

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

CS321 - Nhập môn Thị giác máy tính

BÁO CÁO ĐỒ ÁN CUỐI KÌ

Đề tài: Animals classification

Giảng viên hướng dẫn: TS. Mai Tiến Dũng

Sinh viên thực hiện: 22520004 – Trần Như Cẩm Nguyên

22520361 – Trần Thị Cẩm Giang

Lớp: CS321.O21.KHTN



Mục lục

1. Giới thiệu đề tài

- 1.1. Lý do chọn đề tài
- 1.2. Phát biểu bài toán
- 1.3. Nội dung thực hiện

2. Cơ sở lý thuyết

- 2.1. Histogram Of Oriented Gradient - HOG
- 2.2. Principal Component Analysis - PCA
- 2.3. Support Vector Machine - SVM
- 2.4 k-Nearest Neighbors - KNN
- 2.5. Random Forest - RF

3. Thực nghiệm

- 3.1. Dataset
- 3.2. Cài đặt
- 3.3. Đánh giá

4. Demo



Mục lục

1. Giới thiệu đề tài

- 1.1. Lý do chọn đề tài
- 1.2. Phát biểu bài toán
- 1.3. Nội dung thực hiện

2. Cơ sở lý thuyết

- 2.1. Histogram Of Oriented Gradient - HOG
- 2.2. Principal Component Analysis - PCA
- 2.3. Support Vector Machine - SVM
- 2.4 k-Nearest Neighbors - KNN
- 2.5. Random Forest - RF

3. Thực nghiệm

- 3.1. Dataset
- 3.2. Cài đặt
- 3.3. Đánh giá

4. Demo



Lý do chọn đề tài



Lý do chọn đề tài



Animals Classification (lion, elephant, horse, dog, cat)

- Nền tảng để phát triển hệ thống phân loại nhiều loại động vật hơn.
- Tự động phân loại ảnh động vật từ một tập dữ liệu lớn, để tổ chức và quản lý ảnh trong các cơ sở dữ liệu...
- Áp dụng vào camera trong vườn thú, quản lý động vật.

Phát biểu bài toán

Input

- Ảnh số có chứa 1 đối tượng động vật cần dự đoán
- Bộ dữ liệu động vật được gán nhãn trước (mỗi loài động vật lưu trong một folder riêng với tên folder là tên loài động vật đó)

Animals Classification

Output

Nhãn của ảnh cần dự đoán (thuộc vào tập các nhãn đã được định nghĩa trước trong bộ dữ liệu huấn luyện)

Phát biểu bài toán

Input



Output

**Animals
Classification**

lion

Phát biểu bài toán

Input



Output

**Animals
Classification**

cat

Nội dung thực hiện

1

Rút trích đặc trưng của ảnh bằng **HOG**

2

Chuẩn hóa và giảm chiều dữ liệu bằng **PCA**

3

Sử dụng phương pháp máy học để phân loại (**SVM, KNN, RF**)

4

Chạy thực nghiệm và đánh giá

Mục lục

1. Giới thiệu đề tài

- 1.1. Lý do chọn đề tài
- 1.2. Phát biểu bài toán
- 1.3. Nội dung thực hiện

2. Cơ sở lý thuyết

- 2.1. Histogram Of Oriented Gradient - HOG
- 2.2. Principal Component Analysis - PCA
- 2.3. Support Vector Machine - SVM
- 2.4 k-Nearest Neighbors - KNN
- 2.5. Random Forest - RF

3. Thực nghiệm

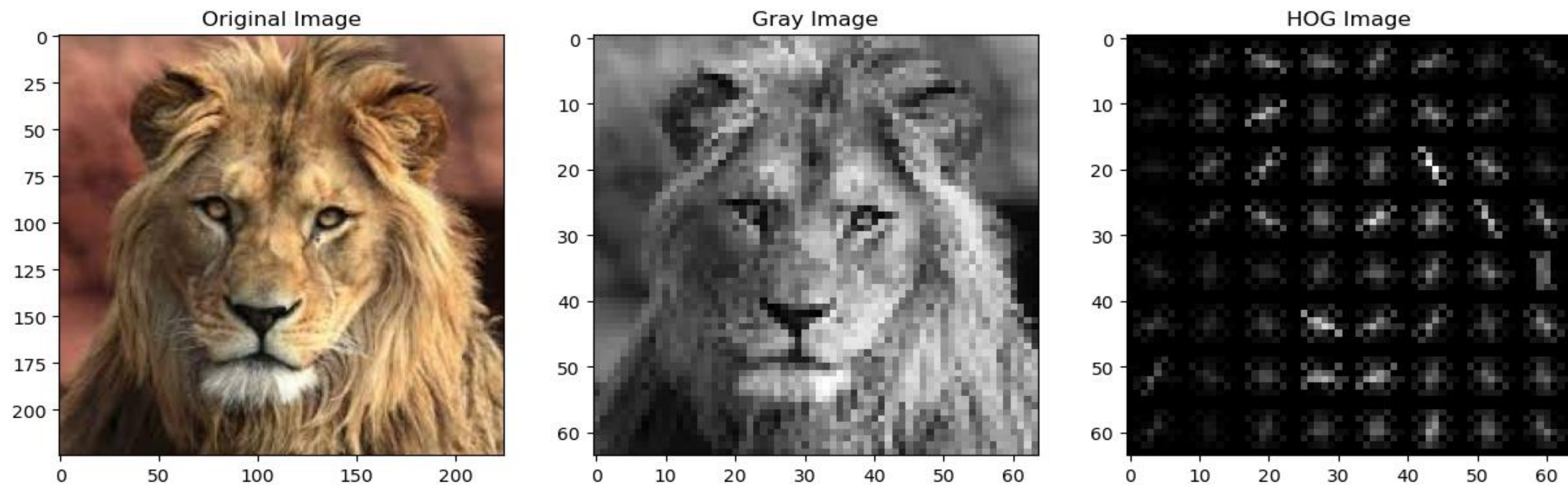
- 3.1. Dataset
- 3.2. Cài đặt
- 3.3. Đánh giá

