**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

-------------------

**BÁO CÁO MACHINE LEARNING**

**ĐỀ TÀI: NHẬN DẠNG KÝ TỰ TRONG ẢNH**

**Giảng viên:** Thầy Vũ Quang Huy

**Sinh viên:** Đoàn Huỳnh Thiện 15110129

Nguyễn Di Thông 15110383

Ngô Đức Tín 15110140

**TP. HỒ CHÍ MINH – 9/2018**

**Mục lục**

[**CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI** 1](#_Toc531636854)

[**1. Đặt vấn đề** 1](#_Toc531636855)

[**2. Lý do chọn đề tài** 1](#_Toc531636856)

[**3. Mục tiêu** 1](#_Toc531636857)

[**Chương 2: LÝ THUYẾT – CONVOLUTIONAL NEURAL NETWORK** 2](#_Toc531636858)

[**1. Định nghĩa** 2](#_Toc531636859)

[**2. Convolution** 2](#_Toc531636860)

[**3. Stride and Padding** 4](#_Toc531636861)

[**4. Pooling** 7](#_Toc531636862)

[**5. Fully Connected** 8](#_Toc531636863)

[**6. CNN Structure** 8](#_Toc531636864)

[**CHƯƠNG 3: TRIỂN KHAI** 9](#_Toc531636865)

[**1. Flowchart** 9](#_Toc531636866)

[**2. Block Diagram** 10](#_Toc531636867)

[**3. Thu thập dữ liệu** 10](#_Toc531636868)

[**4. Preprocessing** 11](#_Toc531636869)

[**4.1 Shape** 11](#_Toc531636870)

[**4.2 Color** 11](#_Toc531636871)

[**4.3 Dataset split** 11](#_Toc531636872)

[**5. Lựa chọn model** 11](#_Toc531636873)

[**5.1 Mô hình 1** 11](#_Toc531636874)

[**5.2 Mô hình 2** 12](#_Toc531636875)

[**5.3 Mô hình 3** 12](#_Toc531636876)

[**6. Demo** 12](#_Toc531636877)

[**CHƯƠNG 4: KẾT LUẬN** 14](#_Toc531636878)

[**TÀI LIỆU THAM KHẢO** 15](#_Toc531636879)

# **CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI**

## **1. Đặt vấn đề**

Hiện này thì kỹ thuật nhận diện ký tự trong ảnh đã trở nên phổ biến và cần thiết trong cuộc sống. Kỹ thuật này giúp ích rất nhiều trong thời buổi cách mạng công nghiệp 4.0. Chúng ta có thể dùng nó để chuyển đổi bức ảnh văn bản thành đoạn văn bản có thể chỉnh sửa, nhận dạng biển số, nhận dạng các biển báo giao thông cho hệ thống xe tự lái… Và để triển khai nhận diện ký tự thì thuật toán được sử dụng rất phổ biến là Convolutional Neural Network.

## **2. Lý do chọn đề tài**

Hiện nay thì kỹ thuật nhận diện ký tự trong ảnh đang xuất hiện xung quanh chúng ta cùng với mong muốn hiểu được cách thuật toán Convolutional Neural Network hoạt động để có thể nhận diện được ký tự trong ảnh nên nhóm đã chọn đề tài này để thực hiện.

## **3. Mục tiêu**

Ứng dụng sẽ có thể nhận diện ký tự từ một bức ảnh đơn giản như sau:



Với bức ảnh trên, ứng dụng sẽ cho kết qua đầu ra là chữ “R”.

