

Trường Đại học Công nghệ thông tin

Đồ án môn học

Nhập môn Nhập môn thị giác máy tính CS232.021

# Nhận diện Cử chỉ

**Giảng viên hướng dẫn: TS. Mai Tiến Dũng**

**Thành viên:**

21522628 - Phan Văn Thiện

21522509 - Hoàng Anh Đức Đăng Quang

21520952 - Hoàng Quang Khải

# Nội dung

01. Giới thiệu

02. Phát biểu bài toán

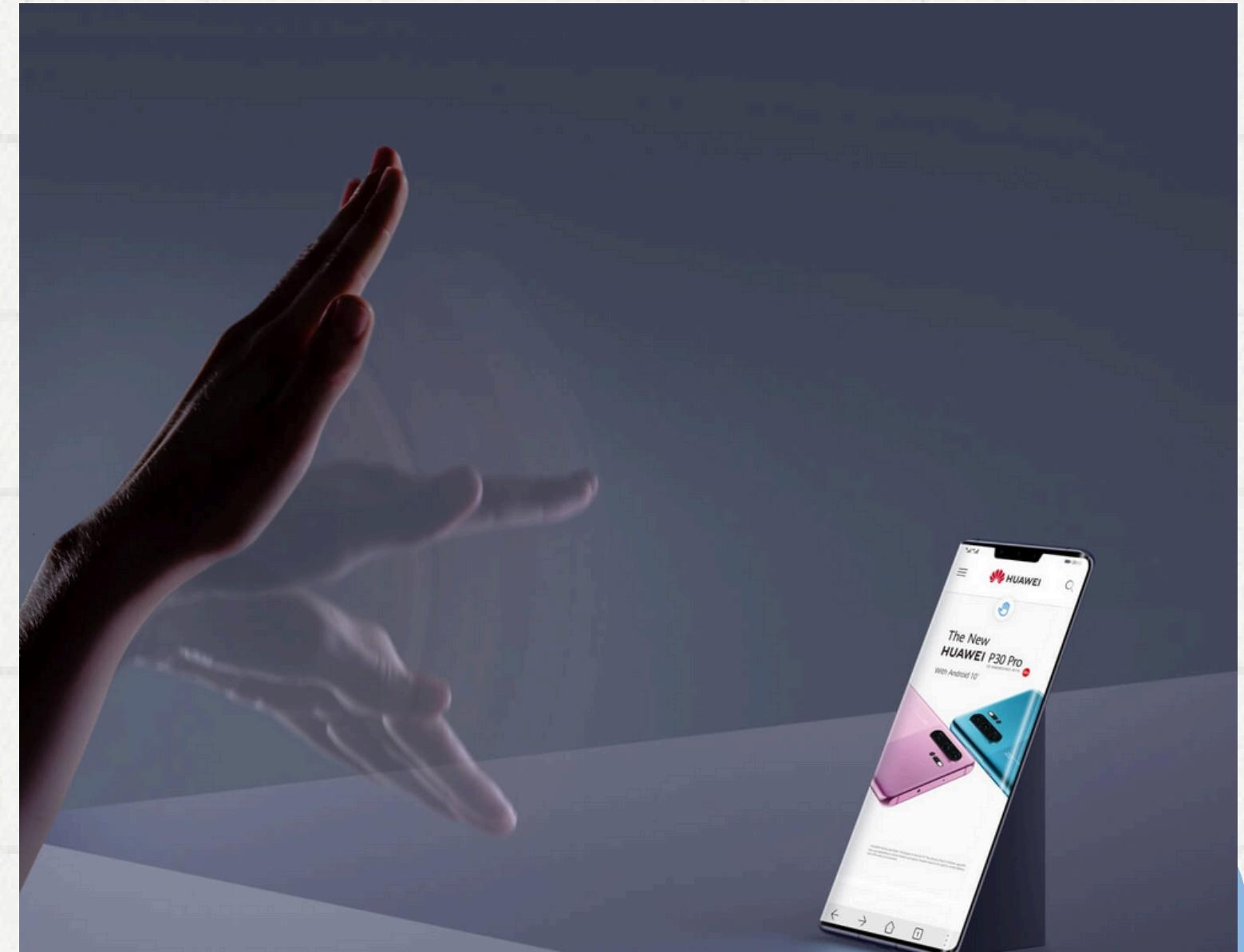
03. Phương pháp

04. Thực nghiệm

# 1. Giới thiệu

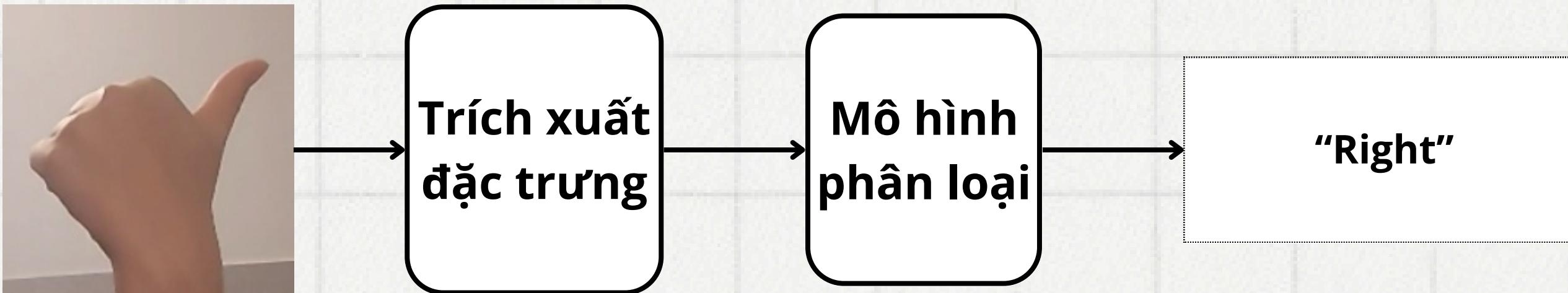
Nhận diện cử chỉ là một bài toán giúp máy tính có thể hiểu được một số cử chỉ tương ứng với các mệnh lệnh đã được định nghĩa từ trước.

Nhận diện cử chỉ có nhiều ứng dụng đa dạng, mang lại những trải nghiệm tương tác sống động và thú vị hơn với người dùng. Đặc biệt, lĩnh vực giải trí và trò chơi, việc điều khiển bằng cử chỉ đem lại nhiều thích thú.



## 2. Phát biểu bài toán

- Input:
  - Tập dữ liệu đã được gán nhãn.
  - Ảnh bàn tay phải của con người (đã được xén ảnh)
- Output:
  - Nhãn tương ứng với cử chỉ được nhận diện thông qua ảnh đầu vào. Bao gồm: "up", "right", "left"



