**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN KỸ THUẬT MÁY TÍNH – VIỄN THÔNG**

|  |  |  |
| --- | --- | --- |
| **Hướng dẫn :** | **TS. TRẦN VŨ HOÀNG** |  |
| **Sinh viên :** | **ĐẶNG HUY HOÀNG** | **MSSV: 17119076** |
|  | **ĐẶNG HUỲNH THI** | **MSSV: 17119103** |

Hồ Chí Minh, Tháng 11/2020

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

XÂY DỰNG HỆ THỐNG NHẬN DIỆN THỦ NGỮ - NGÔN NGỮ KÍ HIỆU TAY

**BÁO CÁO**



# LỜI CẢM ƠN

Để hoàn thành đề tài nghiên cứu này, lời đầu tiên tôi xin chân thành cảm ơn các thầy cô giáo Trường Đại học Sư Phạm Kỹ Thuật TP.HCM nói chung và các thầy cô trong Khoa Điện - Điện tử nói riêng, những người đã dạy dỗ, trang bị cho tôi những kiến thức nền tảng và chuyên ngành bổ ích, giúp tôi có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ tôi trong quá trình học tập.

Đặc biệt, tôi xin chân thành cảm ơn thầy Trần Vũ Hoàng đã tận tình giúp đỡ, trực tiếp chỉ bảo, đưa ra những định hướng nghiên cứu cũng như hướng giải quyết một số vấn đề cho tôi trong suốt thời gian nghiên cứu đề tài. Trong thời gian làm việc với thầy, tôi không ngừng tiếp thu thêm nhiều kiến thức, thái độ nghiên cứu khoa học nghiêm túc, hiệu quả, đây là điều rất cần thiết trong quá trình học tập và công tác sau này.

Đồng thời tôi cũng gửi lời cảm ơn tới bạn bè, các anh chị trong Khoa Điện - Điện tử đã góp ý, chia sẻ kinh nghiệm, hỗ trợ trong quá trình nghiên cứu và thực hiện đề tài tốt nghiệp.

Tôi xin chân thành cảm ơn!

# MỤC LỤC

TP.HCM, ngày 29 tháng 11 năm 2020

Người thực hiện đề tài

[LỜI CẢM ƠN i](#_Toc61100763)

[MỤC LỤC i](#_Toc61100764)

[DANH MỤC HÌNH iii](#_Toc61100765)

[DANH MỤC BẢNG iv](#_Toc61100766)

[Chương 1 MỤC TIÊU CỦA PROJECT 1](#_Toc61100767)

[Chương 2 CÁC THỬ THÁCH CỦA PROJECT 2](#_Toc61100768)

[2.1. Mô hình để train nặng (VGG16) 2](#_Toc61100769)

[2.2. Nhiễu do background 2](#_Toc61100770)

[2.3. Ánh sáng 2](#_Toc61100771)

[2.4. Kích thước bàn tay 2](#_Toc61100772)

[Chương 3 NỘI DUNG THỰC HIỆN 3](#_Toc61100773)

[Chương 4 KẾT QUẢ 4](#_Toc61100774)

[4.1. Tập dữ liệu sử dụng 4](#_Toc61100775)

[4.2. Phương pháp huấn luyện mô hình 4](#_Toc61100776)

[4.3. Đánh giá mô hình sau khi train mô hình cuối cùng 5](#_Toc61100777)

[Chương 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 7](#_Toc61100778)

[Chương 6 CÁC PHƯƠNG PHÁP ĐÃ ĐƯỢC ĐỀ XUẤT 8](#_Toc61100779)

[6.1. CNN (Convolutional Neural Network) 8](#_Toc61100780)

[6.2. EGM (Elastic Graph Matching) 9](#_Toc61100781)

[6.3. SLLE (Supervised Locally Linear Embedding) 9](#_Toc61100782)

[Chương 7 LỰA CHỌN GIẢI PHÁP 11](#_Toc61100783)

[Chương 8 TẬP DỮ LIỆU DỰ KIẾN SẼ SỬ DỤNG 15](#_Toc61100784)

[8.1. Những vấn đề đã giải quyết trong đồ án 15](#_Toc61100785)

[Chương 9 PHƯƠNG PHÁP ĐÁNH GIÁ DỰ KIẾN SẼ SỬ DỤNG 15](#_Toc61100786)

[9.1. Cross Validation 16](#_Toc61100787)

[9.2. Bias and variance 16](#_Toc61100788)

[9.3. Precision, Recall 17](#_Toc61100789)

# DANH MỤC HÌNH

[Hình 4.1. Bộ dữ liệu. 4](#_Toc61100790)

[Hình 4.2.Đường cong học tập khi huấn luyện mô hình theo kiểu chia dữ liệu train:val:test = 64:16:20. 4](#_Toc61100791)

[Hình 4.3. K-Fold Cross Validation với K=5. 5](#_Toc61100792)

[Hình 4.4. Đường cong học tập khi huấn luyện mô hình theo kiểu chia train:test = 80:20, và áp dụng K-Fold Cross Validation để đánh giá chéo. 5](#_Toc61100793)

[Hình 4.5. Bảng Confusion Matrix cho mô hình cuối cùng. 6](#_Toc61100794)

[Hình 4.1. Sơ đồ cấu trúc mạng CNN. 8](#_Toc61100795)

[Hình 4.2. Cách tính tích chập của một ảnh. 8](#_Toc61100796)

[Hình 4.3. Mô tả hoạt động của EGM. 9](#_Toc61100797)

[Hình 4.4. Mô tả hoạt động của SLLE, nâng cấp của LLE. 9](#_Toc61100798)

[Hình 4.5. Mô tả hoạt động của thuật toán LLE. 10](#_Toc61100799)

[Hình 5.1. Biểu đồ độ chính xác. 11](#_Toc61100800)

[Hình 5.2. Biểu đồ số lượng class của các phương pháp. 11](#_Toc61100801)

[Hình 5.3. Nhận diện trên nền background bị nhiễu dùng phương pháp CNN. 12](#_Toc61100802)

