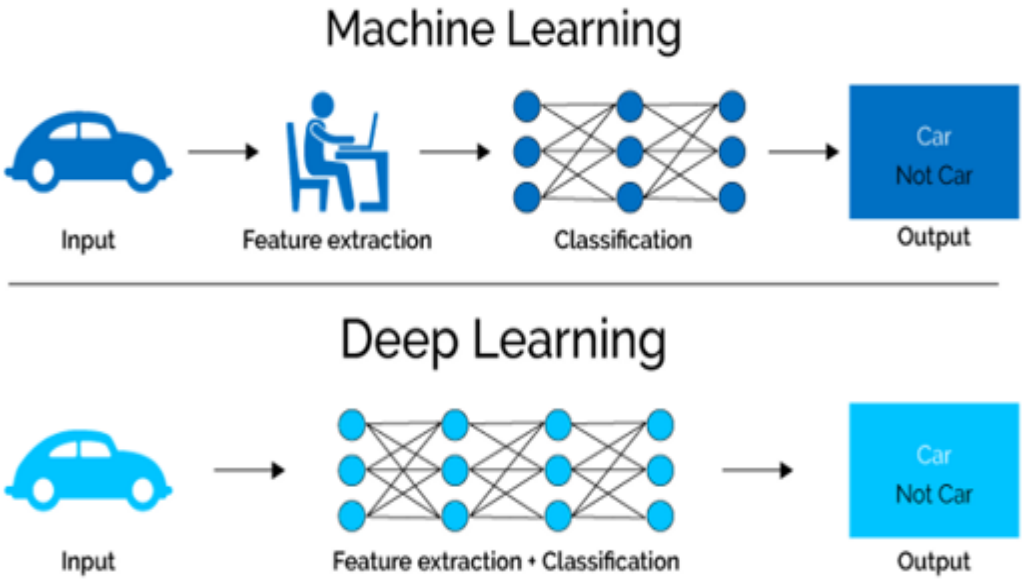
Hình 2: Mối quan hệ giữa học sâu, học máy và trí tuệ nhân tạo.

Deep Learning học cách sử dụng giống cấu trúc của bộ não con người. Tương tự như cách con người đưa ra quyết định, các thuật toán DL học cách đưa ra kết luận dựa trên việc phân tích dữ liệu với một cấu trúc logic nhất định. DL làm được điều này bằng cách ứng dụng mạng thần kinh, tức là sử dụng cấu trúc nhiều lớp của các thuật toán.



Hình 3: Một mạng thần kinh điển hình.

Mạng nơron nhân tạo mô phỏng cấu trúc và cách hoạt động của não người [2], sử dụng các lớp riêng lẻ như bộ lọc để phân tích thông tin từ tổng thể đến chi tiết, giúp phát hiện và đưa ra kết quả chính xác hơn. So với ML, DL không cần trích xuất tính năng, trong khi các thuật toán truyền thống như Cây quyết định, Support Vector Machine (SVM) và Naïve Bayes đòi hỏi bước này. DL cho phép các mạng nơron tự học cách biểu diễn dữ liệu thô, nén và trừu tượng hóa dữ liệu qua nhiều lớp, từ đó tạo ra kết quả như phân loại dữ liệu thành các lớp khác nhau.



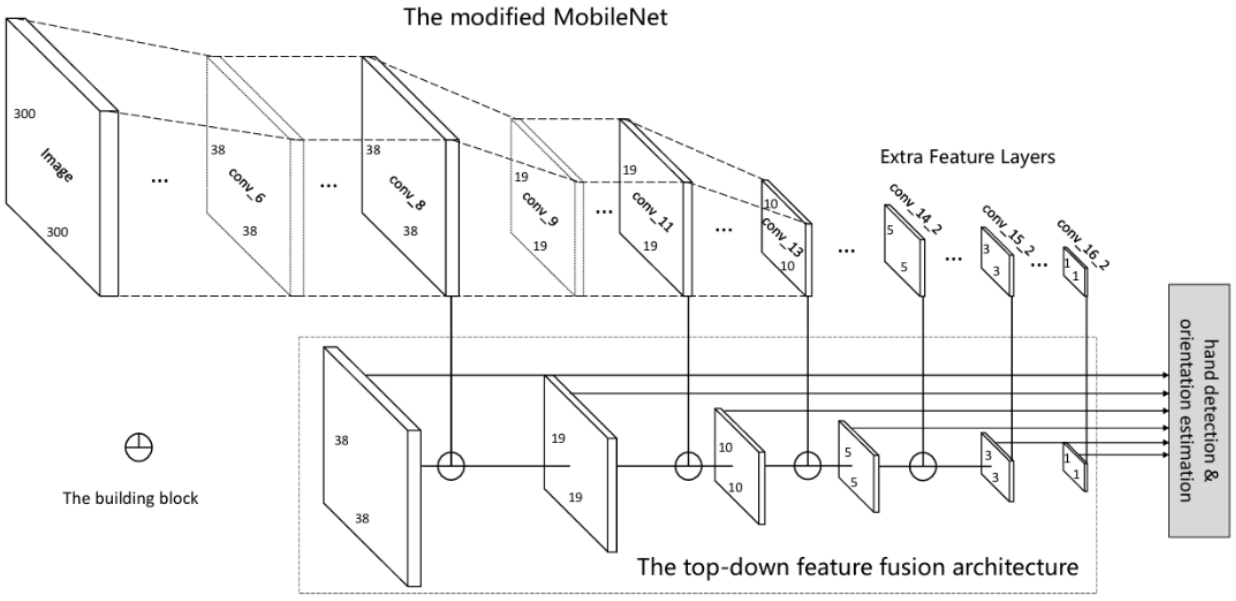
Hình 4: Tính năng trích xuất chỉ được yêu cầu cho các thuật toán ML.

Các mô hình DL tự động thực hiện quá trình trích xuất tính năng, giảm thiểu hoặc loại bỏ nhu cầu can thiệp thủ công. Trong khi ML yêu cầu con người xác định và cung cấp các đặc điểm cụ thể của đối tượng để mô hình phân loại, DL tự động nhận diện các đặc điểm này từ dữ liệu đầu vào. Một ưu điểm quan trọng khác của DL là khả năng xử lý lượng dữ liệu khổng lồ, với khoảng 2,5 tỷ tỷ byte dữ liệu được tạo ra mỗi ngày từ các nguồn như mạng xã hội, phương thức liên lạc và dịch vụ điện tử, cung cấp nguồn tài nguyên lớn cho sự phát triển của DL.

1.2. Các nghiên cứu nước ngoài về phát hiện bàn tay

1.2.1. Phương pháp phát hiện và ước lượng tư thế tay dựa trên CNN

Tôi đã nghiên cứu bài báo "A Light CNN Based Method for Hand Detection and Orientation Estimation" của các tác giả Li Yang và các cộng sự [3]. Bài báo giới thiệu một phương pháp dựa trên mạng nơ-ron tích chập (CNN) để phát hiện và ước tính hướng bàn tay trong ảnh RGB, tập trung vào việc cân bằng giữa độ chính xác và hiệu suất xử lý. Phương pháp sử dụng cấu trúc Single Shot Multibox Detector (SSD) kết hợp với MobileNet đã được tùy chỉnh, tạo ra sáu bản đồ đặc trưng với các độ phân giải khác nhau, hỗ trợ phát hiện bàn tay ở nhiều kích thước. Để cải thiện khả năng nhận diện trong các trường hợp khó khăn như bàn tay nhỏ hoặc bị che khuất, một kiến trúc hợp nhất đặc trưng từ trên xuống được áp dụng, tích hợp thông tin ngữ cảnh trên nhiều mức độ.



Hình 5: Kiến trúc mạng tổng thể của phương pháp.

Hình trên mô tả mô hình mạng nơ-ron sâu dựa trên MobileNet dùng để phát hiện bàn tay và ước lượng hướng bàn tay.

Cấu trúc chính:

- Modified MobileNet: Trích xuất đặc trưng từ hình ảnh đầu vào.
- Hợp nhất đặc trưng: Kết hợp thông tin từ nhiều lớp để cải thiện hiệu quả.
- Lớp bổ sung: Tăng độ chính xác với các đặc trưng sâu hơn.
- Đầu ra: Dự đoán vị trí và hướng bàn tay.

Việc phát hiện bàn tay được thực hiện thông qua dự đoán xác suất và vị trí các hộp giới hạn mặc định, sau đó lọc kết quả bằng thuật toán Non-Maximum Suppression. Phương pháp cũng ước tính hướng bàn tay bằng cách xác định hộp giới hạn xoay qua hai vector vuông góc đại diện cho trục chính và phụ. Thay vì dự đoán trực tiếp góc hoặc vector, mô hình tính toán các phép chiếu của vector lên trục ngang và dọc, kết hợp với vị trí cổ tay để xác định hướng.

