



Phân loại chữ số trong kí hiệu tay

GVHD: TS. Mai Tiến Dũng

SINH VIÊN: Nguyễn Trần Khương An
Hồng Phúc Hải
Tăng Mỹ Hân

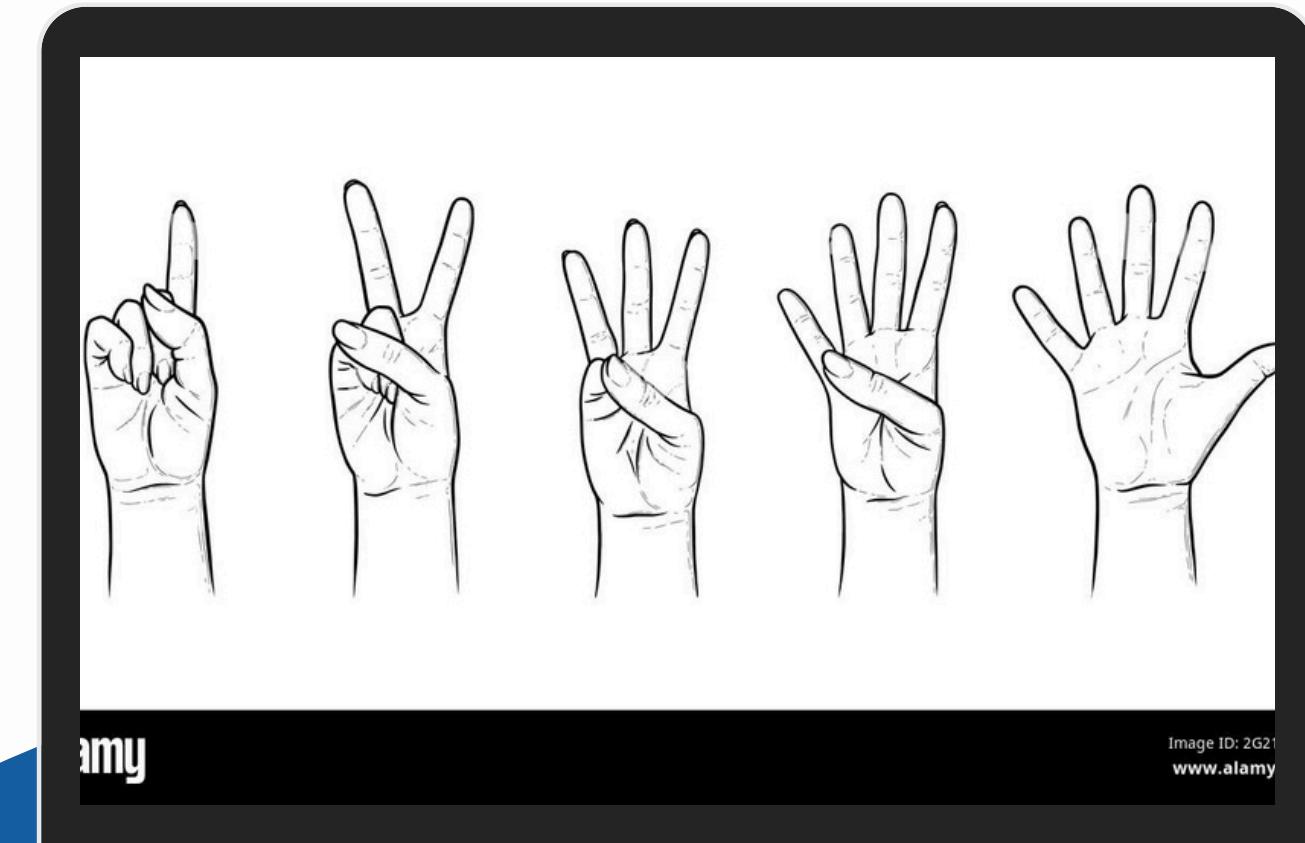


Image ID: 2G21
www.alamy.com

MỤC LỤC

- 01** **Giới thiệu**
 - Lý do chọn đề tài
 - Phát biểu bài toán
- 02** **Rút trích đặc trưng**
 - Histogram of Oriented Gradient
 - Bag of visual word
- 03** **Các mô hình phân loại**
 - SVM
 - KNN
 - Random forest
- 04** **Thực nghiệm**
 - Dataset
 - Đánh giá
- 05** **DEMO**

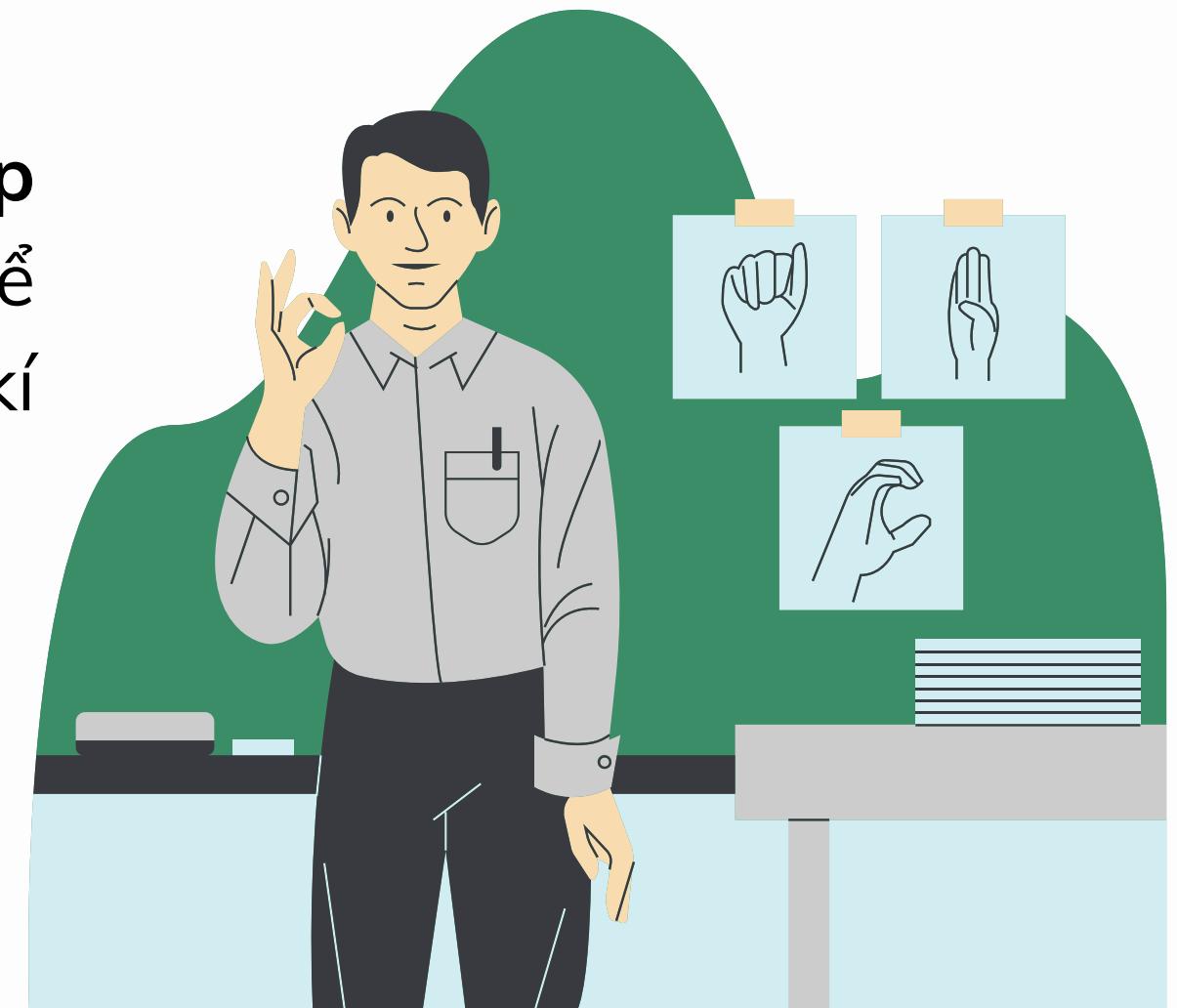
01

Giới thiệu

I. Giới thiệu

Lí do chọn đề tài

- **Hỗ trợ người khiếm thính trong giao tiếp hàng ngày :** trao đổi số điện thoại, thương thảo giá cả, tính toán,...
- **Làm nền tảng cho những nghiên cứu và ứng dụng phức tạp hơn:** các nhóm nghiên cứu có thể mở rộng dự án của mình để giải quyết các bài toán phức tạp hơn như diễn đạt ngôn ngữ ký hiệu thành câu, đoạn văn



I. Giới thiệu

Phát biểu bài toán

Input

- Bộ dữ liệu chứa các ảnh kí hiệu tay về chữ số đã được gán nhãn
- Ảnh số chưa kí hiệu tay cần dự đoán



Output

- Nhãn của ảnh cần được dự đoán

I. Giới thiệu

Ràng buộc đầu vào: mỗi ảnh đầu vào chỉ chứa một bàn tay được chụp cách màn hình một khoảng vừa đủ để giữ cho hình ảnh rõ nét.





02

Rút trích đặc trưng

II. Rút trích đặc trưng

Histogram of Oriented Gradient



Histograms of Oriented Gradients for Human Detection

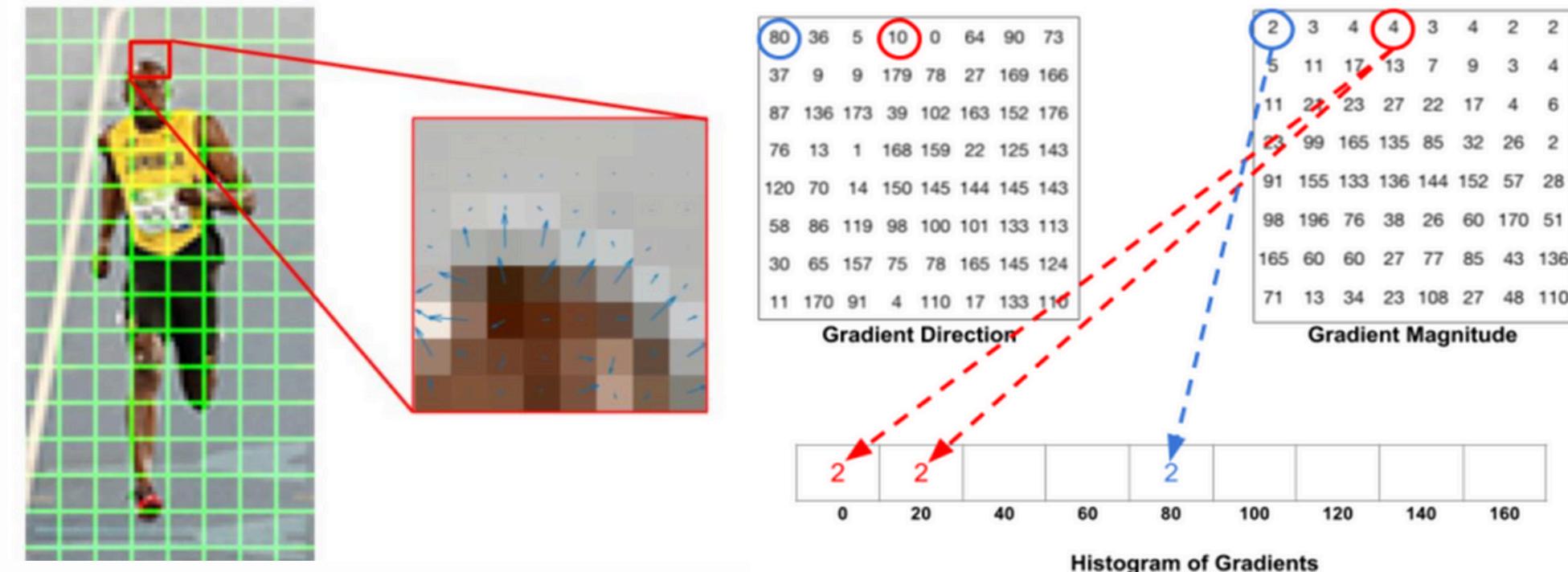
Navneet Dalal and Bill Triggs

INRIA Rhône-Alps, 655 avenue de l'Europe, Montbonnot 38334, France
{Navneet.Dalal,Bill.Triggs}@inrialpes.fr, <http://lear.inrialpes.fr>

II. Rút trích đặc trưng

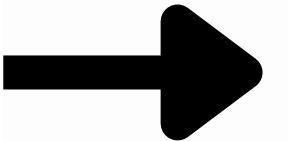
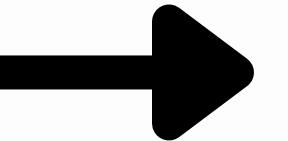
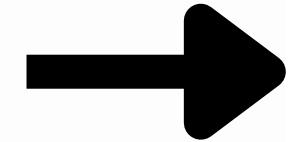
Histogram of Oriented Gradient

Ý tưởng chính của Histogram of Oriented Gradients (HOG) là trích xuất các đặc trưng cục bộ từ hình ảnh dựa trên gradient hướng của các pixel. Bằng cách phân tích các hướng gradient trong các vùng nhỏ (ô) của hình ảnh, HOG có thể nắm bắt được các thông tin quan trọng về hình dạng và cạnh của đối tượng, bất kể điều kiện ánh sáng hoặc màu sắc thay đổi.



II. Rút trích đặc trưng

Histogram of Oriented Gradient



Ảnh gốc

Ảnh resize

Ảnh Xám

HOG

IV. Thực nghiệm

HOG

```
# thu nghiem các tham so nod
orientations_list = [2, 6, 9, 12]
pixels_per_cell_list = [(4,4),(8, 8), (16, 16)]
cells_per_block_list = [(2, 2), (3, 3), (4,4)]
```

IV. Thực nghiệm

HOG



orientations=2,
pixels_per_cell=(8, 8),
cells_per_block=(4, 4)



orientations=2,
pixels_per_cell=(4, 4),
cells_per_block=(2, 2)



orientations=12,
pixels_per_cell=(8, 8),
cells_per_block=(4, 4)

II. Rút trích đặc trưng

SIFT

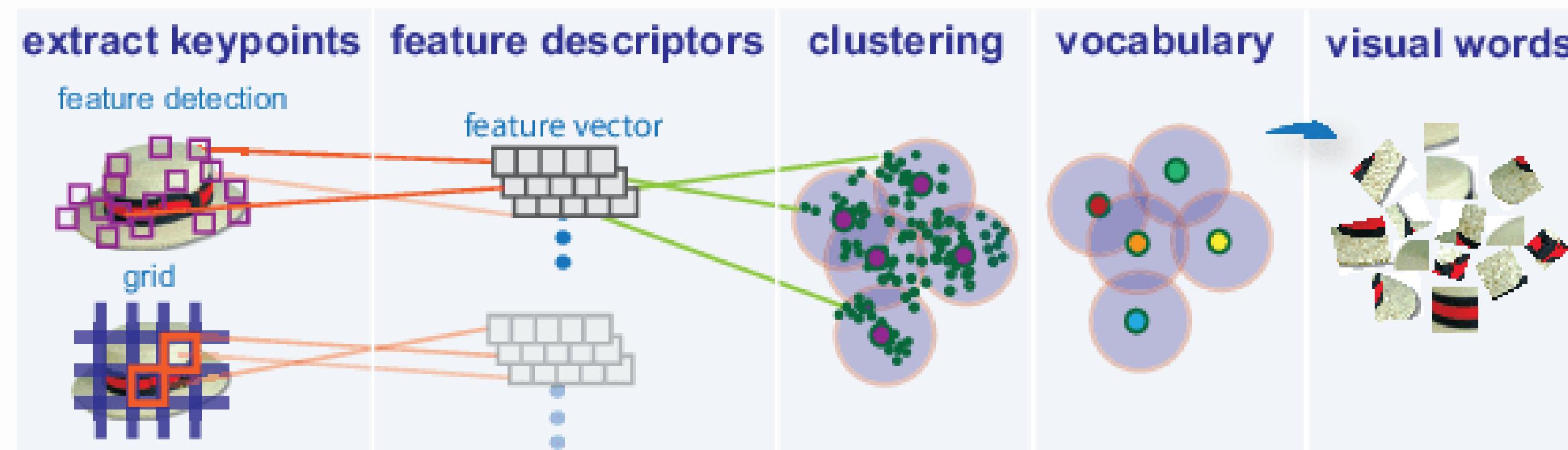
SIFT(Scale Invariant Feature Transform) là một thuật toán phát hiện đặc trưng trong thị giác máy tính để phát hiện và mô tả các đặc trưng cục bộ trong hình ảnh. Nó được xuất bản bởi **David Lowe** vào năm 1999 trong bài báo "Object recognition from local scale-invariant features"[5] và sau đó đã được tinh chỉnh vào năm 2004, mô tả chi tiết trong bài báo "Distinctive image features from scale-invariant keypoints".