# 3. Phương pháp

01. VGG net

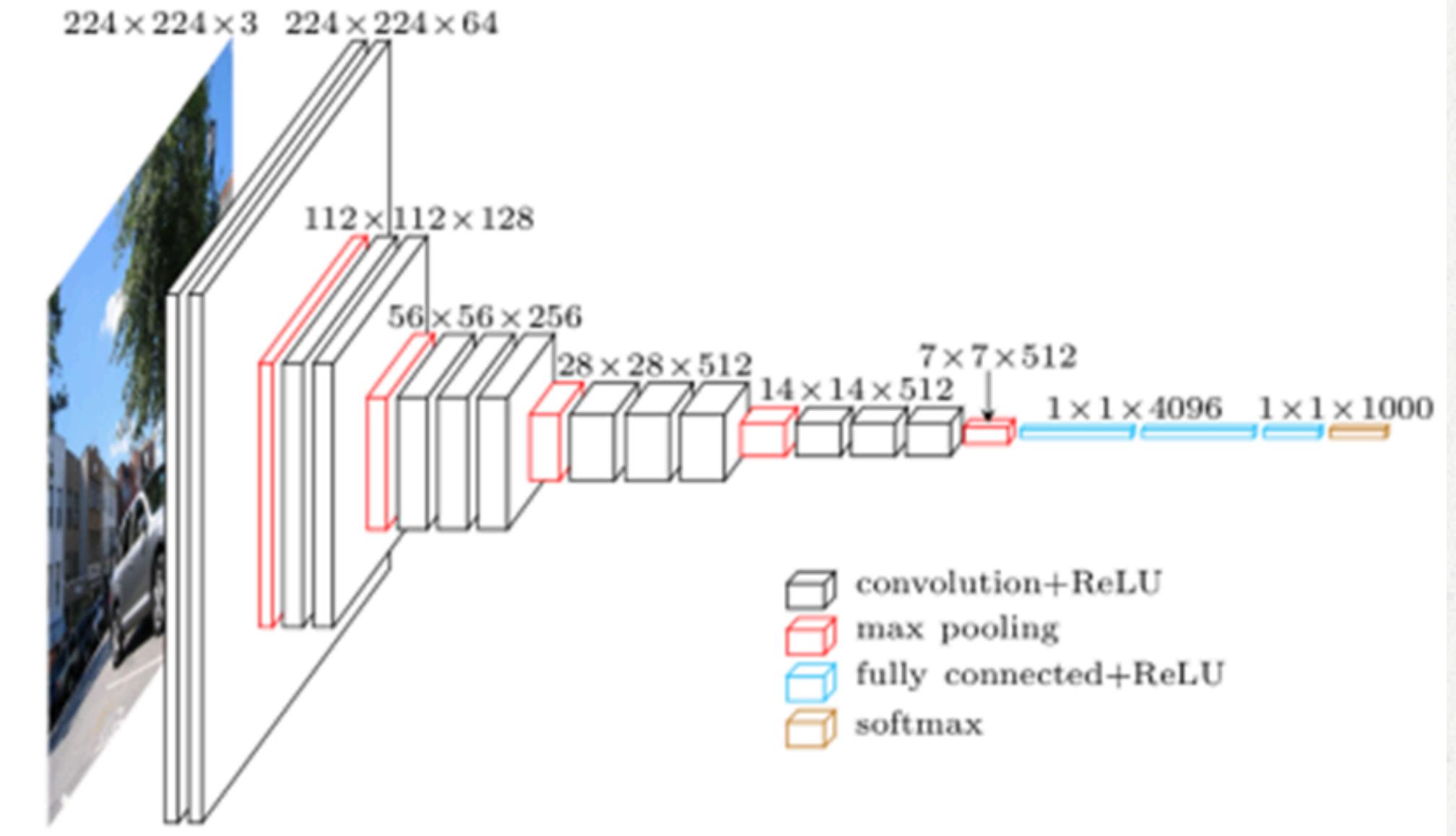
02. Random Forest

03. Support Vector Machine

04. Logistic Regression

# 3.1. VGG

- VGGNet [1] là một trong những mô hình pretrain phổ biến, được giới thiệu lần đầu tại ILSVRC 2014 bởi Karen Simonyan và Andrew Zisserman
- Ưu điểm của VGGNET là được thiết kế theo các mô-đun dễ dàng mở rộng: VGG11, VGG13, VGG16, ...
- Mỗi mô-đun gồm:  
(Conv+Relu+Maxpooling).



## 3.2. Random Forest

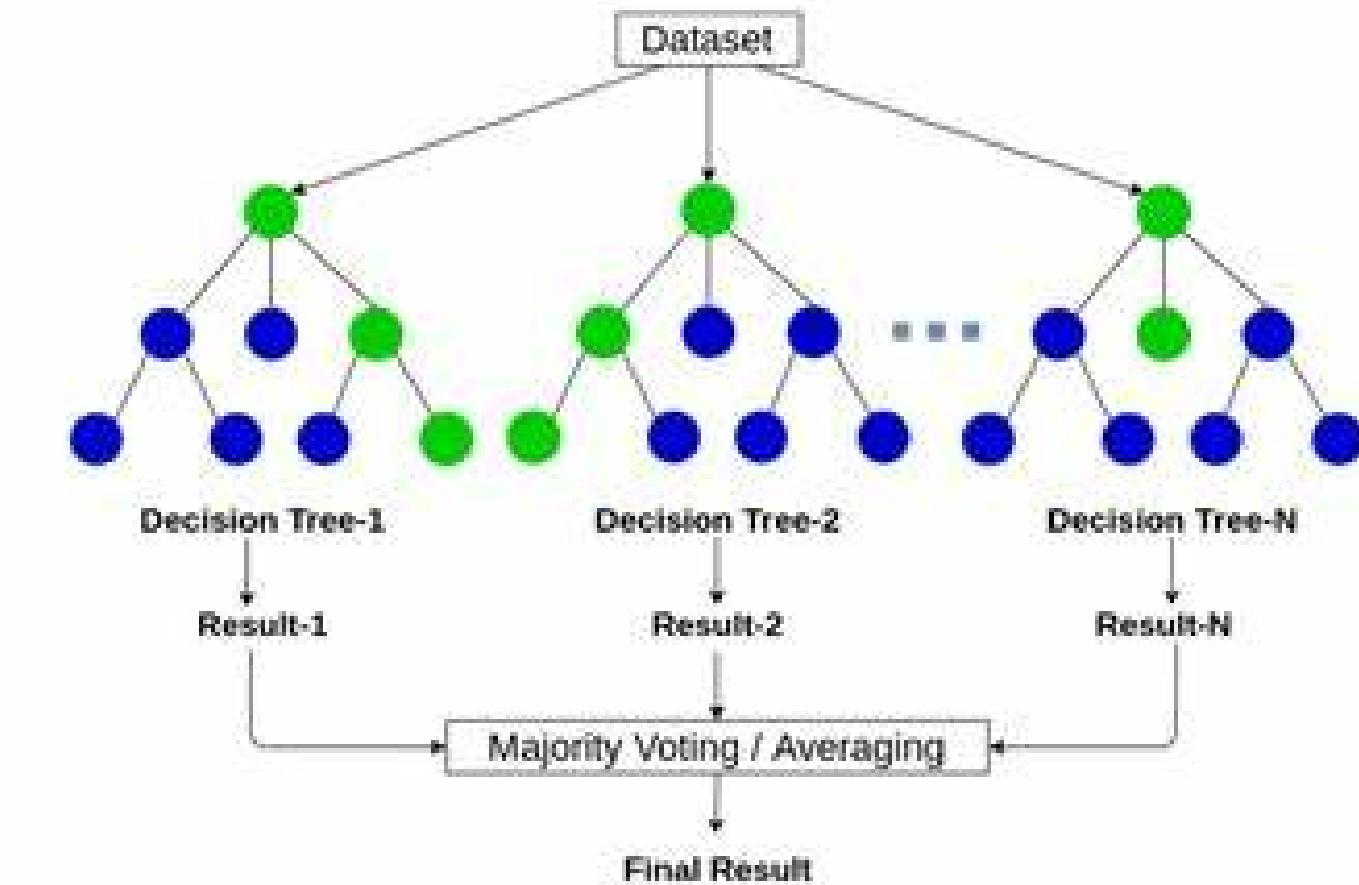
- Random Forest [2] là một phương pháp học máy được sử dụng cho cả phân loại và hồi quy.
- Random Forest được đề xuất bởi Leo Breiman vào năm 2001.
- Ý tưởng và các nguyên tắc cơ bản của Random Forest được trình bày chi tiết trong bài báo khoa học "Random Forests" của Leo Breiman, xuất bản trên tạp chí "Machine Learning" vào năm 2001.