4. Demo



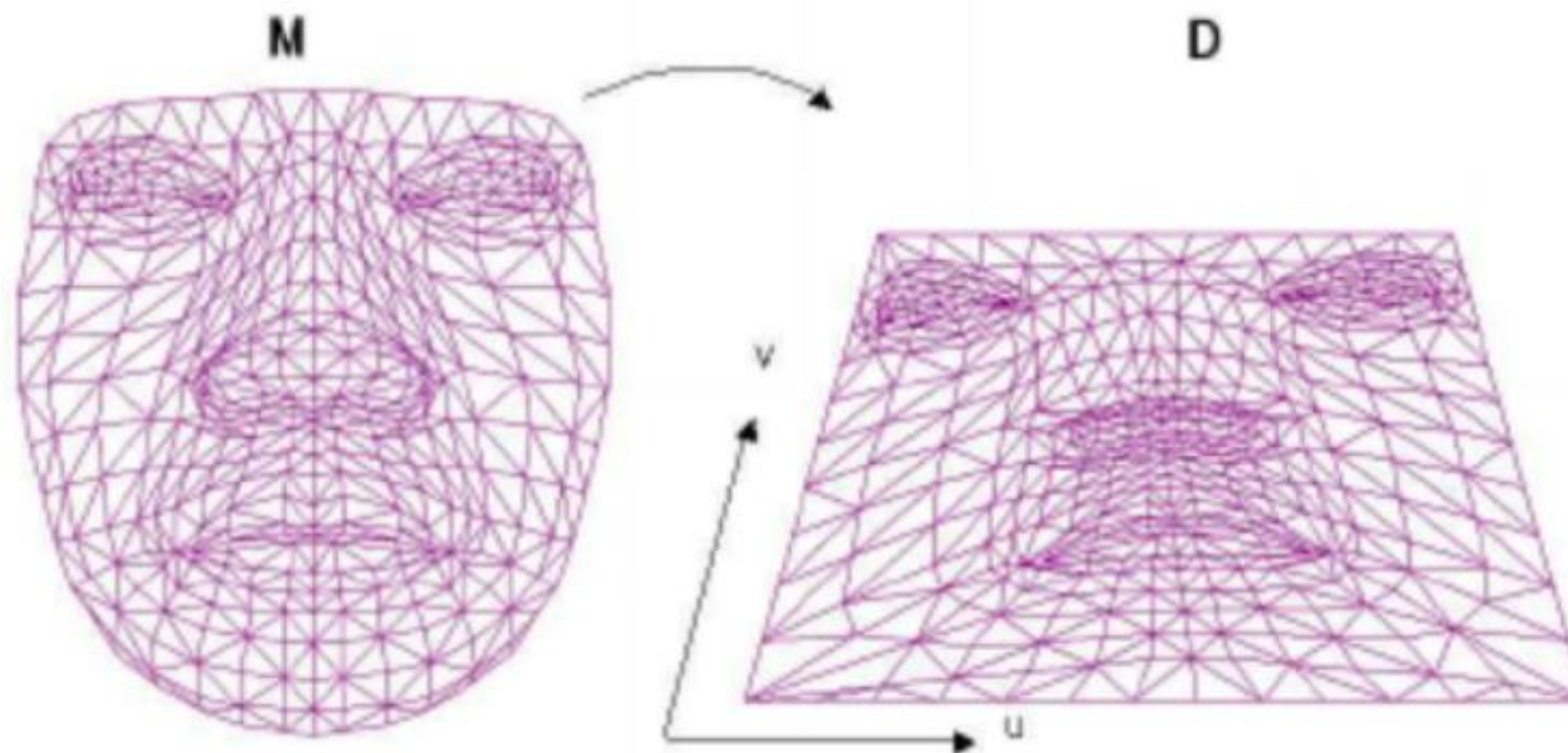
Histogram of Oriented Gradient (HOG)