II. Rút trích đặc trưng

Bag of Visual Word

Phương pháp **Bag of Visual Words (BoVW)** là một kỹ thuật phổ biến trong lĩnh vực xử lý hình ảnh và trí tuệ nhân tạo, thường được sử dụng trong các ứng dụng nhận dạng, phân loại và truy vấn hình ảnh. Phương pháp này được giới thiệu bởi **Gabriela Csurka, Christopher R. Dance, Lixin Fan, và Jutta Willamowski** trong bài báo “**Visual Categorization with Bags of Keypoints**” năm 2004 [1].

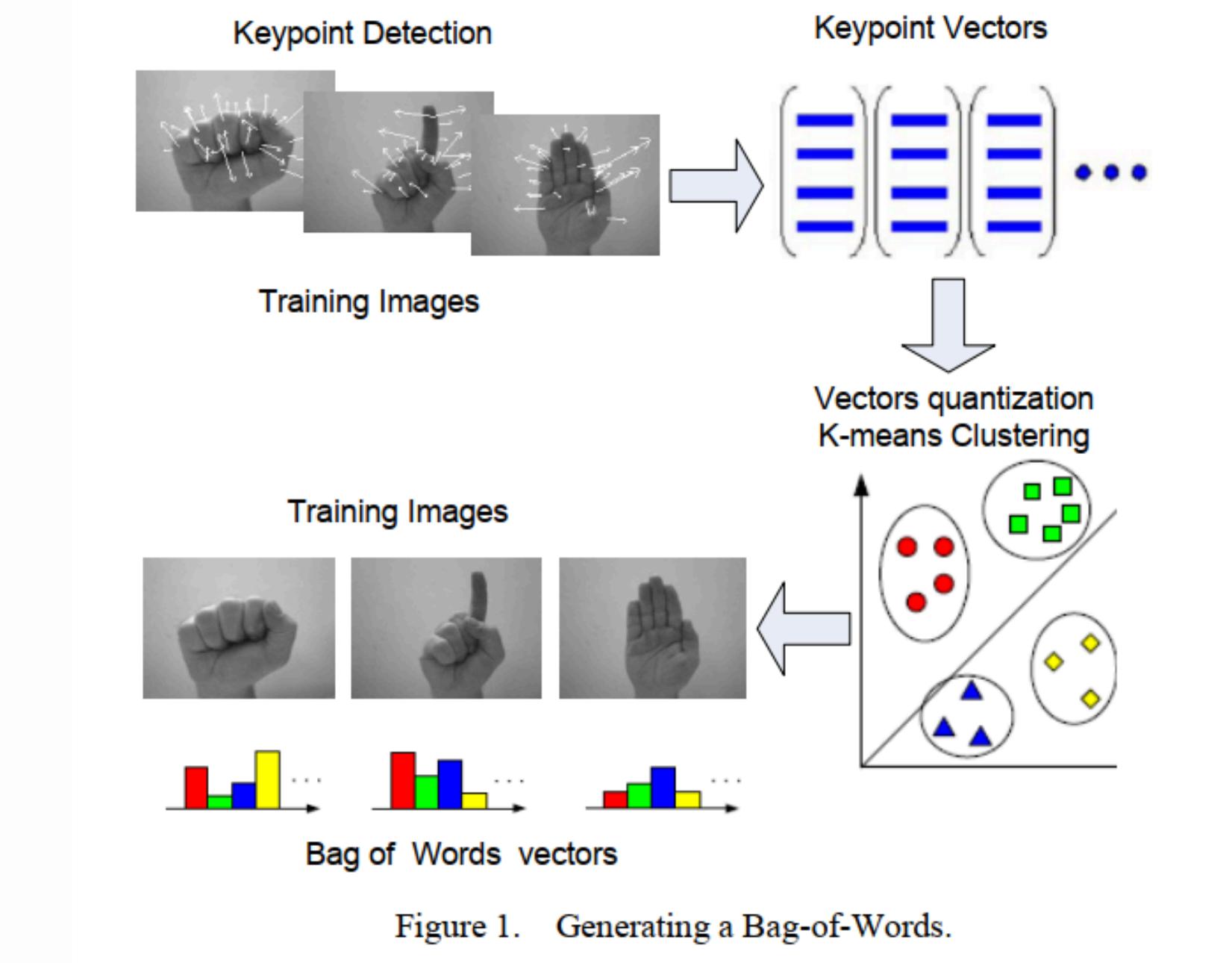


[1] Paper: Visual Categorization with Bags of Keypoints Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, Cédric Bray Xerox Research Centre Europe 6, chemin de Maupertuis 38240 Meylan, France

III. Rút trích đặc trưng

Bag of Visual Word

Ý tưởng cơ bản của BoVW được lấy cảm hứng từ kỹ thuật Bag of Words (BoW) trong xử lý ngôn ngữ tự nhiên, trong đó mỗi tài liệu được biểu diễn dưới dạng một vectơ tần số xuất hiện của các từ trong từ điển. Tương tự, trong BoVW, mỗi hình ảnh được biểu diễn dưới dạng một vectơ tần số xuất hiện của các "từ" hình ảnh, gọi là "visual words"



03

Các mô hình phân loại

II. Các mô hình phân loại

Support Vector Machines



Machine Learning, 20, 273–297 (1995)

© 1995 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

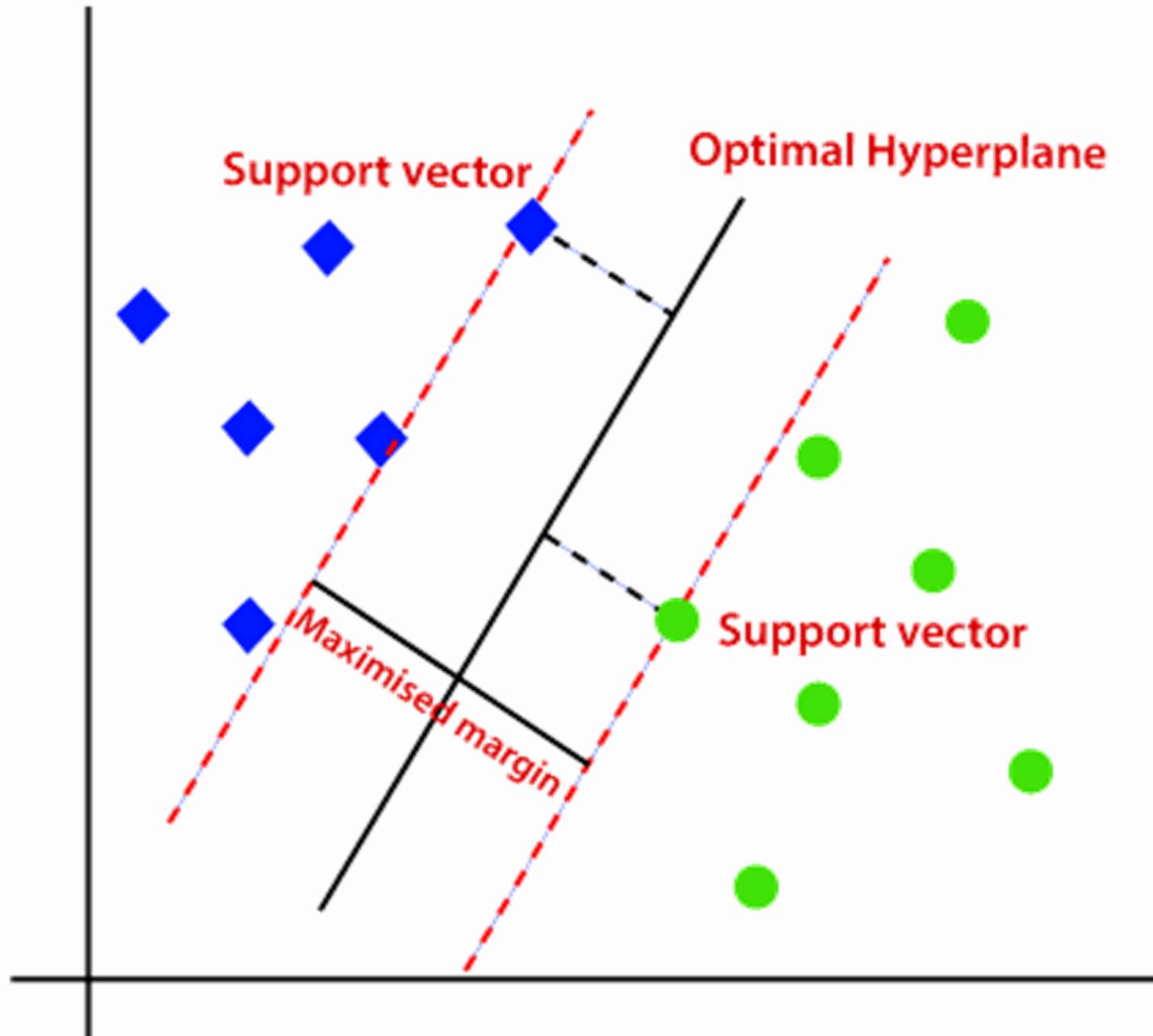
Support-Vector Networks

CORINNA CORTES
VLADIMIR VAPNIK
AT&T Bell Labs., Holmdel, NJ 07733, USA

corinna@neural.att.com
vlad@neural.att.com

II. Các mô hình phân loại

Support Vector Machines

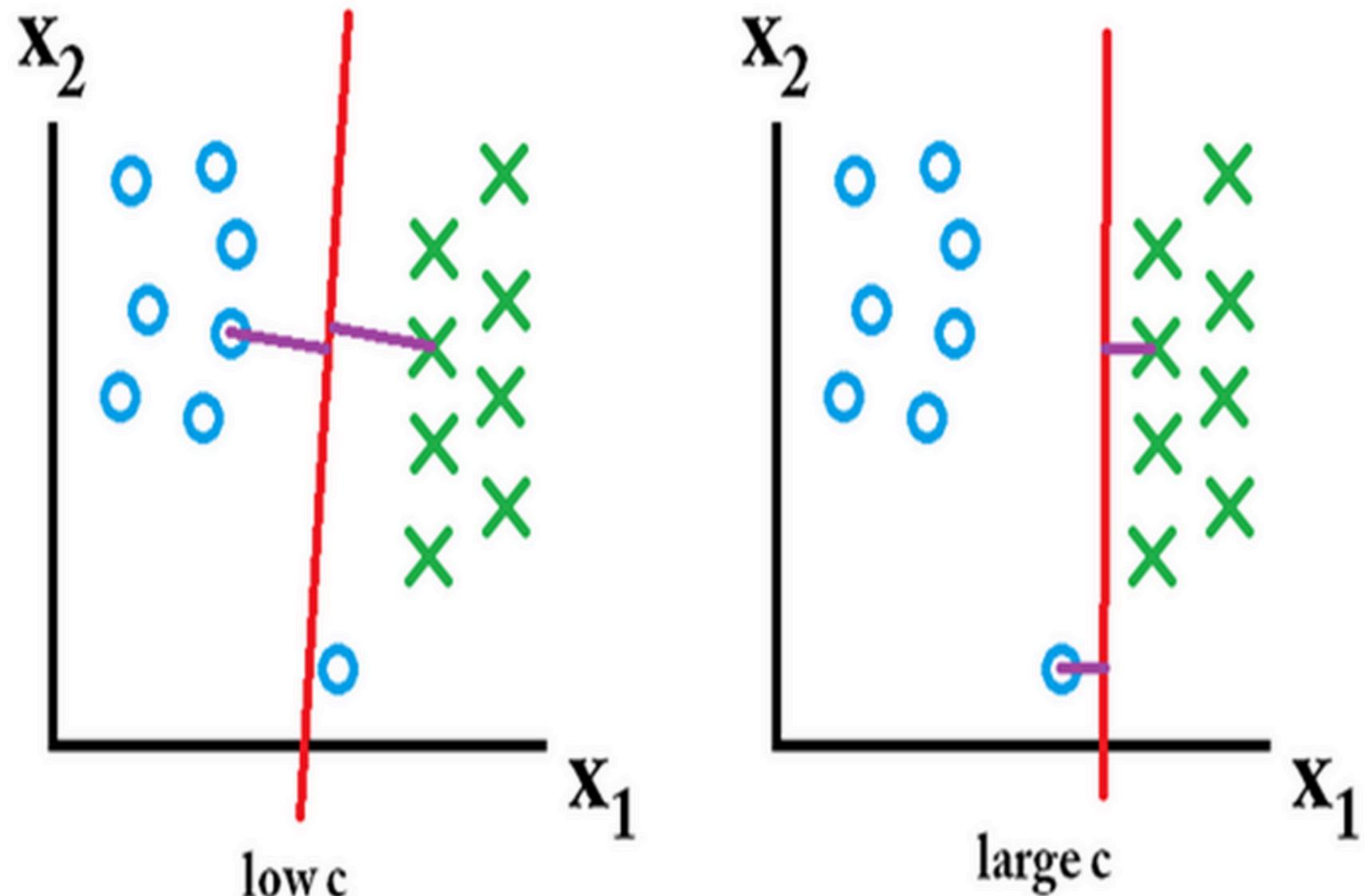


Tìm một đường ranh giới (**hyperplane**) tốt nhất chia hai lớp đó sao cho khoảng cách từ các điểm dữ liệu gần nhất tới đường này là lớn nhất.

