

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

---



**ĐỒ ÁN CUỐI KÌ**  
**DỰ ĐOÁN CHỮ CÁI KÝ HIỆU THỦ**  
**NGỮ BẰNG THUẬT TOÁN MÁY HỌC**

**Môn học:** Máy học

**Giáo viên hướng dẫn:**

PGS.TS Lê Đình Duy

THS. Phạm Nguyễn Trường An

**Sinh viên:**

Đồng Quốc Tuấn - MSSV: 18520185

**Thành phố Hồ Chí Minh, tháng 8 năm 2020**

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



**ĐỒ ÁN CUỐI KÌ**  
**DỰ ĐOÁN CHỮ CÁI KÝ HIỆU THỦ**  
**NGŨ BẰNG THUẬT TOÁN MÁY HỌC**

**Môn học:** Máy học

**Giáo viên hướng dẫn:**

PGS.TS Lê Đình Duy

THS. Phạm Nguyễn Trường An

**Sinh viên:**

Đồng Quốc Tuấn - MSSV: 18520185

**Thành phố Hồ Chí Minh, tháng 8 năm 2020**

# MỤC LỤC


MỤC LỤC.....	3
I. GIỚI THIỆU .....	4
1. Giới thiệu bài toán.....	4
2. Ứng dụng và mục tiêu.....	5
II. DỮ LIỆU .....	6
III. TIỀN XỬ LÝ DỮ LIỆU .....	6
1. Chuẩn hóa dữ liệu: .....	6
2. Rút trích đặc trưng: .....	8
IV. CÁC THUẬT TOÁN MÁY HỌC .....	9
1. K-Nearest-Neighbors (k-NN) .....	9
2. Logistic Regression với kỹ thuật one – vs – rest (OvR) .....	10
3. Softmax Regression .....	10
V. TÍNH CHÍNH THAM SỐ.....	11
1. K-Nearest-Neighbors .....	11
.....	12
2. Logistic Regression với kỹ thuật one-vs-rest (OvR) .....	12
3. Softmax Regression .....	13
VI. KẾT QUẢ VÀ ĐÁNH GIÁ.....	14
VII. KẾT LUẬN .....	15
TÀI LIỆU THAM KHẢO .....	17

# I. GIỚI THIỆU

## 1. Giới thiệu bài toán

Ngôn ngữ ký hiệu hay ngôn ngữ dấu hiệu, thủ ngữ là ngôn ngữ dùng những biểu hiện của bàn tay thay cho âm thanh của tiếng nói. Ngôn ngữ ký hiệu của người khiếm thính được tạo ra nhằm giúp họ có thể giao tiếp với nhau trong cộng đồng của mình và tiếp thu tri thức của xã hội. Ngôn ngữ ký hiệu còn là một thước đo để người bình thường có thể hiểu và giao tiếp với người khiếm thính. Mỗi ngôn ngữ đều có bảng chữ cái riêng, ngôn ngữ ký hiệu cũng thế. Ngôn ngữ ký hiệu đa dạng với từng nước và đôi khi khác nhau ở từng vùng miền ở một nước, nhưng đa số là tương tự nhau <sup>[1]</sup>.

Việc giao tiếp giữa người bình thường và người khiếm thính luôn có những khó khăn nhất định. Đồ án sử dụng thuật toán máy học như K-Nearest-Neighbor, Logistic Regression nhận dạng hình ảnh ký hiệu bàn tay, nhằm dự đoán thuộc ký hiệu nào trong bảng chữ cái.

 A a	 B b	 C c	 D d	 Đ đ	 E e	 F f
 G g	 H h	 I i	 J j	 K k	 L l	 M m
 N n	 O o	 P p	 Q q	 R r	 S s	 T t
 U u	 V v	 W w	 X x	 Y y	 Z z	
 Horn ơ	 Breve ô	 Circumflex ô	<b>Vietnamese Sign Language</b> Combining Diacritic Marks			
 Hook ơ	 Acute ơ	 Grave ơ	 Tilde ơ	 Underdot ơ		

## 2. Ứng dụng và mục tiêu

Trên thế giới, đã có nhiều mô hình nghiên cứu phát triển các dịch vụ thông dịch ngôn ngữ ký hiệu và các sản phẩm công nghệ để hỗ trợ người khiếm thính như: [MotionSavvy tablet](#)<sup>[2]</sup>, [Tencent YouTu Lab platform](#)<sup>[3]</sup>,... Tuy nhiên trên thị trường vẫn chưa thấy những sản phẩm phục vụ cho người Việt.

Mục tiêu của đề án là đưa ra kết quả chữ cái được dự đoán sau khi đưa vào một hình ảnh ký hiệu bàn tay. Đề án với mong muốn tìm kiếm đặc trưng, phương pháp hợp lý để có thể dự đoán chính xác nhằm tạo bước tiến trong việc phát triển ứng dụng hỗ trợ giao tiếp với người khiếm thính.

## II. DỮ LIỆU

Ngôn ngữ ký hiệu có hai dạng đó là tĩnh và động (cần động tác). Đa số ngôn ngữ ký hiệu tĩnh khá tương đồng giữa các quốc gia, trong đó ngôn ngữ ký hiệu Mỹ (ASL)<sup>[4]</sup> được sử dụng rộng rãi ở nhiều nước trên thế giới. Ngôn ngữ ký hiệu Việt Nam (VSL)<sup>[5]</sup> bao gồm các ký hiệu có dấu nên bảng chữ cái ký hiệu có nhiều ký hiệu hơn ngôn ngữ ký hiệu Mỹ. Bộ dữ liệu tĩnh bao gồm 25 ký tự bao gồm 23 ký hiệu ngoại trừ cái ký tự F,J,W,Z từ ASL và 2 ký hiệu dấu "râu","mũ".