Phương pháp **Histogram of Oriented Gradient (HOG)** là một kỹ thuật trích xuất đặc trưng, phát triển bởi **Navneet Dalal** và **Bill Triggs** giải quyết bài Human Detection (2005). HOG sử dụng phương gradient và độ lớn gradient để mô tả các đặc trưng của hình ảnh.



Principal Component Analysis (PCA)

Phương pháp **phân tích thành phần chính (PCA)**, ý tưởng ban đầu của **Person (1901)**, sau đó **Hotelling (1933)** có những đóng góp quan trọng. PCA là phương pháp giảm chiều dữ liệu trong khi vẫn giữ lại hầu hết các đặc trưng quan trọng, tăng tốc độ xử lý, giảm độ phức tạp dữ liệu, cải thiện độ chính xác của mô hình

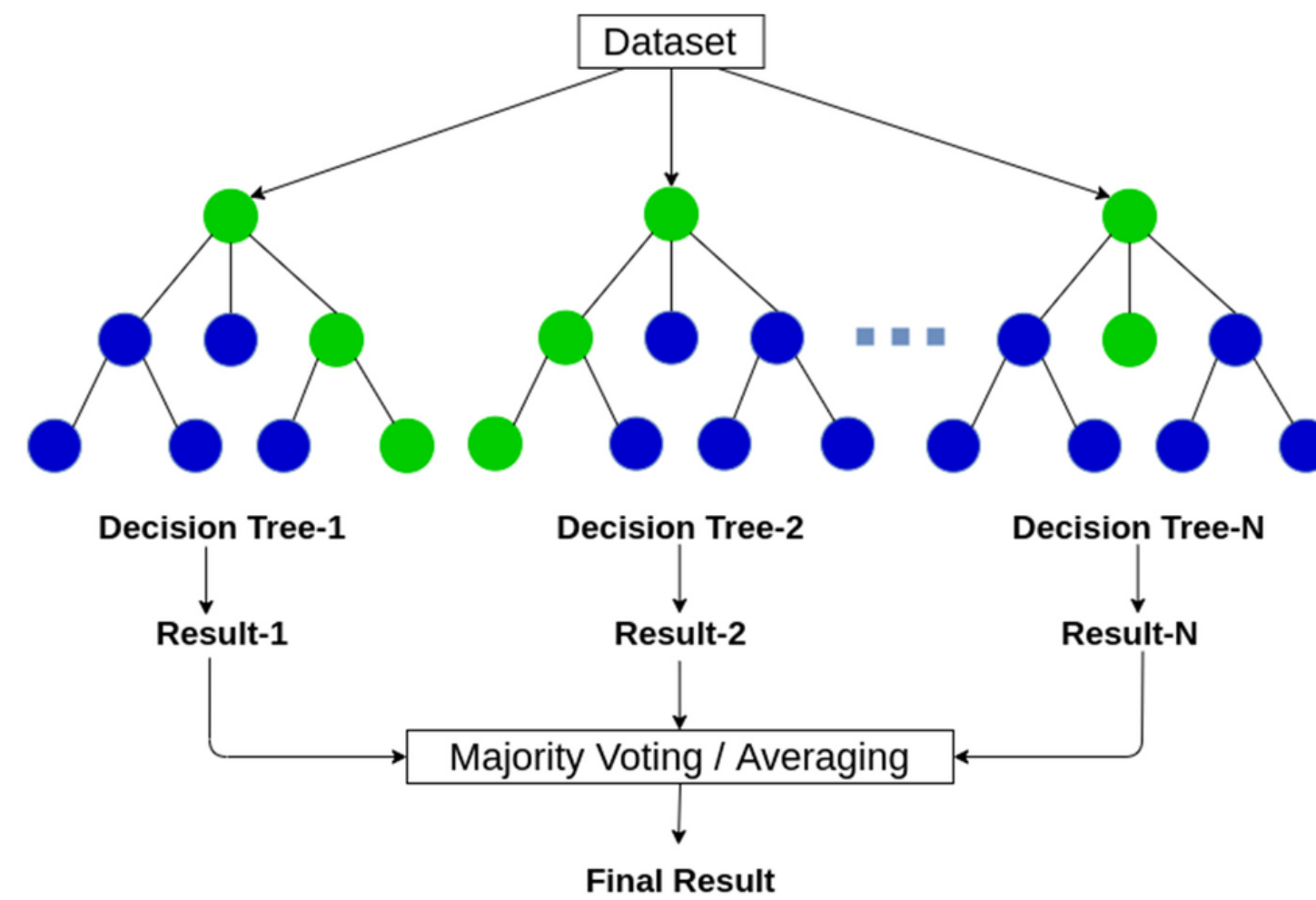


Random Forest (RF)

Random Forest (RF) được giới thiệu bởi **Leo Breiman** và **Adele Cutler (2001)**. Với ý tưởng xây dựng nhiều cây quyết định, tuy nhiên mỗi cây sẽ khác nhau (có yếu tố random), kết quả dự đoán sẽ được tổng hợp từ các cây quyết định.