II. Các mô hình phân loại

Support Vector Machines

Tham số C

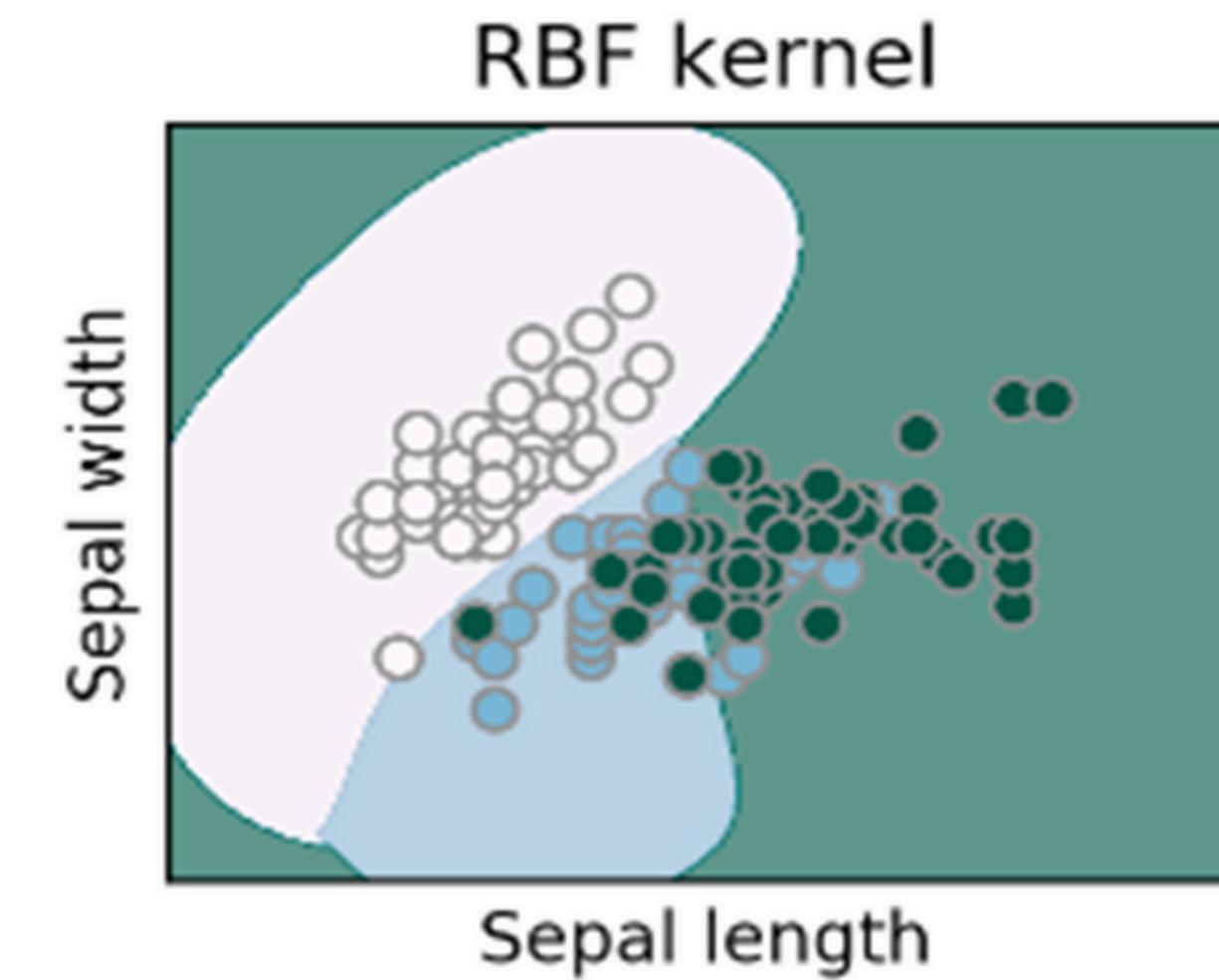
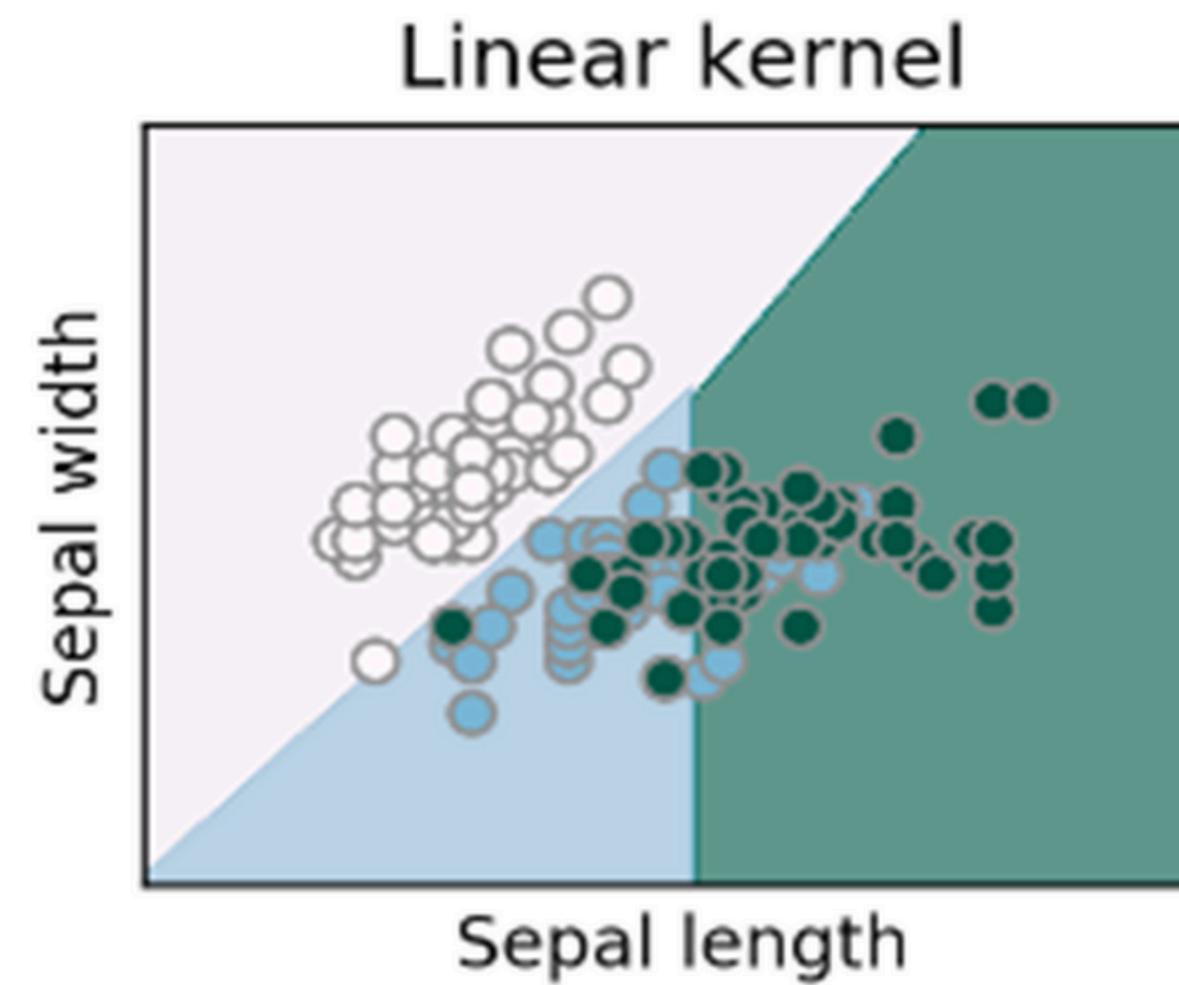


- Xác định sự cân bằng giữa việc đạt được low training error và low testing error.
- Kiểm soát mức độ phân loại sai có thể được chấp nhận trong tập huấn luyện
 - Khi C tăng, mô hình sẽ cố gắng tối đa hóa việc phân loại đúng các điểm dữ liệu huấn luyện
 - Khi C nhỏ, mô hình sẽ chấp nhận một số lượng lớn hơn các điểm dữ liệu bị vi phạm mềm

II. Các mô hình phân loại

Support Vector Machines

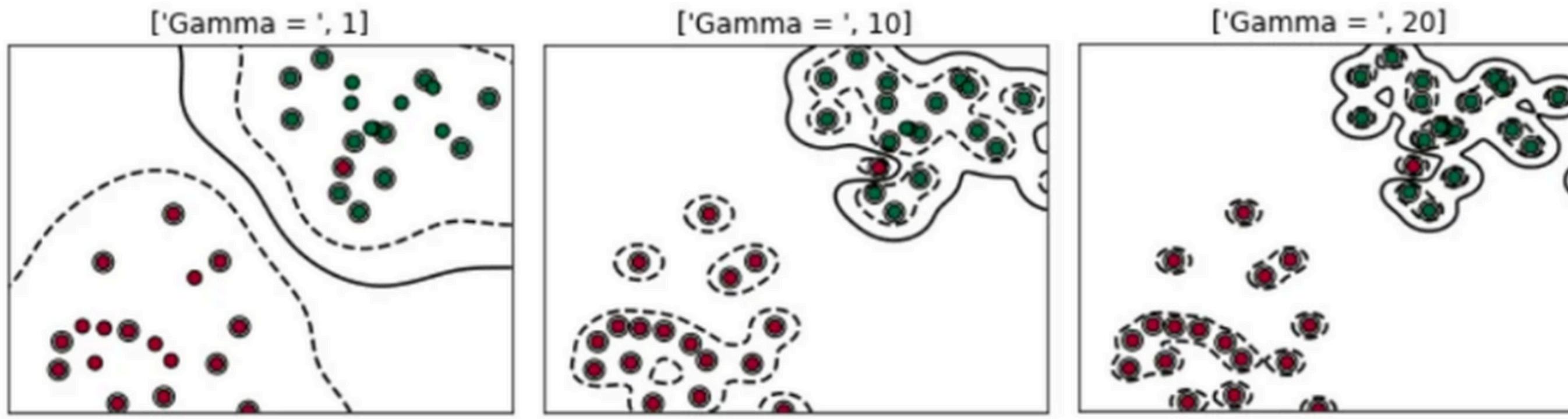
kernel



II. Các mô hình phân loại

Support Vector Machines

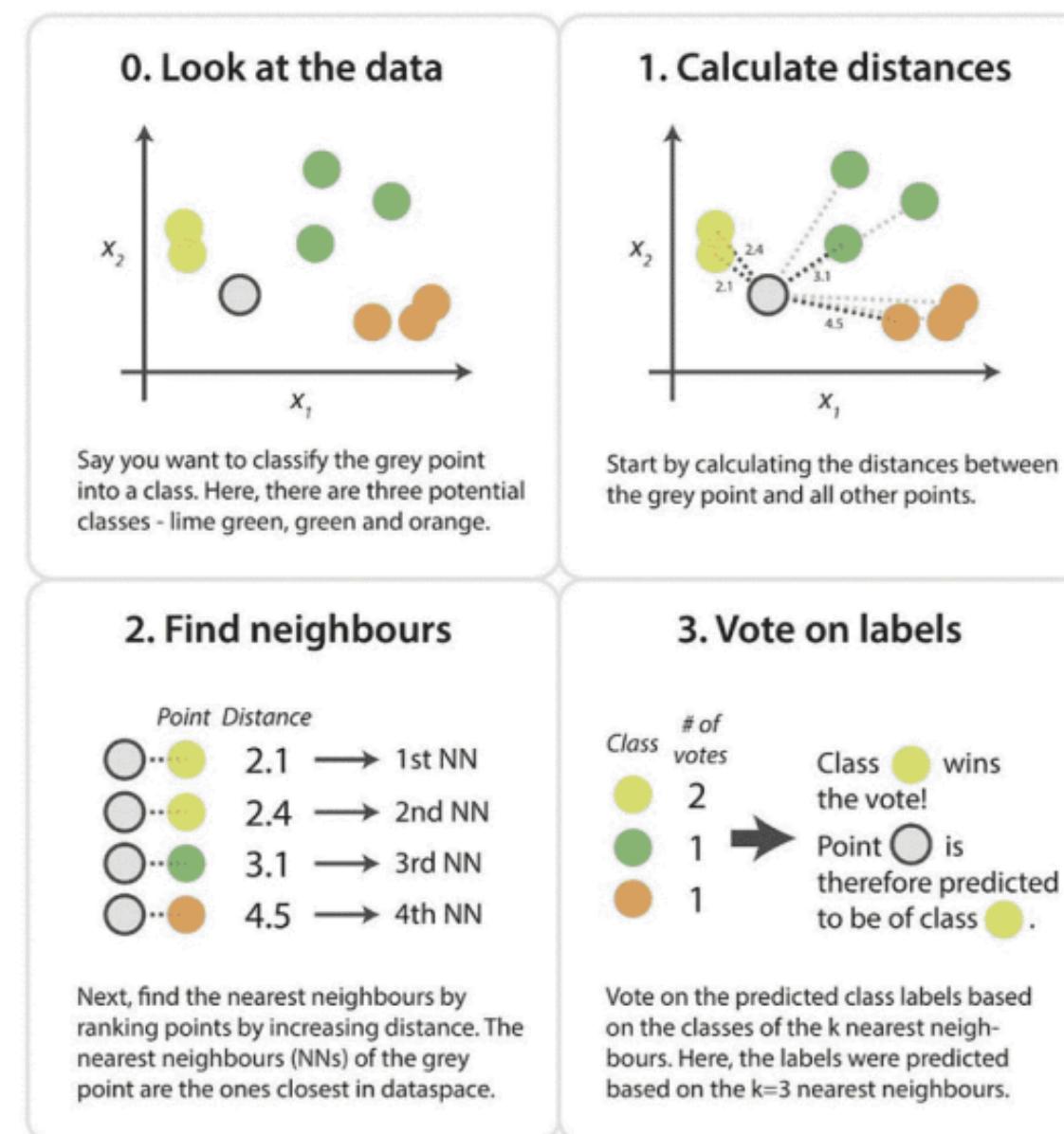
Tham số gamma



Tham số kiểm soát hình dạng của ranh giới quyết định

II. Các mô hình phân loại

k-Nearest Neighbors (KNN)



Thuật toán k-Nearest Neighbor (kNN) là một phương pháp học có giám sát phi tham số thường dùng cho bài toán phân loại và hồi quy do **Evelyn Fix và Joseph Hodges** phát triển lần đầu vào năm **1951**, và sau đó được cải tiến bởi Thomas Cover.

Các bước tính toán:

- Tính toán khoảng cách từ điểm dữ liệu mới đến tất cả các điểm dữ liệu trong tập huấn luyện.
- Chọn K điểm dữ liệu gần nhất (theo khoảng cách) đến điểm dữ liệu mới.
- Gán nhãn cho điểm dữ liệu mới dựa trên nhãn chiếm đa số hoặc trọng số của K điểm dữ liệu gần nhất.