# **Chương 2: LÝ THUYẾT – CONVOLUTIONAL NEURAL NETWORK**

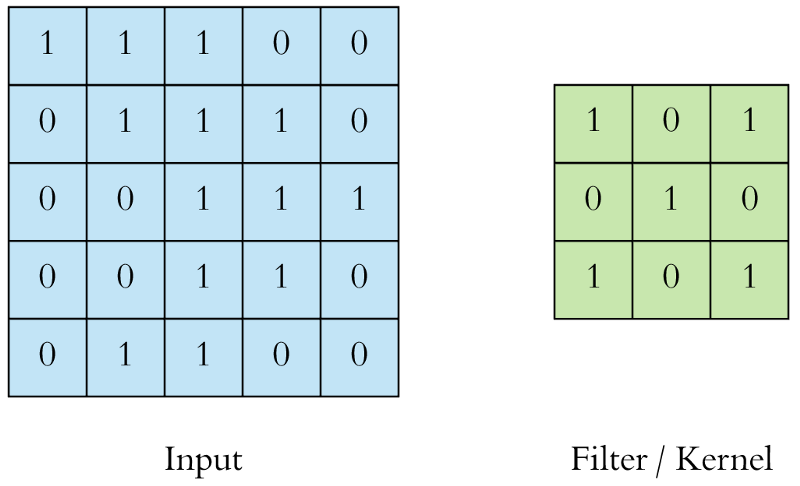
## **1. Định nghĩa**

Convolutional Neural Network (CNN – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay như hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động. CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh. Để tìm hiểu tại sao thuật toán này được sử dụng rộng rãi cho việc nhận dạng (detection).

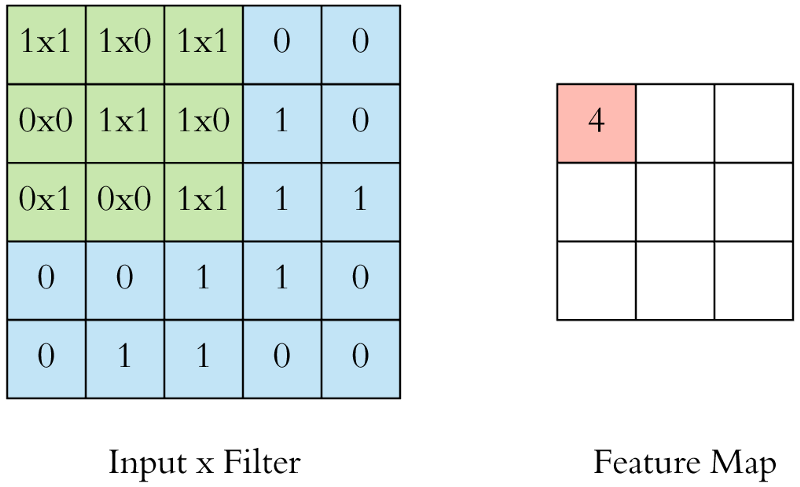
Convolutional Neural Network, đây là một deep neural network architecture. Hiểu đơn giản, nó cũng chính là một dạng Artificial Neural Network, một Multiplayer Perceptron nhưng mang thêm 1 vài cải tiến, đó là Convolution và Pooling.

## **2. Convolution**

Convolution gồm 2 khái niệm khác là Convolution Filter và Convolutional Layer. Trong mạng neural network thông thường, từ input, ta cho qua các hidden layer rồi ra được output. Với CNN, Convolutional Layer cũng chính là hidden layer, khác ở chỗ, Convolutional Layer là một tập các feature map và mỗi feature map này là một bản scan của input ban đầu, nhưng được trích xuất ra các feature/đặc tính cụ thể. Scan như thế nào thì lại dựa vào Convolution Filter hay kernel. Đây là một ma trận sẽ quét qua ma trận dữ liệu đầu vào, từ trái qua phải, trên xuống dưới, và nhân tương ứng từng giá trị của ma trận đầu vào mà ma trận kernel rồi cộng tổng lại, đưa qua activation funciton (sigmoid, relu, elu, ... ), kết quả sẽ là một con số cụ thể, tập hợp các con số này lại là 1 ma trận nữa, chính là feature map. Hãy nhìn vào ví dụ sau cho dễ hiểu. Có một ma trận đầu vào và một kernel 5x5.