[Hình 5.4. Nhận diện trên nền background bị nhiễu dùng phương pháp EGM. 12](#_Toc61100803)

[Hình 5.5. Nhận diện trên nền background bị nhiễu dùng phương pháp SLLE 13](#_Toc61100804)

[Hình 6.1. Bộ dữ liệu. 15](#_Toc61100805)

[Hình 7.1. Phương pháp k-fold Cross val được sử dụng cho các tập data có ít dữ liệu. 16](#_Toc61100806)

[Hình 7.2. High variance và high bias 16](#_Toc61100807)

[Hình 7.3. Công thức tính precision và recall. 17](#_Toc61100808)

# DANH MỤC BẢNG

[Bảng 5.1. Bảng số liệu về các vấn đề khác. 12](#_Toc61100035)

[Bảng 5.2. Bảng tổng kết kết quả so sánh. 12](#_Toc61100036)

# MỤC TIÊU CỦA PROJECT

Sự ra đời và phát triển của máy tính đã giúp ích rất nhiều cho công việc và cuộc sống của con người. Với máy tính, con người có thể làm việc và giải trí theo nhu cầu sử dụng của cá nhân... Tuy nhiên, việc giao tiếp giữa con người và máy tính phụ thuộc chủ yếu vào bàn phím và chuột, hầu như con người luôn phải ngồi thao tác trực tiếp với máy tính. Dần dần, các nhà sản xuất thấy được sự bất tiện và đã tạo ra bàn phím và chuột không dây với mong muốn mang lại sự tự do hơn cho người dùng. Tuy nhiên,với bàn phím không dây thì con người vẫn phải tương tác với máy tính bằng các phím cơ học. Con người chỉ thật sự được thoải mái khi việc tương tác với máy tính được thực hiện thông qua ngôn ngữ cử chỉ và giọng nói. Đó chính là vấn đề đặt ra cho bài toán phát hiện bàn tay.

Trong cuộc sống hàng ngày, phát hiện bàn tay có thể giúp cho việc giao tiếp giữa người với máy tính, vì máy tính sẽ chuyển ngôn ngữ cử chỉ thành chữ viết. Trong công nghiệp và sản xuất, chỉ cần trang bị cho các robot hệ thống camera, việc điều khiển robot sẽ trở nên đơn giản hơn bao giờ hết. Trong lĩnh vực đồ họa 3 chiều, ta có thể dùng một số động tác yêu cầu máy tính xoay mô hình theo ý muốn của mình. Trong công việc văn phòng, phát hiện bàn tay giúp ta có thể yêu cầu máy tính thực thi một chương trình, mở một bài hát, gửi một lá thư ... chỉ với một vài cử chỉ ra hiệu từ xa. Trong lĩnh vực giải trí, các trò chơi thực tế ảo (Virtual Reallity, người chơi sẽ điều khiển hành động nhân vật bằng chính hành động của mình. Với những yêu cầu thực tế và ứng dụng đem lại của lĩnh vực phát hiện bàn tay, mục tiêu của đề tài này là là tìm hiểu về phương pháp phát hiện bàn tay, từ đó xây dựng được mô hình phù hợp cho bài toán phát hiện bàn tay, áp dụng vào điều khiển chương trình máy tính.

# CÁC THỬ THÁCH CỦA PROJECT

## Mô hình để train nặng (VGG16)

Project nhận dạng cử chỉ tay được thực hiện bằng việc sử dụng mạng CNN, model nổi tiếng VGG16. Do đó để tăng cường độ chính xác cho project thì mạng CNN phải đủ phức tạp và đủ feature. Điều này dẫn đến yêu cầu về phần cứng cao, những phần cứng cơ bản thông thường không thể thực hiện được hoặc có thể thực hiện nhưng sẽ có hiện tượng giật lag, độ trễ lớn, delay nhiều.

## Nhiễu do background

Hầu hết các bài toán liên quan đến nhận dạng xử lý ảnh đề phải gặp vấn đề về việc nhiễm do background và đề này này cũng không ngoại lệ. Để có thể chiết xuất lấy phần cánh tay và loại bỏ phần backgroud sẽ khá phức tạp do background sẽ không có định, tùy vào mỗi người mà cammera nhìn thấy thì background sẽ thay đổi. Nếu background có màu đen hoàn toàn , hay một màu nào đó tương phản với màu da và đồng màu trên cả khung hình thì việc xử lý sẽ dễ dàng hơn. Nhưng đời không như là mơ, hầu hết các trường hợp khi ứng dụng thực tế sẽ chẳng có nơi nào như vậy. Môi trường hoạt động của project sẽ là những nơi đông người,trong nhà …. Vì vậy xử lý nhiễu background là vấn đề bắt buộc phải quan tâm nếu muốn đề tài có thể ứng dụng được.

## Ánh sáng

Giống như background thì ánh sáng cũng là yếu tố quan trọng ảnh hưởng lớn để kết quả đánh giá của model. Việc thiếu hay thừa ánh sáng sẽ dẫn đến độ tương phản giữa bàn tay và background không rõ ràng, việc này dẫn đến nhận dạng sai ký tự.

## Kích thước bàn tay

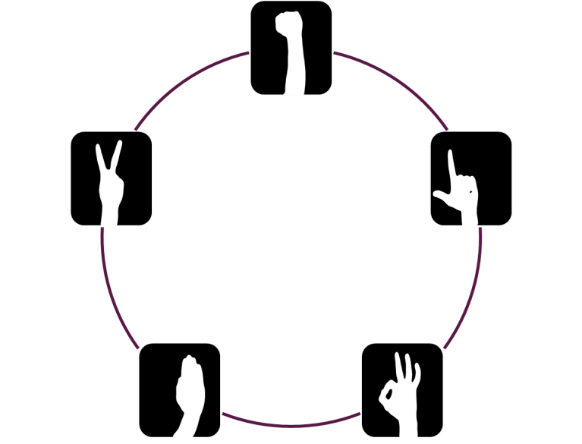
Trong khi nhận dạng, việc xác định tư thế các ngón tay sẽ tương ứng với cử chỉ nào sẽ phụ thuộc phần lớn vào dataset của chúng ta. Nếu dataset đủ lớn với các dữ liệu về một tư thế tay với nhiều kích thước to nhỏ khác nhau sẽ giúp vấn đề này được giải quyết. Nhưng dataset thường sẽ không có như vậy và nếu có thì sẽ khiến cho lượng dataset tăng lên đáng kể và khiến cho việc train tốn rất nhiều thời gian.