Huấn luyện mô hình trên bộ dữ liệu Oxford Hand Dataset, phương pháp sử dụng các kỹ thuật tăng cường dữ liệu và tối ưu hóa hàm mất mát bao gồm các thành phần phân loại, định vị, và ước tính vector. Kết quả cho thấy mô hình đạt độ chính xác trung bình (Average Precision - AP) 83.2%, vượt trội hơn các phương pháp trước đó (Le et al.: 75.1%, Deng et al.: 57.7%) và tốc độ xử lý 139 fps trên GPU Nvidia Titan X, nhanh hơn gần 30 lần so với phương pháp tốt nhất trước đó. Mô hình cũng đạt 63.36% độ chính xác trong việc ước tính hướng bàn tay với sai lệch không quá 10°.

Bảng 1: So sánh hiệu suất phát hiện bàn tay bằng nhiều phương pháp khác nhau.

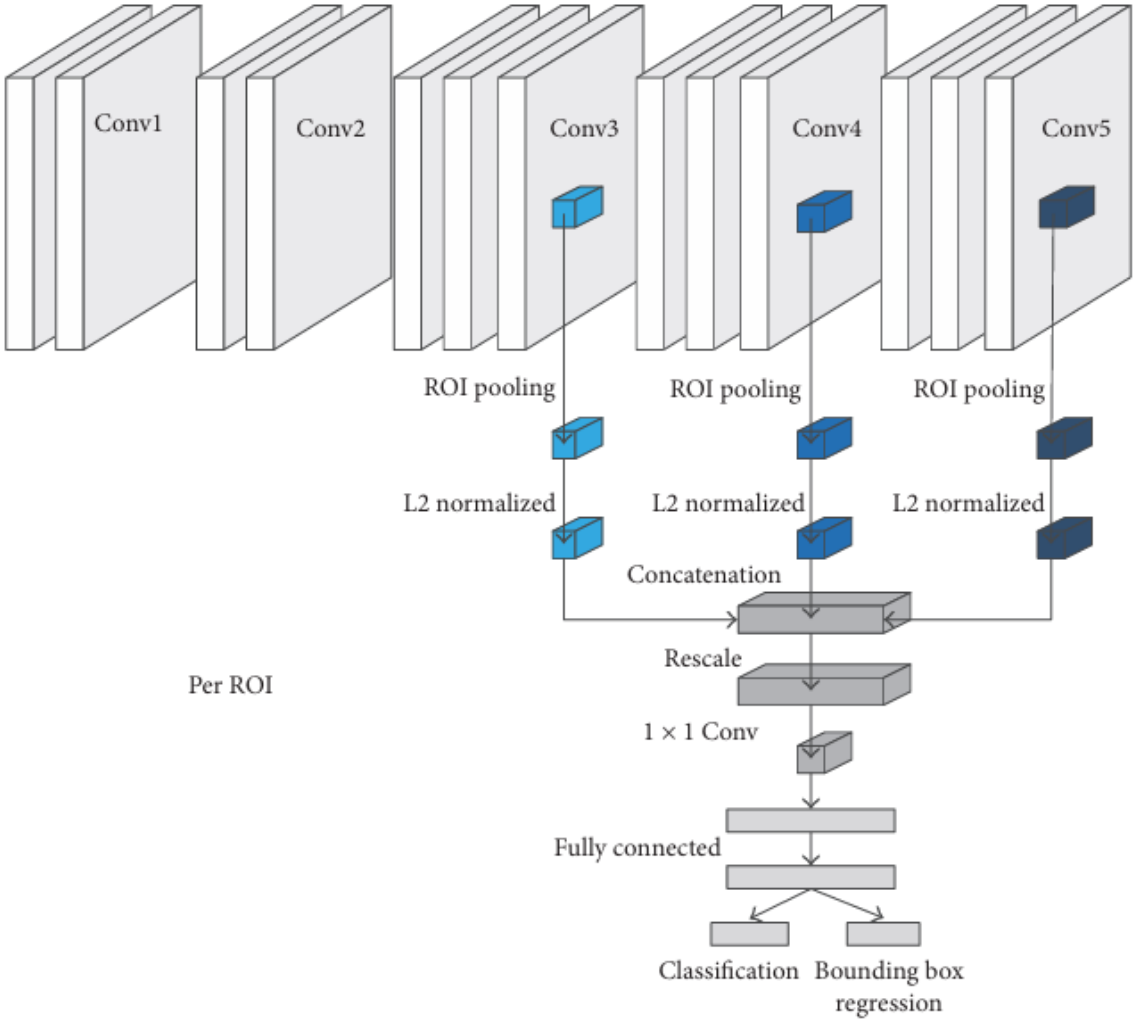
Phương pháp	AP	Thời gian	Môi trường thử nghiệm
Mittal et al. [4]	48.2%	2 phút	quad-core 2.5 GHz CPU
Deng et al. [5]	57.7%	0.1 giây	quad-core 2.9 GHz CPU with Nvidia Titan X
Le et al. [6]	75.1%	0.2150 giây	6 cores 3.5 GHz CPU with Nvidia Titan X
This method	83.2%	0.0072 giây	8 cores 3.0 GHz CPU with Nvidia Titan X

Bảng trên so sánh hiệu suất phát hiện bàn tay giữa phương pháp được đề xuất và các phương pháp trước đó. Kết quả cho thấy phương pháp của bài báo đạt độ chính xác trung bình (AP) cao nhất với 83.2%, vượt xa các phương pháp Mittal et al. (48.2%), Deng et al. (57.7%), và Le et al. (75.1%). Về thời gian xử lý, phương pháp này cũng đạt hiệu suất vượt trội với chỉ 0.0072 giây, nhanh hơn đáng kể so với Le et al. (0.2150 giây) và Deng et al. (0.1 giây), đồng thời rút ngắn thời gian so với Mittal et al. vốn yêu cầu tới 2 phút trên CPU.

Phương pháp này đánh dấu một bước tiến quan trọng trong việc phát hiện và ước tính hướng bàn tay, với tiềm năng ứng dụng lớn trong các lĩnh vực như nhận diện cử chỉ và tương tác giữa người và máy.

1.2.2. Mạng nơ-ron tích chập đa tỉ lệ cho phát hiện bàn tay

Shiyang Yan và các cộng sự [7] đề xuất một phương pháp phát hiện bàn tay trong ảnh tĩnh, sử dụng mô hình học sâu đa tỉ lệ để xử lý các vấn đề về biến đổi kích thước và ngữ cảnh phức tạp. Phương pháp này dựa trên mô hình VGG16 được cải tiến, kết hợp các đặc trưng từ nhiều lớp tích chập để tận dụng thông tin đa tỉ lệ. Hệ thống sử dụng cơ chế Region of Interest (RoI) pooling từ các lớp tích chập khác nhau, cho phép phát hiện các bàn tay nhỏ và xử lý hiệu quả các đối tượng có sự biến đổi lớn về hình dạng và kích thước.



Hình 6: Cấu trúc mô hình của các mạng.

Hình trên mô tả một kiến trúc mạng nơ-ron sâu sử dụng cơ chế RoI pooling để phát hiện đối tượng. Đầu vào được trích xuất đặc trưng qua các lớp Conv1 đến Conv5. Các vùng quan tâm (RoIs) được lấy từ các lớp Conv3, Conv4 và Conv5 thông qua RoI pooling, sau đó chuẩn hóa bằng L2 normalization. Đặc trưng từ các RoIs được kết hợp, chuẩn hóa kích thước (Rescale), và xử lý qua một lớp convolution 1x1 trước khi đưa vào các lớp fully connected. Đầu ra cuối cùng gồm phân loại nhãn (Classification) và dự đoán vị trí hộp giới hạn (Bounding box regression). Kiến trúc này tận dụng thông tin từ nhiều cấp độ đặc trưng để tăng hiệu quả phát hiện.

Cụ thể, thay vì chỉ sử dụng lớp cuối cùng của CNN để biểu diễn đặc trưng, phương pháp này tích hợp các đặc trưng từ các lớp Conv3, Conv4, và Conv5 của VGG16. Kỹ thuật chuẩn hóa L2 được áp dụng để cân bằng giá trị đặc trưng giữa các lớp, kết hợp với các lớp tích chập 1x1 để giảm chiều dữ liệu. Phương pháp sau đó thực hiện phân loại và hồi quy vị trí hộp giới hạn thông qua các lớp fully connected, sử dụng framework Fast R-CNN làm nền tảng.

Hệ thống được đánh giá trên hai bộ dữ liệu benchmark là Oxford Hand Detection Dataset và VIVA Hand Detection Challenge. Trên Oxford dataset, mô hình đạt độ chính xác trung bình (AP) 58.4%, vượt qua các phương pháp trước đó (Mittal et al.: 48.2%, VGG16 baseline: 56.8%). Trên VIVA dataset, phương pháp đạt AP 92.8% ở cấp độ dễ (L1) và 84.7% ở cấp độ khó (L2), trong đó mức L2 bao gồm cả các bàn tay nhỏ và góc nhìn phức tạp. Các kết quả này khẳng định hiệu quả vượt trội của việc sử dụng thông tin đa tỉ lệ trong phát hiện bàn tay, đặc biệt đối với các đối tượng có kích thước nhỏ.

Bảng 2: Độ chính xác trung bình (AP) trên bộ dữ liệu VIVA L1 và L2 và so sánh với các phương pháp trước đó.