Quét kernel qua từng phần tử của input. Và tính toán như trên: nhân tương ứng, rồi cộng tổng kết quả, đưa qua activation function, ta thu được một giá trị tại feature map.



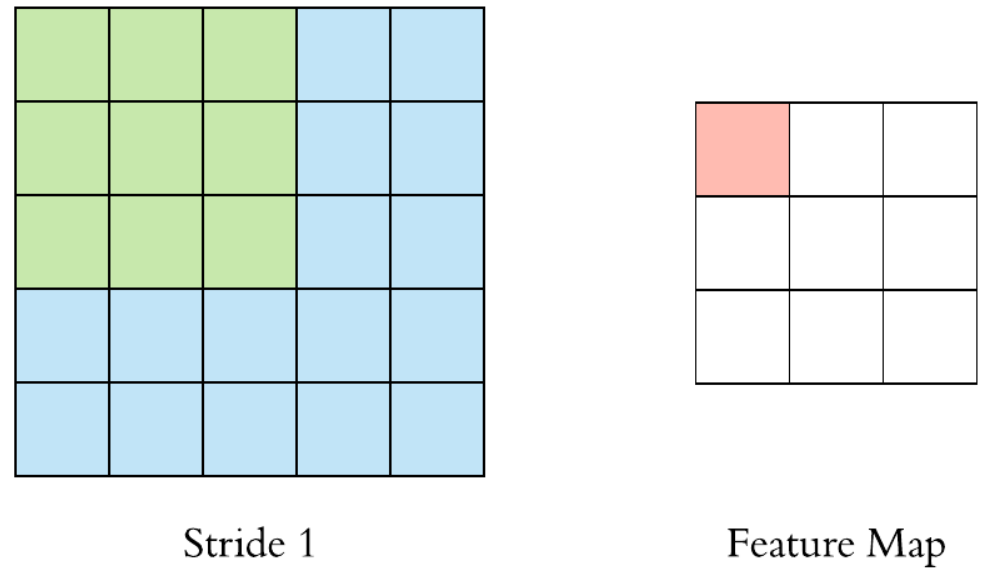
Thực hiện lần lượt trái qua phải, trên xuống dưới cho hết.

Thông thường các giá trị size kernel là 3x3 hoặc là 5x5.