# NỘI DUNG THỰC HIỆN

# KẾT QUẢ

## Tập dữ liệu sử dụng

Việc thu thập dataset cho toàn bộ ký tự trong bảng chữ cái khá khó khăn và việc train hết dữ liệu đó trên máy tính chúng tôi không thể được. Để tối ưu hơn về thời gian train để lựa chọn thông số, ở bài này chúng tôi sử dụng 2750 ảnh cho 5 kiểu ký tự tay, với mỗi kiểu ký tự là 550 ảnh.



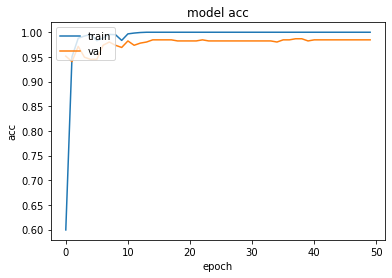
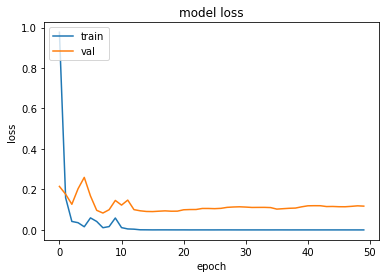
Hình 4.1. Bộ dữ liệu.

## Phương pháp huấn luyện mô hình

Ban đầu, dataset sẽ được chia làm 2 tập train, test. Với tập train là 80% ,tương ứng 2200 ảnh và 20% test 110 ảnh. Tập train tiếp tục chia thành tập train và vali với tỷ lệ 8:2. Công việc huấn luyện có thể hình dung như sau:

*Huấn luyện trên tập train và val, sau đó lấy giá trị loss rồi vẽ đường learning curve (đường cong học tập) của mô hình để xác định xem nên dừng việc train ở epoch bao nhiêu.*

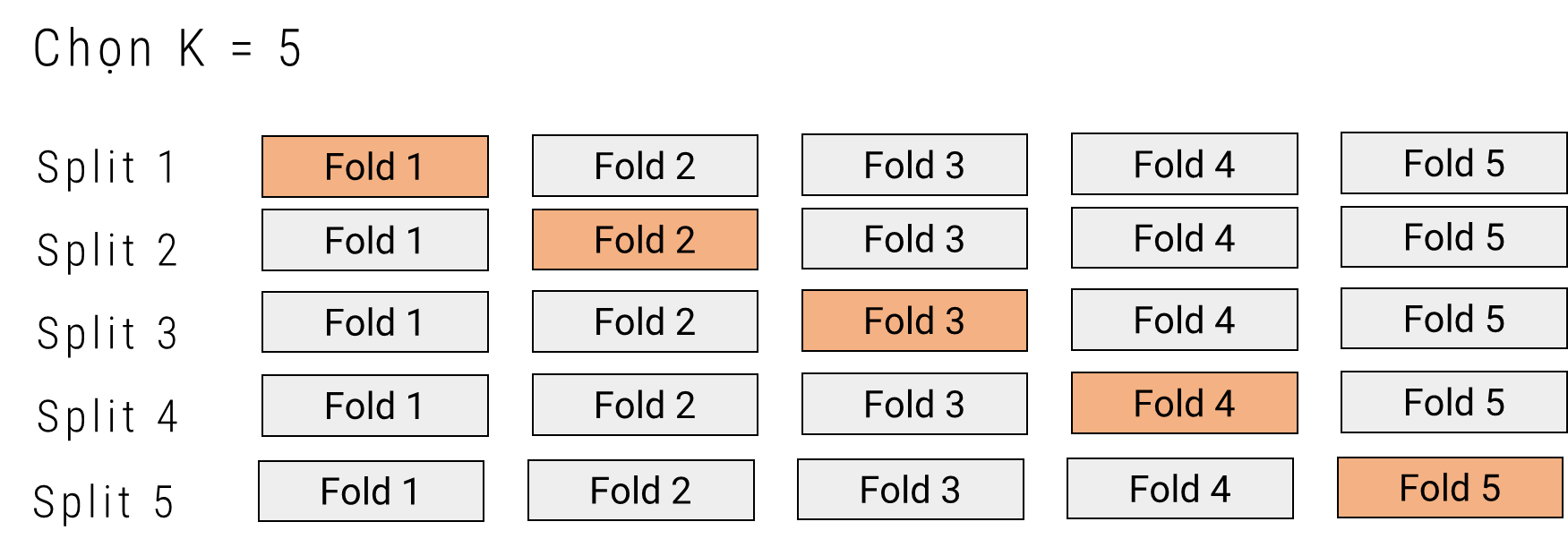
Kết quả sau khi huấn luyện được thể hiện như hình sau:



Hình 4.2.Đường cong học tập khi huấn luyện mô hình theo kiểu chia dữ liệu train:val:test = 64:16:20.

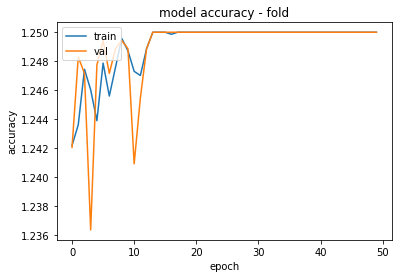
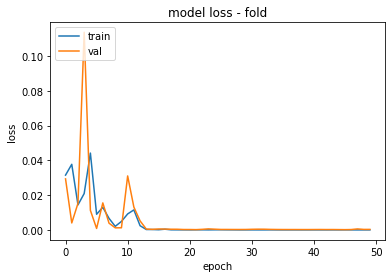
Nhận xét: nhìn vào đường loss line (hình phía bên trái), ta nhận thấy vẫn còn khoảng cách lớn giữa hai đường loss line và khoảng cách này có xu hướng tăng dần. Lúc này mô hình đang rơi vào trường hợp “tập train không thể hiện hết toàn bộ tập dữ liệu” (Unpresentative train dataset).