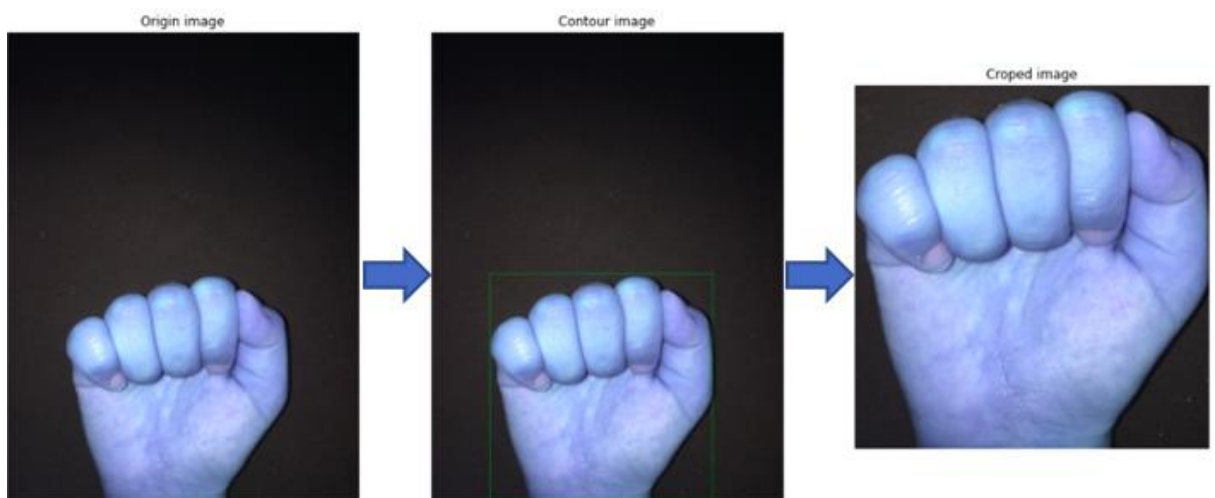
Bộ dữ liệu của đề án bao gồm 22 ký tự không bao gồm ký tự Đ,J,W,Z,"râu","mũ". Mỗi ký tự bao gồm 50 hình ảnh đều được chụp bằng điện thoại, dưới ánh đèn Flash và nền đen và 5 hình ảnh khác nhau về góc chụp, hướng chụp,... dùng để kiểm tra độ chính xác của model vừa train.

Kích thước của bộ dữ liệu bao gồm 1100 ảnh dùng để [train](#) và 110 ảnh dùng để [kiểm tra](#)<sup>[6]</sup>.

## III. TIỀN XỬ LÝ DỮ LIỆU

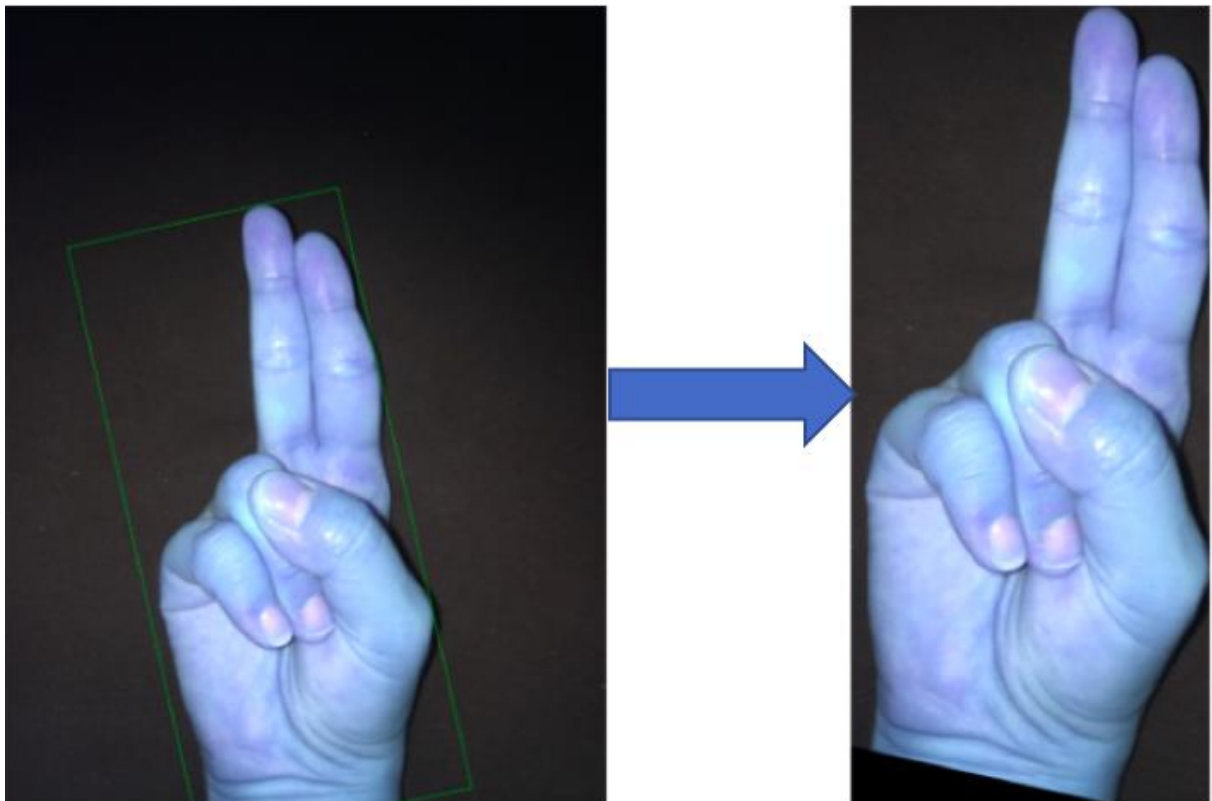
### 1. Chuẩn hóa dữ liệu:

Hình ảnh thô bao gồm những thông tin không cần thiết như phần rìa đen của background. Mục đích của quá trình xử lý là lấy ra thông tin cần thiết của bàn tay và loại bỏ phần không cần thiết.



Quá trình xử lý bao gồm 2 bước:

- Bước 1: Ảnh bằng cách đưa về ảnh gray -> lọc nhiễu bằng bộ lọc Gaussian -> áp dụng thresh hold để chuyển về ảnh binary (vì màu da và màu nền đối nhau). Tìm contour lớn nhất của ảnh binary và bounding box tương ứng (khung màu xanh của hình Countour image ở trên).
- Bước 2: Vì sẽ có một số bounding box có góc xoay, nên cần thực hiện một số phép biến đổi hình học để đưa về góc thẳng.



Hình sau khi thực hiện các phép biến đổi, thu được hình chứa thông tin cần thiết của ảnh gốc. Để dễ dàng cho việc tính toán sau này, quá trình xử lý này được thực hiện ở [file colab](#)<sup>[7]</sup> và hình ảnh của các ký tự sau khi xử lý được lưu vào thư mục [cropped](#)<sup>[8]</sup>.

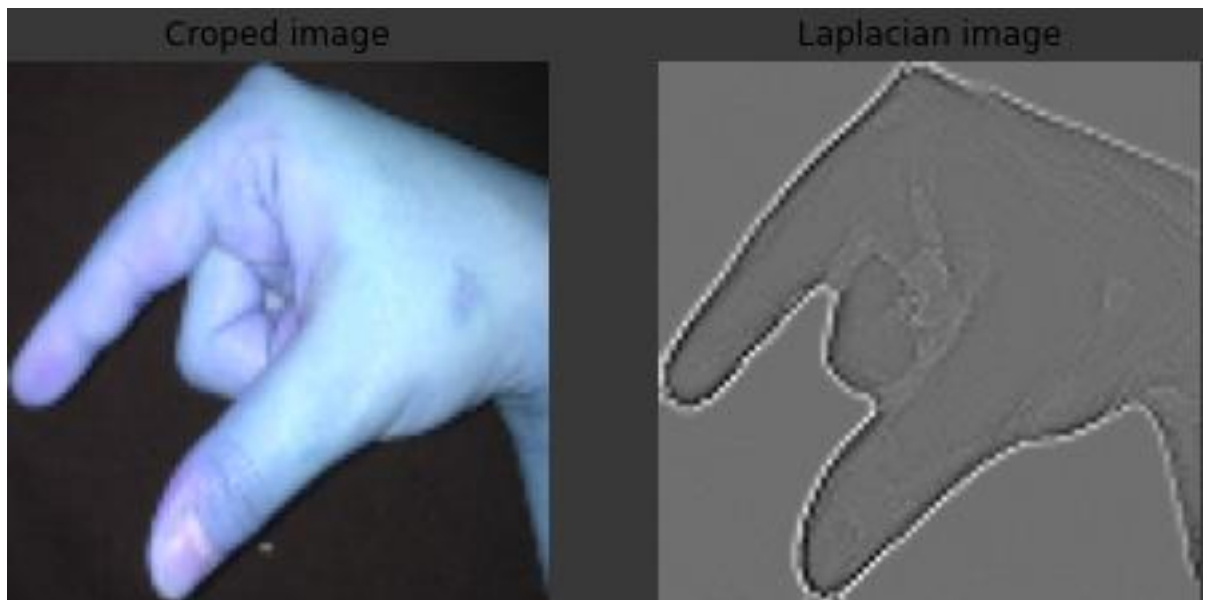
Sau khi cắt ảnh dựa vào bounding box thì mỗi ảnh sẽ có size khác nhau. Để dễ dàng cho việc rút trích đặc trưng và train model, sẽ tiến hành resize ảnh về (90x90).



## 2. Rút trích đặc trưng:

- Màu da mỗi người là khác nhau, có người tay to, tay nhỏ => không xét màu da, kích thước bàn tay là đặc trưng.
- Đa số mọi người đều thuận tay phải, nên chọn tay phải là một đặc trưng cho bài toán.
- Đa số màu da của người là màu sáng (trừ người da đen) so với nền màu tối, và hình dáng biểu diễn ngôn ngữ ký hiệu là tương tự => Xem xét chọn hình dáng, cạnh của bàn tay là đặc trưng.
- Để có thể lấy được đặc trưng của bàn tay, đầu tiên cần đưa về ảnh xám và qua bộ lọc để khử nhiễu, sau đó dùng toán tử Laplace<sup>[9]</sup> để có thể lấy được cạnh của bàn tay. Cuối cùng đưa hình ảnh Laplacian về vector có số chiều 8100. Vector đặc trưng này cũng chính là input của model xem xét bên dưới.





