**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](#_Toc61285550)

[MỤC LỤC i](#_Toc61285551)

[DANH MỤC HÌNH iii](#_Toc61285552)

[Chương 1 MỤC TIÊU CỦA PROJECT 1](#_Toc61285553)

[Chương 2 CÁC THỬ THÁCH CỦA PROJECT 2](#_Toc61285554)

[2.1. Mô hình để train nặng (VGG16) 2](#_Toc61285555)

[2.2. Nhiễu do background 2](#_Toc61285556)

[2.3. Ánh sáng 2](#_Toc61285557)

[2.4. Kích thước bàn tay 2](#_Toc61285558)

[Chương 3 NỘI DUNG THỰC HIỆN 3](#_Toc61285559)

[3.1. Lưu đồ nhận diện bàn tay (detection) 3](#_Toc61285560)

[3.2. Chọn model để thực hiện nhận diện ký tự từ ảnh 3](#_Toc61285561)

[3.2.1. Transfer learning 4](#_Toc61285562)

[Chương 4 KẾT QUẢ 7](#_Toc61285563)

[4.1. Tập dữ liệu sử dụng 7](#_Toc61285564)

[4.2. Phương pháp huấn luyện mô hình 7](#_Toc61285565)

[4.3. Đánh giá mô hình sau khi train mô hình cuối cùng 8](#_Toc61285566)

[Chương 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 10](#_Toc61285567)

[5.1. Tiêu chí: 10](#_Toc61285568)

[5.2. Thử thách 10](#_Toc61285569)

[5.3. Hạn chế 10](#_Toc61285570)

[5.4. Hướng phát triển 10](#_Toc61285571)

# DANH MỤC HÌNH

[Hình 3.1. Bộ dữ liệu. 3](#_Toc61285541)

[Hình 3.2. Phân loại transfer learning. 4](#_Toc61285542)

[Hình 3.2. Bảng xếp hạng top các mạng CNN. 5](#_Toc61285543)

[Hình 3.4. Cấu trúc các lớp CNN sẽ sử dụng. 5](#_Toc61285544)

[Hình 4.1. Bộ dữ liệu. 7](#_Toc61285545)

[Hình 4.2.Đường cong học tập khi huấn luyện mô hình theo kiểu chia dữ liệu train:val:test = 64:16:20. 7](#_Toc61285546)

[Hình 4.3. K-Fold Cross Validation với K=5. 8](#_Toc61285547)

[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. 8](#_Toc61285548)

[Hình 4.5. Bảng Confusion Matrix cho mô hình cuối cùng. 9](#_Toc61285549)