Phương pháp	L1 set	L2 set
CNNRegionSampling [8]	66.8%	57.8%
ACFDepth4 [9]	70.1%	60.1%
YOLO [10]	76.4%	69.5%
FRCNN [11]	90.7%	**86.5%**
Mô hình đề xuất (Multiscale Fast R-CNN)	**92.8%**	8 4. 7%

Bảng 3: Độ chính xác trung bình (AP) trên bộ dữ liệu phát hiện bàn tay Oxford và so sánh với các phương pháp trước đó. (Chỉ các trường hợp bàn tay lớn).

Phương pháp	AP
Multiple proposals [12]	48.2%
VGG16	56.8%
Mô hình đề xuất	58.4%

Bảng 2 và Bảng 3 cung cấp so sánh hiệu suất của mô hình đa tỉ lệ được đề xuất với các phương pháp trước đó trên hai bộ dữ liệu khác nhau. Trên bộ dữ liệu VIVA, mô hình đạt độ chính xác trung bình (AP) cao nhất ở cấp độ dễ (L1) với 92.8%, vượt trội so với YOLO (76.4%) và Fast R-CNN (90.7%). Ở cấp độ khó (L2), mô hình đạt AP 84.7%, chỉ thấp hơn một chút so với Fast R-CNN (86.5%)

nhưng vẫn vượt xa các phương pháp khác như YOLO (69.5%). Trên bộ dữ liệu phát hiện bàn tay Oxford, mô hình đạt AP 58.4%, cao hơn so với VGG16 (56.8%) và phương pháp Multiple Proposals (48.2%). Những kết quả này chứng minh ưu điểm của việc sử dụng thông tin đa tỉ lệ, đặc biệt trong việc phát hiện các bàn tay nhỏ hoặc trong các tình huống ngữ cảnh phức tạp.

Tuy nhiên, phương pháp này vẫn gặp khó khăn trong việc phân biệt các đối tượng có hình dáng tương tự bàn tay như cánh tay, bàn chân, hoặc các chi tiết trên quần áo. Các tác giả đề xuất rằng việc tích hợp thêm thông tin ngữ cảnh có thể giúp cải thiện khả năng phân biệt này trong tương lai.

1.3. Nghiên cứu trong nước về phát hiện bàn tay

Tôi đã nghiên cứu bài báo "Nhận dạng cử chỉ bàn tay dùng mạng nơ-ron chập" của nhóm nghiên cứu trường Đại học Sư phạm thành phố Hồ Chí Minh [13]. Trong bài báo này, nhóm tác giả trình bày một thiết kế mạng nơ-ron tích chập cho bài toán nhận dạng cử chỉ bàn tay. Dữ liệu hình ảnh cử chỉ tay được thu thập qua camera và trải qua các bước tiền xử lý, bao gồm chuyển đổi sang thang xám để giảm độ phức tạp, loại bỏ nhiễu và chuẩn hóa kích thước ảnh nhằm đảm bảo tính nhất quán trong đầu vào cho mô hình.

Hệ thống CNN được thiết kế với các lớp chính như lớp tích chập (convolutional layers) để trích xuất đặc trưng không gian, tập trung vào các yếu tố như đường viền và hình dạng của cử chỉ tay. Sau đó, lớp gộp (pooling layers) được sử dụng để giảm kích thước dữ liệu, giúp giảm thiểu số lượng tham số và cải thiện hiệu quả tính toán. Cuối cùng, các lớp kết nối hoàn toàn (fully connected layers) chịu trách nhiệm phân loại các cử chỉ thành các nhãn tương ứng. Thuật toán tối ưu hóa Stochastic Gradient Descent (SGD) và hàm mất mát được sử dụng trong quá trình huấn luyện để cập nhật trọng số, cải thiện độ chính xác của dự

đoán. Dữ liệu được chia thành tập huấn luyện và kiểm tra để đảm bảo đánh giá mô hình một cách khách quan.

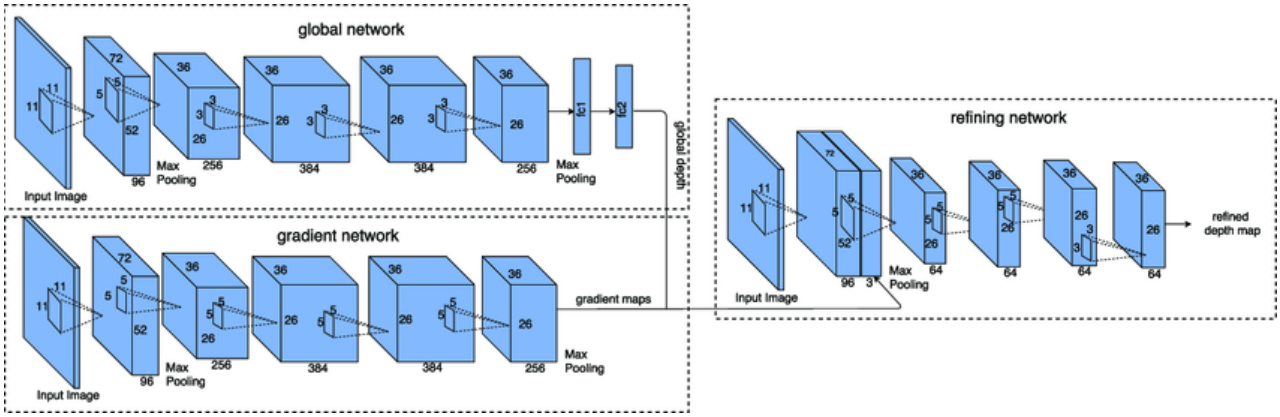
Kết quả cho thấy hệ thống đạt được độ chính xác trung bình rất cao, lên đến 98.6% trong điều kiện thích hợp, vượt trội so với các phương pháp truyền thống. Đặc biệt, trong môi trường nhiều nhiễu hoặc điều kiện ánh sáng không ổn định, mô hình vẫn duy trì được hiệu suất ổn định, chứng minh khả năng thích ứng tốt với các biến động thực tế. Các chỉ số như độ nhạy (sensitivity) và độ đặc hiệu (specificity) cũng được phân tích chi tiết, khẳng định rằng hệ thống không chỉ phân loại chính xác các cử chỉ phổ biến mà còn nhận diện tốt các trường hợp khó.

Ngoài ra, so với các phương pháp trước đây, hệ thống này mang lại lợi thế nhờ khả năng khai thác đặc trưng không gian hiệu quả từ hình ảnh, đồng thời có thể mở rộng ứng dụng trong các lĩnh vực như điều khiển thiết bị thông qua cử chỉ, giao tiếp không chạm, hoặc hỗ trợ các giải pháp thông minh trong nhà. Kết quả nghiên cứu nhấn mạnh tiềm năng của CNN trong việc cải thiện hiệu quả các hệ thống nhận diện cử chỉ tay hiện nay.

1.4. Các phương pháp ước lượng khoảng cách đối với ảnh RGB

1.4.1. Mạng ước lượng độ sâu

Mạng ước lượng độ sâu (Depth Estimation Networks) [14] là phương pháp sử dụng học sâu (deep learning) để dự đoán bản đồ chiều sâu (depth map) từ ảnh RGB. Đây là cách tiên tiến để suy luận thông tin về khoảng cách từ ảnh 2D, vốn không chứa dữ liệu độ sâu trực tiếp.



Hình 7: Cấu trúc mạng ước lượng độ sâu.

Kiến trúc mạng nơ-ron sâu gồm ba thành phần chính: mạng toàn cục (global network), mạng gradient (gradient network) và mạng tinh chỉnh (refining network). Mạng toàn cục trích xuất các đặc trưng tổng thể từ ảnh đầu vào, tạo ra bản đồ độ sâu thô. Mạng gradient tập trung vào việc phân tích các đặc trưng chi tiết, như các biên và sự thay đổi cường độ, để tạo ra bản đồ gradient. Cuối cùng, mạng tinh chỉnh kết hợp thông tin từ mạng toàn cục và gradient để cải thiện độ chính xác của bản đồ độ sâu, tạo ra kết quả hoàn thiện hơn. Kiến trúc này kết hợp các đặc trưng toàn cục và cục bộ, mang lại khả năng ước lượng độ sâu chính xác trong các tình huống phức tạp.

Phân loại mạng ước lượng độ sâu

- Ước lượng độ sâu từ ảnh đơn (Monocular Depth Estimation) [15]

Dự đoán bản đồ chiều sâu chỉ từ một ảnh RGB đầu vào.

+ Nguyên lý: Mạng học các đặc trưng phức tạp trong ảnh như phối cảnh, kích thước tương đối, và ngữ cảnh để suy ra khoảng cách tương đối đến từng điểm ảnh.
+ Ưu điểm: Chỉ cần một camera RGB thông thường, dễ áp dụng trong các hệ thống di động.

+ Hạn chế: Chỉ ước lượng tương đối, không tuyệt đối nếu không có thông tin chuẩn (ground truth), kết quả phụ thuộc nhiều vào chất lượng dữ liệu huấn luyện.