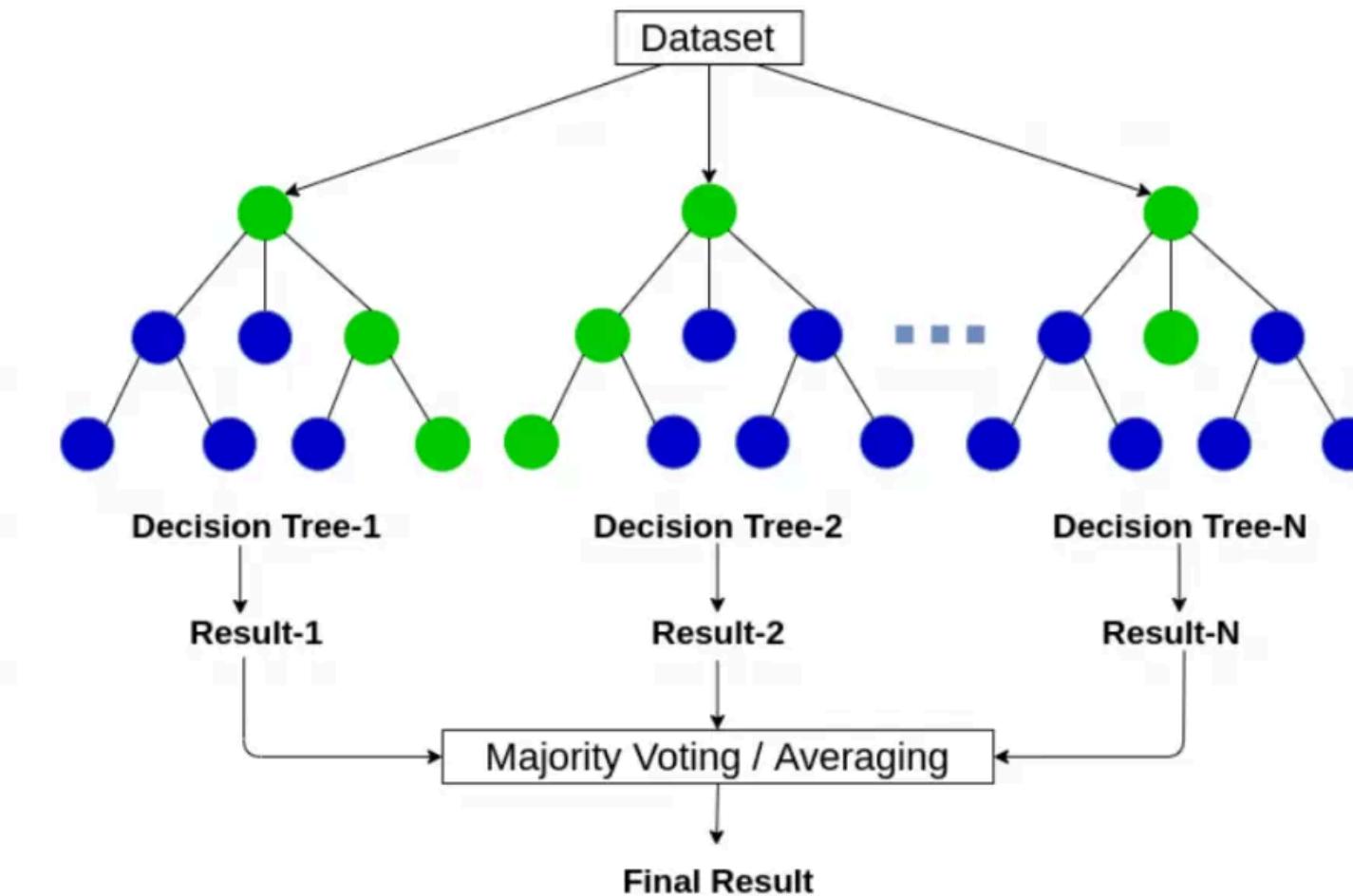
II. Các mô hình phân loại

Random forest

Random Forest là thuật toán học có giám sát thuộc vào nhóm mô hình học kết hợp (Ensemble Learning) được đề xuất bởi Leo Breiman và Adele Cutler vào năm 2001.

Ý tưởng của thuật toán Random Forest cho phân loại dữ liệu:

- Chọn các mẫu ngẫu nhiên từ tập dữ liệu đã cho.
- Xây dựng cây quyết định trên từng tập dữ liệu con.
- Kết hợp các cây: sử dụng chiến lược bình chọn theo số đông với bài toán phân loại hoặc lấy trung bình các giá trị dự đoán từ các cây với bài toán hồi quy.



```
def train(X_train, y_train, k, model_save_path, model_name,clf):
    des_train, kmean = data_preprocessing(X_train, k=k)
    if clf == 'svm':
        classifier = SVC()
        param_grid = {
            'C': [0.001, 0.01, 0.1, 1, 10, 100], # Regularization parameter
            'kernel': ['rbf', 'linear'],
            'gamma':[0.001, 0.01, 0.1, 1, 10, 100] # Kernel coefficient for 'rbf'
        }
    elif clf == 'knn':
        classifier = KNeighborsClassifier()
        param_grid = {'n_neighbors': range(1, 31)}
    elif clf == 'rf':
        classifier = RandomForestClassifier()
        param_grid = {'n_estimators': range(50, 150, 10)}
    else:
        raise ValueError("Invalid classifier specified: {}".format(clf))
    return classifier, param_grid
```



04

Thực nghiệm

IV. Thực nghiệm

Dataset

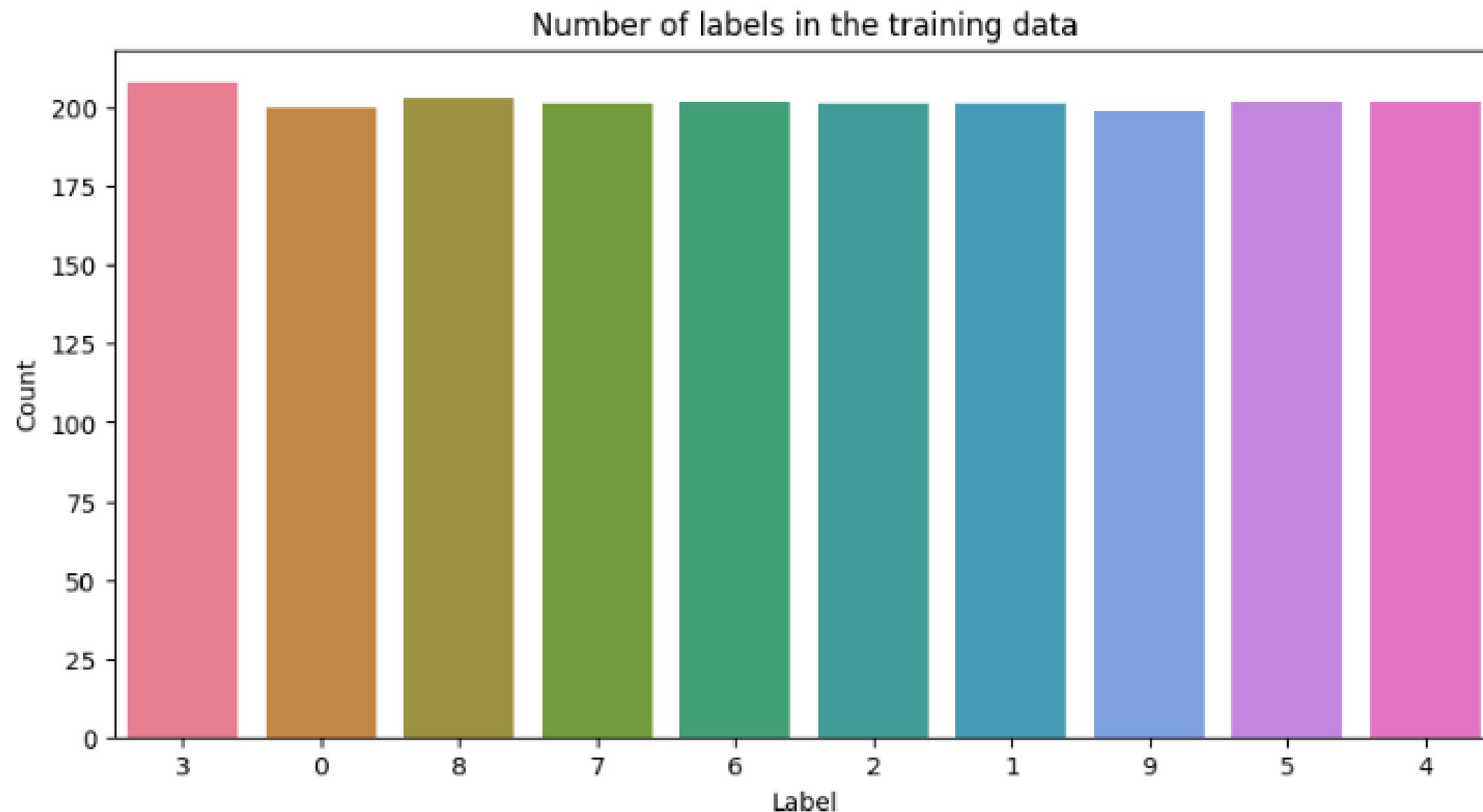
A New Dataset and Proposed Convolutional Neural Network Architecture
for Classification of American Sign Language Digits



(*) Dataset được đăng trên kaggle : [Link](#)

IV. Thực nghiệm

Dataset



Chi tiết về bộ dữ liệu:

Kích thước ảnh: 100 x 100 pixel

Không gian màu: RGB

-Số lượng lớp: 10 (Các chữ số từ 0 đến 9)

-Số lượng học sinh tham gia: 218

-Số mẫu mỗi học sinh: 10

Thực hiện bởi:

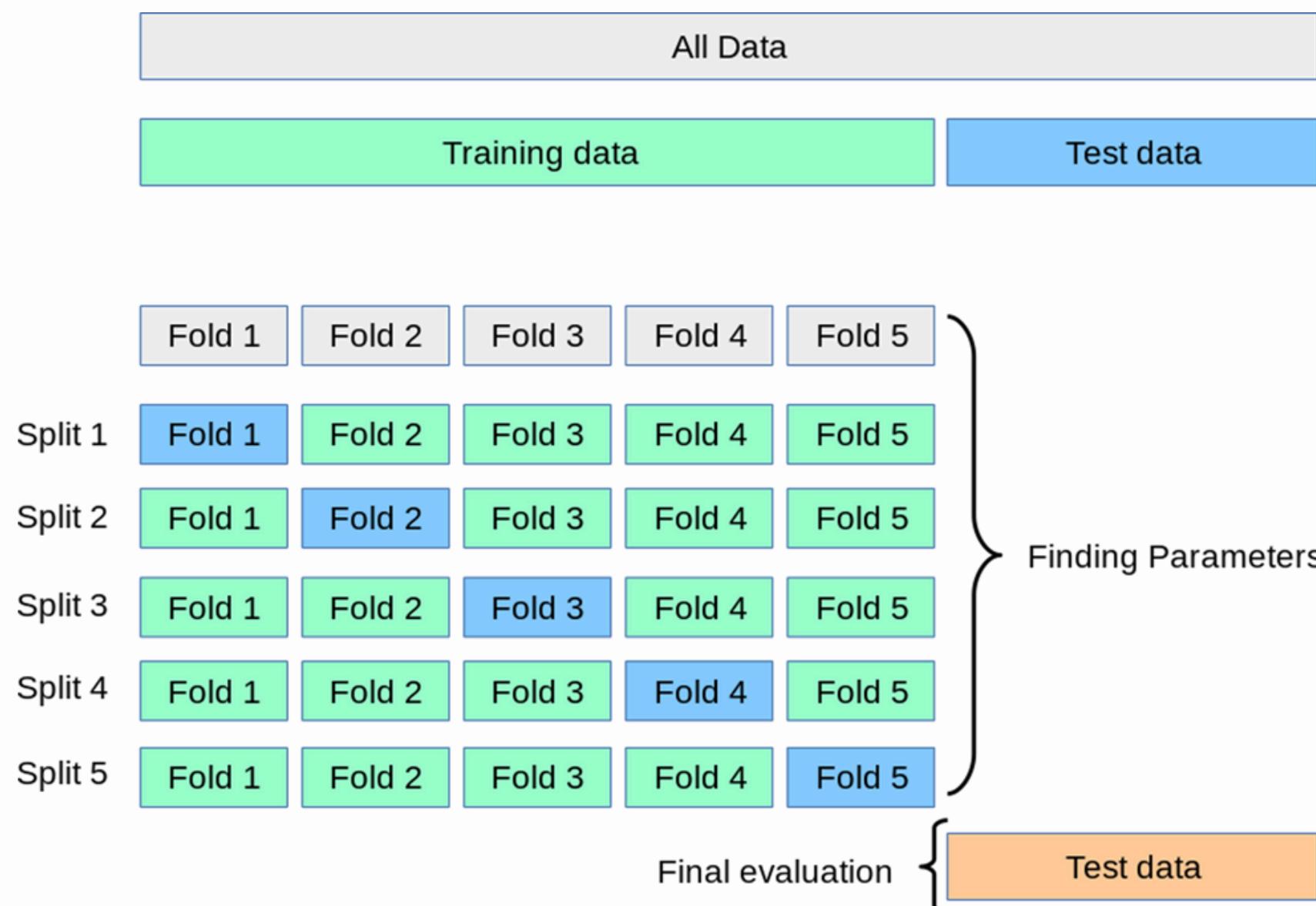
Bộ dữ liệu này được chuẩn bị bởi các học sinh của Trường Trung học Anadolu Ayrancı, Ankara, Thổ Nhĩ Kỳ.

Link GitHub: [Link](#)