# 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

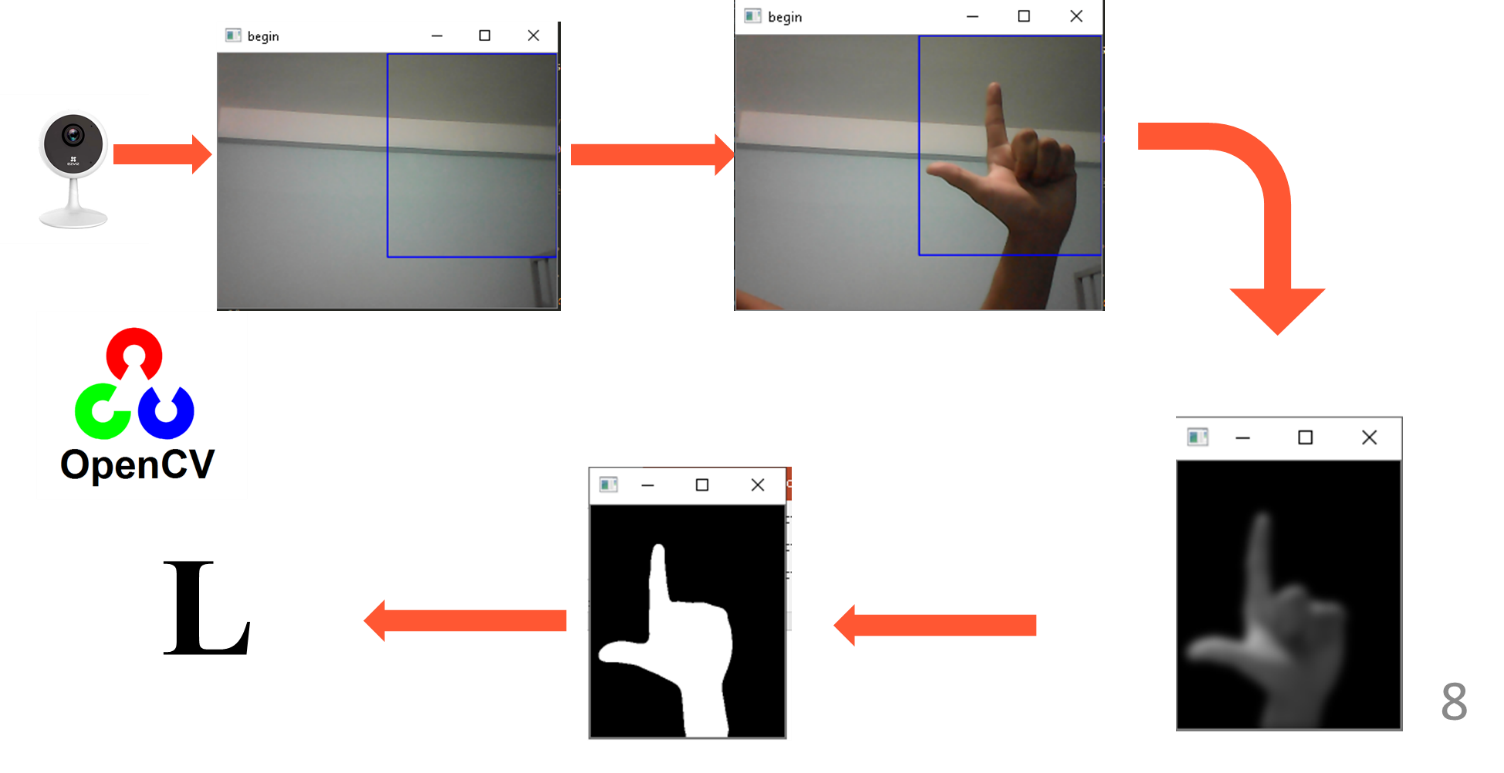
Giống như background thì ánh sáng cũng là yếu tố quan trọng ảnh hưởng lớn để kết quả đánh giá của model. Việc thiếu hay thừa ánh sáng sẽ dẫn đến độ tương phản giữa bàn tay và background không rõ ràng, việc này dẫn đến nhận dạng sai ký tự.

## 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

## Lưu đồ nhận diện bàn tay (detection)



Hình 3.1. Bộ dữ liệu.

Các bước thực hiện

* Chọn background cố định trong khung chữ nhật 🡪 để giúp giải quyết các vấn đề về background nhiễu
* Sau khi chọn khung chữ nhật thì đưa tay vào, ảnh sẽ được làm mịn, đưa về ảnh xám
* Sau đó, từ ảnh xám này chuyển sang ảnh trắng đen và cho chạy mô hình để phân loại
* Trong quá trình nhận diện, nhóm gặp vấn đề về ánh sáng và đưa ra cách giải quyết là tăng giảm độ sáng và độ tương phản bằng phím bấm để điều chỉnh cho thích hợp