## IV. CÁC THUẬT TOÁN MÁY HỌC

Các thuật toán máy học với tham số mặc định được sử dụng từ thư viện `scikit-learn`<sup>[10]</sup>.

### 1. K-Nearest-Neighbors (k-NN)

k mặc định là 5.

- Có thể thấy f1-score khá thấp: 0.4
- Đa số các class có precision và recall chênh lệch khá cao trừ một số class (B,D,R,N,S,T,U)
- Class L có kết quả tệ nhất f1-score: 0

Accuracy on training set: 100.0%				
Accuracy on test set: 40.45454545454545%				
	precision	recall	f1-score	support
A	0.08	0.75	0.14	12
B	1.00	0.78	0.88	9
C	1.00	0.07	0.12	15
D	0.58	0.78	0.67	9
E	0.71	0.56	0.63	9
G	1.00	0.12	0.22	8
H	1.00	0.08	0.14	13
I	1.00	0.33	0.50	9
K	1.00	0.10	0.18	10
L	0.00	0.00	0.00	9
M	0.42	0.73	0.53	11
N	0.78	0.70	0.74	10
O	1.00	0.17	0.29	12
P	1.00	0.07	0.12	15
Q	1.00	0.14	0.25	7
R	0.86	0.86	0.86	7
S	0.75	0.90	0.82	10
T	0.86	0.55	0.67	11
U	0.88	0.88	0.88	8
V	1.00	0.10	0.18	10
X	1.00	0.33	0.50	9
Y	1.00	0.43	0.60	7
accuracy			0.40	220
macro avg	0.81	0.43	0.45	220
weighted avg	0.81	0.40	0.42	220

## 2. Logistic Regression với kỹ thuật one – vs – rest (OvR)

- Accuracy cao: 81.82%.
- f1-score cao: 0.82.
- Một số class có f1-score thấp so với các class còn lại: A(0.62), G(0.4), H(0.58), I(0.63), M(0.64), Q(0.62).
- Class Y,L có f1-score cao nhất: 1.

Accuracy on training set: 100.0%				
Accuracy on test set: 81.818181818183%				
	precision	recall	f1-score	support
A	0.57	0.67	0.62	12
B	0.82	1.00	0.90	9
C	1.00	0.87	0.93	15
D	0.90	1.00	0.95	9
E	1.00	0.89	0.94	9
G	1.00	0.25	0.40	8
H	0.64	0.54	0.58	13
I	0.71	0.56	0.63	9
K	1.00	0.90	0.95	10
L	1.00	1.00	1.00	9
M	0.53	0.82	0.64	11
N	0.67	1.00	0.80	10
O	0.92	1.00	0.96	12
P	0.85	0.73	0.79	15
Q	0.67	0.57	0.62	7
R	1.00	1.00	1.00	7
S	0.82	0.90	0.86	10
T	0.89	0.73	0.80	11
U	0.80	1.00	0.89	8
V	1.00	0.80	0.89	10
X	0.80	0.89	0.84	9
Y	1.00	1.00	1.00	7
accuracy			0.82	220
macro avg			0.84	220
weighted avg			0.84	220

## 3. Softmax Regression

- Accuracy cao: 82.27%.
- f1-score cao: 0.82.
- Một số class có f1-score thấp so với các class còn lại: A(0.59), G(0.36), H(0.67), I(0.63), M(0.69), Q(0.67).
- Class Y,L có f1-score cao nhất: 1.

Accuracy on training set: 100.0%				
Accuracy on test set: 82.272727272728%				
	precision	recall	f1-score	support
A	0.53	0.67	0.59	12
B	0.90	1.00	0.95	9
C	1.00	0.87	0.93	15
D	0.90	1.00	0.95	9
E	1.00	0.89	0.94	9
G	0.67	0.25	0.36	8
H	0.73	0.62	0.67	13
I	0.71	0.56	0.63	9
K	0.90	0.90	0.90	10
L	1.00	1.00	1.00	9
M	0.60	0.82	0.69	11
N	0.67	1.00	0.80	10
O	0.92	1.00	0.96	12
P	0.79	0.73	0.76	15
Q	0.80	0.57	0.67	7
R	0.88	1.00	0.93	7
S	0.90	0.90	0.90	10
T	0.90	0.82	0.86	11
U	0.80	1.00	0.89	8
V	1.00	0.70	0.82	10
X	0.80	0.89	0.84	9
Y	1.00	1.00	1.00	7
accuracy			0.82	220
macro avg			0.84	220
weighted avg			0.83	220

Nhìn chung, dựa vào kết quả dự đoán của tất cả model thì KNN có kết quả thấp nhất, trong khi đó Softmax Regression và one-vs-rest có kết quả gần giống nhau. Ký tự G có kết quả dự đoán thấp nhất ở cả Softmax Regression và one-vs-rest.