IV. Thực nghiệm

Độ đo đánh giá

Sử dụng 5-fold Cross-Validation



TRAIN

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

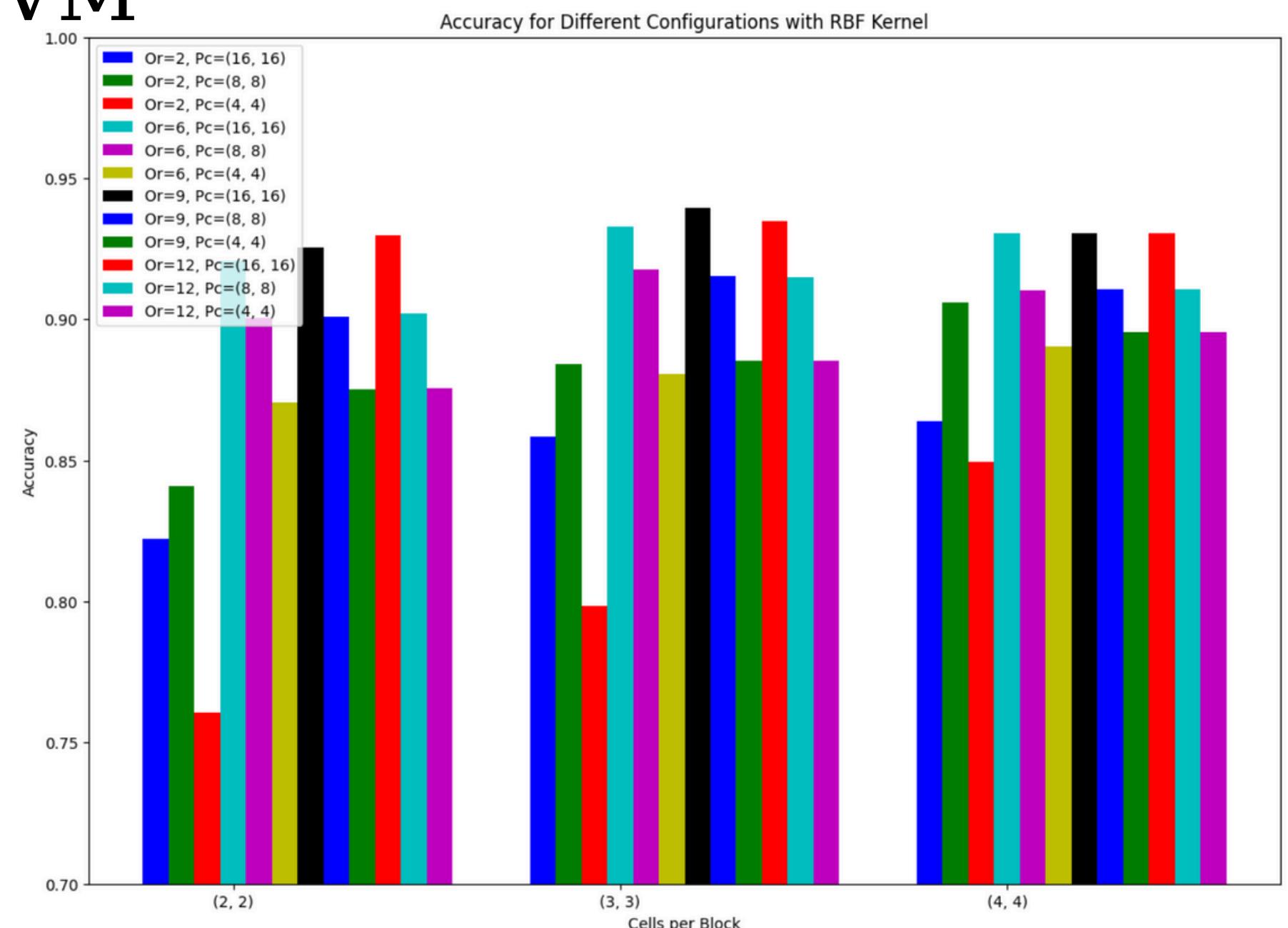
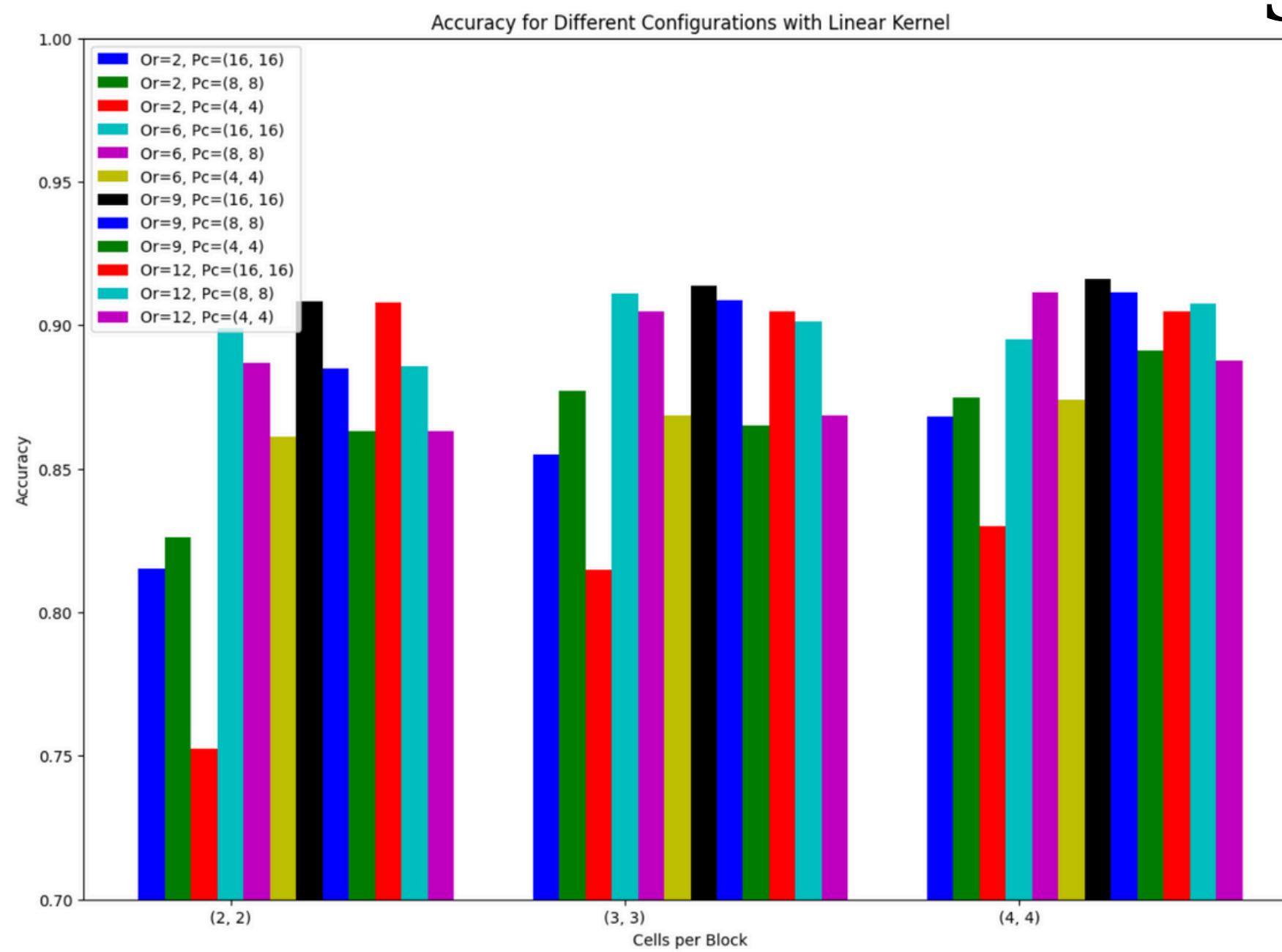
Test

	precision	recall	f1-score	support
0	0.77	0.86	0.81	37584
1	0.84	0.75	0.79	37577
accuracy				0.80
macro avg	0.81	0.80	0.80	75161
weighted avg	0.81	0.80	0.80	75161

Histogram of Oriented Gradient

IV. Thực nghiệm

HOG SVM

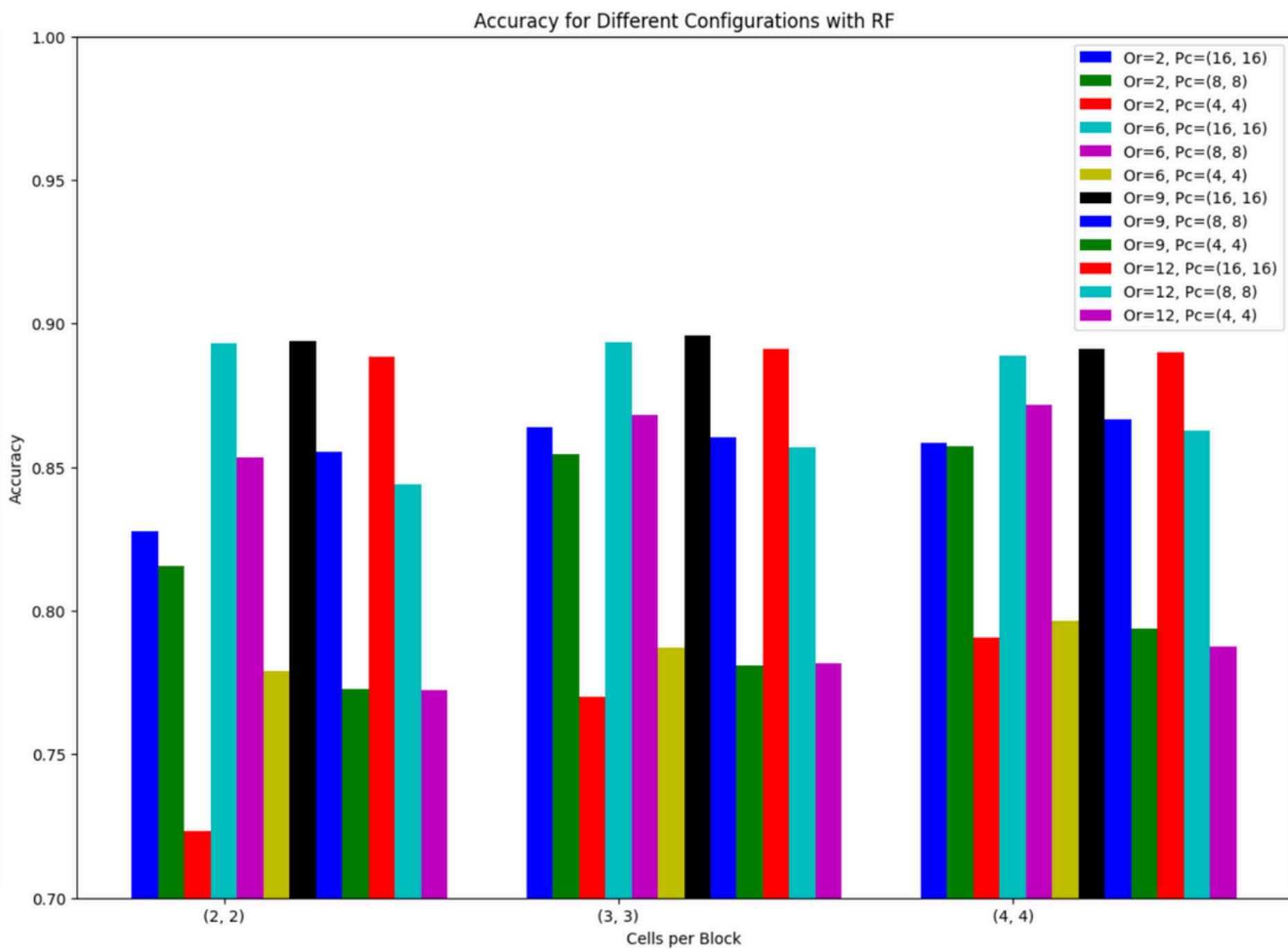
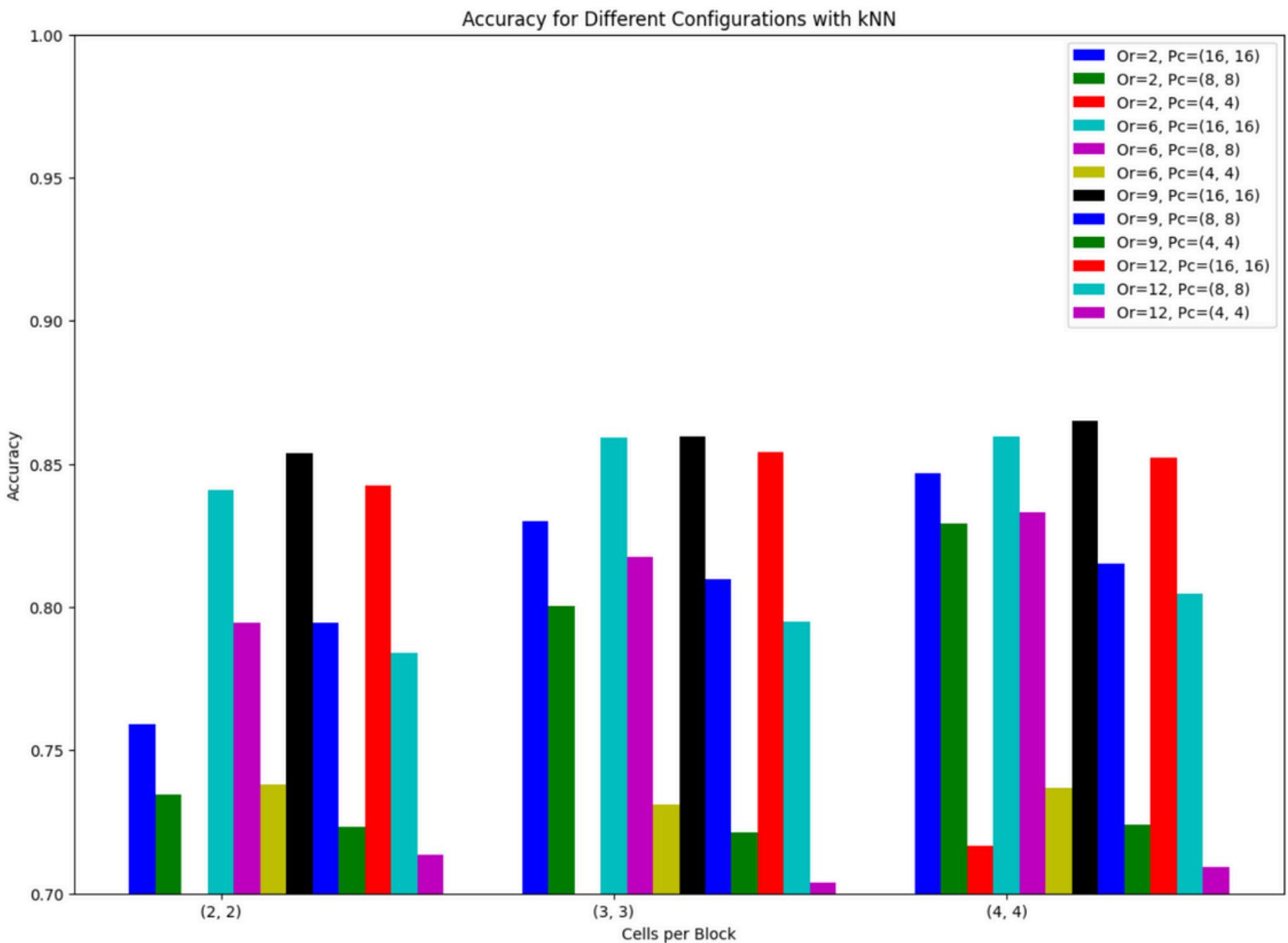