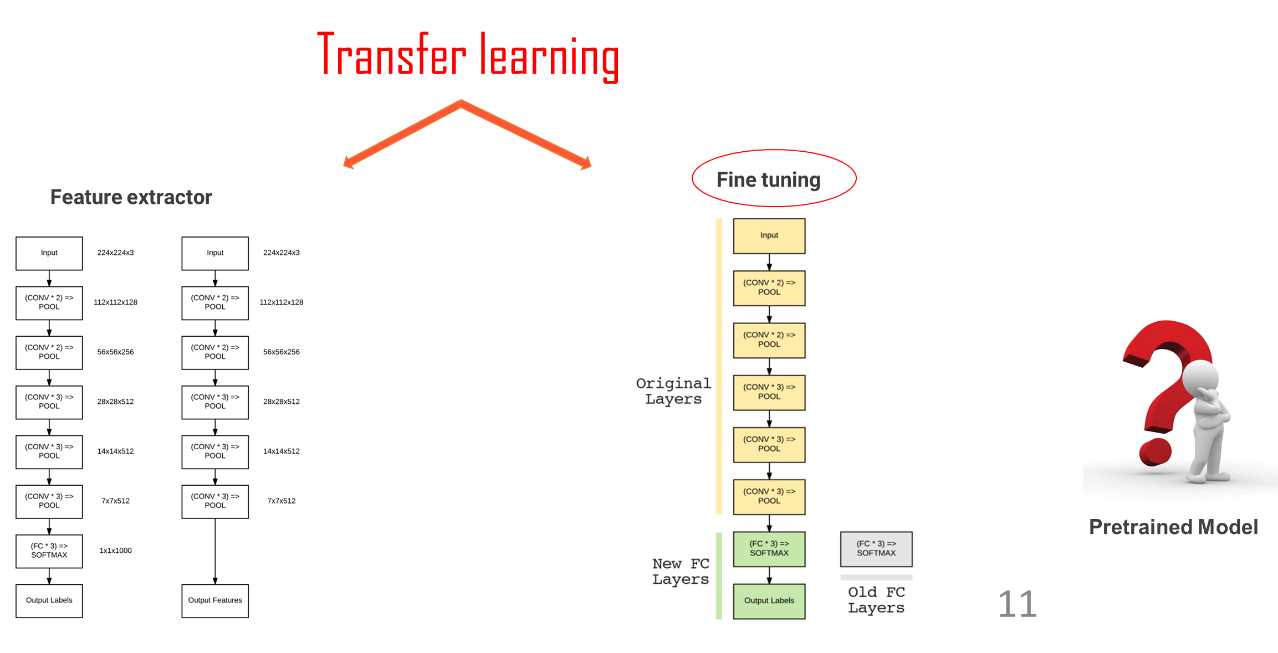
## Chọn model để thực hiện nhận diện ký tự từ ảnh

* Nhóm quyết định dùng “Transfer learning” và mạng CNN để thực hiện train model
* Transfer learning là một kỹ thuật giúp chúng ta kế thừa các Pretrained Model đã có sẵn, sau đó học lên để có được model tốt hơn dựa trên cái đã có sẵn. Nhóm lựa chọn kỹ thuật này vì lượng dữ liệu mà chúng tôi có theo như báo cáo giữa kỳ là nhỏ nên sẽ sử dụng kỹ thuật này để khắc phục

### Transfer learning

Có 2 loại transfer learning:

* Feature extractor: Sau khi lấy ra các đặc điểm của ảnh bằng việc sử dụng ConvNet của pre-trained model, thì ta sẽ dùng linear classifier (linear SVM, softmax classifier,..) để phân loại ảnh. Hiểu đơn giản thì các đặc điểm ảnh (tai, mũi, tóc,…) giờ như input của bài toán linear regression hay logistic regression.
* Fine tuning: Sau khi lấy ra các đặc điểm của ảnh bằng việc sử dụng ConvNet của pre-trained model, thì ta sẽ coi đây là input của 1 CNN mới bằng cách thêm các ConvNet và Fully Connected layer. Lý do là ConvNet của VGGFace 2 model có thể lấy ra được các thuộc tính của mặt người nói chung nhưng người Việt Nam có nhưng đặc tính khác nên cần thêm 1 số ConvNet mới để học thêm các thuộc tính của người Việt Nam.