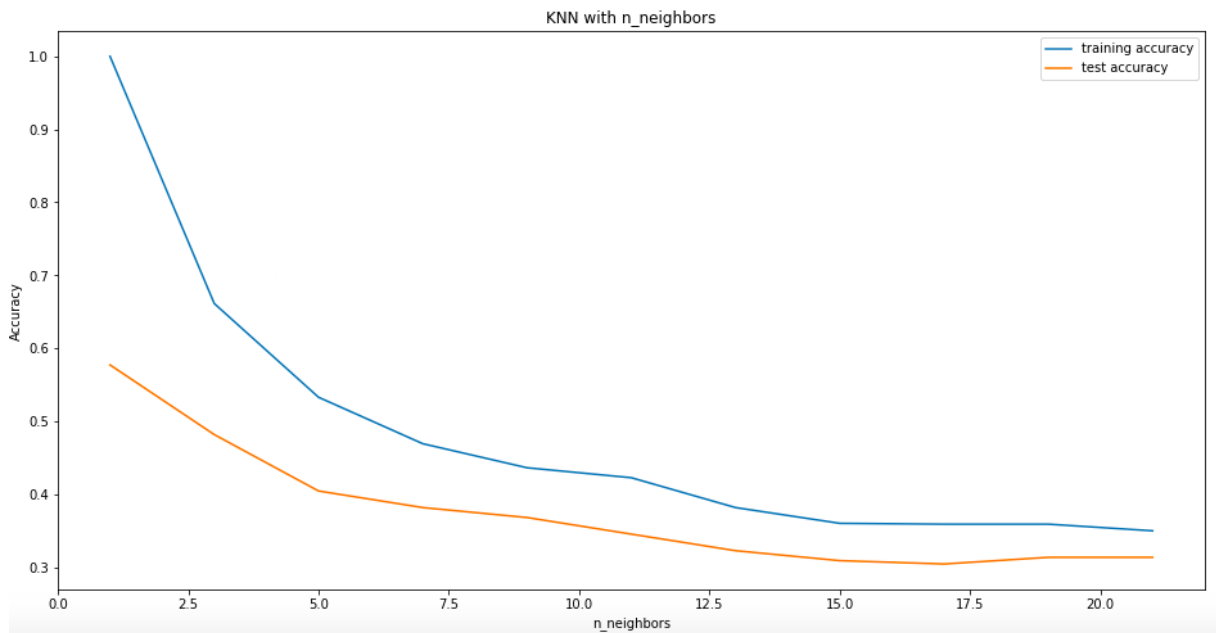
## V. TÍNH CHỈNH THAM SỐ

### 1. K-Nearest-Neighbors

Các tham số quan trọng cần xem xét:

- k: số điểm lân cận.
- metric: thuật toán tính khoảng cách giữa 2 điểm.

Vì chỉ cần dự đoán 22 ký tự nên xem xét k trong khoảng từ 1 - 22



*Dựa vào đồ thị, tham số k trong khoảng từ 15-22 sẽ cho kết quả tốt.*

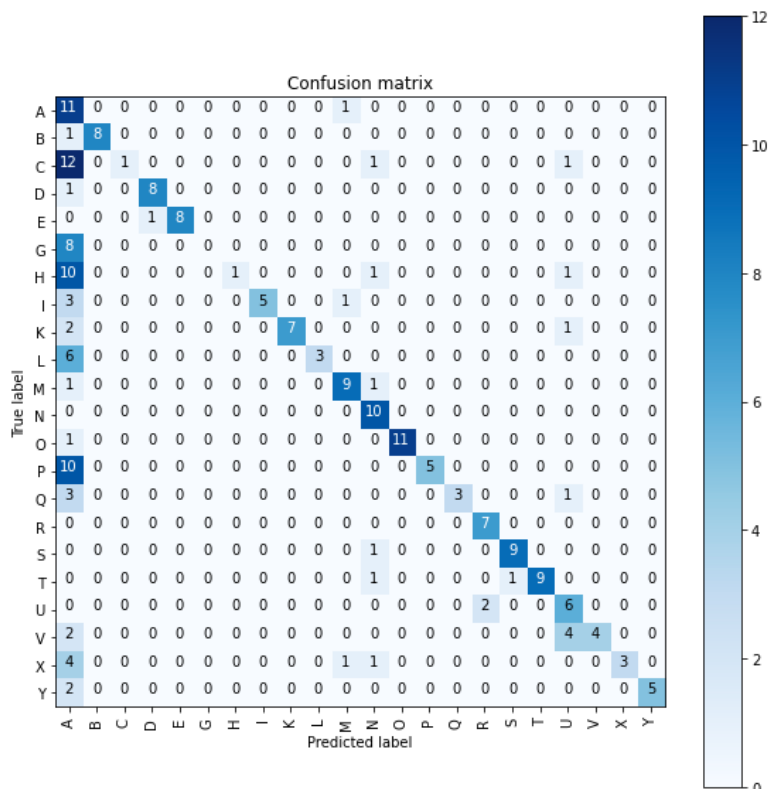
Các tham số tốt nhất sau khi tuning:

```
Best KNN model:  
Best performance: 0.589773 using {'metric': 'manhattan', 'n_neighbors': 15}
```

Accuracy on training set: 66.818181818183%

Accuracy on test set: 60.454545454545%

	precision	recall	f1-score	support
A	0.14	0.92	0.25	12
B	1.00	0.89	0.94	9
C	1.00	0.07	0.12	15
D	0.89	0.89	0.89	9
E	1.00	0.89	0.94	9
G	0.00	0.00	0.00	8
H	1.00	0.08	0.14	13
I	1.00	0.56	0.71	9
K	1.00	0.70	0.82	10
L	1.00	0.33	0.50	9
M	0.75	0.82	0.78	11
N	0.62	1.00	0.77	10
O	1.00	0.92	0.96	12
P	1.00	0.33	0.50	15
Q	1.00	0.43	0.60	7
R	0.78	1.00	0.88	7
S	0.90	0.90	0.90	10
T	1.00	0.82	0.90	11
U	0.43	0.75	0.55	8
V	1.00	0.40	0.57	10
X	1.00	0.33	0.50	9
Y	1.00	0.71	0.83	7
accuracy			0.60	220
macro avg	0.84	0.62	0.64	220
weighted avg	0.85	0.60	0.62	220



### Kết quả sau khi tuning trên tập test

Hiệu quả dự đoán được tăng, f1-score tăng từ: 0.4 lên 0.6. Mặc dù vậy vẫn có class L có f1-score = 0. Sự chênh lệch precision và recall giữa các class vẫn cao và f1-score ở 1 số class vẫn thấp.