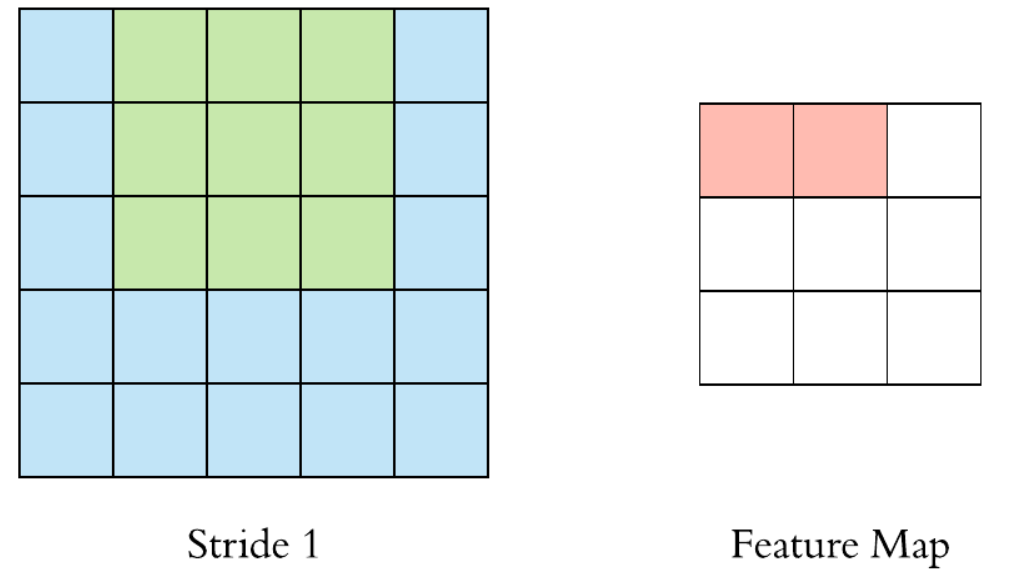
## **3. Stride and Padding**

Stride là khoảng cách giữa 2 kernel khi quét. Với stride = 1, kernel sẽ quét 2 ô ngay cạnh nhau, nhưng với stride = 2, kernel sẽ quét ô số 1 và ô số 3. Bỏ qua ô ở giữa. Điều này nhằm tránh việc lặp lại giá trị ở các ô bị quét stride = 1.

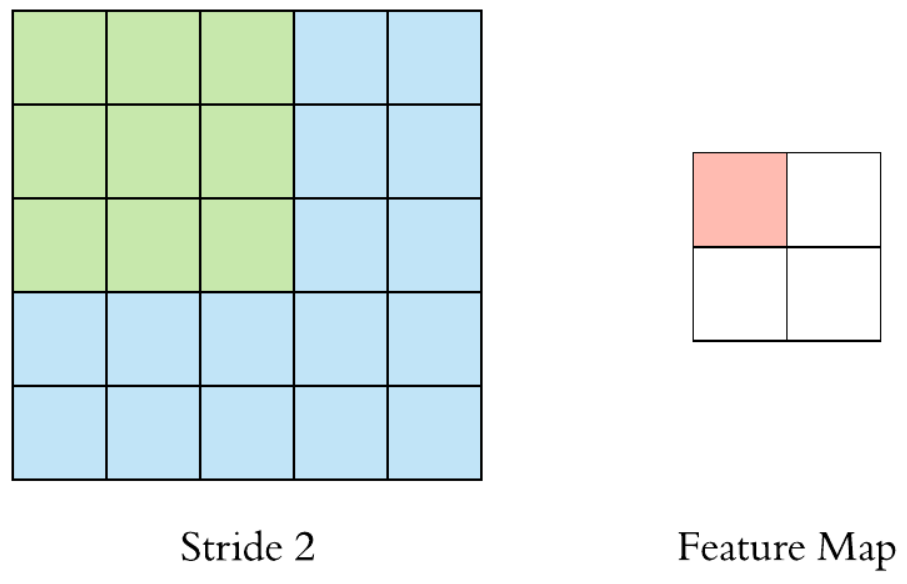
Ví dụ stride = 1.



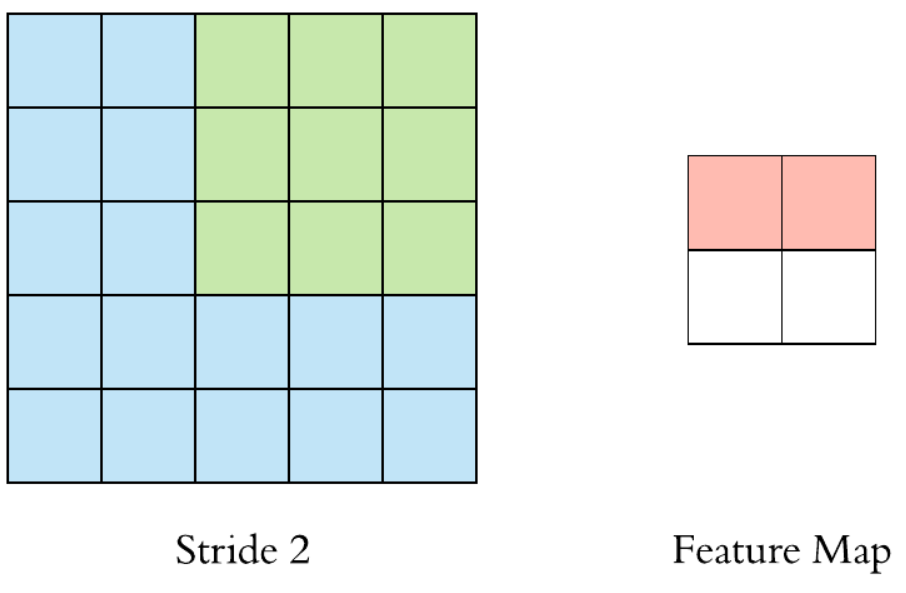
Stride = 1 thì sẽ quét qua ô thứ 2.