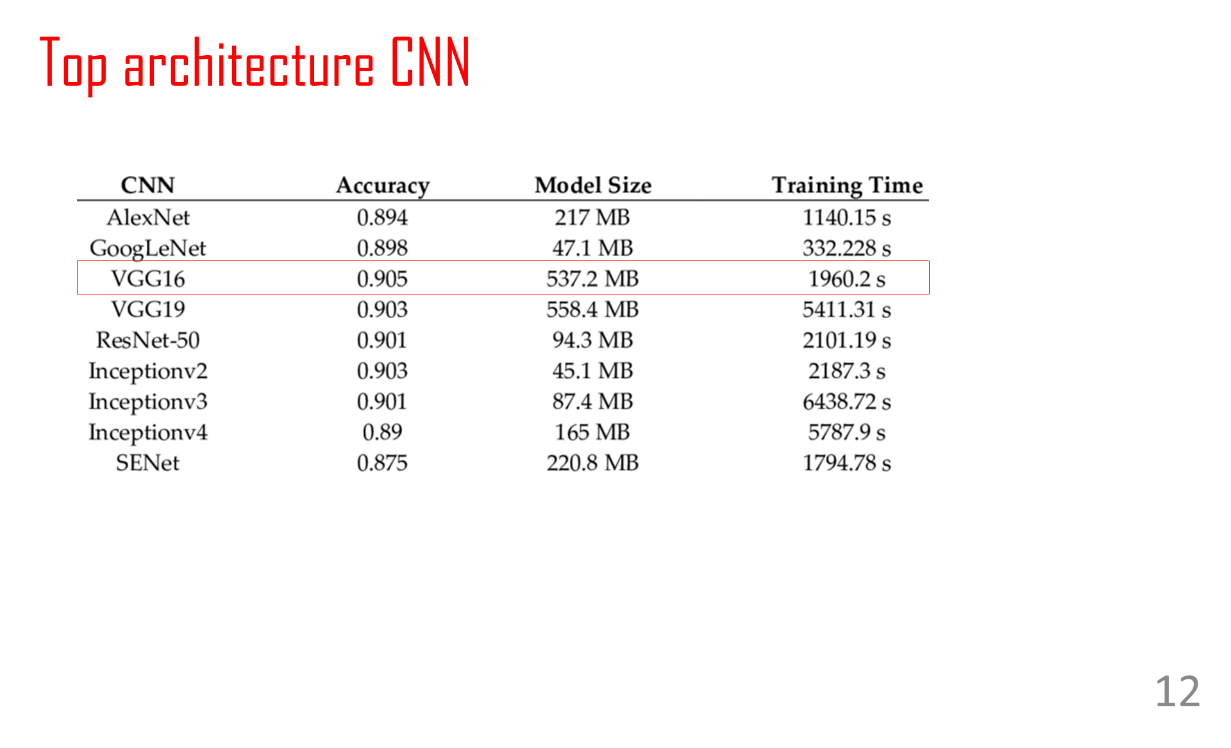


Hình 3.2. Phân loại transfer learning.

🡪 Dựa theo đặc trưng của đề tài, nhóm quyết định chọn fine tuning. Vì lý do là ta cần mô hình học thêm các đặc tính của tập dữ liệu.

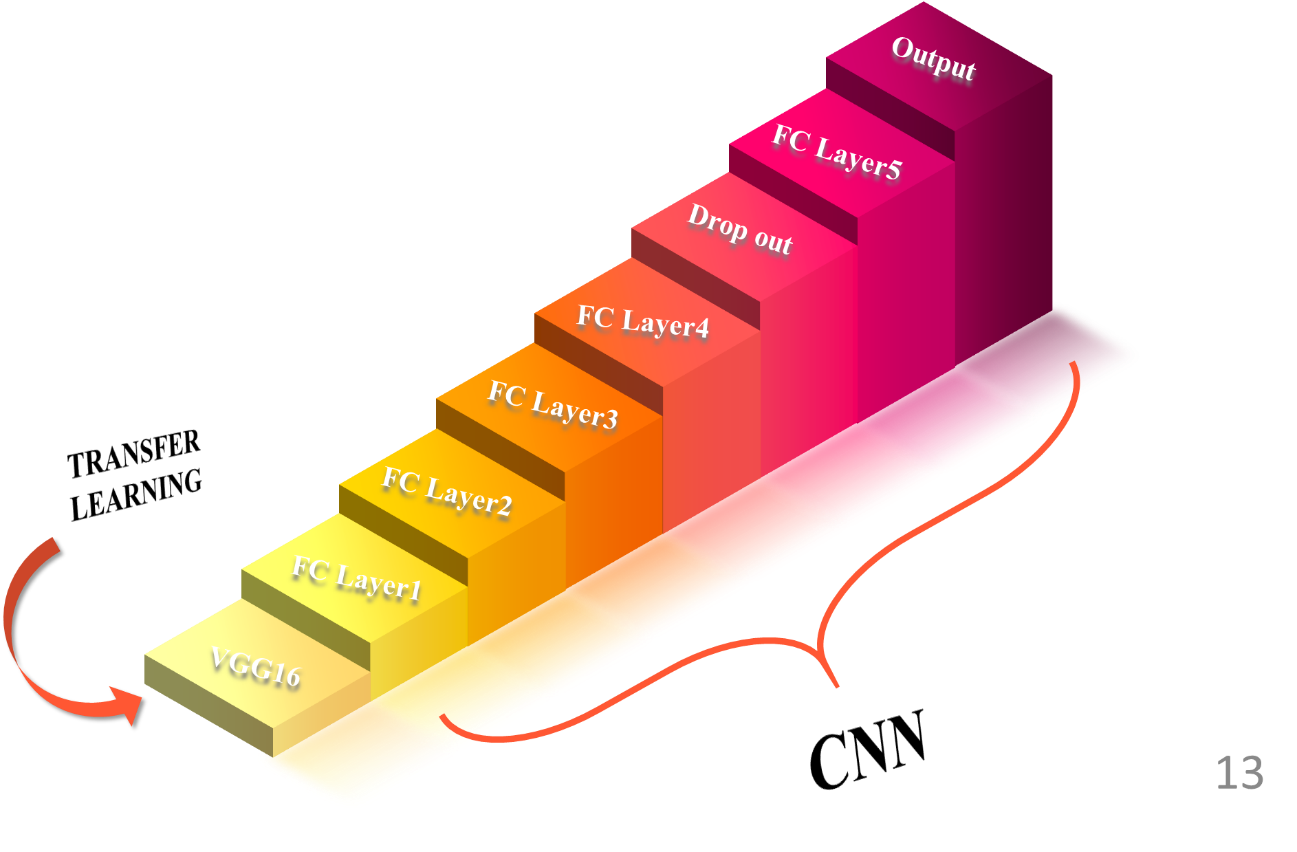
🡪 Cần chọn Pretrained Model nào?