IV. Thực nghiệm

KNN

HOG

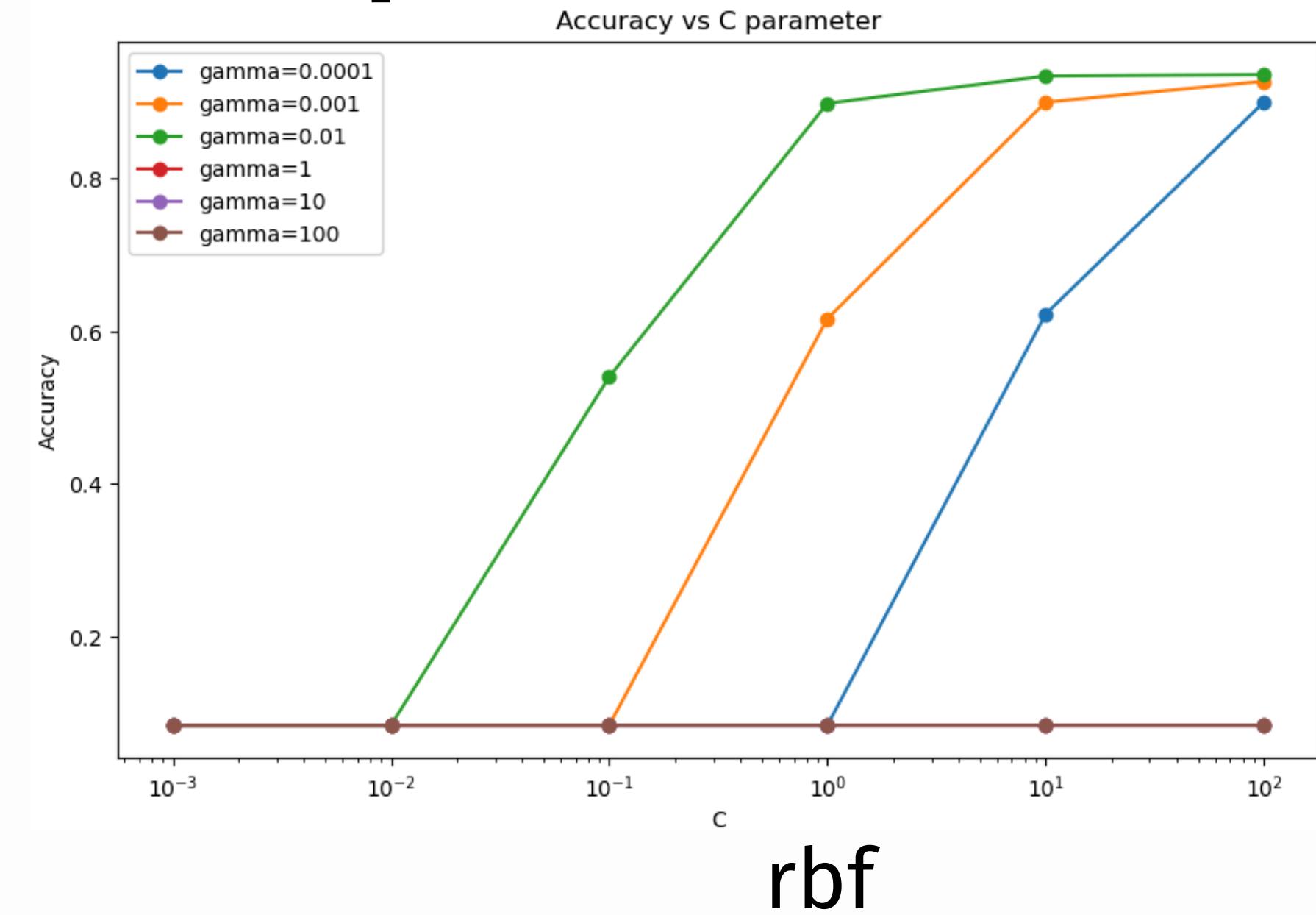
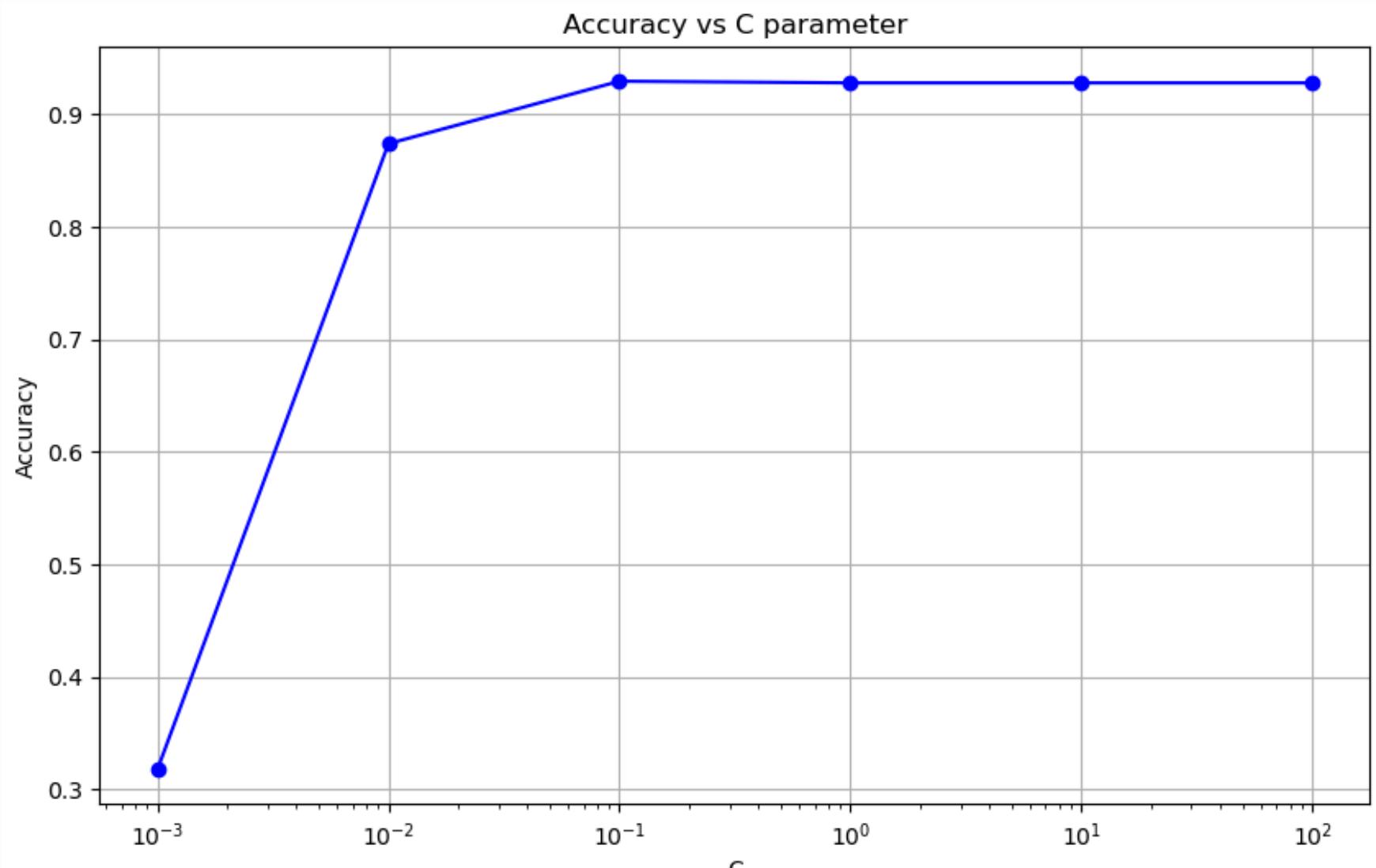
RF



IV. Thực nghiệm

SVM

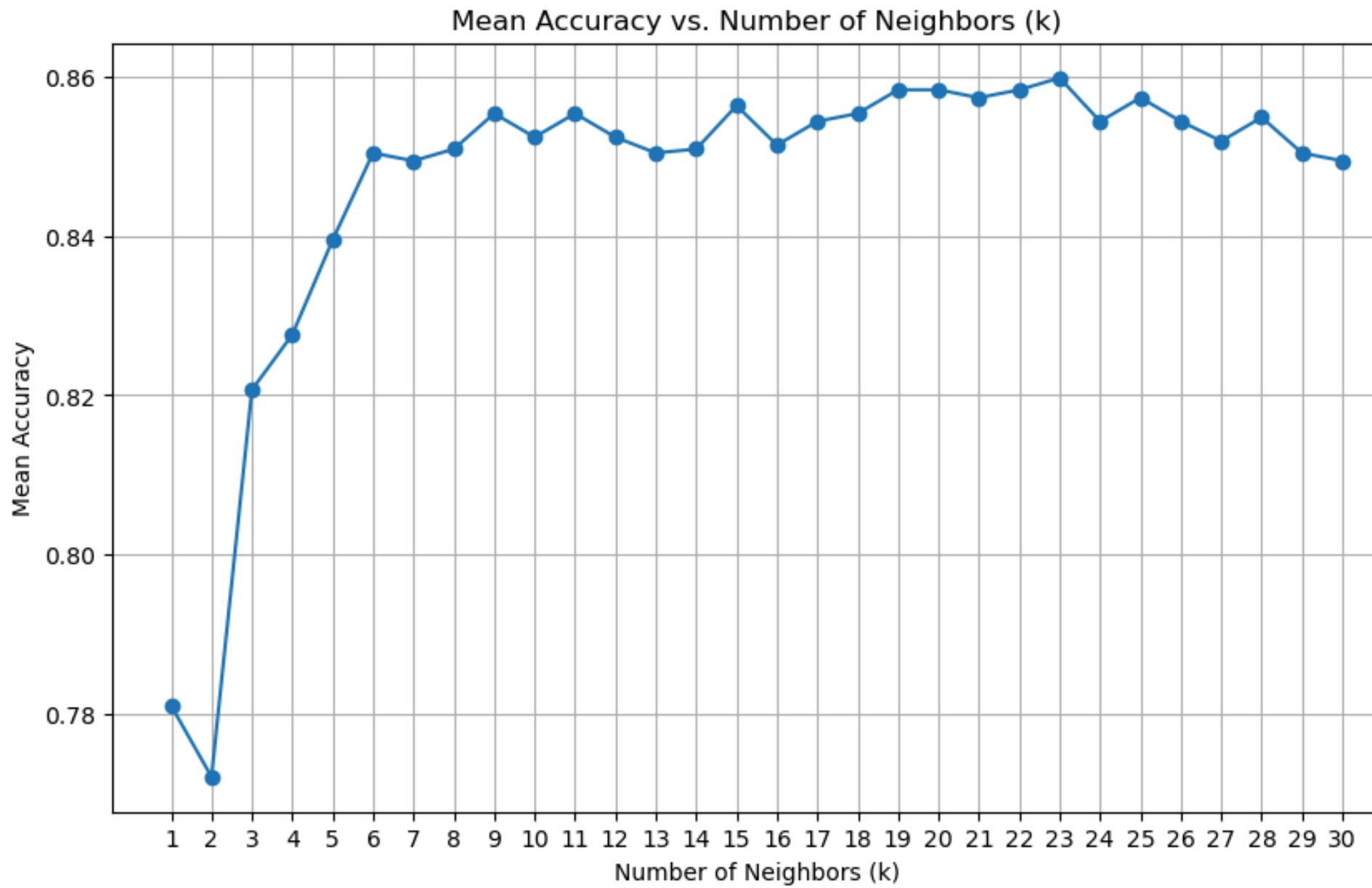
HOG(ori=9,ppc=(16,16),cpb=(3,3)



IV. Thực nghiệm

kNN

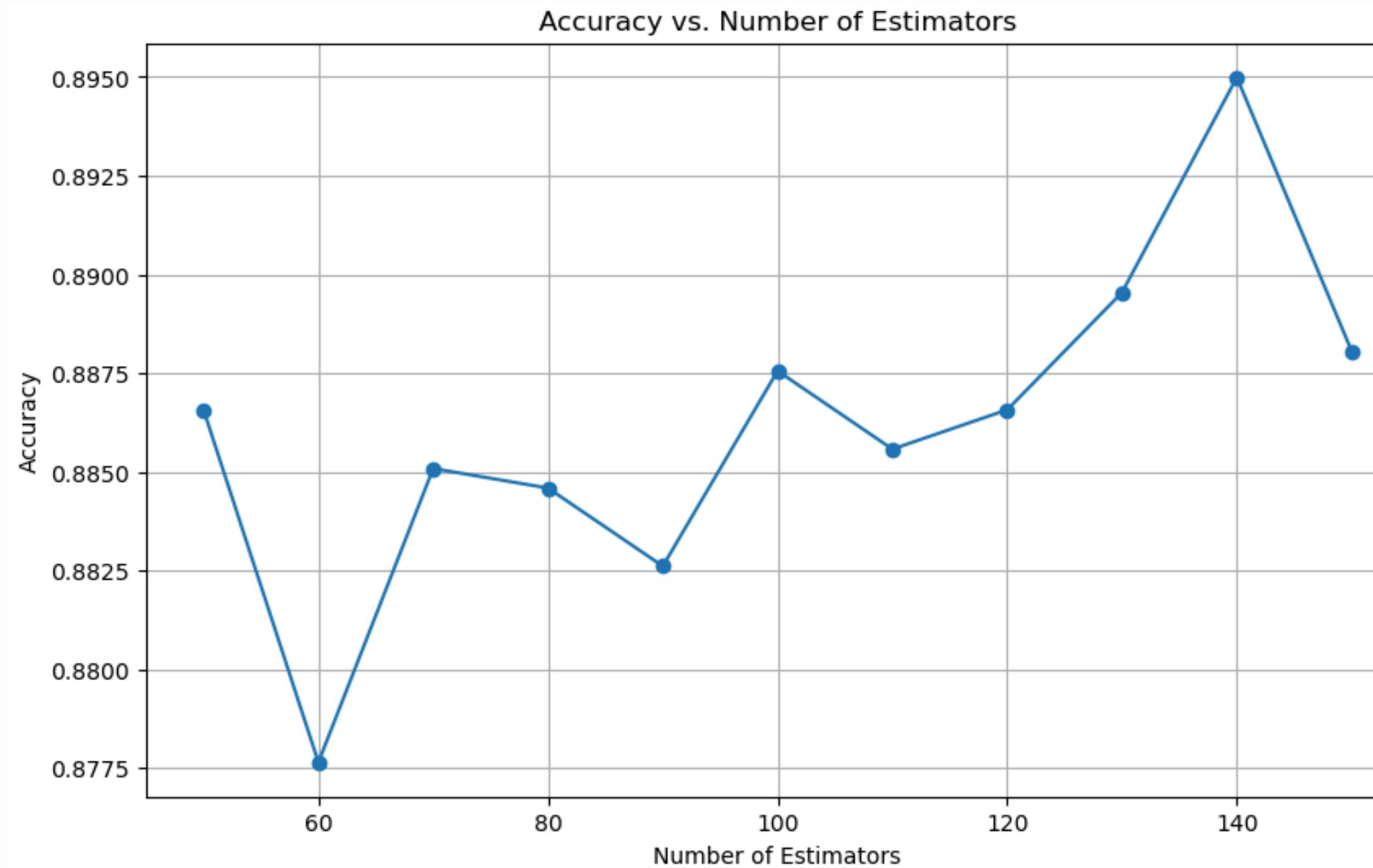
HOG(ori=9,ppc=(16,16),cpb=(3,3)



IV. Thực nghiệm

Random Forest

HOG(ori=9,ppc=(16,16),cpb=(3,3)



IV. Thực nghiệm

Kết quả trên Test

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5
1	1.00	0.89	0.89	5
2	0.83	1.00	0.91	5
3	1.00	0.89	0.89	5
4	0.62	1.00	0.77	5
5	1.00	1.00	1.00	5
6	0.80	0.89	0.89	5
7	1.00	0.49	0.57	5
8	0.80	0.89	0.89	5
9	0.83	1.00	0.91	5
accuracy			0.86	50
macro avg	0.89	0.86	0.85	50
weighted avg	0.89	0.86	0.85	50

KNN
(k =23)

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5
1	1.00	1.00	1.00	5
2	1.00	1.00	1.00	5
3	0.80	0.80	0.80	5
4	0.50	0.60	0.55	5
5	1.00	0.80	0.89	5
6	0.80	0.80	0.80	5
7	0.75	0.60	0.67	5
8	1.00	1.00	1.00	5
9	0.83	1.00	0.91	5
accuracy			0.86	50
macro avg	0.87	0.86	0.86	50
weighted avg	0.87	0.86	0.86	50

Random Forest
(n = 140)

IV. Thực nghiệm

Kết quả trên Test

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5
1	1.00	1.00	1.00	5
2	1.00	1.00	1.00	5
3	1.00	1.00	1.00	5
4	0.67	0.80	0.73	5
5	1.00	1.00	1.00	5
6	0.80	0.80	0.80	5
7	0.75	0.60	0.67	5
8	0.80	0.80	0.80	5
9	1.00	1.00	1.00	5
accuracy			0.90	50
macro avg	0.90	0.90	0.90	50
weighted avg	0.90	0.90	0.90	50

SVM
(C=0.1, kernel ='linear')

	Classification Report for Best Model:			
	precision	recall	f1-score	support
0	1.00	1.00	1.00	5
1	1.00	1.00	1.00	5
2	1.00	1.00	1.00	5
3	1.00	1.00	1.00	5
4	0.57	0.80	0.67	5
5	1.00	1.00	1.00	5
6	0.80	0.80	0.80	5
7	0.67	0.40	0.50	5
8	0.80	0.80	0.80	5
9	1.00	1.00	1.00	5
accuracy			0.88	50
macro avg	0.88	0.88	0.88	50
weighted avg	0.88	0.88	0.88	50