## 2. Logistic Regression với kỹ thuật one-vs-rest (OvR)

Các tham số quan trọng cần xem xét:

- penalty: lựa chọn regularization
- solver: thuật toán dùng tối ưu hóa vấn đề.

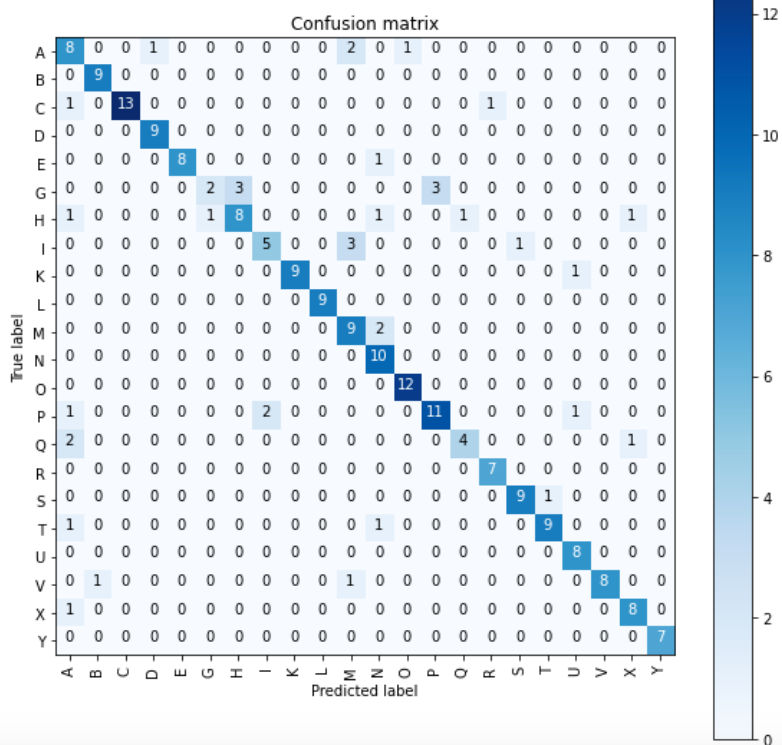
Các tham số tốt nhất sau khi tuning:

```
Best OvR model:
Best performance: 0.814773 using {'C': 50, 'penalty': 'l2'}
```

Accuracy on training set: 100.0%

Accuracy on test set: 82.727272727273%

	precision	recall	f1-score	support
A	0.53	0.67	0.59	12
B	0.90	1.00	0.95	9
C	1.00	0.87	0.93	15
D	0.90	1.00	0.95	9
E	1.00	0.89	0.94	9
G	0.67	0.25	0.36	8
H	0.73	0.62	0.67	13
I	0.71	0.56	0.63	9
K	1.00	0.90	0.95	10
L	1.00	1.00	1.00	9
M	0.60	0.82	0.69	11
N	0.67	1.00	0.80	10
O	0.92	1.00	0.96	12
P	0.79	0.73	0.76	15
Q	0.80	0.57	0.67	7
R	0.88	1.00	0.93	7
S	0.90	0.90	0.90	10
T	0.90	0.82	0.86	11
U	0.80	1.00	0.89	8
V	1.00	0.80	0.89	10
X	0.80	0.89	0.84	9
Y	1.00	1.00	1.00	7
accuracy			0.83	220
macro avg	0.84	0.83	0.82	220
weighted avg	0.84	0.83	0.82	220



*Kết quả sau khi tuning trên tập test*

Hiệu quả dự đoán được cải thiện một tí, f1-score tăng từ: 0.82 lên 0.83. Tuy nhiên, vẫn còn 1 số class có f1-score thấp hơn với tất cả các class còn lại như G(0.36), A(0.59).

### 3. Softmax Regression

Cũng tương tự one-vs-rest, các tham số quan trọng cần xem xét như:

- penalty: lựa chọn regularization
- solver: thuật toán dùng tối ưu hóa vấn đề.