- Ước lượng độ sâu từ ảnh đôi (Stereo Depth Estimation) [16]

Sử dụng hai ảnh chụp từ hai camera hoặc hai góc nhìn khác nhau để tính toán độ sâu dựa trên sự chênh lệch (disparity) giữa các điểm tương ứng.

+ Nguyên lý: Tam giác hóa (Triangulation) được sử dụng để tính khoảng cách từ camera dựa vào chênh lệch giữa các điểm tương ứng trên hai ảnh.
+ Ưu điểm: Độ sâu tuyệt đối và chính xác hơn so với ảnh đơn. không phụ thuộc quá nhiều vào dữ liệu huấn luyện.
+ Hạn chế: Yêu cầu hệ thống camera stereo được căn chỉnh chính xác, khó áp dụng trong trường hợp có nhiều vật che khuất.

- Ước lượng độ sâu từ video (Structure from Motion) [17]

Sử dụng chuỗi ảnh hoặc video (nhiều khung hình từ một camera duy nhất) để tính toán độ sâu thông qua chuyển động của camera.

+ Nguyên lý: Từ sự thay đổi vị trí của các đối tượng trong các khung hình, mạng học cách suy ra độ sâu thông qua cấu trúc chuyển động.
+ Ưu điểm: Không yêu cầu dữ liệu stereo, kết hợp hiệu quả thông tin thời gian (temporal information).
+ Hạn chế: Cần thông tin về chuyển động của camera hoặc đối tượng, khó hoạt động trong môi trường tĩnh hoặc phức tạp.

Nguyên lý hoạt động của mạng ước lượng độ sâu

Đầu vào: Một ảnh RGB đơn hoặc nhiều ảnh từ góc nhìn khác nhau.

Xử lý bởi mạng: Mạng CNN hoặc transformer trích xuất đặc trưng không gian và ngữ cảnh. Các mô hình tích hợp khả năng nhận biết ngữ cảnh, phối cảnh, và các yếu tố hình học để dự đoán độ sâu cho từng điểm ảnh.

Đầu ra: Bản đồ chiều sâu (depth map), trong đó mỗi giá trị tại một điểm ảnh tương ứng với khoảng cách tương đối hoặc tuyệt đối từ camera đến điểm đó.

1.4.2. Phương pháp hình học (Geometric Methods)

Dựa trên các nguyên lý hình học và phối cảnh để tính toán khoảng cách. [18]

Tam giác hóa (Triangulation) [19]

- Nguyên lý: Sử dụng hai camera hoặc hai góc nhìn khác nhau (hệ stereo camera) để tìm sự chênh lệch vị trí của cùng một điểm (disparity).
- Công thức:

$$Z = \frac{f.B}{disparity} \quad (1)$$

Trong đó:

Z: Khoảng cách từ camera đến điểm ảnh.

f: Tiêu cự của camera.

B: Khoảng cách giữa hai camera (baseline).

$disparity$: Sự chênh lệch giữa hai vị trí điểm ảnh.

Tỷ lệ đối tượng (Object Scaling) [20]

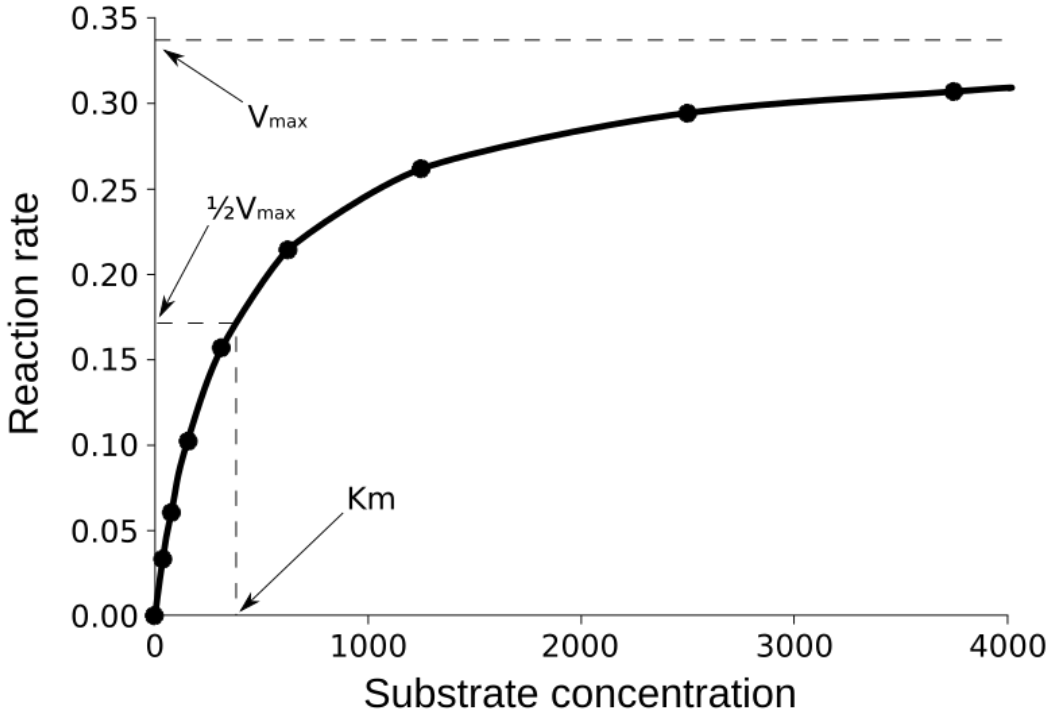
- Nguyên lý: Nếu biết kích thước thực tế của một đối tượng (ví dụ: bàn tay), khoảng cách có thể được suy ra từ kích thước của đối tượng trong ảnh.
- Ứng dụng: Áp dụng khi đối tượng có kích thước cố định (ví dụ: bàn tay có chiều dài trung bình).

CHƯƠNG 2: NGHIÊN CỨU PHÁT TRIỂN MÔ HÌNH ƯỚC LƯỢNG KHOẢNG CÁCH BÀN TAY DỰA TRÊN KHÁI NIỆM HỒI QUY PHI TUYẾN TÍNH

2.1. Khái niệm hồi quy phi tuyến tính và mô hình bình phương của hồi quy phi tuyến tính

2.1.1. Hồi quy phi tuyến tính là gì?

Hồi quy phi tuyến tính(nonlinear regression) [21] là một phương pháp phân tích thống kê dùng để mô hình hóa mối quan hệ giữa một biến phụ thuộc y và một hoặc nhiều biến độc lập x, trong đó quan hệ giữa chúng không phải là một hàm tuyến tính. Thay vào đó, mối quan hệ được thể hiện qua các hàm phi tuyến, ví dụ như lũy thừa, logarit, hàm mũ, hoặc các hàm khác.



Hình 8: Hình ảnh minh họa một đồ thị hồi quy phi tuyến tính điển hình.

Hồi quy phi tuyến tính thường được sử dụng khi dữ liệu có xu hướng không thể được biểu diễn chính xác bởi mô hình tuyến.

Dạng tổng quát của hồi quy phi tuyến tính có thể viết như sau:

$$y = f(x, \theta) + \epsilon \quad (2)$$

$f(x, \theta)$: là một hàm phi tuyến tính của các biến độc lập x, tham số θ là các hệ số cần ước lượng.

ϵ: là nhiễu hoặc sai số ngẫu nhiên.

Có nhiều phương pháp hồi quy khác nhau có thể được áp dụng để mô hình hóa tập dữ liệu, chẳng hạn như hồi quy bậc hai, hồi quy bậc ba và các dạng hồi quy bậc cao hơn, tùy thuộc vào yêu cầu cụ thể của bài toán.

Những giả định trong hồi quy phi tuyến tính tương tự như trong hồi quy tuyến tính nhưng có thể mang những ý nghĩa tinh chỉnh hơn do tính phi tuyến của mô hình. Dưới đây là các giả định chính trong hồi quy phi tuyến tính:

- Hình thức hàm số (Functional Form) [22]: Mô hình phi tuyến tính được chọn phải phản ánh chính xác mối quan hệ thực tế giữa biến phụ thuộc và biến độc lập.
- Tính độc lập (Independence) [23]: Các quan sát được giả định là độc lập với nhau.
- Tính đồng nhất phương sai (Homoscedasticity) [24]: Phương sai của phần dư (chênh lệch giữa giá trị quan sát và giá trị dự đoán) phải không đổi ở mọi mức của biến độc lập.
- Tính phân phối chuẩn (Normality) [23]: Phần dư được giả định tuân theo phân phối chuẩn.
- Đa cộng tuyến (Multicollinearity) [25]: Các biến độc lập không được hoàn toàn tương quan với nhau.

Trong học máy, hồi quy phi tuyến tính được phân thành hai loại chính:

- Hồi quy phi tuyến tính tham số (Parametric Nonlinear Regression) [26]: Loại này giả định rằng mối quan hệ giữa biến phụ thuộc và biến độc lập có thể được mô hình hóa bằng một hàm toán học cụ thể. Ví dụ, mối quan hệ giữa dân số và thời gian có thể được mô hình hóa bằng hàm số mũ.
- Hồi quy phi tuyến tính phi tham số (Non - parametric Nonlinear Regression) [27]: Loại này không giả định rằng mối quan hệ giữa biến phụ thuộc và biến độc lập có thể được mô hình hóa bằng một hàm toán học cụ thể. Thay vào đó, các thuật toán học máy được sử dụng để tìm hiểu mối quan hệ này từ dữ liệu.