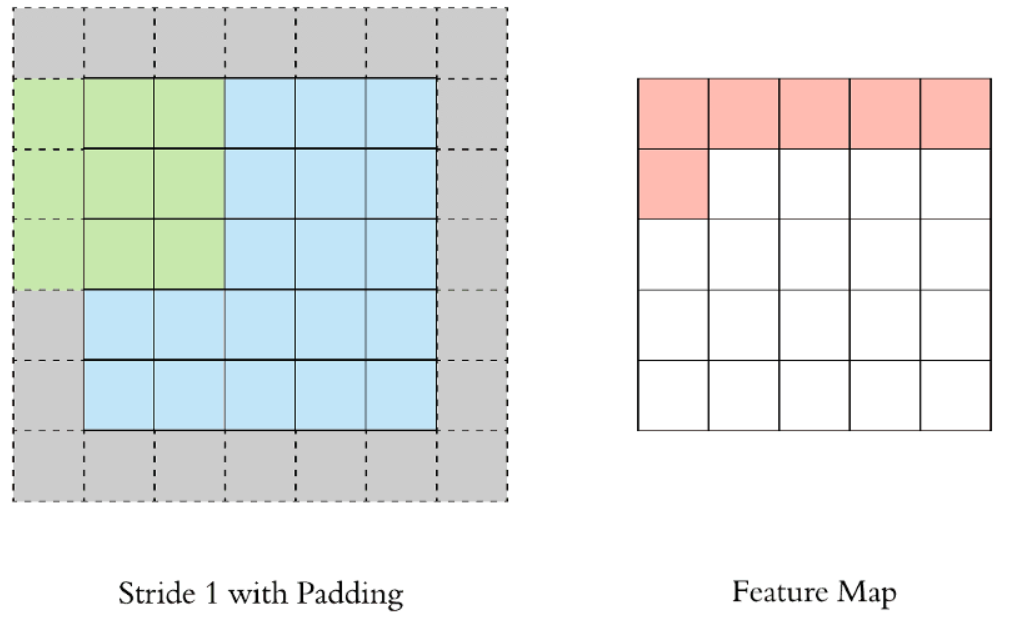
Ví dụ stride = 2.



Stride = 2 thì sẽ bỏ qua ô 2, quét từ ô 3 trở đi.



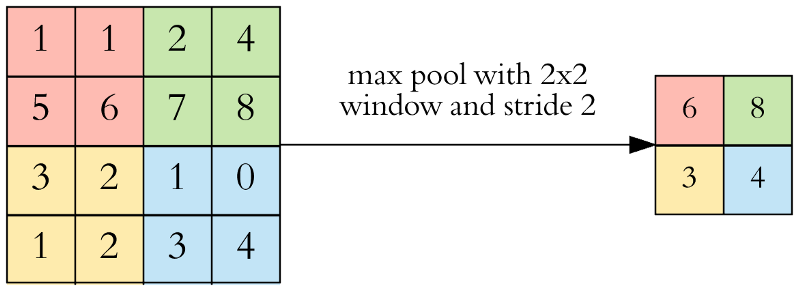
Chúng ta chọn stride và size của kernel càng lớn thì size của feature map càng nhỏ, một phần lý do đó là bởi kernel phải nằm hoàn toàn trong input. Có một cách để giữ nguyên kích cỡ của feature map so với ban đầu. Đấy là Padding. Khi ta điều chỉnh padding = 1, tức là ta đã thêm 1 ô bọc xung quanh các cạnh của input, muốn phần bọc này càng dày thì ta cần phải tăng padding lên. Hãy nhìn vào ví dụ sau, ta xét padding = 1:



Phần màu xám chính là phần bọc thêm vào input.