Dưới đây là bảng xếp hạng các model CNN thuộc top



Hình 3.2. Bảng xếp hạng top các mạng CNN.

🡪 Nhóm quyết định chọn VGG16 vì lý do nhóm cho rằng thư viện keras hỗ trợ VGG16 tốt và đáp ứng được nhu cầu khi thực hiện



Hình 3.4. Cấu trúc các lớp CNN sẽ sử dụng.

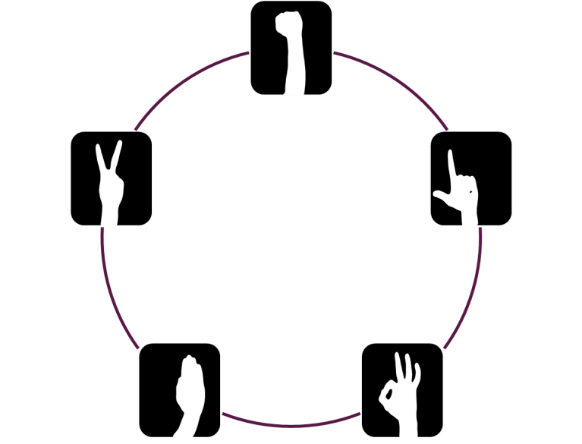
Hình trên mô tả cấu trúc CNN sẽ sử dụng:

* VGG16 để tiền xử lý, trích xuất các đặc trưng cơ bản
* Các lớp CNN phía sau sẽ tập trung trích xuất các đặc trưng của bàn tay
* Lớp dropout giúp tránh overfitting

# 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.



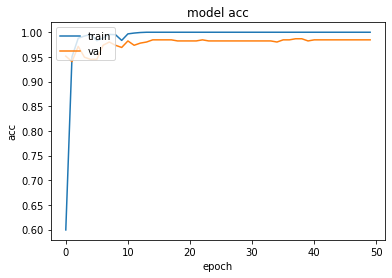
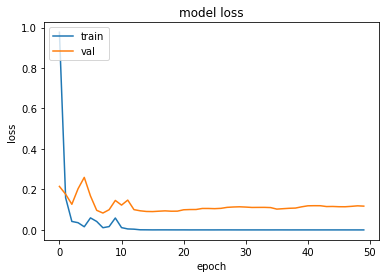
Hình 4.1. Bộ dữ liệu.