Khắc phục: xác định lại cách chia dữ liệu. Ta sẽ chia tập dữ liệu thành hai tập: tập train (chiếm 80%) và tập test (chiếm 20%). Sau đó áp dụng K-Fold Cross Validation (nhóm chọn K = 5) để đánh giá chéo.



Hình 4.3. K-Fold Cross Validation với K=5.

Kết quả huấn luyện sau khi chia lại tập dữ liệu:



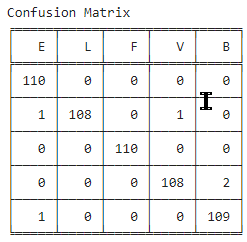
Hình 4.4. Đường cong học tập khi huấn luyện mô hình theo kiểu chia train:test = 80:20, và áp dụng K-Fold Cross Validation để đánh giá chéo.

Nhận xét: với cách chia dữ liệu như thế này, đồ thị đường cong học tập rất tốt và ta cũng nhận thấy ở đồ thị ổn định bắt đầu từ epoch 30 trở đi, vì vậy ta chọn epoch 30 làm điểm dừng khi train mô hình cuối cùng trên toàn bộ 80% tập dữ liệu.

## Đánh giá mô hình sau khi train mô hình cuối cùng

Sau khi có được mô hình cuối cùng, nhóm áp dụng Confusion Matrix lên tập test và tính độ chính xác dựa theo confusion matrix đó.

Kết quả sau khi lập bảng confusion matrix:



Hình 4.5. Bảng Confusion Matrix cho mô hình cuối cùng.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

# CÁC PHƯƠNG PHÁP ĐÃ ĐƯỢC ĐỀ XUẤT

Theo khảo sát của chúng tôi qua các bài báo từ nước ngoài, có rất nhiều phương pháp được đề xuất và thực hiện trước đây. Trong bài nguyên cứu này chúng tôi sẽ chọn ra 3 phương pháp để đánh giá.

* Học có giám sát: CNN, SLLE
* Học không giám sát: EGM

## CNN (Convolutional Neural Network)



Hình 4.1. Sơ đồ cấu trúc mạng CNN.

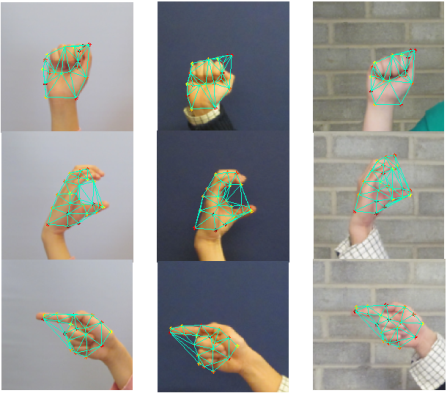
Là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node.



Hình 4.2. Cách tính tích chập của một ảnh.

Các convolutional layer(cửa sổ trươc) có các parameter(kernel) đã được học để tự điều chỉnh lấy ra những thông tin chính xác nhất mà không cần chọn các feature. Subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).

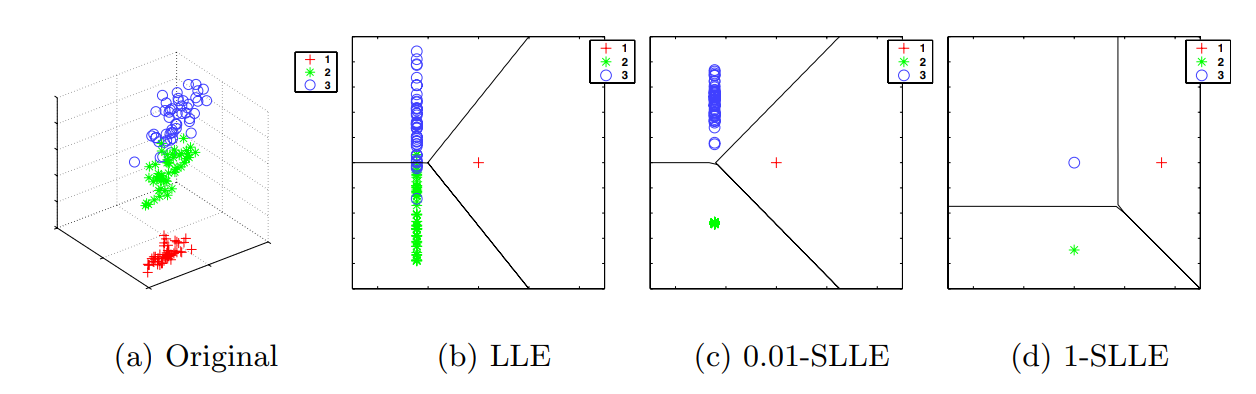
## EGM (Elastic Graph Matching)



Hình 4.3. Mô tả hoạt động của EGM.

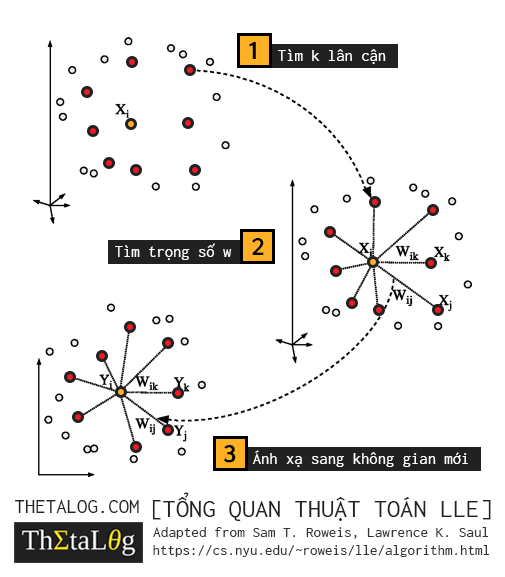
Là một kiến trúc nhận dạng mẫu lấy cảm hứng từ thần kinh. EGM có khả năng cố hữu để xử lý các biến dạng hình học, không yêu cầu hình ảnh đầu vào được phân đoạn hoàn hảo và có thể biểu diễn một cách trang nhã các phương sai về hình dạng đối tượng .Thông qua các bộ lọc wavelet Gabor để tìm các nút, các nút này đại diện cho kết cấu của cử chỉ tay.