## 3.2. Random Forest

- Ý tưởng Random Forest giống với ý tưởng của The Wisdom of Crowds [3] được đề xuất bởi James Surowiecki vào năm 2004.
- The Wisdom of Crowds nói rằng thông thường tổng hợp thông tin từ 1 nhóm sẽ tốt hơn từ một cá nhân.
- Ở thuật toán Random Forest mình cũng tổng hợp thông tin từ 1 nhóm các cây quyết định và kết quả cho ra tốt hơn thuật toán Decision Tree với 1 cây quyết định

### Random Forest



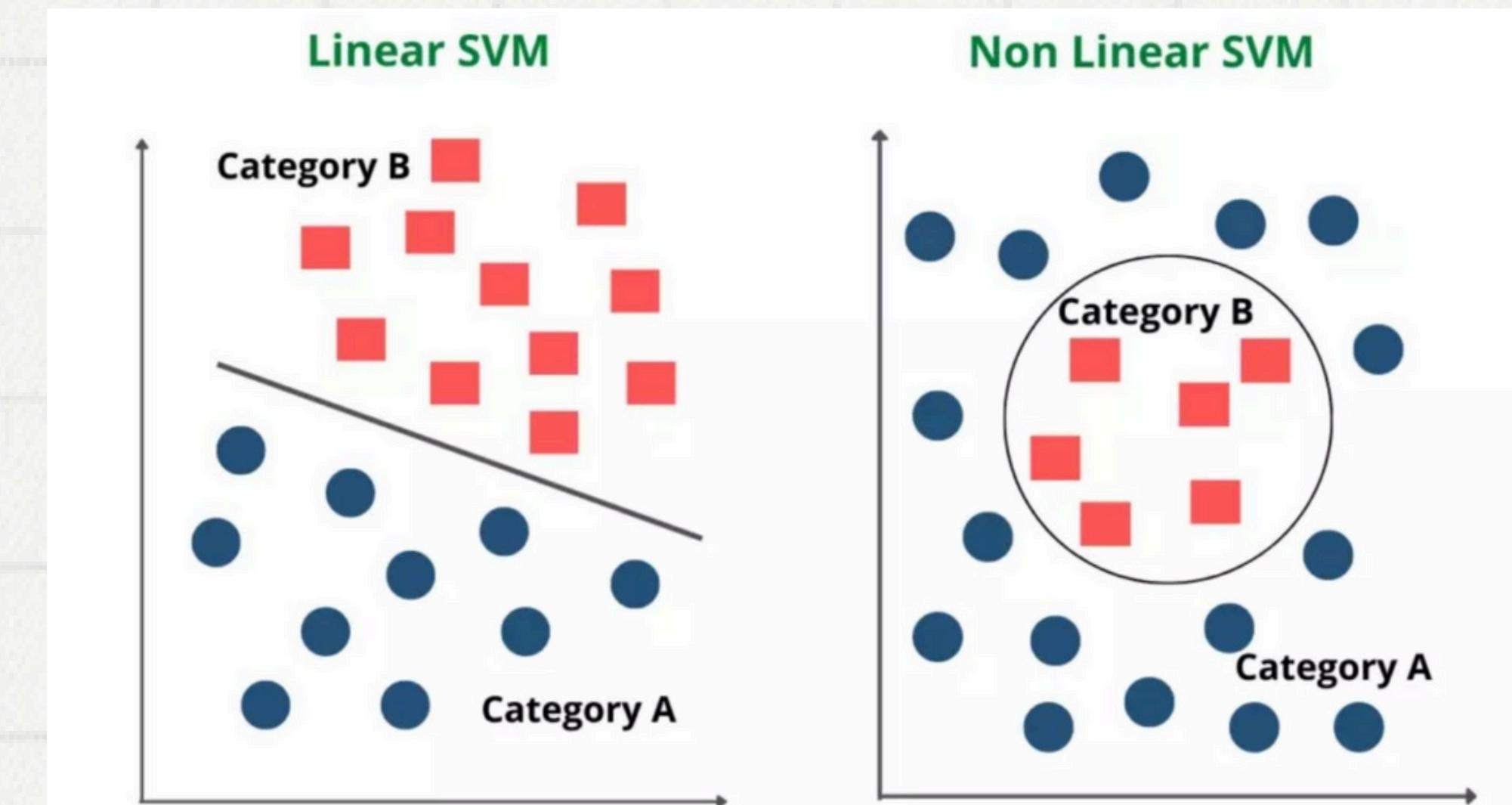
### 3.3. Support Vector Machine

- SVM (Support Vector Machine) [4] là một thuật toán học máy mạnh mẽ được sử dụng cho các bài toán phân loại và hồi quy.
- SVM lần đầu tiên được đề xuất bởi Vladimir Vapnik và Alexey Chervonenkis. Những công trình quan trọng và cơ sở lý thuyết của SVM được mô tả chi tiết trong cuốn sách "The Nature of Statistical Learning Theory" của Vladimir Vapnik, xuất bản lần đầu năm 1995.



# 3.3. Support Vector Machine

- Ý tưởng chính của SVM là tìm kiếm một siêu phẳng (hyperplane) tốt nhất phân tách các lớp dữ liệu trong không gian đặc trưng.
- Siêu phẳng này được xác định sao cho khoảng cách giữa nó và các điểm dữ liệu gần nhất (gọi là các support vectors) là lớn nhất, nhằm tối đa hóa lề phân cách giữa các lớp.

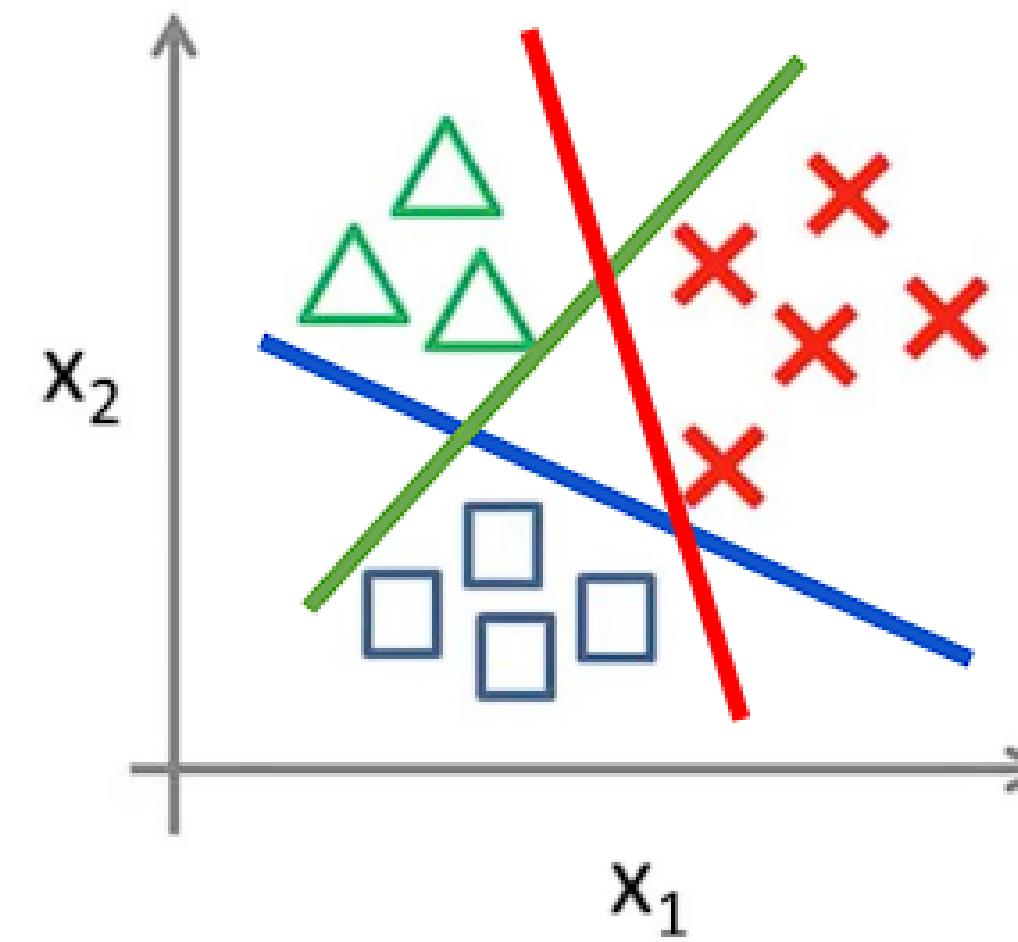


# 3.3. Logistic Regression

- Logistic Regression [5] là 1 thuật toán phân loại được dùng để gán các đối tượng cho 1 tập hợp giá trị rời rạc.
- Ý tưởng cơ bản của Logistic Regression là dự đoán xác suất của một sự kiện xảy ra bằng cách ánh xạ các biến đầu vào thông qua một hàm logistic.
- Với phân loại đa lớp: mô hình sẽ fit sử dụng cross-entropy loss để dự đoán giá trị số nguyên, với mỗi số nguyên đã được encoded dựa trên label.