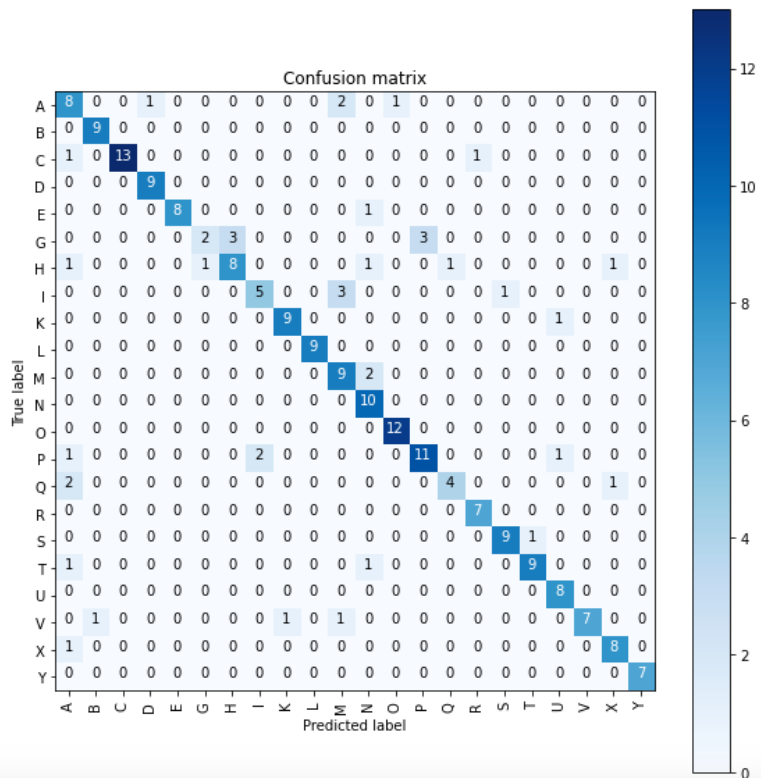
Các tham số tốt nhất sau khi tuning:

```
Best Softmax Regression model:
Best performance: 0.812500 using {'penalty': 'l2', 'solver': 'newton-cg'}
```

Accuracy on training set: 100.0%

Accuracy on test set: 82.27272727272728%

	precision	recall	f1-score	support
A	0.53	0.67	0.59	12
B	0.90	1.00	0.95	9
C	1.00	0.87	0.93	15
D	0.90	1.00	0.95	9
E	1.00	0.89	0.94	9
G	0.67	0.25	0.36	8
H	0.73	0.62	0.67	13
I	0.71	0.56	0.63	9
K	0.90	0.90	0.90	10
L	1.00	1.00	1.00	9
M	0.60	0.82	0.69	11
N	0.67	1.00	0.80	10
O	0.92	1.00	0.96	12
P	0.79	0.73	0.76	15
Q	0.80	0.57	0.67	7
R	0.88	1.00	0.93	7
S	0.90	0.90	0.90	10
T	0.90	0.82	0.86	11
U	0.80	1.00	0.89	8
V	1.00	0.70	0.82	10
X	0.80	0.89	0.84	9
Y	1.00	1.00	1.00	7
accuracy			0.82	220
macro avg	0.84	0.83	0.82	220
weighted avg	0.83	0.82	0.82	220



*Kết quả sau khi tuning trên tập test*

Hiệu quả dự đoán sau khi tuning không khác gì nhiều với model mặc định.

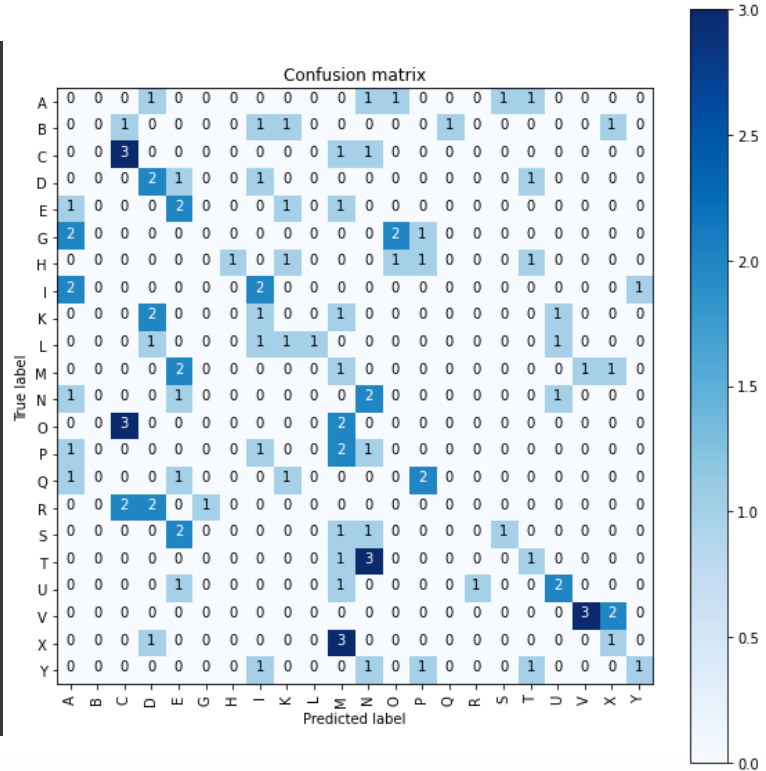
## VI. KẾT QUẢ VÀ ĐÁNH GIÁ

Sau khi xây dựng và kiểm thử với 3 model trên, model sau khi tuning cho kết quả tốt nhất là Logistic Regression với kỹ thuật one-vs-rest (f1-score = 0.83).

Bên cạnh đó Softmax Regression cho ra kết quả khá cao f1-score = 0.82. Thấp nhất là K-Nearest-Neighbors với f1-score = 0.6.