## SLLE (Supervised Locally Linear Embedding)



Hình 4.4. Mô tả hoạt động của SLLE, nâng cấp của LLE.

Là mở rộng của LLE (Locally Linear Embedding). LLE sẽ giảm chiều dữ liệu dựa vào khoảng cách của các điểm đến các điểm k lân cận.



Hình 4.5. Mô tả hoạt động của thuật toán LLE.

Ở hình trên ,chúng tôi thấy nó chạy khá giống với thuật toán giảm chiều

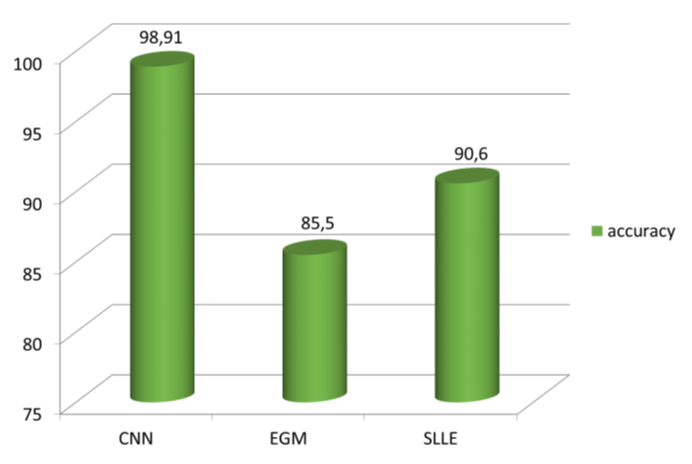
* Đầu tiên tìm lân cận k( điểm màu đỏ).
* Xét khoảng cách từ điểm dữ liệu (điểm màu vàng) đến điểm k lân cận tính trọng số w.
* Ánh xạ sang không gian mới.

Nếu ta tăng giả khoảng cách thì các điểm này sẽ dần cách xa ra và đó là ý tưởng giải quyết bài toán phân loại.

# LỰA CHỌN GIẢI PHÁP

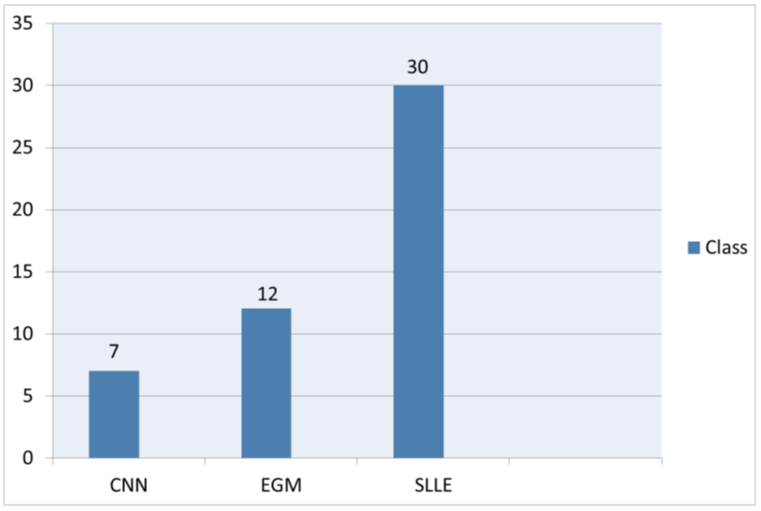
Để chọn một giải pháp phù hợp với tiêu chí mà chúng tôi đã đặt ra cho đề tài này nên chúng tôi đã làm bảng đánh giá, so sánh các giải pháp với nhau để lựa chọn giải pháp tốt nhất.

* Về độ chính xác



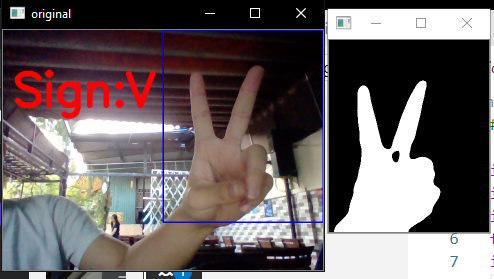
Hình 5.1. Biểu đồ độ chính xác.

* Về số lượng class



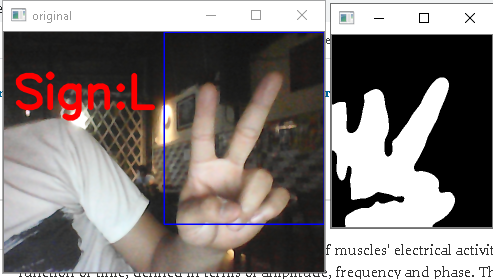
Hình 5.2. Biểu đồ số lượng class của các phương pháp.

* Về nhiễu do background
* CNN



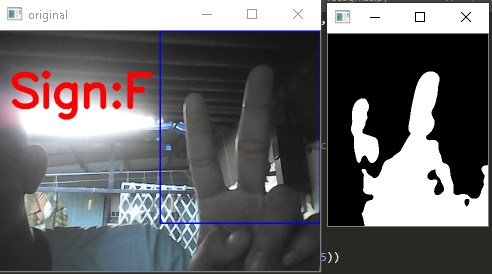
Hình 5.3. Nhận diện trên nền background bị nhiễu dùng phương pháp CNN.

* EGM



Hình 5.4. Nhận diện trên nền background bị nhiễu dùng phương pháp EGM.