## 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.*

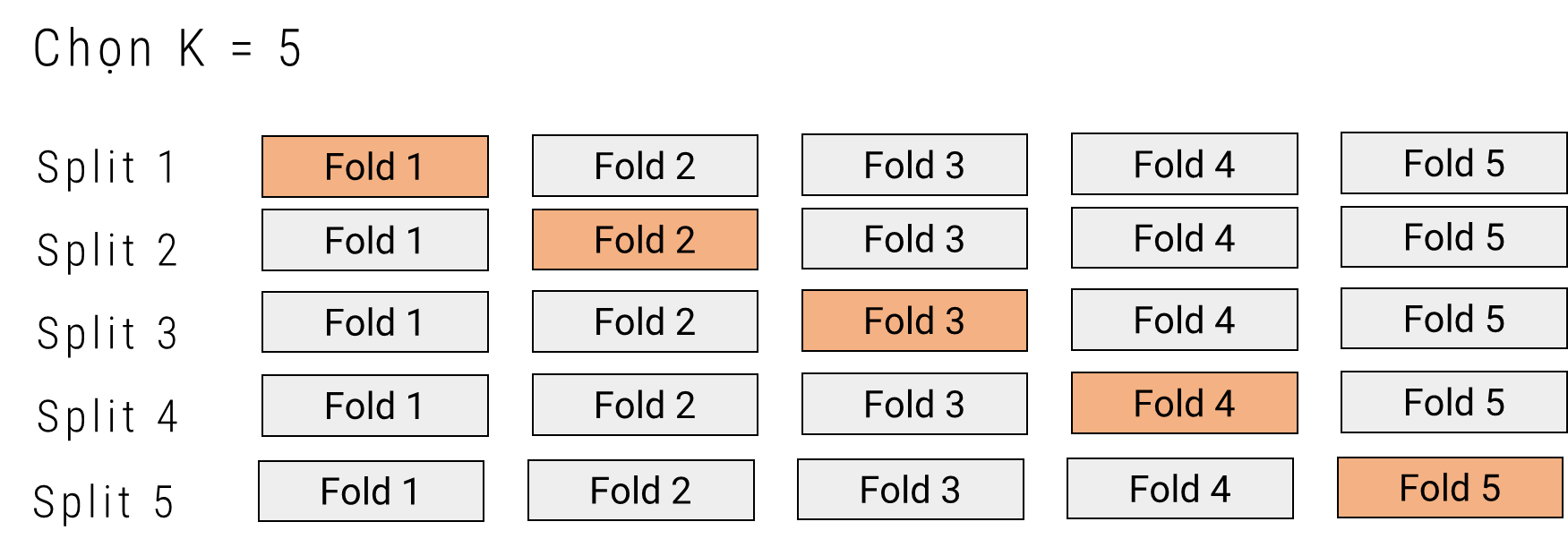
Kết quả sau khi huấn luyện được thể hiện như hình sau:



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

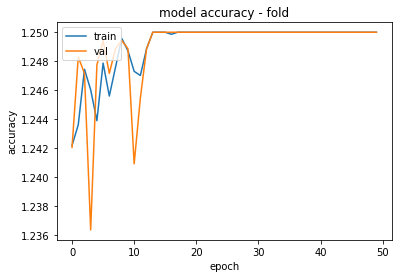
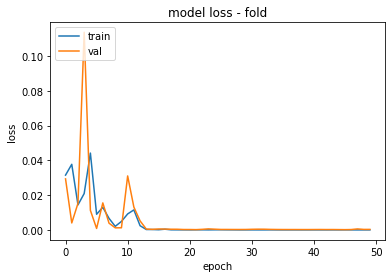
Nhận xét: nhìn vào đường loss line (hình phía bên trái), ta nhận thấy vẫn còn khoảng cách lớn giữa hai đường loss line và khoảng cách này có xu hướng tăng dần. Lúc này mô hình đang rơi vào trường hợp “tập train không thể hiện hết toàn bộ tập dữ liệu” (Unpresentative train dataset).

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:



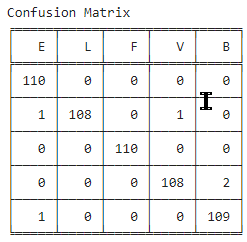
Hình 4.4. Đường cong học tập khi huấn luyện mô hình theo kiểu chia train:test = 80:20, và áp dụng K-Fold Cross Validation để đánh giá chéo.

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

## Tiêu chí:

* Độ chính xác cao - Thỏa mãn
* Mô hình không quá nặng - Không thỏa mãn
* Không bị nhiễu bởi background - Thỏa mãn
* Dễ tiếp cận, tìm hiểu - Thỏa mãn

## Thử thách

* Nhiễu
* Điều kiện ánh sáng
* Kích thước tay thay đổi

## Hạn chế

* Vị trí tay nhận dạng ở một vị trí cố định
* Khi thêm dữ liệu mới, cần phải train lại từ đầu
* Background là cảnh tĩnh

## Hướng phát triển

* Nhận diện bàn tay theo chuyển động
* Thay đổi mạng VGG16