2.1.2. Mô hình bình phương của hồi quy phi tuyến tính

Trong hồi quy phi tuyến tính, mô hình bình phương (quadratic model) [28] là một loại mô hình đặc biệt trong đó mối quan hệ giữa biến phụ thuộc và biến độc lập là một đa thức bậc hai.

Dạng tổng quát của mô hình bình phương:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon \quad (3)$$

Trong đó:

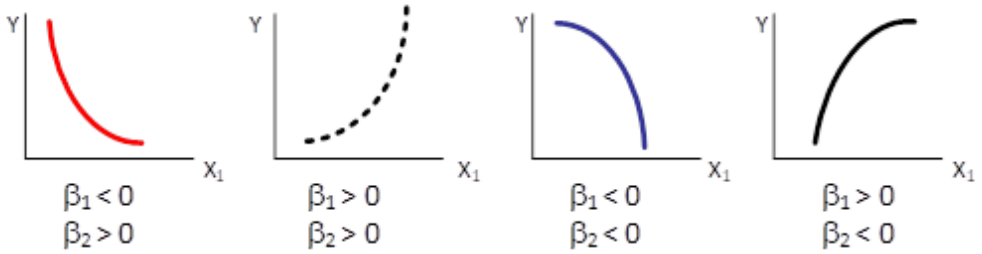
y: biến phụ thuộc

x: biến độc lập

$\beta_0, \beta_1, \beta_2$: các tham số hồi quy

ϵ: nhiễu hoặc sai số ngẫu nhiên

Mô hình này phù hợp khi mối quan hệ giữa y và x không phải là đường thẳng mà có dạng parabol. Tùy vào giá trị của β_2, đồ thị của y theo x có thể lõm xuống ($\beta_2 > 0$) hoặc lõm lên ($\beta_2 < 0$).



Hình 9: Các kiểu dữ liệu dạng mô hình bình phương.

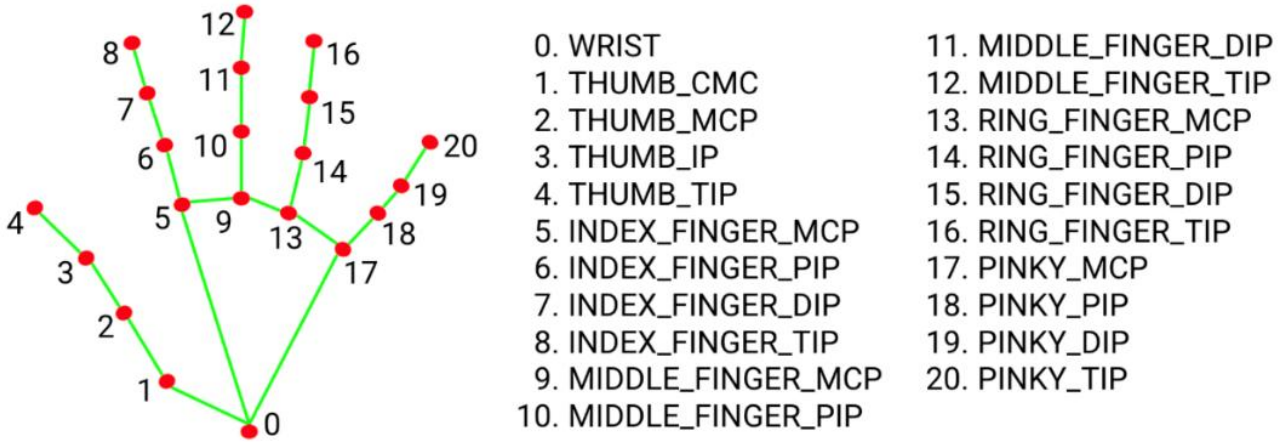
2.2. Phát hiện điểm mốc trên bàn tay bằng MediaPipe

Mediapipe là một framework mã nguồn mở do Google phát triển [29], dùng để xây dựng các ứng dụng xử lý video, âm thanh, và dữ liệu cảm biến thời gian thực. Framework này cung cấp các giải pháp mạnh mẽ dựa trên AI và DL giúp lập trình viên dễ dàng triển khai các chức năng xử lý dữ liệu đa phương tiện, đặc biệt trong lĩnh vực thị giác máy tính (Computer Vision) và nhận diện chuyển động.

Solution	Android	Web	Python	iOS	Customize model
LLM Inference API	●	●		●	●
Object detection	●	●	●	●	●
Image classification	●	●	●	●	●
Image segmentation	●	●	●		
Interactive segmentation	●	●	●		
Hand landmark detection	●	●	●	●	
Gesture recognition	●	●	●	●	●
Image embedding	●	●	●		
Face detection	●	●	●	●	
Face landmark detection	●	●	●		
Face stylization	●	●	●		●
Pose landmark detection	●	●	●		
Image generation	●				●
Text classification	●	●	●	●	●
Text embedding	●	●	●		
Language detector	●	●	●		
Audio classification	●	●	●		

Hình 10: Các giải pháp được cung cấp và tính khả dụng trên các nền tảng.

MediaPipe cung cấp nhiều giải pháp tiên tiến, trong đó có MediaPipe Hands [30], một mô-đun chuyên dụng để phát hiện và nhận diện 21 điểm mốc (landmarks) trên bàn tay. Những điểm mốc này bao gồm các vị trí quan trọng như đầu ngón tay, lòng bàn tay và các khớp nối, đóng vai trò thiết yếu trong việc phân tích cử động bàn tay. Mô hình này được xây dựng dựa trên tập dữ liệu huấn luyện lớn với hơn 30,000 hình ảnh thực tế, kết hợp với dữ liệu tay tổng hợp từ nhiều môi trường và cảnh vật khác nhau, đảm bảo khả năng nhận diện chính xác và ổn định trong nhiều điều kiện ánh sáng và góc nhìn.



Hình 11: Danh sách các điểm mốc mà mô-đun Hands của MediaPipe xác định được.

Mỗi khung hình từ video sẽ được chuyển đổi sang định dạng RGB và gửi qua mô-đun MediaPipe Hands. Mô-đun này sẽ xác định và đánh dấu các điểm mốc trên bàn tay và trả về một landmarks list chứa thông tin về các điểm mốc này dưới dạng tọa độ 3D (x, y, z). Các tọa độ này sau đó được sử dụng để tính toán các khoảng cách giữa các điểm mốc.

MediaPipe Hands có khả năng xử lý dữ liệu video thời gian thực, mang lại độ chính xác cao và hiệu suất vượt trội. Điều này giúp nó trở thành một công cụ lý tưởng trong các ứng dụng như điều khiển bằng cử chỉ, theo dõi chuyển động tay hoặc tích hợp vào các giao diện tương tác người - máy (Human-Machine Interface - HMI)

2.3. Ứng dụng mô hình bình phương của hồi quy phi tuyến tính vào ước lượng lượng khoảng cách từ bàn tay đến camera 2D

2.3.1. Lý do ứng dụng mô hình bình phương vào ước lượng khoảng cách

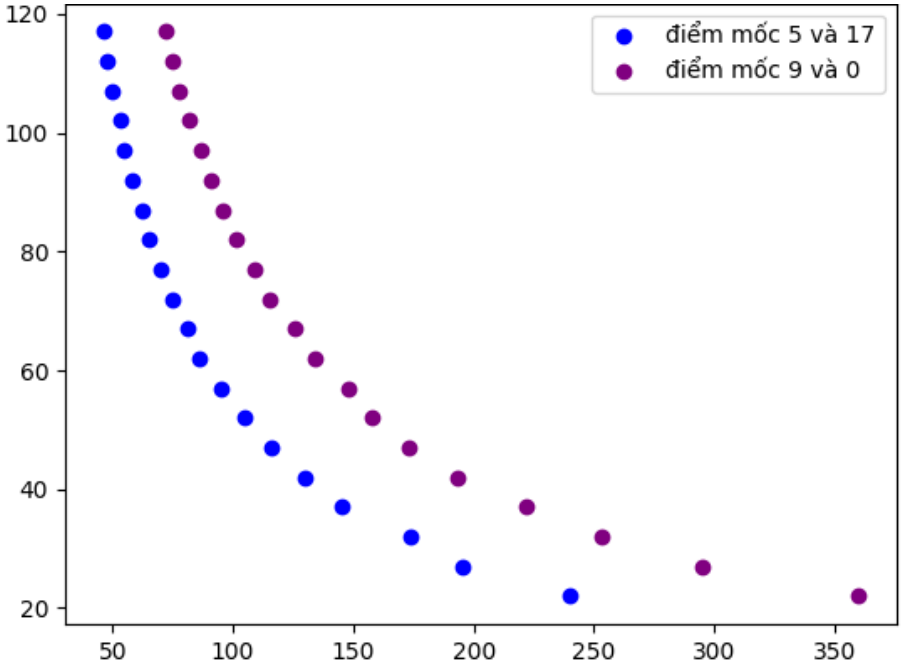
Trong nghiên cứu này, tôi đã tiến hành phân tích sự thay đổi của khoảng cách giữa các điểm mốc trên bàn tay khi vị trí của bàn tay thay đổi so với camera. Mục tiêu của việc phân tích là tìm ra một mô hình phù hợp để ước lượng khoảng cách thực tế giữa bàn tay và camera dựa trên thông tin thu được từ hình ảnh 2D. Sau quá trình phân tích và thử nghiệm, tôi quyết định sử dụng mô hình bình phương của hồi quy phi tuyến, với các lý do chính như sau:

Tôi lựa chọn cặp điểm mốc số 5 và 17 (xem Hình 11) dựa trên tính chất ổn định của khoảng cách giữa chúng trong các chuyển động phức tạp của bàn tay, chẳng hạn như động tác nắm hoặc xòe bàn tay. Tuy nhiên, khi bàn tay thực hiện các động tác xoay cổ tay, tôi nhận thấy khoảng cách giữa hai điểm này có sự thay đổi đáng kể, gây ảnh hưởng đến độ chính xác của phép đo. Để khắc phục vấn đề này, tôi đã bổ sung thêm cặp điểm mốc số 9 và 0, với mục tiêu sử dụng cặp điểm 5 và 17 làm trục ngang, và cặp điểm 9 và 0 làm trục dọc của bàn tay. Sau khi phân tích dữ liệu, khoảng cách giữa các cặp điểm này chứng minh được tính ổn định cao ngay cả khi bàn tay thực hiện nhiều chuyển động linh hoạt. Nhờ đó, chúng trở thành các chỉ báo cố định và đáng tin cậy để ước tính khoảng cách từ bàn tay đến camera.

Một phát hiện quan trọng là hiệu ứng phối cảnh có tác động lớn đến việc đo đạc khoảng cách trong hình ảnh 2D. Khi bàn tay di chuyển gần camera, khoảng cách giữa các điểm mốc trên bàn tay (được tính theo tọa độ 2D) có xu hướng tăng lên. Ngược lại, khi bàn tay di chuyển xa camera, khoảng cách này giảm đi. Đây là đặc điểm chính giúp tôi xác định được mối quan hệ phi tuyến giữa khoảng cách từ bàn tay đến camera và khoảng cách giữa các điểm mốc.

Qua phân tích thực nghiệm, tôi nhận thấy rằng mối quan hệ giữa khoảng cách từ tay đến camera và khoảng cách giữa các điểm mốc trên bàn tay có dạng tương tự một phần của đồ thị hàm số bậc hai (nửa trái của hình parabol với hệ số $a > 0$). Đặc điểm này xuất hiện rõ ràng khi bàn tay ở các vị trí gần hoặc xa camera, giúp tôi dễ dàng xây dựng một mô hình dự đoán chính xác dựa trên dạng quan hệ này.

Dựa trên các lý do nêu trên, việc áp dụng mô hình hồi quy bậc hai là một lựa chọn hợp lý và khoa học để ước lượng khoảng cách thực tế từ bàn tay đến camera.



Hình 12: Mối quan hệ giữa khoảng cách từ tay đến màn hình với khoảng cách giữa hai điểm mốc (5 – 17 và 9 – 0).

2.3.2. Mô hình ước lượng được đề xuất

Trong nghiên cứu này, tôi đã xác định sử dụng mô hình hồi quy phi tuyến tính bậc hai làm cơ sở để ước lượng khoảng cách từ bàn tay đến màn hình camera. Đây là một lựa chọn được cân nhắc kỹ lưỡng sau khi phân tích thực nghiệm về sự thay đổi của khoảng cách giữa các điểm mốc trên bàn tay khi bàn tay di chuyển

trong không gian 3D so với vị trí của camera. Cụ thể, mô hình được đề xuất có dạng phương trình bậc hai như sau:

$$y = ax^2 + bx + c + \epsilon \quad (4)$$

Trong đó:

y: khoảng cách thực tế từ bàn tay đến camera cần được ước lượng.

x: là khoảng các pixel giữa các cặp điểm mốc 5 – 17 và 9 – 0.

a, b, c: là hệ số của hàm bậc hai

ϵ: là sai số ngẫu nhiên

Khoảng cách pixel giữa các cặp điểm mốc 5 – 17 và 9 – 0 được tính toán dựa trên công thức Euclid:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (5)$$

Trong đó:

x_1, x_2: là tọa độ của điểm mốc thứ nhất

y_1, y_2: là tọa độ của điểm mốc thứ hai

Phương pháp này cho phép tính toán khoảng cách pixel giữa các điểm mốc trên hình ảnh 2D, tạo tiền đề cho việc áp dụng mô hình phi tuyến tính bậc hai để ước lượng khoảng cách thực tế.

Để xác định các hệ số a, b, c tôi sử dụng hàm polyfit của thư viện Python numpy. Hàm này áp dụng phương pháp bình phương tối thiểu (least squares) để tối ưu hóa các hệ số sao cho tổng bình phương sai số giữa các giá trị thực tế và giá trị dự đoán là nhỏ nhất. Cụ thể, quá trình thực hiện bao gồm các bước sau:

Bước 1: Thu thập dữ liệu thực nghiệm, bao gồm cặp giá trị:

- Khoảng cách thực tế từ bàn tay đến camera (y) được đo bằng thước dây.
- Khoảng cách pixel giữa các điểm mốc trên hình ảnh 2D (x) được tính toán từ tọa độ các điểm mốc nhận dạng được qua ảnh.

Bước 2: Áp dụng hàm polyfit để xác định các hệ số a, b, c đảm bảo rằng mô hình phù hợp tốt nhất với dữ liệu thực nghiệm.

Bước 3: Kiểm tra tính chính xác của mô hình bằng cách so sánh giá trị dự đoán với giá trị thực tế trên một tập dữ liệu kiểm tra riêng biệt, nhằm đảm bảo khả năng tổng quát hóa của mô hình.

Trong quá trình ước lượng khoảng cách, tôi tính toán khoảng cách pixel giữa các cặp điểm mốc tại mỗi khung hình và áp dụng vào mô hình bậc hai để ước lượng khoảng cách thực tế. Kết quả cuối cùng được xác định bằng cách lấy giá trị nhỏ nhất trong hai dự đoán từ hai cặp điểm:

$$D_{final} = \min(D_{5-17}, D_{9-0}) \quad (6)$$

Cách tiếp cận này giúp giảm thiểu sai số do ảnh hưởng của các yếu tố ngoại cảnh như nhiễu trong quá trình nhận dạng điểm mốc hoặc sự thay đổi góc xoay của bàn tay.

Việc sử dụng mô hình hồi quy bậc hai không chỉ mang lại khả năng ước lượng chính xác mà còn đảm bảo tính hiệu quả trong triển khai thực tế. Với dạng quan hệ phi tuyến tương tự một phần của đồ thị parabol, mô hình đã tận dụng được đặc điểm phối cảnh tự nhiên của hình ảnh 2D khi bàn tay di chuyển xa hoặc gần camera. Sự đơn giản trong cấu trúc toán học nhưng hiệu quả trong dự đoán của mô hình này làm cho nó trở thành một công cụ đáng tin cậy trong các ứng dụng yêu cầu ước lượng khoảng cách từ hình ảnh 2D.

2.4. Cài đặt mô hình và triển khai mô hình

2.4.1. Cấu hình máy tính, môi trường

Cấu hình máy tính được sử dụng: Gigabyte G5 GD, 11th Gen Intel® Core™ i5 Processor H-Series (6 cores, 12 threads, 4,5GHz), 16GB RAM, NVIDIA® GeForce RTX™ 30 Series Laptop GPUs. Hệ điều hành và ngôn ngữ sử dụng: Windows 11 home, 64-bit, python 3.10.15. Môi trường: Anaconda 9.0.

Các thư viện:

- opencv-python 4.5.5
- mediapipe 0.10.8
- numpy 1.26.4
- matplotlib 3.9.2

2.4.2. Thu thập data

Việc thu thập dữ liệu thực nghiệm về khoảng cách thực tế và khoảng cách pixel giữa các điểm mốc trên bàn tay đóng vai trò quan trọng trong việc xây dựng một mô hình hồi quy bậc hai chính xác. Quá trình thu thập dữ liệu thực nghiệm được thực hiện một cách hệ thống và chặt chẽ, nhằm đảm bảo tính khách quan và độ tin cậy của dữ liệu. Quy trình thu thập dữ liệu được chia thành các bước cụ thể như sau:

Bước 1: Ghi hình video: Dữ liệu được thu thập thông qua việc ghi lại video trong điều kiện ánh sáng đầy đủ, nhằm đảm bảo chất lượng hình ảnh sắc nét và độ chính xác cao trong việc xác định các điểm mốc trên bàn tay. Các thông số kỹ thuật cụ thể như sau:

- Độ phân giải: 1280 x 720 pixels
- Tốc độ khung hình: 30 fps

Bước 2: Đặt các vị trí cố định: Để đảm bảo tính chính xác của dữ liệu thu thập, bàn tay được đặt ở các khoảng cách cố định từ camera. Các khoảng cách này được lựa chọn trong phạm vi từ 22 cm đến 117 cm, với các bước đo là 5 cm (Xem Hình 13).