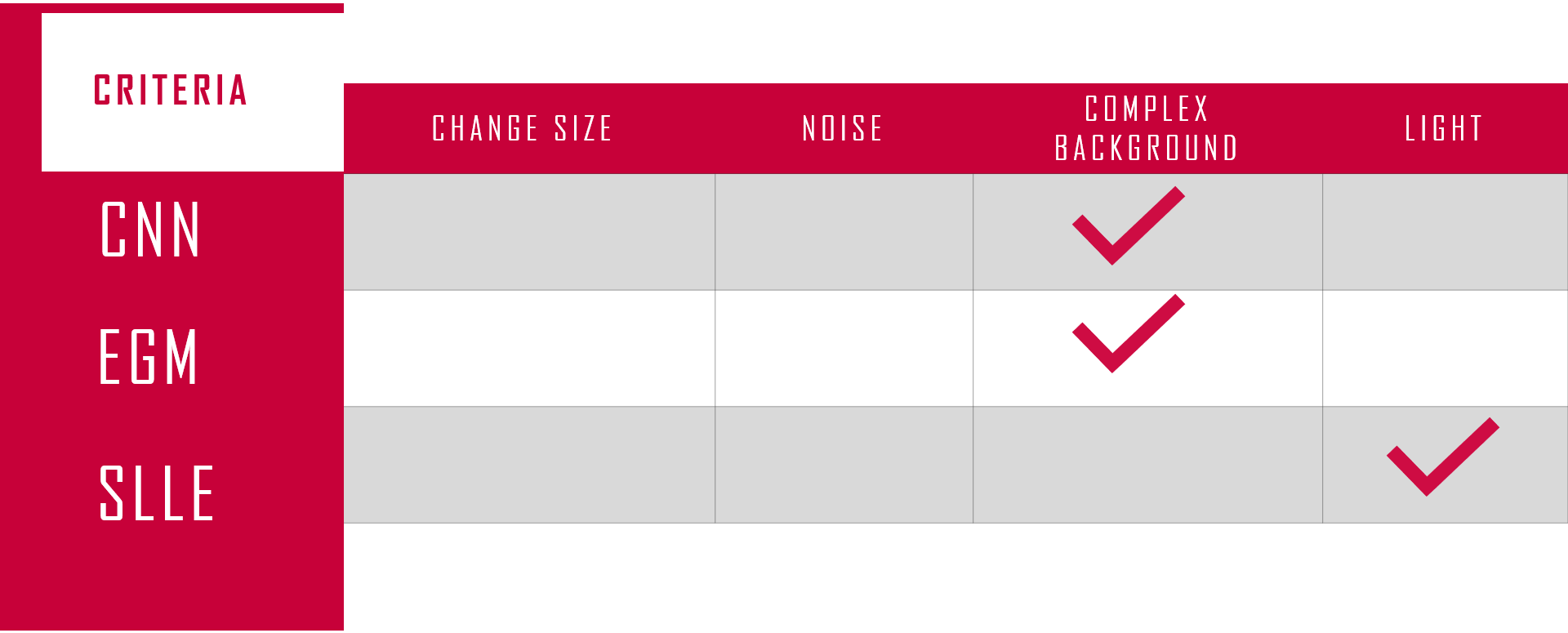
* SLLE



Hình 5.5. Nhận diện trên nền background bị nhiễu dùng phương pháp SLLE

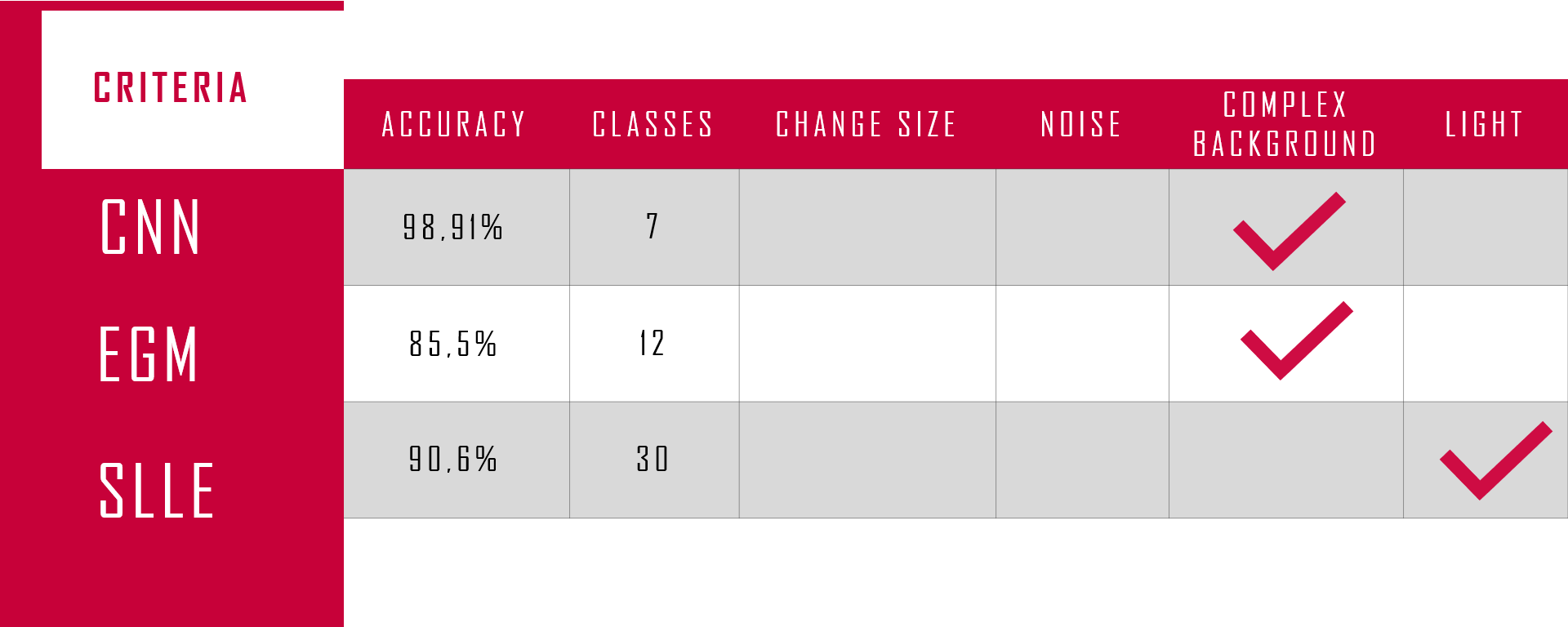
* Các vấn đề khác

Bảng 5.1. Bảng số liệu về các vấn đề khác.