SVM
(C=100, gamma =0.01, kernel ='rbf')

BAG OF VISUAL WORD

IV. Thực nghiệm

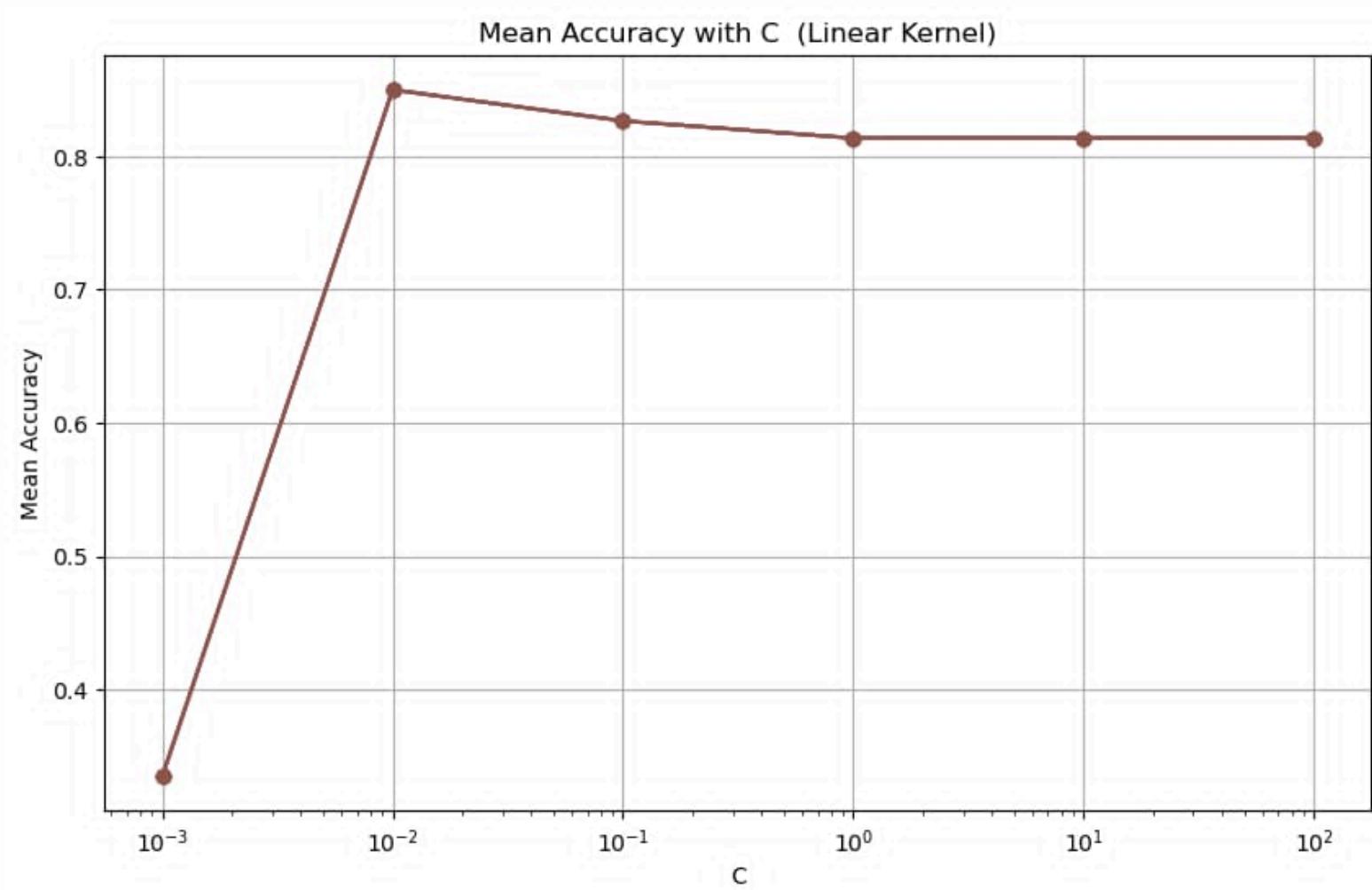
KMean BOVW

Accuracy using best_param in Train									
Số cụm	Model		SVM		RF		KNN		Average
	Accuracy	Tham số	Accuracy	Tham số	Accuracy	Tham số	Accuracy	Tham số	
150	0,8086	1,0.01,rbf	0,7887	120	0,5939	28	0,7304		
200	0,8265	10,0.001, rbf	0,8022	130	0,5825	20	0,737067		
250	0,8568	1,0.01,rbf	0,8315	120	0,5805	27	0,756267		
300	0,8633	10,0.001,rbf	0,824	150	0,5169	15	0,734733		
350	0,8678	0.1,0.001, linear	0,8225	110	0,4935	21	0,727933		
400	0,8772	10,0.001,rbf	0,823	140	0,4711	9	0,723767		

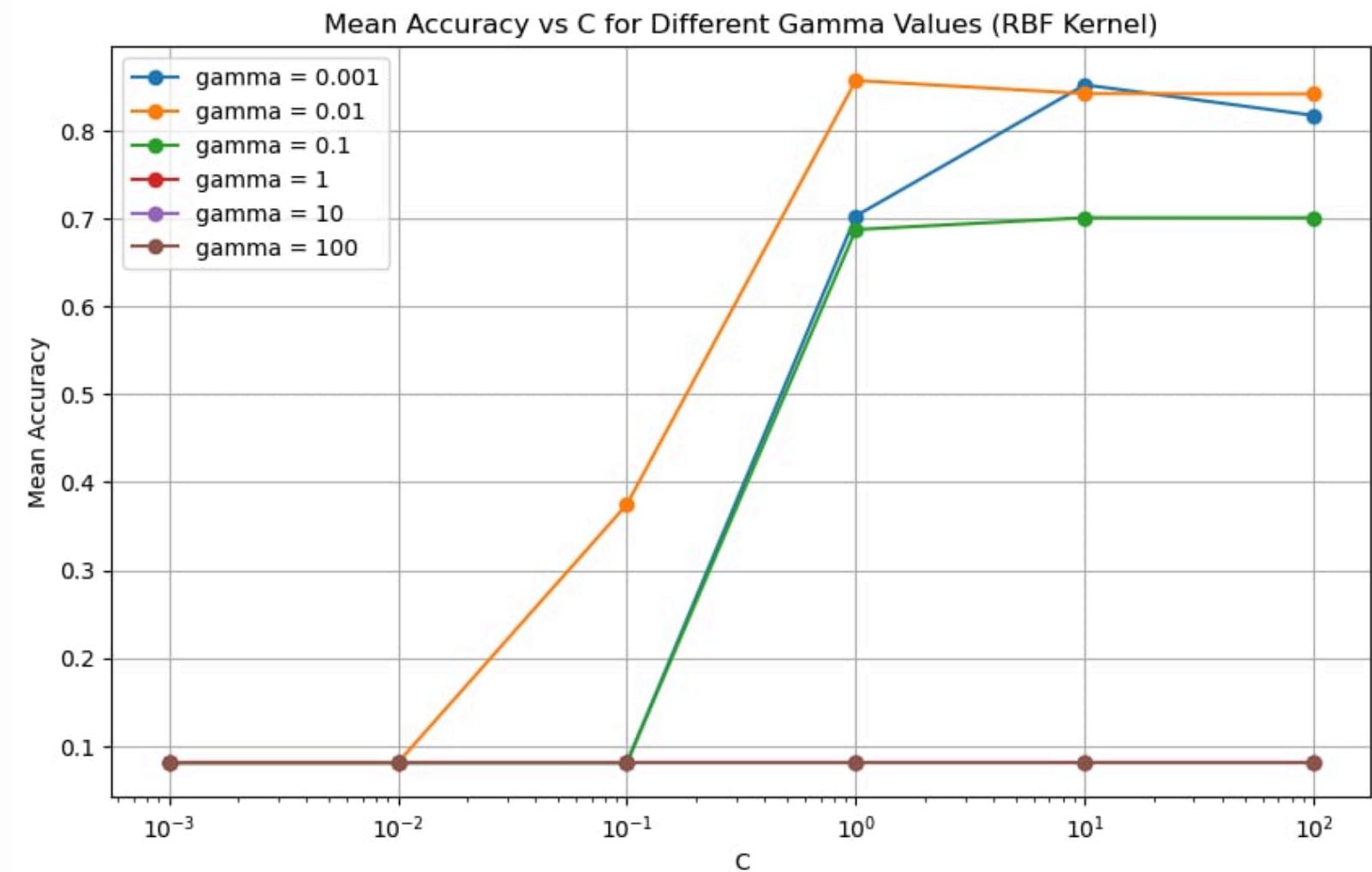
IV. Thực nghiệm

SVM

BOVW(KMean K = 250)



Linear

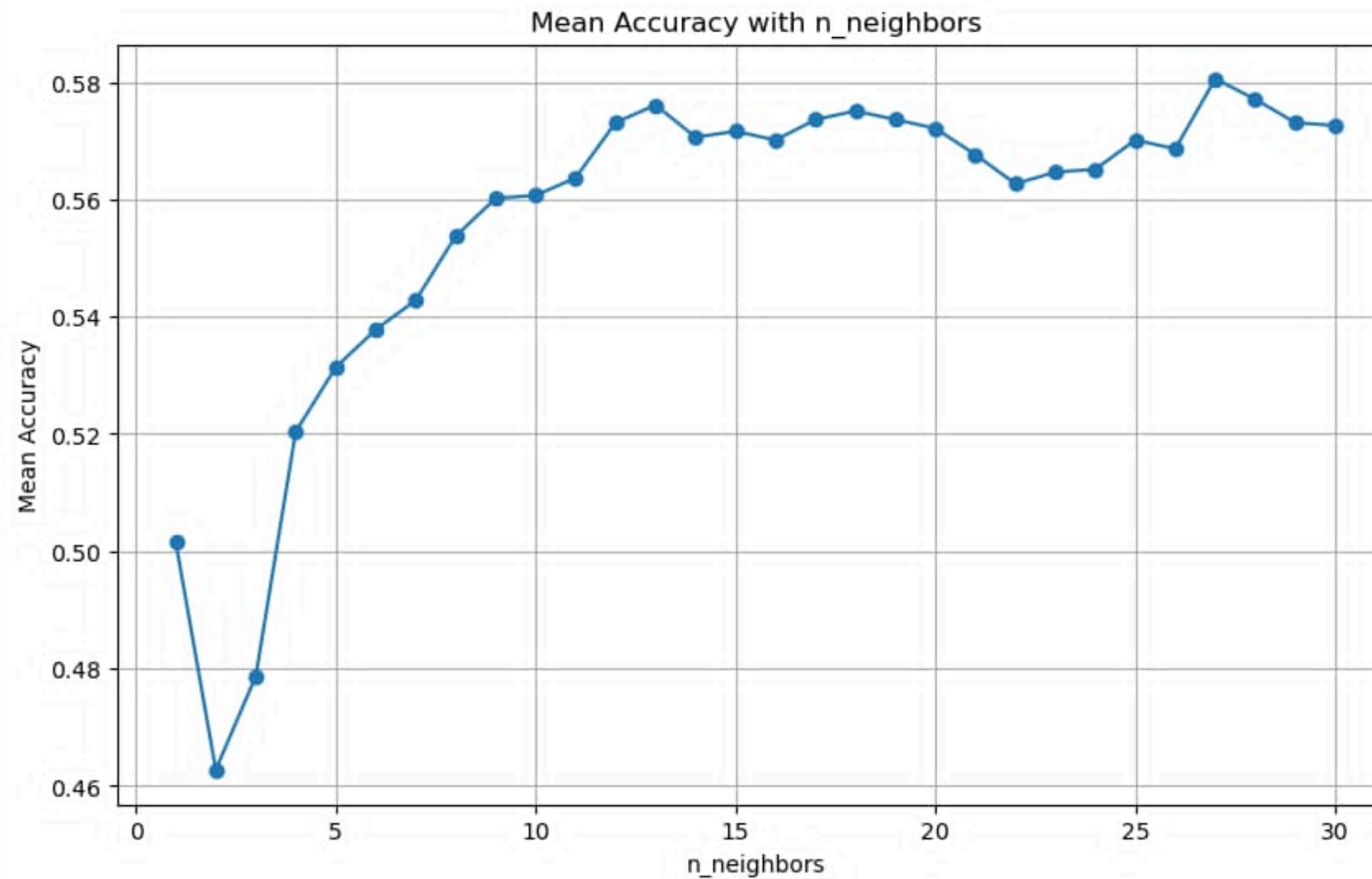


rbf

IV. Thực nghiệm

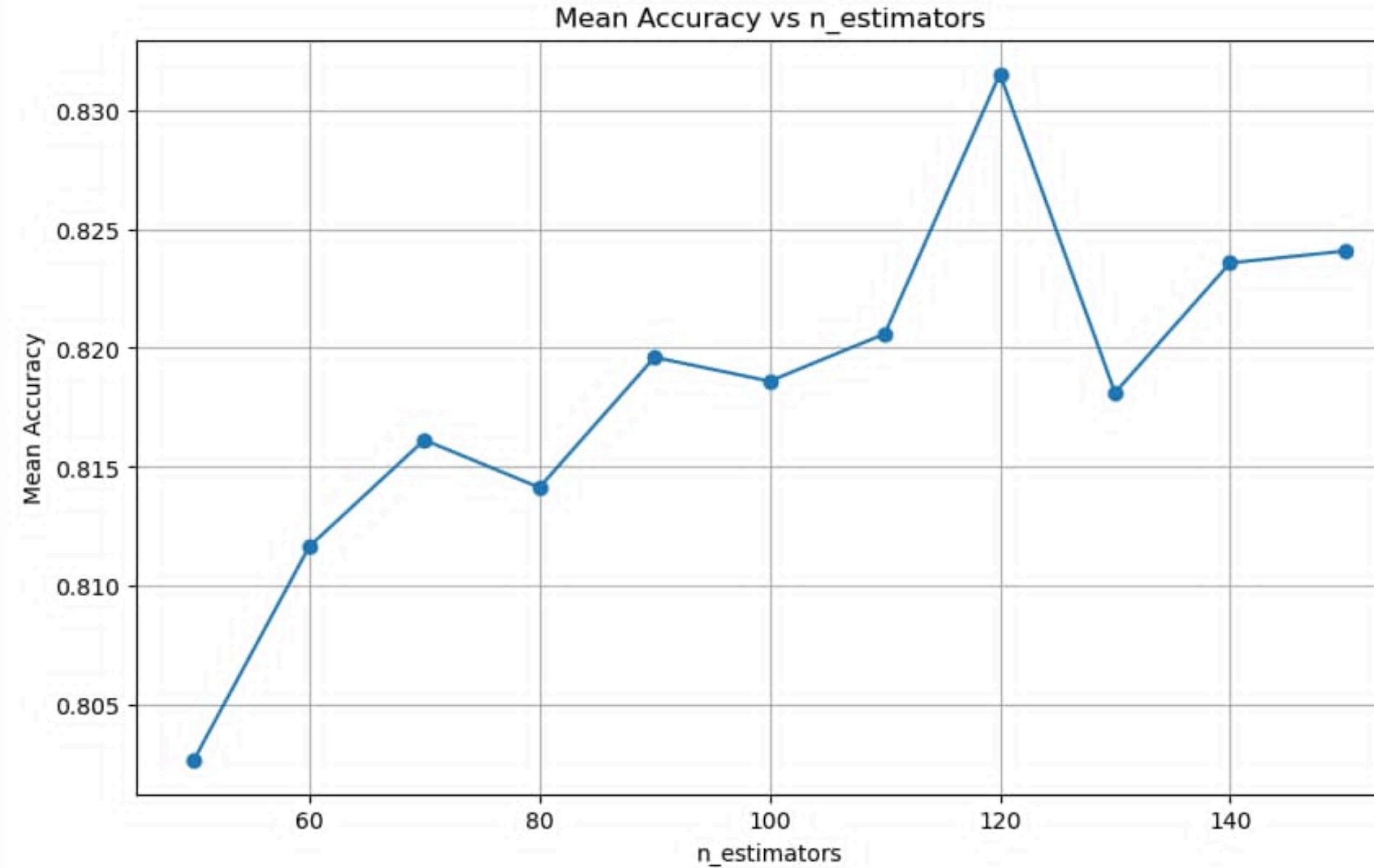
kNN

BOVW (KMean K = 250)