## **4. Pooling**

Mục đích của pooling rất đơn giản, nó làm giảm số hyperparameter mà ta cần phải tính toán, từ đó giảm thời gian tính toán, tránh overfitting. Loại pooling ta thường gặp nhất là max pooling, lấy giá trị lớn nhất trong một pooling window. Pooling hoạt động gần giống với convolution, nó cũng có 1 cửa sổ trượt gọi là pooling window, cửa sổ này trượt qua từng giá trị của ma trận dữ liệu đầu vào (thường là các feature map trong convolutional layer), chọn ra một giá trị từ các gía trị nằm trong cửa sổ trượt (với max pooling ta sẽ lấy giá trị lớn nhất). Hãy cùng nhìn vào ví dụ sau, tôi chọn pooling window có kích thước là 2 \* 2, stride = 2 để đảm bảo không trùng nhau, và áp dụng max pooling:

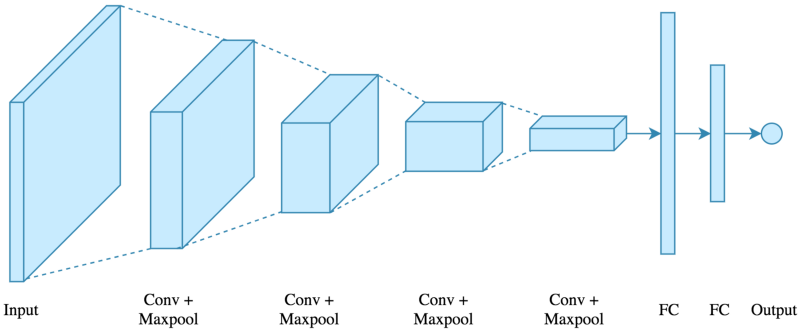


## **5. Fully Connected**

Tại lớp mạng này, mỗi một nơ-ron của layer này sẽ liên kết tới mọi nơ-ron của lớp khác. Để đưa ảnh từ các layer trước vào mạng này, buộc phải dàn phẳng bức ảnh ra thành 1 vector thay vì là mảng nhiều chiều như trước. Tại layer cuối cùng sẽ sử dụng 1 hàm kinh điển trong học máy mà bất kì ai cũng từng sử dụng đó là softmax để phân loại đối tượng dựa vào vector đặc trưng đã được tính toán của các lớp trước đó.

## **6. CNN Structure**

Hình vẽ dưới đây đã biểu diễn rất rõ ràng kiến trúc của 1 mạng CNN



Ta có ảnh input đầu vào. Qua hàng loạt các Convolutional Layer cùng Max Pool Layer (thường pooling sẽ theo ngay sau 1 convolutional layer), cuối cùng là 2 fully connected.

# **CHƯƠNG 3: TRIỂN KHAI**

## **1. Flowchart**

Input

Preprocessing

CNN

Classifier

Start

Stop

Yes

No

Output

Valid?

## **2. Block Diagram**

1

Training Data

Pre Processing

Feature Extraction

Predict Model

CNN

Input Image

Classifier

Pre Processing

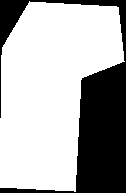
Feature Extraction

CNN

## **3. Thu thập dữ liệu**

Bộ dữ liệu có 7704 tấm hình bao gồm các ký tự từ a-z, A-Z và từ 0-9. Mỗi dữ liệu sẽ có kèm thêm mask của riêng nó.

Ví dụ:



## **4. Preprocessing**

### **4.1 Shape**

Tất cả các hình ảnh có shape và size khác nhau. Vì vậy sẽ phải chuyển chúng thành 1 shape chung để thuận lợi trong việc training. Các bức ảnh sẽ được chuyển về kích thước 64x64.