* Bảng tổng kết kết quả so sánh

Bảng 5.2. Bảng tổng kết kết quả so sánh.



Qua khảo sát phương pháp dùng CNN có tỷ lệ accuracy cao nhất, số lượng class lại ít nhất. Việc background phức tạp đã được xử lý nhưng các vấn đề về độ sáng, nhiễu do tay nhiều người hay thay đổi kích thước tay vẫn chưa được xử lý. Tuy nhiên độ sáng có thể được khắc phục bằng cách thay đổi vị trí sao cho phù hợp chứ không cần phải xử lý code phức tạp, việc thay đổi kích thước bàn tay có thể tăng dataset lên nhưng sẽ khiến cho việc train trở nên khó khăn và mất thời gian hơn.

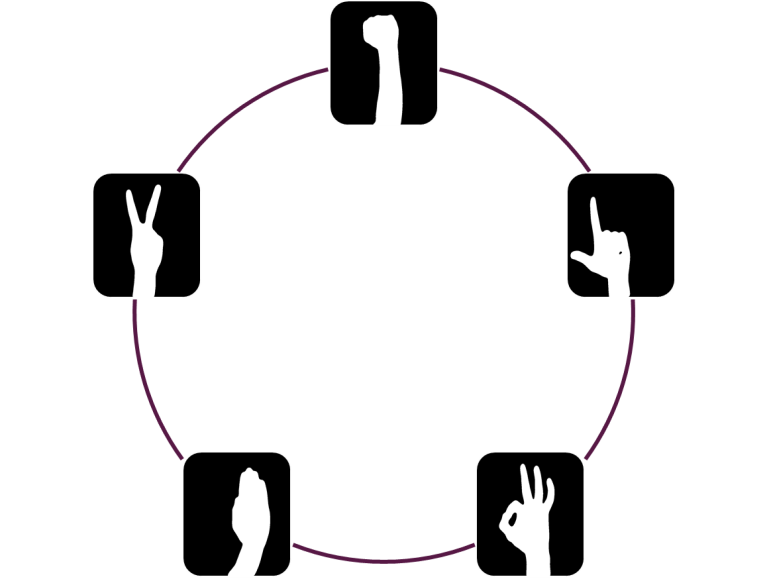
Sau khi xem xét với mục tiêu và yêu cầu đặt ra, chúng tôi quyết định chọn phương pháp sử dụng mạng CNN làm phương pháp sử dụng trong đề tài này.

Sau quá trình nghiên cứu và thiết kế hệ thống, tôi đã đưa ra mô hình của sản phẩm, mô hình gồm 2 phần chính: mạch chạy và ứng dụng trên laptop.

# TẬP DỮ LIỆU DỰ KIẾN SẼ SỬ DỤNG

## Những vấn đề đã giải quyết trong đồ án

Việc thu thập dataset cho toàn bộ ký tự trong bảng chữ cái khá khó khăn và việc train hết dữ liệu đó trên máy tính chúng tôi không thể được. Để tối ưu hơn về thời gian train để lựa chọn thông số, ở bài này chúng tôi sử dụng 2750 ảnh cho 5 kiểu ký tự tay, với mỗi kiểu ký tự là 550 ảnh.



Hình 6.1. Bộ dữ liệu.

Dataset sẽ được chia làm 2 tập train, test. Với tập train là 80% ,tương ứng 2200 ảnh và 20% test 110 ảnh.

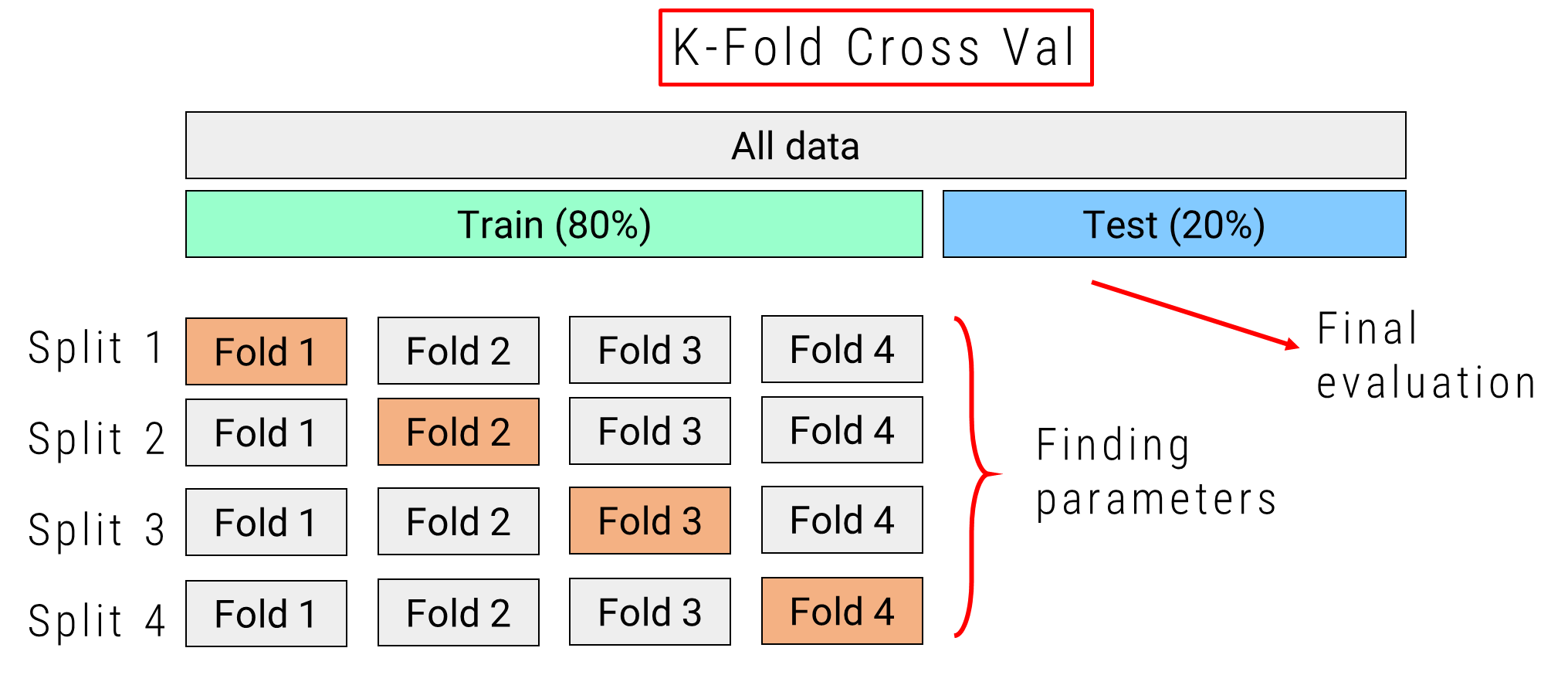
Tập train tiếp tục chia thành tập train và vali với tỷ lệ 8:2.