IV. Thực nghiệm

Random Forest BOVW (KMean K = 250)



IV. Thực nghiệm

Kết quả trên Test BOVW(KMean = 250)

	precision	recall	f1-score	support
0	0.71	1.00	0.83	5
1	0.57	0.80	0.67	5
2	0.75	0.60	0.67	5
3	0.50	0.20	0.29	5
4	0.50	0.20	0.29	5
5	0.25	1.00	0.40	5
6	1.00	0.40	0.57	5
7	1.00	0.40	0.57	5
8	1.00	0.60	0.75	5
9	1.00	0.20	0.33	5
accuracy			0.54	50
macro avg	0.73	0.54	0.54	50
weighted avg	0.73	0.54	0.54	50

KNN
(k =27)

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5
1	0.80	0.80	0.80	5
2	0.71	1.00	0.83	5
3	0.80	0.80	0.80	5
4	0.67	0.80	0.73	5
5	1.00	1.00	1.00	5
6	0.50	0.60	0.55	5
7	1.00	0.40	0.57	5
8	0.80	0.80	0.80	5
9	1.00	0.80	0.89	5
accuracy			0.80	50
macro avg	0.83	0.80	0.80	50
weighted avg	0.83	0.80	0.80	50

Random Forest
(n = 120)

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5
1	0.83	1.00	0.91	5
2	0.83	1.00	0.91	5
3	1.00	1.00	1.00	5
4	1.00	1.00	1.00	5
5	0.83	1.00	0.91	5
6	0.80	0.80	0.80	5
7	1.00	0.40	0.57	5
8	0.83	1.00	0.91	5
9	1.00	0.80	0.89	5
accuracy			0.90	50
macro avg	0.91	0.90	0.89	50
weighted avg	0.91	0.90	0.89	50

SVM
(C=1, gamma =0.01, kernel ='rbf')



05

Demo

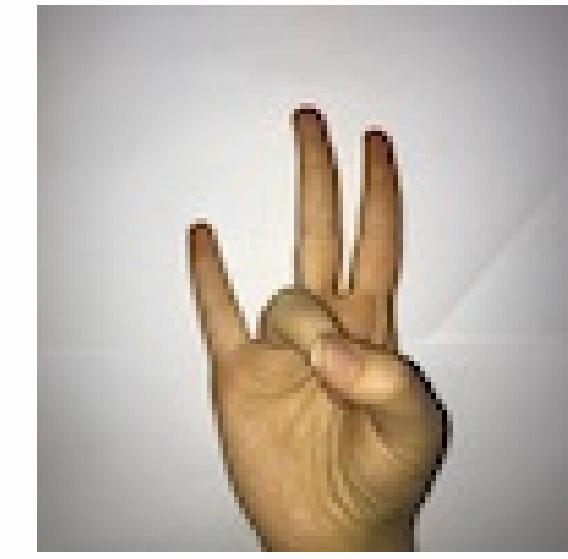
Ảnh dự đoán đúng



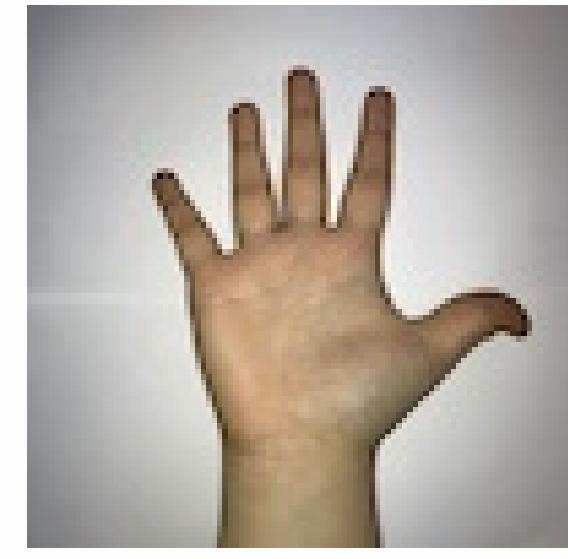
True label: 2
hog_predict: 2



True label: 3
hog_predict: 3



True label: 7
hog_predict: 7



True label: 5
hog_predict: 5

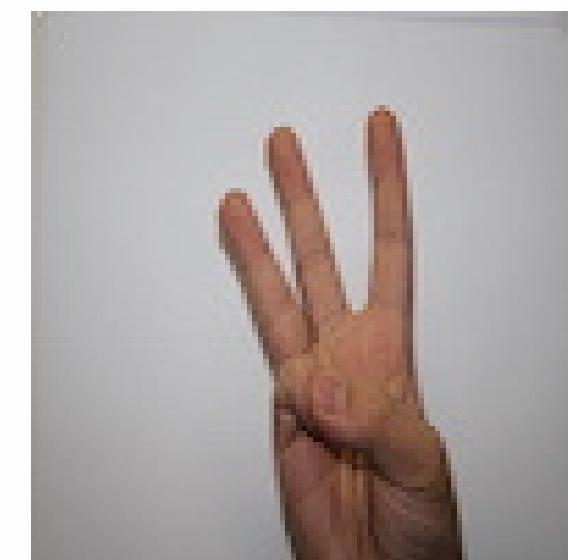
Ảnh dự đoán sai



True label: 4
hog_predict: 7

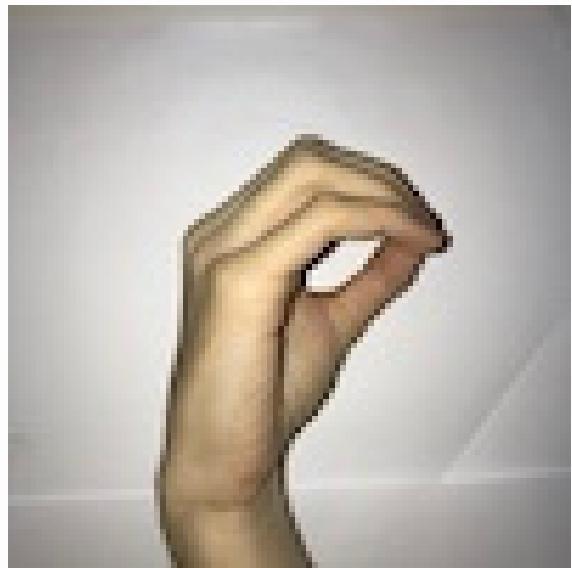


True label: 7
hog_predict: 6



True label: 6
hog_predict: 4

Ảnh dự đoán đúng



True label: 0
bovw_predict: 0



True label: 1
bovw_predict: 1



True label: 2
bovw_predict: 2



True label: 4
bovw_predict: 4

Ảnh dự đoán sai



True label: 9
bovw_predict: 5



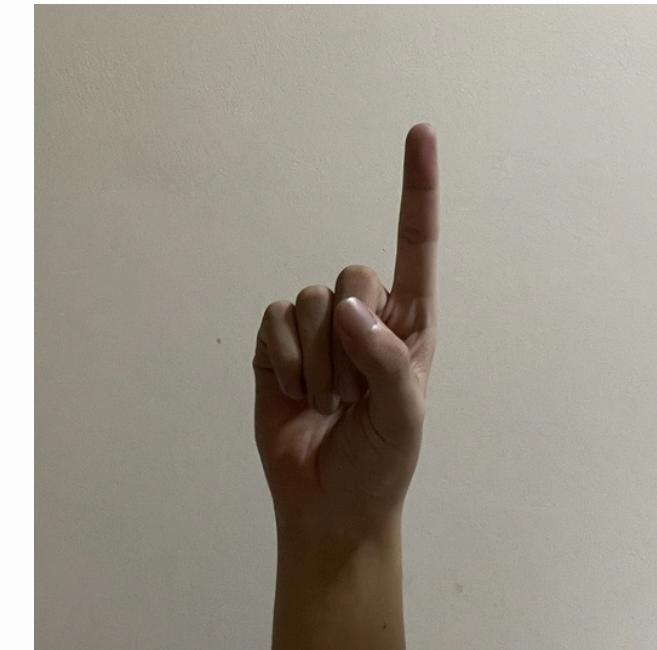
True label: 7
bovw_predict: 2



True label: 6
bovw_predict: 8



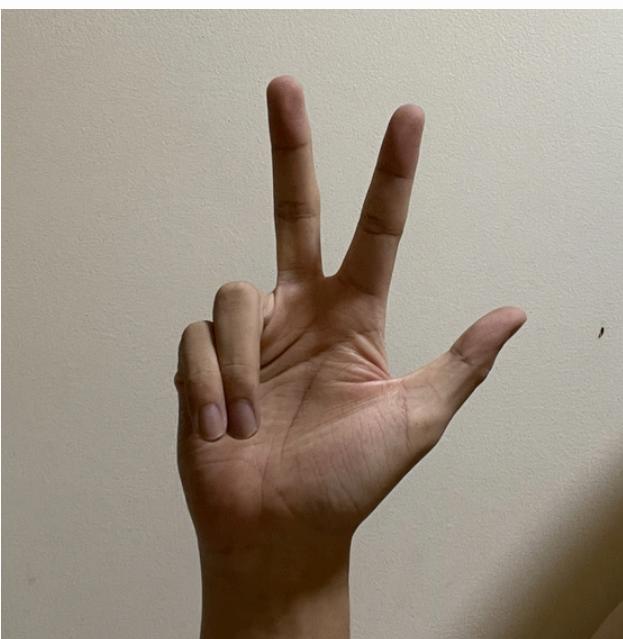
True label: 0
Hog_predict: 0
bovw_predict: 0



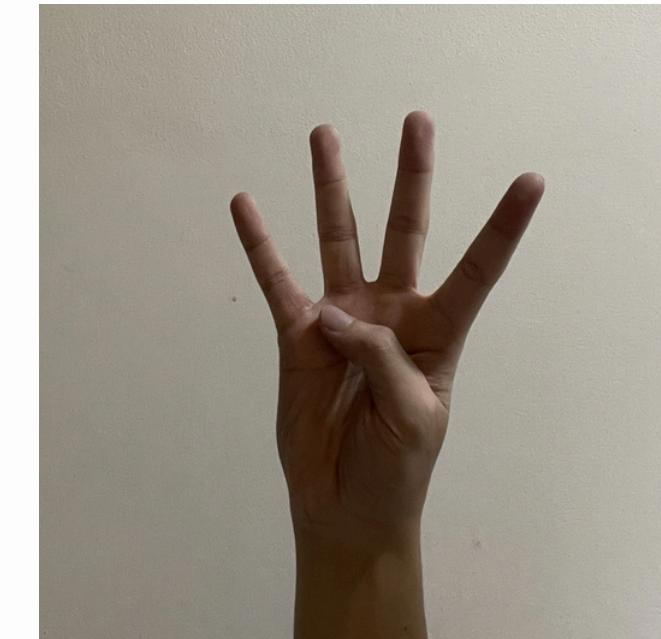
True label: 1
Hog_predict: 1
bovw_predict: 1



True label: 2
Hog_predict: 2
bovw_predict: 2



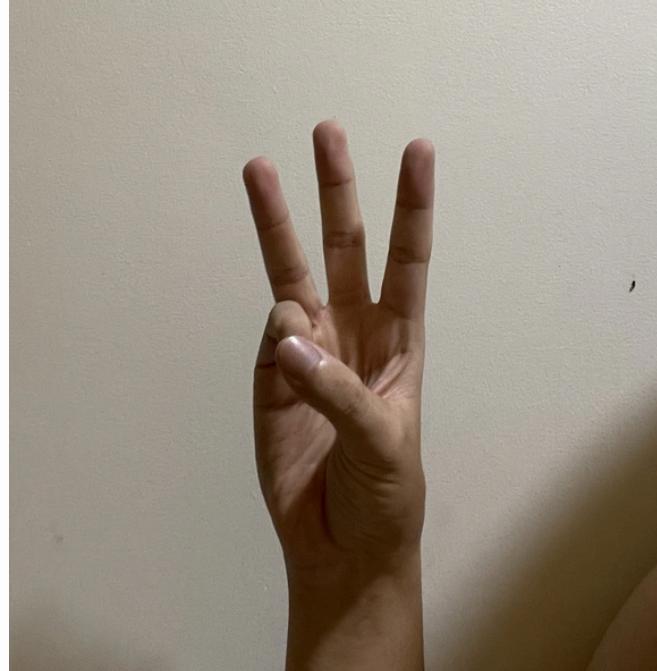
True label: 3
Hog_predict: 3
bovw_predict: 2



True label: 4
Hog_predict: 4
bovw_predict: 8



True label: 5
Hog_predict: 3
boww_predict: 6



True label: 6
Hog_predict: 6
boww_predict: 4



True label: 7
Hog_predict: 7
boww_predict: 3



True label: 8
Hog_predict: 8
boww_predict: 8



True label: 9
Hog_predict: 9
boww_predict: 9

REFERENCES

- [1] Paper: **Visual Categorization with Bags of Keypoints** Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, Cédric Bray Xerox Research Centre Europe 6, chemin de Maupertuis 38240 Meylan, France
- [2] Paper: **Hand Gesture Recognition via Bag of Visual Words** Guoming Chen ,Qiang Chen ,Yiqun Chen,Xiongyong Zhu Department of Computer Science, Guangdong University of Education Guangdong 510303,China
- [3] Paper: **Hand Gesture Recognition Using Bag-of-Features** and Multi-Class Support Vector Machine Nasser Dardas, Qing Chen, Nicolas D. Georganas, Fellow, IEEE, and Emil M. Petriu, Fellow, IEEE
- [4] Paper: **Hand Gesture Recognition using Support Vector Machine and Bag of Visual Words model** Vipul Mehra May 13, 2018
- [5] Paper: **Object Recognition from Local Scale-Invariant Features** David G. Lowe Computer Science Department University of British Columbia Vancouver, B.C., V6T 1Z4, Canada
- [6] <https://www.pinecone.io/learn/series/image-search/bag-of-visual-words/>
- [7] github: <https://github.com/talhaimran96/Image-features-Detection-and-Matching-Image-Classification-Using-Bag-of-Visual-Words/tree/main>
- [8] Paper: Cortes, Corinna; Vapnik, Vladimir (1995). "**Support-vector networks**" (PDF)
- [9] Navneet Dalal and Bill Triggs. **Histograms of oriented gradients for human detection.** IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 1:886–893, 2005

THANK YOU!