## Multiclass Classification

Involves more than 2 classes



# 4. Thực nghiệm

1. Bộ dữ liệu
2. Độ đo đánh giá
3. Cấu hình thực nghiệm
4. Kết quả
5. Minh họa



# 4.1. Dữ liệu

- Bộ dữ liệu do nhóm thực hiện thu thập. Gồm hình ảnh bàn tay của 3 thành viên trong nhóm. Ảnh được chụp ở nhiều địa điểm khác nhau với các góc sáng, góc bàn tay khác nhau nhằm tạo nên sự đa dạng trong dữ liệu.
- Bộ dữ liệu gồm 2550 mẫu được chia thành 2 tập
  - Tập Train: 2250 mẫu
  - Tập Test: 300 Mẫu
- Bộ dữ liệu đã được xử lý xén ảnh bàn tay trước để thuận tiện trong quá trình huấn luyện và sử dụng.

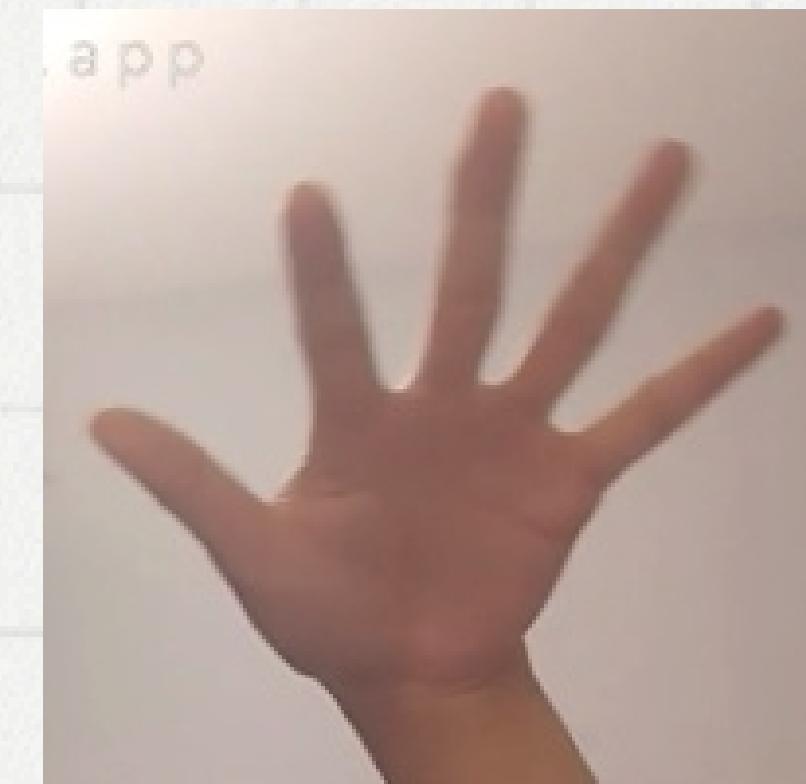
# 4.1. Dữ liệu



right



left



up

# 4.1. Độ đo đánh giá

**Precision** là tỷ lệ giữa số mẫu được tính là True Positive (TP) với tổng số mẫu được phân loại là Positive (bằng TP + FP)

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

**Recall** là tỷ lệ giữa các điểm positive thực được nhận đúng trên tổng điểm positive thực.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

		Model dự đoán	
		Positive	Negative
Thực tế	Positive	True Positive (Dự đoán Đúng là Positive)	False Negative (Dự đoán Sai là Negative)
	Negative	False Positive (Dự đoán Sai là Positive)	True Negative (Dự đoán Đúng là Negative)

# 4.1. Độ đo đánh giá

**F1 score** : kết hợp giữa precision và recall vào một chỉ số duy nhất để đánh giá cả sự chính xác và khả năng bao quát của mô hình

$$F1 = 2 \frac{Precision * Recall}{Precision + Recall}$$

**Độ chính xác** là một số liệu thống kê lường tỉ lệ các dự đoán chính xác của một mô hình so với tổng số mẫu dữ liệu

$$\text{Accuracy} = \frac{\text{Số lượng dự đoán đúng}}{\text{Tổng số mẫu dữ liệu}}$$

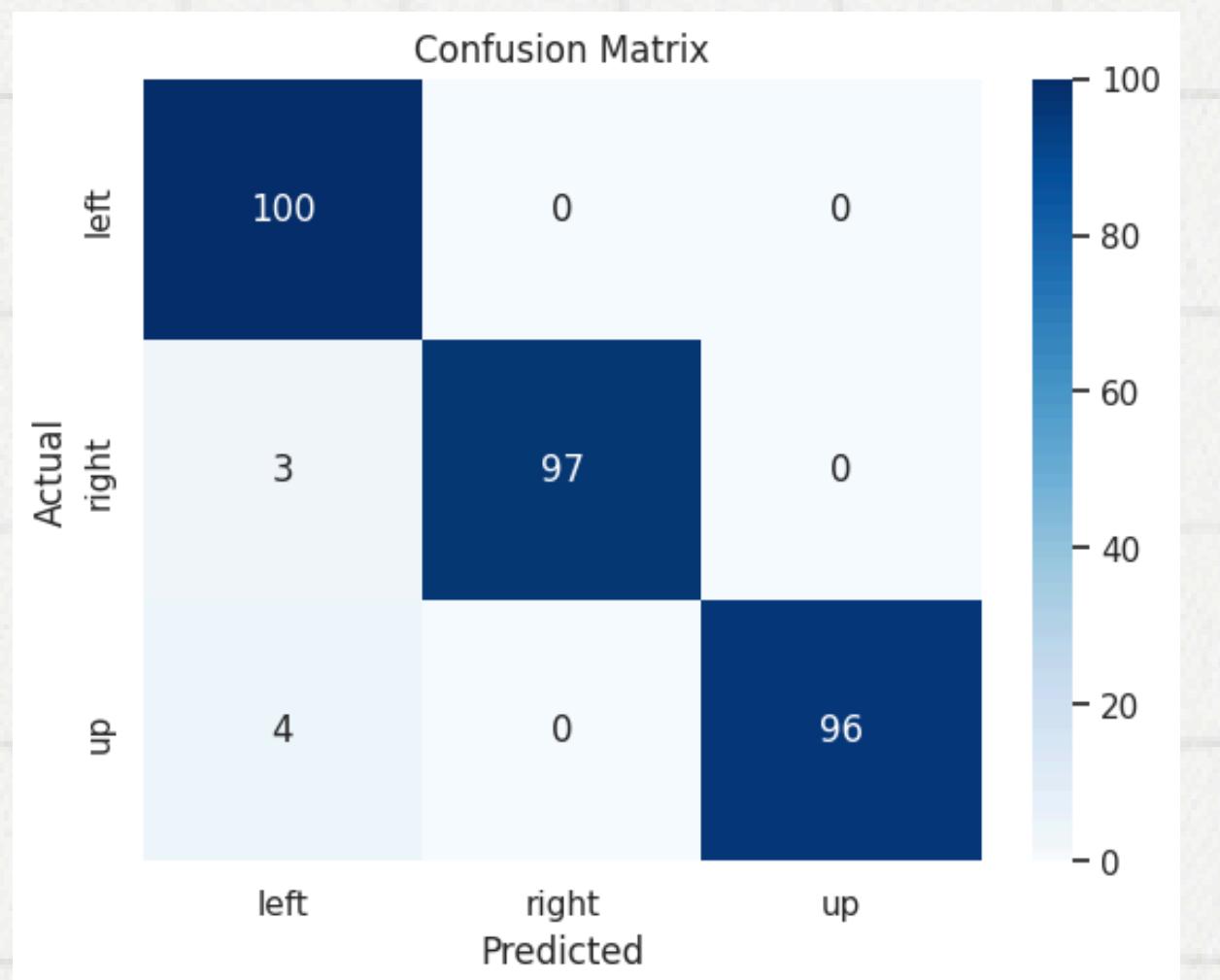
## 4.2. Cấu hình thực nghiệm

- Tiền xử lý:
  - xén ảnh bàn tay (sử dụng thư viện cvzone[6])
  - resize về (224,224)
- Trích xuất đặc trưng - VGG16:
  - Bộ trọng số: imagenet [\[7\]](#)
  - Sử dụng output lớp maxpooling cuối cùng của VGG.
  - Sau khi được Flatten,kết quả này nó sẽ là input của các mô hình phân loại