Bước 3: Ghi nhận khoảng cách pixel: Các điểm mốc được lựa chọn là các điểm mốc 5 - 17 và 0 - 9 trên bàn tay, nhằm đảm bảo tính chính xác cao trong việc đo đạc khoảng cách giữa các điểm này. Khoảng cách pixel được xác định thông qua việc phân tích hình ảnh thu được từ camera, từ đó xây dựng mối quan hệ giữa khoảng cách thực tế và khoảng cách pixel (Xem Hình 13).

Bước 4: Xác định khoảng cách thực tế: Đối với mỗi giá trị khoảng cách pixel đã ghi nhận, khoảng cách thực tế từ camera đến bàn tay được đo và ghi lại chính xác. Quá trình đo này được thực hiện bằng các công cụ đo lường chính xác, đảm bảo ít sai số nhất có thể trong quá trình thu thập. Mỗi giá trị khoảng cách thực tế này được lưu trữ và liên kết với giá trị khoảng cách pixel tương ứng, tạo thành bộ dữ liệu chuẩn xác để sử dụng trong mô hình (Xem Hình 13).

Hình 13: Quá trình thu thập dữ liệu.

Bước 5: Lặp lại quy trình thu thập dữ liệu: Để đảm bảo tính chính xác và độ tin cậy của mô hình, quy trình thu thập dữ liệu được lặp lại nhiều lần nhằm giúp kiểm tra tính ổn định của dữ liệu và phát hiện, loại bỏ những sai lệch có thể xảy ra trong quá trình thu thập.

Sau khi hoàn thành quy trình thu thập dữ liệu, kết quả thu được là một tập hợp các giá trị khoảng cách pixel và khoảng cách thực tế.

Bảng 4: Bảng dữ liệu tập hợp các giá trị khoảng cách pixel và khoảng cách thực tế.

Khoảng cách thực tế (cm)	Khoảng cách pixel (5 - 17)	Khoảng cách pixel (9 - 0)	**Khoảng cách thực tế (cm)**	Khoảng cách pixel (5 - 17)	Khoảng cách pixel (9 - 0)
22	240	360	**72**	75	115
27	195	295	**77**	70	109
32	174	253	**82**	65	101
37	145	222	**87**	62	96
42	130	193	**92**	58	91
47	116	173	**97**	55	87
52	105	158	**102**	53	82
57	95	148	**107**	50	78
62	86	134	**112**	48	75
67	81	126	**117**	46	72

2.5. Kết quả

Kết quả hiệu suất tiêu thụ:

Lượng RAM tiêu thụ: 176.6 MB

% CPU tiêu thụ: 20.5%

Điện năng tiêu thụ: rất cao

Kết quả về hiệu suất cho thấy mô hình hoạt động ổn định trong điều kiện thực nghiệm trên máy tính cấu hình trung bình, phù hợp với các ứng dụng thực tiễn yêu cầu tài nguyên phần cứng vừa phải.

Thử nghiệm với 3 video ở các môi trường khác nhau với các thông số kỹ thuật được đề cập ở mục 2.4.2. để đánh giá khả năng ứng dụng của mô hình trong các điều kiện thực tế khác nhau, thử nghiệm được thực hiện với 3 môi trường:

Person1 - Đầy đủ ánh sáng: Trong phòng, ánh sáng đồng đều.

Person2 - Thiếu sáng: Trong phòng tối, ánh sáng hạn chế.

Person3 - Ngoài trời: Ánh sáng tự nhiên thay đổi tùy thời điểm và góc quay.

Hình 14: Thử nghiệm với 3 môi trường ánh sáng khác nhau.

Kết quả trả về của mô hình:

Bảng 5: So sánh kết quả trả về của mô hình với khoảng cách thực tế.

Khoảng cách thực tế (cm)	Person1 (cm)	Person2 (cm)	Person3 (cm)	**Khoảng cách thực tế (cm)**	Person1 (cm)	Person2 (cm)	Person3 (cm)
30	22.71	25.46	22.72	**70**	71.40	76.68	75.47
35	28.00	33.09	29.53	**75**	78.49	85.09	80.33
40	34.52	41.83	36.67	**80**	81.27	87.04	84.28
45	41.83	47.34	42.96	**85**	85.08	93.07	88.02
50	48.43	55.41	50.79	**90**	90.02	97.24	92.36
55	54.36	63.21	56.38	**95**	93.07	99.37	94.10
60	62.43	67.89	64.59	**100**	95.74	101.53	98.30
65	65.59	74.00	69.03				

Dựa vào bảng kết quả so sánh giữa khoảng cách thực tế và khoảng cách đo được của mô hình cho ba tình huống ánh sáng khác nhau (Person1, Person2, Person3), tôi có nhận xét như sau:

Person1 (Đầy đủ ánh sáng):

- Kết quả đo của mô hình khá chính xác khi ánh sáng đều, với sai số nhỏ so với khoảng cách thực tế.
- Sai số thường dao động dưới 10% và ổn định, cho thấy mô hình hoạt động tốt trong điều kiện ánh sáng lý tưởng.

Person2 (Thiếu sáng):

- Sai số lớn hơn đáng kể so với điều kiện ánh sáng đầy đủ.
- Mô hình gặp khó khăn trong việc đo chính xác khi ánh sáng hạn chế, đặc biệt ở các khoảng cách từ 55 cm đến 90 cm, có thể do ảnh hưởng của nhiễu ánh sáng.

Person3 (Ngoài trời):

- Kết quả khá tốt nhưng không ổn định bằng điều kiện đầy đủ ánh sáng trong nhà.
- Ở các khoảng cách xa (trên 80 cm), sai số nhỏ. Tuy nhiên, ở khoảng cách gần hơn (dưới 40 cm), sai số tang.

Kết quả trả về của mô hình hồi quy tuyến tính:

Bảng 6: So sánh kết quả trả về của mô hình hồi quy tuyến tính với khoảng cách thực tế.

Khoảng cách thực tế (cm)	Person1 (cm)	Person2 (cm)	Person3 (cm)	**Khoảng cách thực tế (cm)**	Person1 (cm)	Person2 (cm)	Person3 (cm)
30	28.28	35.68	27.94	**70**	77.14	80.11	79.38
35	39.71	47.71	42.07	**75**	80.60	84.06	81.40
40	48.45	54.51	51.14	**80**	82.58	85.55	83.75
45	56.19	61.90	57.53	**85**	84.56	89.52	86.04
50	62.30	67.62	62.91	**90**	87.53	90.50	88.12
55	66.26	72.19	67.95	**95**	88.52	90.99	89.01
60	71.70	75.01	72.00	**100**	89.51	92.47	90.81
65	74.67	78.13	75.35				

Đánh giá sai số của mô hình hồi quy phi tuyến tính (HQPTT) và mô hình hồi quy tuyến tính (HQTT):

Bảng 7: Đánh giá sai số của mô hình hồi quy phi tuyến tính và mô hình hồi quy tuyến tính trên từng mốc khoảng cách thực tế.

Khoảng cách thực tế (cm)	Sai số mô hình HQPTT	Sai số mô hình HQTT	**Khoảng cách thực tế (cm)**	Sai số mô hình HQPTT	Sai số mô hình HQTT
30	24.3%	5.73%	**70**	2%	10.2%
35	20%	13.46%	**75**	4.65%	7.47%
40	13.7%	21.13%	**80**	1.59%	3.22%
45	07.04%	24.87%	**85**	0.09%	0.52%
50	3.14%	24.6%	**90**	0.02%	2.74%
55	1.16%	20.47%	**95**	02.03%	6.82%
60	04.05%	19.5%	**100**	4.26%	10.49%
65	0.91%	14.88%			

Mô hình hồi quy tuyến tính có sai số thấp ở các khoảng cách gần, đạt mức cao nhất là 24.87% ở khoảng cách 45 cm, nhưng sai số giảm khi khoảng cách thực tế lớn hơn. Ngược lại, mô hình hồi quy phi tuyến tính cho thấy sai số cao ở các khoảng cách ngắn, giảm dần khi khoảng cách thực tế tăng lên, đạt mức thấp nhất là 0.02% ở khoảng cách 90 cm. Nhìn chung, mô hình hồi quy phi tuyến tính cho kết quả chính xác hơn ở hầu hết các mốc khoảng cách, đặc biệt là ở các khoảng cách lớn, trong khi mô hình hồi quy tuyến tính cho kết quả ổn định hơn nhưng sai số cao ở hầu hết các mốc.

2.6. Đánh giá

Ưu điểm:

- Thuật toán đơn giản, dễ triển khai: Mô hình được xây dựng với thuật toán hồi quy bậc hai của hồi quy phi tuyến, đảm bảo khả năng tính toán nhanh và yêu cầu tài nguyên thấp.
- Khả năng hoạt động trên cấu hình phổ thông: Mô hình có thể triển khai trên các hệ thống máy tính không yêu cầu cấu hình cao.
- Độ chính xác tương đối cao: Mô hình đạt sai số ở mức chấp nhận được trong hầu hết các môi trường thử nghiệm, ngay cả trong điều kiện thiếu sáng hoặc ánh sáng thay đổi.