Áp dụng model tốt nhất trên bộ dữ liệu [kiểm tra](#):

Accuracy: 20.909090909090907%				
	precision	recall	f1-score	support
A	0.00	0.00	0.00	5
B	0.00	0.00	0.00	5
C	0.33	0.60	0.43	5
D	0.22	0.40	0.29	5
E	0.20	0.40	0.27	5
G	0.00	0.00	0.00	5
H	1.00	0.20	0.33	5
I	0.25	0.40	0.31	5
K	0.00	0.00	0.00	5
L	1.00	0.20	0.33	5
M	0.07	0.20	0.11	5
N	0.20	0.40	0.27	5
O	0.00	0.00	0.00	5
P	0.00	0.00	0.00	5
Q	0.00	0.00	0.00	5
R	0.00	0.00	0.00	5
S	0.50	0.20	0.29	5
T	0.20	0.20	0.20	5
U	0.40	0.40	0.40	5
V	0.75	0.60	0.67	5
X	0.20	0.20	0.20	5
Y	0.50	0.20	0.29	5
accuracy			0.21	110
macro avg	0.26	0.21	0.20	110
weighted avg	0.26	0.21	0.20	110



- Đạt được f1-score rất thấp = 0.21
- Xảy ra hiện tượng overfitting. Model dự đoán có hiệu quả kém với bộ dữ liệu kiểm tra được chụp ở góc chụp, hướng chụp khác nhau.
- Nguyên nhân có thể là do bộ dữ liệu train không đa dạng về điều kiện môi trường hình ảnh. Một nguyên nhân có thể xem xét là đặc trưng lựa chọn không hợp lí. Cũng có thể model không đáp ứng được các đặc trưng lựa chọn.

## VII. KẾT LUẬN

Đồ án đưa ra được phương pháp để loại bỏ đi những thông tin không cần thiết trong ảnh nhằm lấy được thông tin quan trọng. Tuy nhiên, phương pháp này chỉ hiệu quả khi background màu tối (tương phản với màu da). Vì thế, cần

có sự thay đổi về tập dữ liệu train cũng như phương pháp để lấy được bàn tay khi background hỗn loạn.

Đồ án lựa chọn hình dáng, góc cạnh của bàn tay phải là đặc trưng của bài toán. Sẽ không hiệu quả khi áp dụng với bàn tay trái và cần cân nhắc lựa chọn đặc trưng khác để tăng hiệu quả của model.

Đồ án lựa chọn 3 model chính để xem xét: K-Nearest-Neighbors, Logistic Regression với kỹ thuật one-vs-rest và Softmax Regression. Trong đó one-vs-rest và Softmax Regression cho thấy độ hiệu quả khá giống nhau, k-NN có kết quả thấp nhất. Tuy nhiên, có xảy ra hiện tượng overfitting, model cho độ kết quả không cao khi dự đoán trên tập kiểm tra nhưng lại tốt trên tập train.

SVMs, Neural Network là những model được sử dụng rộng rãi vì độ hiệu quả cao của nó. Để có thể nâng cao độ chính xác cho bài toán thì đây chính là những model cần cân nhắc.

Hướng phát triển:

- Sử dụng các model có thể tự rút trích đặc trưng như CNN, VGG16,... để tăng sự chính xác của dự đoán.
- Nghiên cứu phương pháp dự đoán những ngôn ngữ ký hiệu động, đồng thời áp dụng việc dự đoán trên thời gian thực. Xa hơn nữa có thể tạo ứng dụng hỗ trợ phiên dịch thủ ngữ nhằm hỗ trợ việc giao tiếp với người khiếm thính.



## TÀI LIỆU THAM KHẢO

1. Ngôn ngữ kí hiệu  
<[https://vi.wikipedia.org/wiki/Ng%C3%B4n\\_ng%E1%BB%AF\\_k%C3%BD\\_hi%E1%BB%87u](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_k%C3%BD_hi%E1%BB%87u)>
2. MotionSavvy  
<<https://techcrunch.com/2014/06/06/motionsavvy-is-a-tablet-app-that-understands-sign-language/>>
3. Tencent YouTu Lab  
<<https://www.youtube.com/watch?v=XYnwcIzZBGQ>>
4. Ngôn ngữ ký hiệu Mỹ (ASL)  
<[https://en.wikipedia.org/wiki/American\\_Sign\\_Language](https://en.wikipedia.org/wiki/American_Sign_Language)>
5. Ngôn ngữ ký hiệu Việt Nam (VSL)  
<[https://vi.wikipedia.org/wiki/Ng%C3%B4n\\_ng%E1%BB%AF\\_k%C3%BD\\_hi%E1%BB%87u\\_Vi%E1%BB%87t\\_Nam](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_k%C3%BD_hi%E1%BB%87u_Vi%E1%BB%87t_Nam)>
6. Bộ dataset sử dụng trong đồ án  
<[https://drive.google.com/drive/folders/1ULJv-gxbBxwoW3QO1MGZKvjK\\_0tvfQbR?usp=sharing](https://drive.google.com/drive/folders/1ULJv-gxbBxwoW3QO1MGZKvjK_0tvfQbR?usp=sharing)>
7. File Colab tiền xử lý hình ảnh  
<<https://colab.research.google.com/drive/18iVInC5O7qi9uAWoOWpzPj7J7-txv9Of?usp=sharing>>
8. Folder hình ảnh được xử lý  
<[https://drive.google.com/drive/folders/1X7P4noJyEeRBEJ\\_ysoh\\_9YQVhFTB92Ho?usp=sharing](https://drive.google.com/drive/folders/1X7P4noJyEeRBEJ_ysoh_9YQVhFTB92Ho?usp=sharing)>
9. Toán tử Laplace  
<[https://docs.opencv.org/3.4/d5/db5/tutorial\\_laplace\\_operator.html](https://docs.opencv.org/3.4/d5/db5/tutorial_laplace_operator.html)>
10. Scikit-learn  
<<https://scikit-learn.org>>