## 4.2. Cấu hình thực nghiệm

- Mô hình phân loại:
  - SVM:
    - 'C': 100, 'gamma': 'auto', 'kernel': 'rbf'
  - RandomForest:
    - n\_estimators= [100,300], max\_features=['auto']
  - LR:
    - C=[0.1, 1, 10, 100], solver=["lbfgs", "saga"]
- Sử dụng RandomSearch để tìm bộ tham số tối ưu cho các mô hình phân loại

# 4.3. Kết quả thực nghiệm SVM



Confusion Matrix của SVM

	precision	recall	f1-score	support
0	0.93	1.00	0.97	100
1	1.00	0.97	0.98	100
2	1.00	0.96	0.98	100
<b>accuracy</b>				0.98
<b>macro avg</b>				0.98
<b>weighted avg</b>				0.98

Kết quả Classification Report của SVM

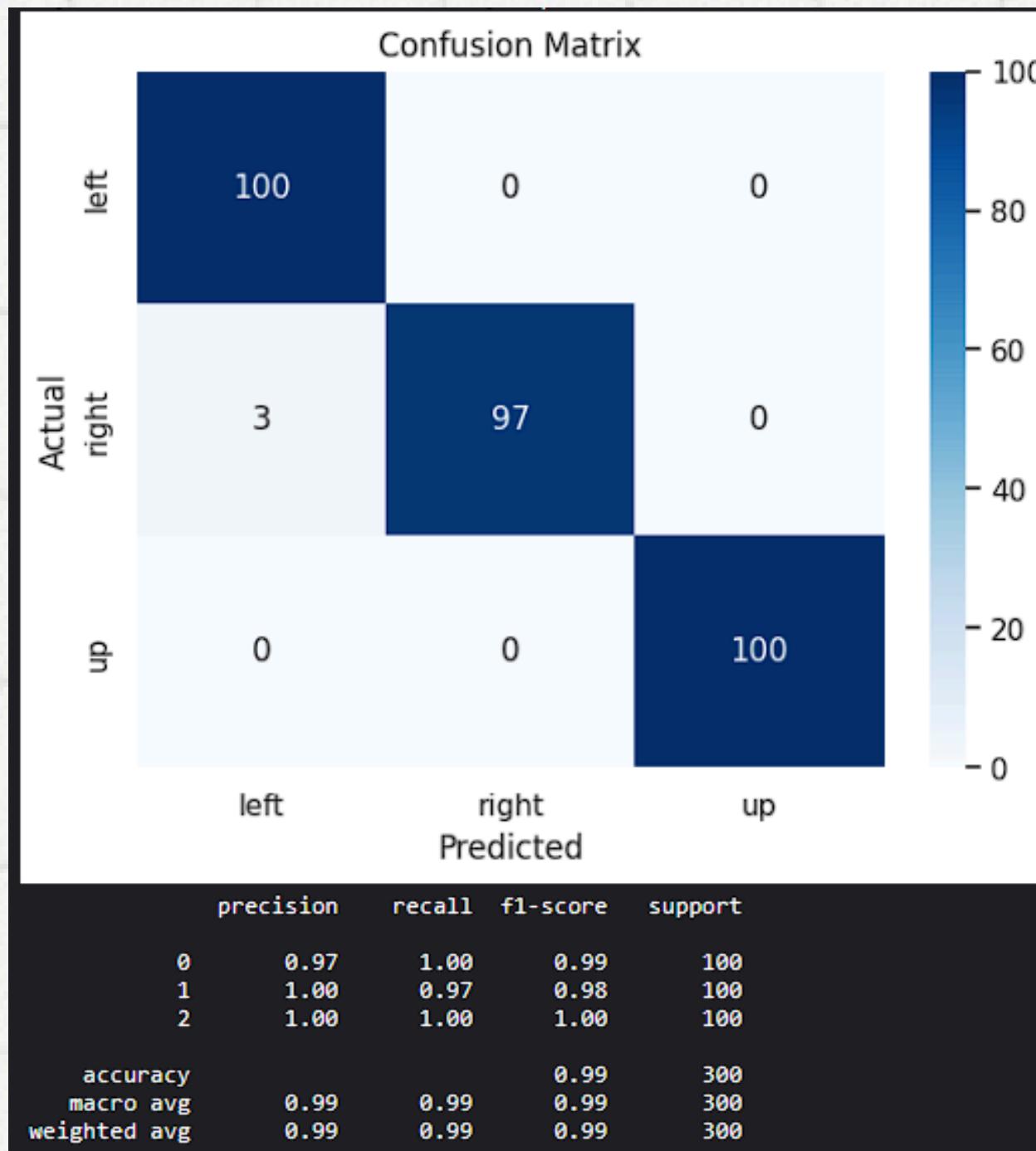
# 4.3. Kết quả thực nghiệm RF

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_RF_n_estimators	param_RF_max_features	params	split0_test_score	mean_test_score	std_test_score	rank_test_score
0	11.318239	1.955931	2.453924	0.821771	300	log2	{'RF_n_estimators': 300, 'RF_max_features': ...}	0.993333	0.995111	0.000889	4
1	9.118676	0.082462	1.956547	0.016101	20	auto	{'RF_n_estimators': 20, 'RF_max_features': ...}	0.991111	0.992000	0.002266	8
2	22.458925	0.159339	2.018767	0.013553	300	auto	{'RF_n_estimators': 300, 'RF_max_features': ...}	0.995556	0.995556	0.000000	1
3	8.592958	0.043772	1.990921	0.018570	10	auto	{'RF_n_estimators': 10, 'RF_max_features': ...}	0.993333	0.992000	0.001778	8
4	17.717626	0.138227	2.008391	0.008782	200	sqrt	{'RF_n_estimators': 200, 'RF_max_features': ...}	0.995556	0.995111	0.000889	6
5	9.055835	0.020315	1.982821	0.026272	20	sqrt	{'RF_n_estimators': 20, 'RF_max_features': ...}	0.995556	0.995111	0.002177	6
6	17.775229	0.144263	2.019146	0.011777	200	auto	{'RF_n_estimators': 200, 'RF_max_features': ...}	0.993333	0.995556	0.001405	1
7	8.261209	0.047962	2.010093	0.010946	10	log2	{'RF_n_estimators': 10, 'RF_max_features': ...}	0.995556	0.991556	0.003266	10
8	10.569565	0.049376	2.009138	0.013911	50	sqrt	{'RF_n_estimators': 50, 'RF_max_features': ...}	0.993333	0.995111	0.000889	4
9	12.994587	0.066386	2.007203	0.014409	100	auto	{'RF_n_estimators': 100, 'RF_max_features': ...}	0.993333	0.995556	0.001405	1