Nhược điểm:

- Quy trình thu thập dữ liệu tốn thời gian: Yêu cầu thực hiện thủ công và nhiều lần lặp lại để đảm bảo độ chính xác.
- Giới hạn ứng dụng: Hiện tại mô hình chỉ hiệu quả với các đối tượng có bounding box rõ ràng.
- Nhạy cảm với điều kiện ánh sáng: Hiệu suất suy giảm khi ánh sáng yếu hoặc không ổn định.

CHƯƠNG 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

3.1. Kết luận

Trong đề tài này, tôi đã tiến hành nghiên cứu và phát triển mô hình dựa trên mô hình bình phương của hồi quy phi tuyến tính nhằm ước lượng khoảng cách từ bàn tay đến camera trong không gian 3D. Các kết quả đạt được có thể tổng kết như sau:

- Thành công trong ước lượng khoảng cách: Tôi đã triển khai mô hình ước lượng khoảng cách bàn tay với độ chính xác tương đối cao, phù hợp với các điều kiện sử dụng thực tế. Mô hình đã được vận hành thành công trên dữ liệu thực nghiệm và hoạt động ổn định trong môi trường xử lý thời gian thực.
- Nền tảng phát triển: Hệ thống được xây dựng và phát triển trên ngôn ngữ lập trình Python, sử dụng các thư viện phổ biến như MediaPipe và NumPy. Mã nguồn của hệ thống được triển khai trên máy chủ sử dụng hệ điều hành Windows, với khả năng mở rộng dễ dàng sang các nền tảng khác nếu cần.
- Kiến thức đạt được:
  + Qua quá trình thực hiện, tôi đã nắm vững hơn về ngôn ngữ lập trình Python, đặc biệt trong các ứng dụng liên quan đến xử lý ảnh và trí tuệ nhân tạo.
  + Đề tài giúp tôi hiểu rõ hơn về các nguyên tắc và ứng dụng của thị giác máy tính, bao gồm phát hiện và phân tích điểm mốc, cũng như các thuật toán hồi quy phi tuyến tính trong việc giải quyết bài toán không gian 2D và 3D.

+ Ngoài ra, tôi còn có cái nhìn sâu sắc hơn về các ứng dụng thực tế của hồi quy phi tuyến, chẳng hạn như trong việc phát hiện, đo lường khoảng cách hoặc định vị các đối tượng trong không gian ảnh.
+ Tóm lại, đề tài không chỉ giúp tôi hoàn thành mục tiêu ban đầu, mà còn mở ra cơ hội để áp dụng kiến thức này vào các dự án lớn hơn trong tương lai, góp phần giải quyết các bài toán thực tế trong lĩnh vực trí tuệ nhân tạo và khoa học máy tính.

3.2. Hướng phát triển

Việc ước lượng với mô hình đề xuất đã mang lại những kết quả khá chính xác và đáng khích lệ trong nhiều trường hợp. Tuy nhiên, trong thực tế, tôi nhận thấy rằng việc triển khai các phương pháp này vào môi trường ứng dụng vẫn gặp phải một số khó khăn và hạn chế đáng kể. Một trong những vấn đề nổi bật là yêu cầu về việc chuẩn hóa dữ liệu, điều này không chỉ đòi hỏi một lượng lớn công sức mà còn tiêu tốn rất nhiều thời gian và tài nguyên. Cụ thể, việc chuẩn hóa dữ liệu đòi hỏi sự can thiệp thủ công, đặc biệt là trong các tình huống có sự biến động về góc nhìn, ánh sáng và các yếu tố ngoại cảnh khác.

Hơn nữa, tính khả thi của phương pháp này hiện tại vẫn còn hạn chế, khi chỉ có thể áp dụng hiệu quả đối với một số đối tượng cụ thể như bàn tay, khuôn mặt và cơ thể người. Việc mở rộng ứng dụng cho các đối tượng khác trong không gian 3D hay các tình huống phức tạp vẫn chưa thể thực hiện được.

Nhằm khắc phục những hạn chế này và hướng đến việc tối ưu hóa khả năng ứng dụng trong thực tế, tôi dự định sẽ kết hợp thuật toán ước lượng với các phương pháp học sâu (Deep Learning) và các thuật toán học máy (Machine Learning) khác. Các kỹ thuật học sâu sẽ giúp cải thiện khả năng nhận diện và phân tích đặc trưng của các đối tượng phức tạp hơn, đồng thời giảm thiểu sự phụ thuộc vào việc chuẩn hóa dữ liệu thủ công. Việc áp dụng các mô hình học sâu không chỉ giúp

nâng cao độ chính xác của các dự đoán mà còn có thể xử lý tốt hơn các biến đổi về môi trường, góc nhìn và điều kiện ánh sáng.

Trong tương lai, tôi kỳ vọng rằng việc tích hợp các kỹ thuật học máy sẽ giúp gia tăng khả năng mở rộng ứng dụng của hệ thống, từ đó hỗ trợ việc tương tác với các vật thể trong môi trường thực tế một cách hiệu quả và linh hoạt hơn. Đây chính là một hướng đi quan trọng, nhằm nâng cao khả năng ứng dụng của hệ thống trong các lĩnh vực như thực tế ảo, giao diện người-máy và các ứng dụng trong ngành công nghiệp 4.0.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Xing Hao, Guigang Zhang, and Shang Ma, “Deep Learning,” 2016.

[2] Krogh, “What are artificial neural networks?,” 2008.

[3] Li Yang, Zhi Qi, Zeheng Liu, Shanshan Zhou, Yang Zhang, Hao Liu, Jianhui Wu, Longxing Shi, “A Light CNN based Method for Hand Detection and Orientation Estimation,” National ASIC System Engineering Research Center, Southeast University, Nanjing, China, 2018.

[4] A. Mittal, A. Zisserman, P.H. Torr, “Hand detection using multiple proposals,” 2011.

[5] X. Deng, Y. Zhang, S. Yang, P. Tan, L. Chang, Y. Yuan, et al., “Joint hand detection and rotation estimation using CNN,” 2017.

[6] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al, “Efficient convolutional neural networks for mobile vision applications,” 2017.

[7] Shiyang Yan, Yizhang Xia, Jeremy S. Smith, Wenjin Lu, and Bailing Zhang, “Multiscale Convolutional Neural Networks for Hand Detection,” 2017.

[8] Bambach S., Lee S., Crandall D. J., Yu C., “Lending a hand: detecting hands and recognizing activities in complex egocentric interactions,” 2015.

[9] Das N., Ohn-Bar E., and Trivedi M. M., “On performance evaluation of driver hand detection algorithms: challenges, dataset, and metrics,” 2015.

[10] Redmon J., Divvala S., Girshick R., Farhadi A., “You only look once: Unified real-time object detection,” 2016.

[11] Zhou T., Pillai P. J., Yalla V. G., “Hierarchical context-aware hand detection algorithm for naturalistic driving,” 2016.

[12] Mittal A., Zisserman A., Torr P., “Hand detection using multiple proposals,” 2011.

[13] Minh-Thanh Le, Van-Ca Phan, Hai-Trang Dang Phuoc, Duy-Tan Do, Ngoc-Son Truong, Hand Gesture Regconition Using Convolutional Neural Network, 2021.

[14] Alican Mertan, Damien Jade Duff, Gozde Unal, “Single image depth estimation: An overview,” 2022.

[15] A. Bhoi, “Monocular Depth Estimation: A Survey,” 2019.

[16] Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, Mohammed Bennamoun, “A Survey on Deep Learning Techniques for Stereo-Based Depth Estimation,” 2022.

[17] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, Amit Singer, “A survey of structure from motion,” 2017.

[18] M.O.Katanaev, “Geometrical methods in mathematical physics,” 2013.

[19] Richard I. Hartley, Peter Sturm, “Triangulation,” 2002.

[20] Edward P.F. Chan, Kevin K.W. Chow, “On multi-scale display of geometric objects,” 2001.

[21] Gallant, “Nonlinear Regression,” 2012.

[22] J. S. Morris, “Functional Regression,” 2015.

[23] Lange, K., Sinsheimer, J. S, “Normal/Independent Distributions and Their Applications in Robust Regression,” 2012.

[24] Kun Yang, Justin Tu, Tian Chen, “Homoscedasticity: an overlooked critical assumption for linear regression,” 2019.

[25] J. I. Daoud, “Multicollinearity and Regression Analysis,” 2017.

[26] C. Büchel, R.J.S. Wise, C.J. Mummery, J.-B. Poline, K.J. Friston, “Nonlinear Regression in Parametric Activation Studies,” 1996.

[27] A. Yatchew, “Nonparametric Regression Techniques in Economics,” 1998.

[28] J. F. Fowler, “The linear-quadratic model,” 2014.

[29] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M.G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, M. Grundmann, “MediaPipe: A Framework for Building Perception Pipelines,” 2019.

[30] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, M. Grundmann, “MediaPipe Hands: On-device Real-time Hand Tracking,” 2020.