### **4.2 Color**

Đầu tiên, các bức ảnh đầu vào sẽ là ở dạng RGB. Sau đó, các bức ảnh sẽ được chuyển về grayscale, điều này không làm ảnh hưởng đến độ chính xác và cũng giúp cho việc training nhanh chóng hơn.

### **4.3 Dataset split**

Dataset sẽ được chia thành 3 nhóm như sau:

* 60% training set. (4623 ảnh)
* 20% validation set. (1541 ảnh)
* 20% test set. (1541 ảnh)

## **5. Lựa chọn model**

Convolutional Neural Network được chọn vì thuật toán này hoạt động tốt trong việc phân tích hình ảnh.

Các model đã được thử nghiệm:

### **5.1 Mô hình 1**

|  |
| --- |
| Conv1 : 3x3 kernels | 1x1 stride | 32 maps | relu  Conv2 : 3x3 kernels | 2x2 stride | 64 maps | relu  Conv3 : 3x3 kernels | 2x2 stride | 128 maps | relu  Conv4 : 5x5 kernels | 2x2 stride | 128 maps | relu  Max Pooling: 2x2 kernels | 2x2 stride  Fully Connected: 512 units | relu  Fully Connected: 62 units | relu |

Accurancy: 51. 2%

Top 3 accurancy: 74.56%

### **5.2 Mô hình 2**

|  |
| --- |
| Conv1 : 3x3 kernels | 1x1 stride | 32 maps | relu  Conv2 : 3x3 kernels | 1x1 stride | 64 maps | relu  Conv3 : 5x5 kernels | 2x2 stride | 128 maps | relu  Max Pooling: 2x2 kernels | 2x2 stride  Fully Connected: 512 units | relu  Fully Connected: 62 units | relu |

Accurancy: 68.72%

Top 3 accurancy: 86.70%

### **5.3 Mô hình 3**

|  |
| --- |
| Conv1 : 3x3 kernels | 1x1 stride | 32 maps | relu  Conv2 : 5x5 kernels | 2x2 stride | 64 maps | relu  Conv3 : 5x5 kernels | 2x2 stride | 128 maps | relu  Max Pooling: 2x2 kernels | 2x2 stride  Fully Connected: 512 units | relu  Fully Connected: 62 units | relu |

Accurancy: 72.03%

Top 3 accurancy: 87.87%

Qua 3 mô hình thì mô hình 3 cho kết quả khả quan nhất. Vì vậy, nhóm sẽ áp dụng mô hình 3 để tiến hành.

## **6. Demo**





Hai ảnh trên là kết quả dự đoán từ chương trình dựa trên thuật toán Convolutional Neural Network.

# **CHƯƠNG 4: KẾT LUẬN**

Trong suốt quá trình tìm hiểu và thực hiện đề tài, nhóm đã đạt được những điều sau:

* Hiểu được Machine Learning và sâu hơn nữa là Deep Learning.
* Hiểu được cách hoạt động của thuật toán Convolutional Neural Network.
* Ứng dụng được những kiến thức về thuật toán Convolutional Neural Network để giải quyết vấn đề về nhận diện ký tự trong ảnh tuy nhiên độ chính xác chưa được cao, đạt mức hơn 70%.

# **TÀI LIỆU THAM KHẢO**

1. Ứng dụng Convolutional Neural Network trong bài toán phân loại ảnh, <https://viblo.asia/p/ung-dung-convolutional-neural-network-trong-bai-toan-phan-loai-anh-4dbZNg8ylYM>

2. Chia sẻ về Mạng nơ-ron tích chập (Convolutional Neural Networks or ConvNEts), <https://ereka.vn/post/chia-se-ve-mang-noron-tich-chap-convolutional-neural-networks-or-convnets-52790224348847566>

3. Kaggle First Steps With Julia (Chars74k): First Place using Convolutional Neural Networks, <http://ankivil.com/kaggle-first-steps-with-julia-chars74k-first-place-using-convolutional-neural-networks/>