# PHƯƠNG PHÁP ĐÁNH GIÁ DỰ KIẾN SẼ SỬ DỤNG

Chúng tôi dự định sử dụng 3 phương pháp đánh giá là :

* Cross Validation

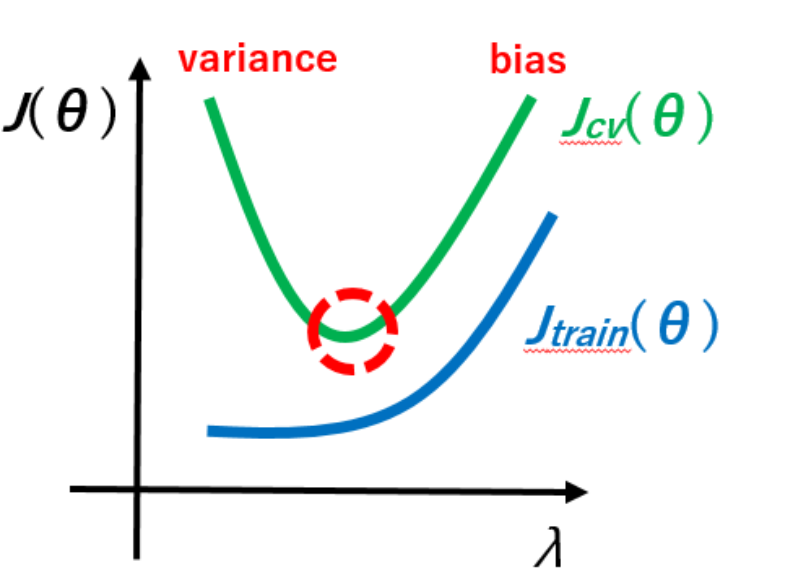
## Cross Validation



Hình 7.1. Phương pháp k-fold Cross val được sử dụng cho các tập data có ít dữ liệu.

Thay vì chia tập train ra làm 2 tập train và val, thì bây giờ chúng ta chia tập train thành k tập bằng nhau. Sau đó lần chọn 1 trong các tập đó làm tập val, các tập còn lại sẽ là tập train đưa vào vòng lặp train để train. Cách này giúp tránh được vấn đề underfit khi tập dữ liệu nhỏ.

## Bias and variance

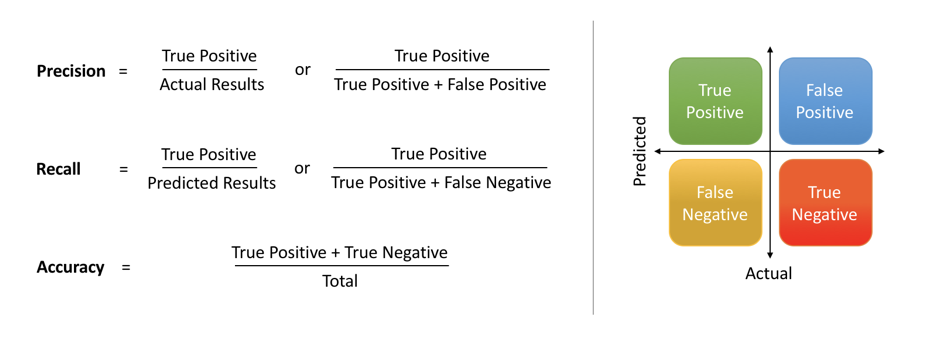


Hình 7.2. High variance và high bias

High bias (tạm dịch là trọng số lớn) là khoảng cách giữa đường J(ፀ) của tập train với đường J(ፀ) = 0 lớn, điều này nghĩa là độ lỗi trên tập train lớn, hiển nhiên sẽ dẫn đến độ lỗi trên tập test cao.

High variance (tạm dịch là phương sai lớn) là khoảng cách giữa đường J(ፀ) của tập train và đường J(ፀ) của tập val lớn, điều này phản ánh rằng độ lỗi trên tập train tuy thấp nhưng khi đưa vào thực tế thì model của mình chạy không đúng vì quá phù hợp với tập train (overfitting)

## Precision, Recall



Hình 7.3. Công thức tính precision và recall.

Tùy vào đặc trưng của model và các ứng dụng của nó mà ta chọn precision hay recall.

Precision được chọn cho các ứng dụng đề cao độ chính xác. Đề tài này cũng là một trong số đó. Trong khi đó, recall được chọn cho các ứng dụng không đề cao độ chính xác, tập trung vào tốc độ chạy hơn.