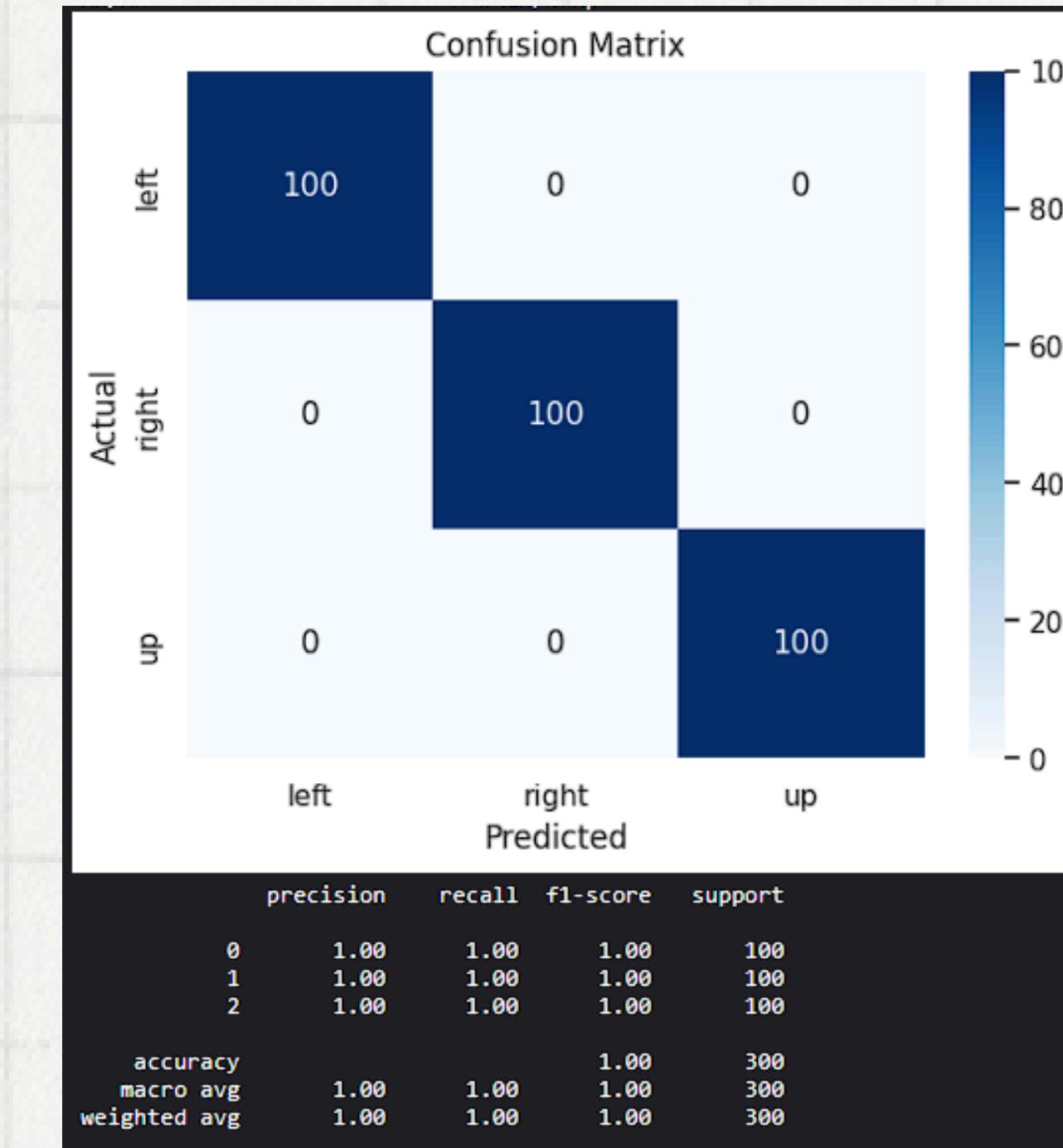
Bảng score RandomForest với các params khác nhau

Theo bảng bảng trên ta thấy max\_feature="auto" và n\_estimator >=100 cho kết quả tốt nhất

# 4.3. Kết quả thực nghiệm RF

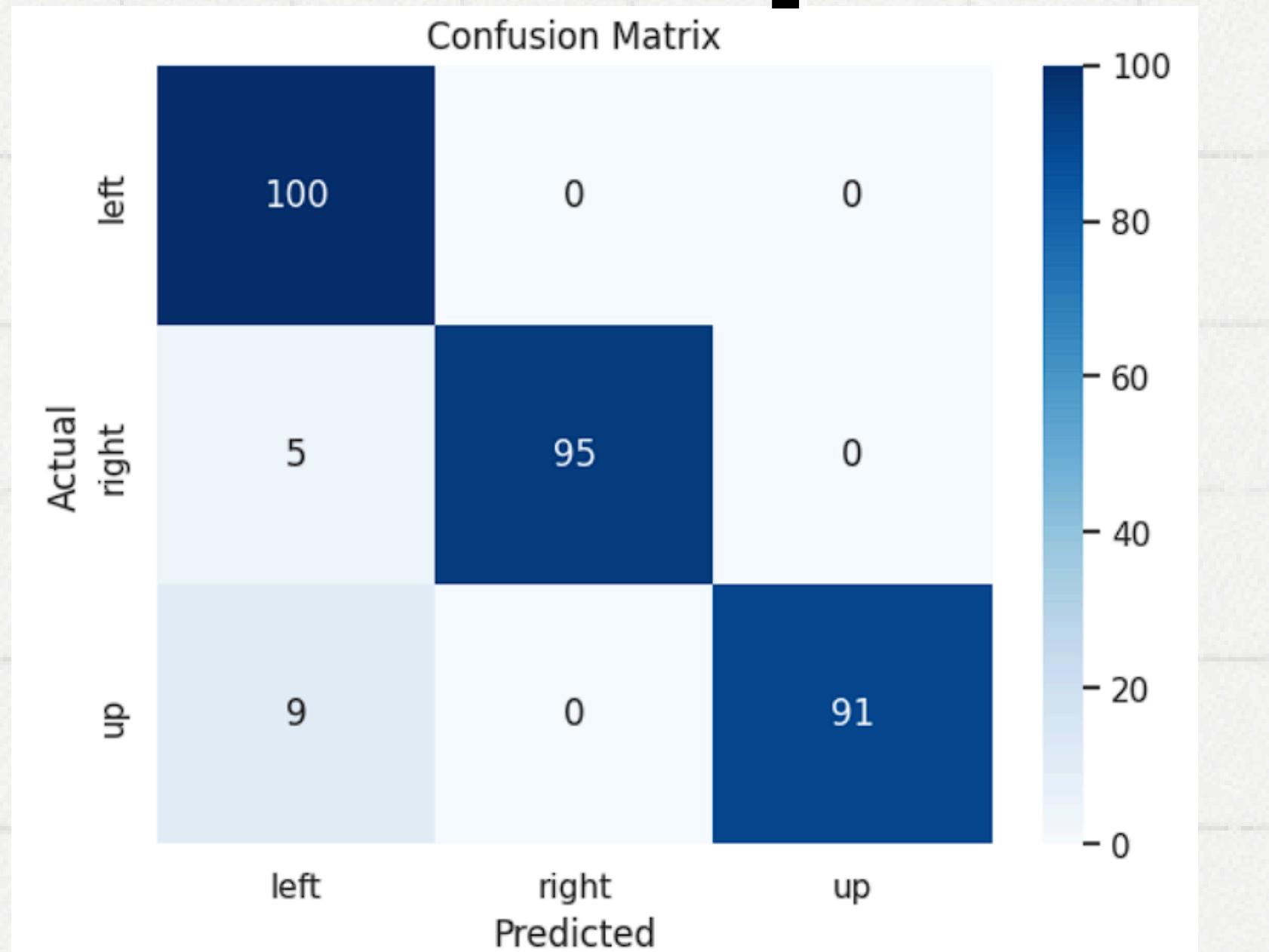


n\_estimators=100

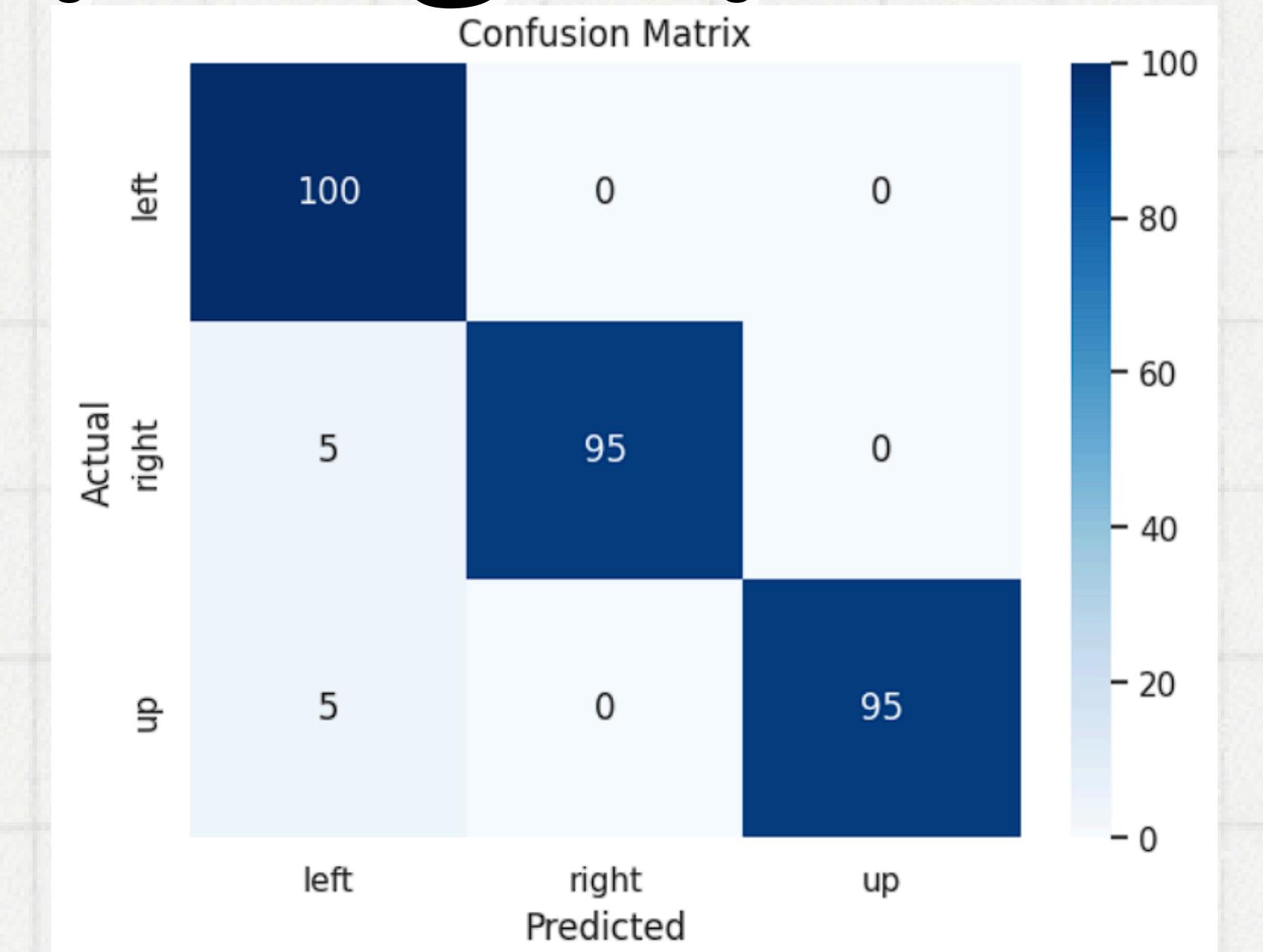


n\_estimators=300

# 4.3. Kết quả thực nghiệm LR

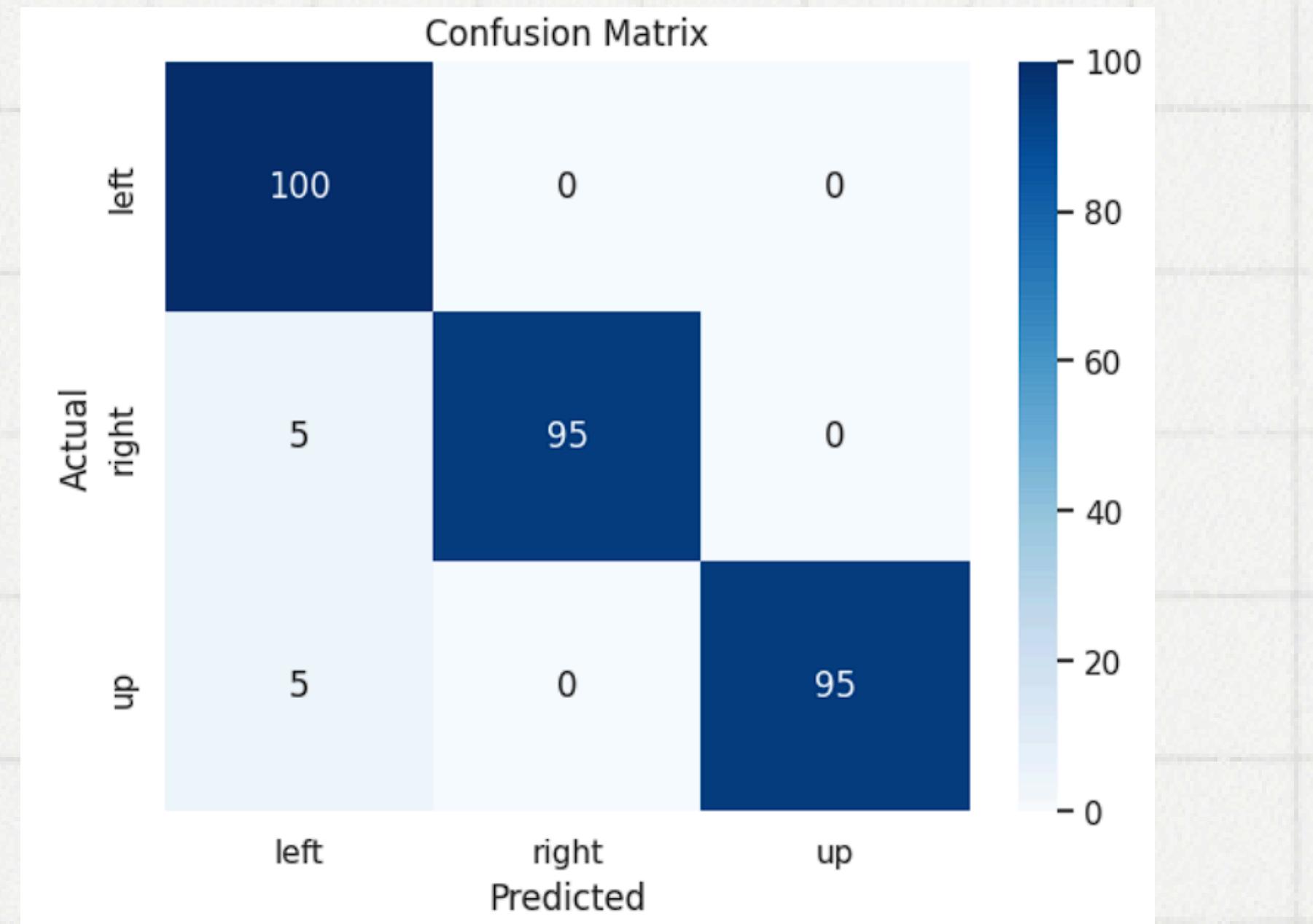


$C=10$

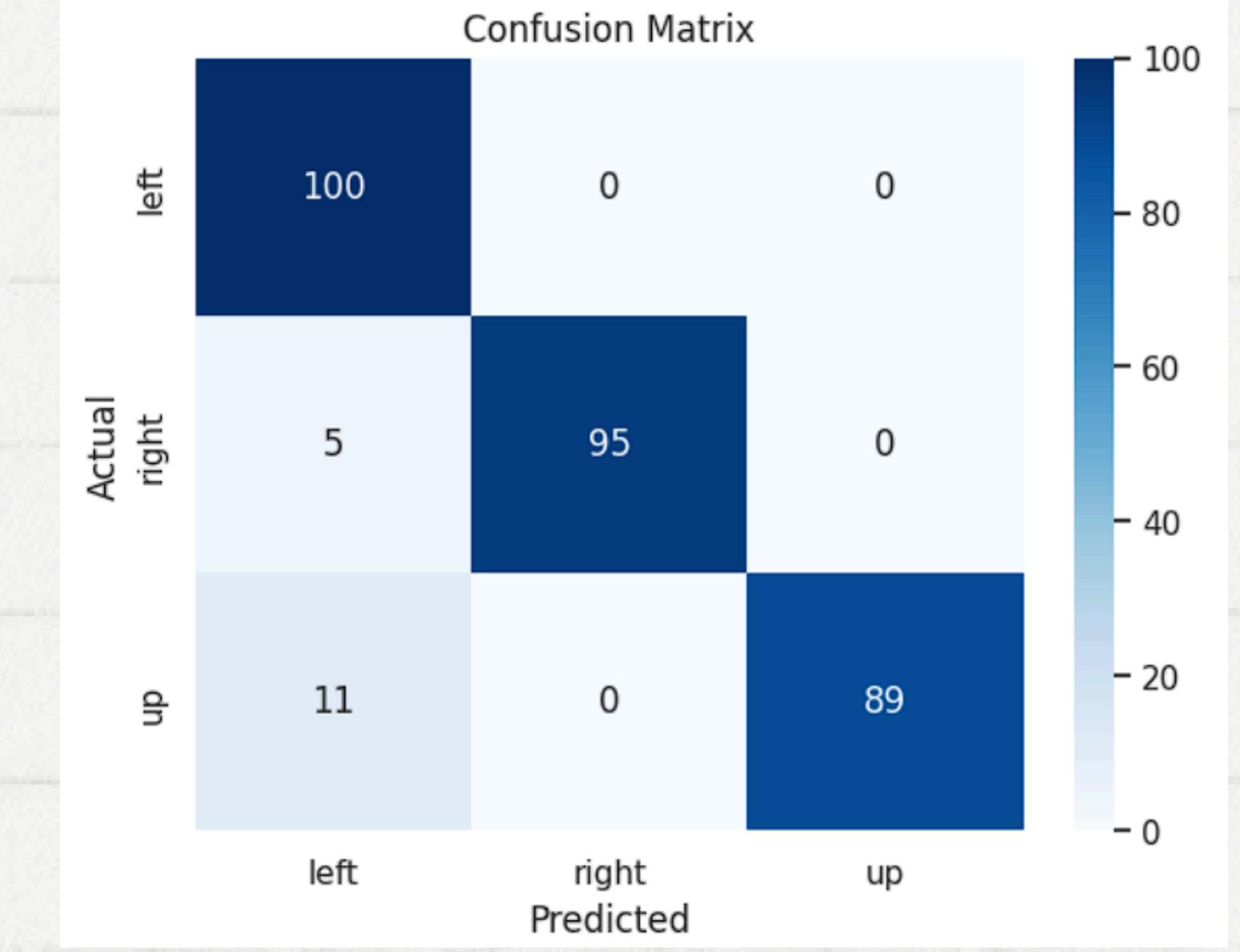


$C=0.1$

# 4.3. Kết quả thực nghiệm LR



$C=1$



$C=100$

# 4.3. Kết quả thực nghiệm LR

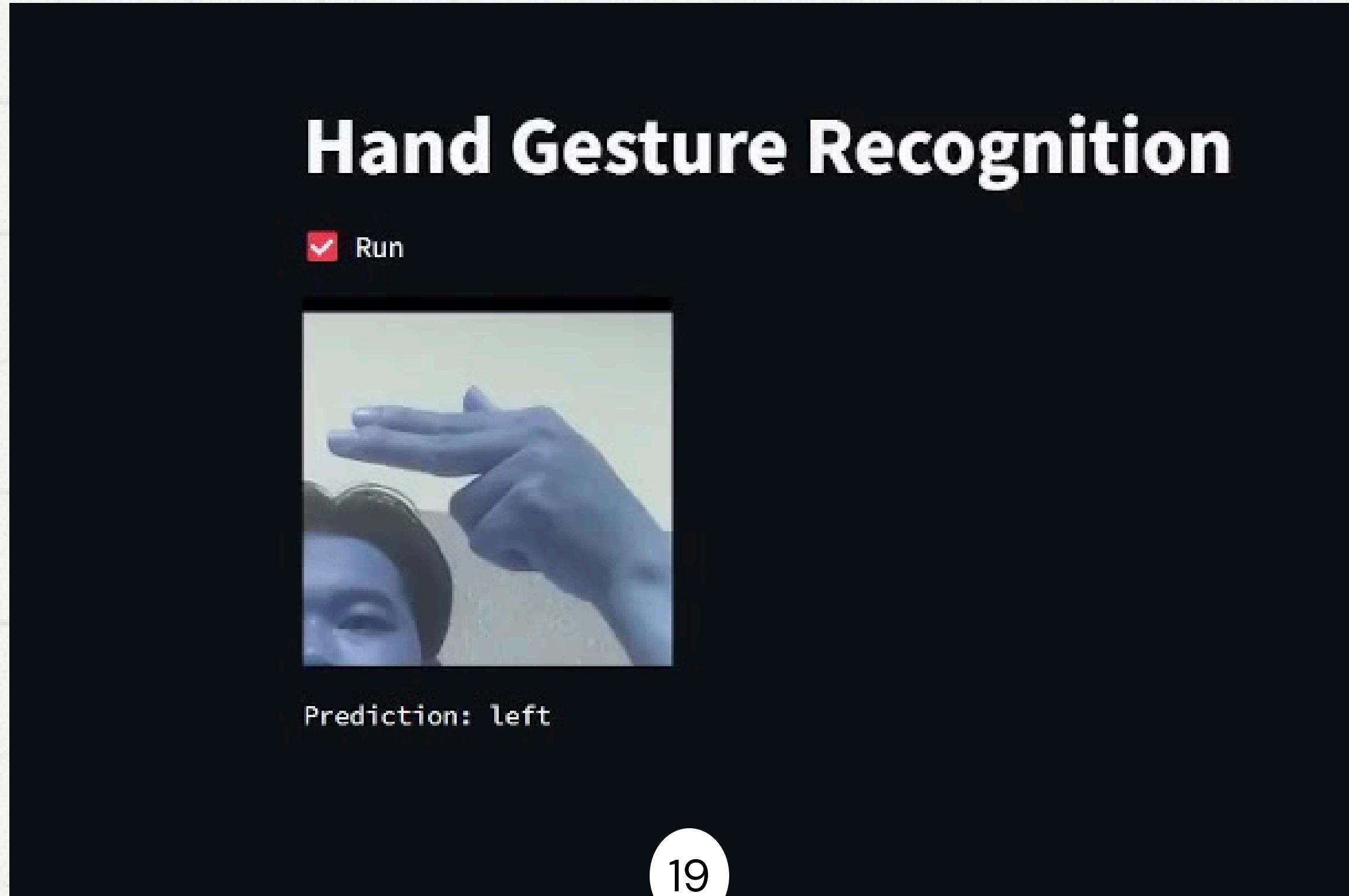
C	solver	acc	precision	recall	f1-score
0.1	lbfgs	0.97	0.91;1;1	1;0.95;0.95	0.95;0.97;0.97
1	lbfgs	0.97	0.91;1;1	1;0.95;0.95	0.95;0.97;0.97
10	lbfgs	0.95	0.88;1;1	1;0.95;0.91	0.93;0.97;0.95
100	lbfgs	0.95	0.86;1;1	1;0.95;0.89	0.93;0.97;0.94
1	saga	0.97	0.91;1;1	1;0.95;0.95	0.95;0.97;0.97

## 4.3. Kết quả thực nghiệm

Mô hình	acc	precision	recall	f1-score
SVM	0.98	0.98	0.98	0.98
RF	1	1	1	1
LR	0.97	0.97	0.97	0.97

Kết quả tổng hợp của các cấu hình tốt nhất trong 3 mô hình

## 4.4. Minh họa



# TÀI LIỆU THAM KHẢO

1. Karen Simonyan, Andrew Zisserman: Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR 2015
2. Cutler, Adele & Cutler, David & Stevens, John. (2011). Random Forests. 10.1007/978-1-4419-9326-7\_5.
3. Surowiecki, James (2004). The Wisdom of Crowds. Anchor Books. pp. xv. ISBN 978-0-385-72170-7.
4. Vapnik, Vladimir. (1995). The Nature of Statistical Learning Theory. 10.1007/978-1-4757-3264-1\_1.
5. Maalouf, Maher. (2011). Logistic regression in data analysis: An overview. International Journal of Data Analysis Techniques and Strategies. 3. 281-299. 10.1504/IJDATS.2011.041335.
6. <https://github.com/cvzone/cvzone>
7. <https://www.kaggle.com/datasets/crawford/vgg16>

# TÀI NGUYÊN

**VGG+LG:** Kaggle LR

**VGG+SVM:** Kaggle SVM

**VGG+RF:** Kaggle RF

**Thanks for  
attention**