Tham số **n_estimators** là số cây quyết định sẽ được tạo ra

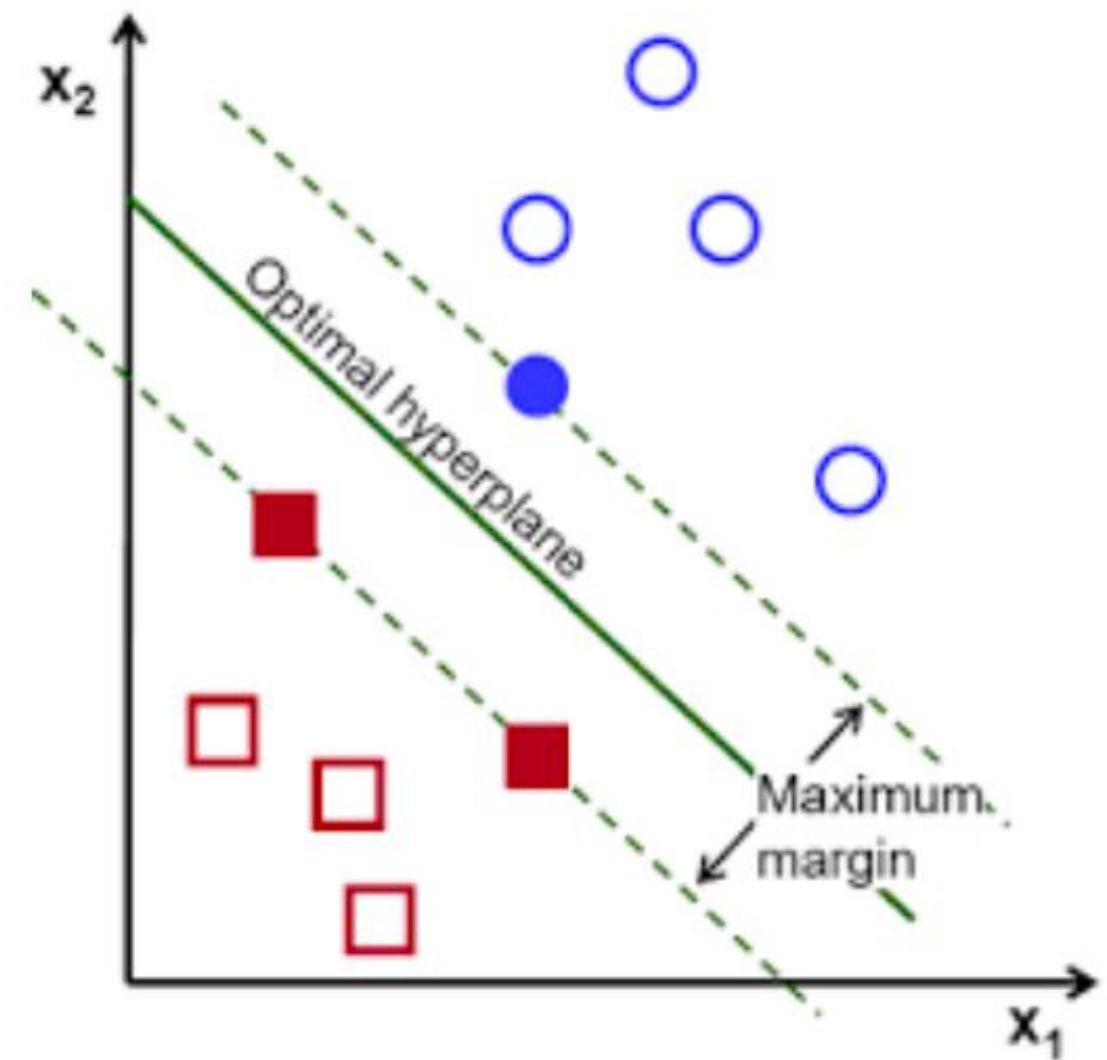
Cây quyết định (decision tree) bao gồm một tập hợp các nút, mỗi nút đại diện cho một quyết định hoặc một dự đoán, được xây dựng dựa trên đặc trưng từ bộ dữ liệu huấn luyện



Support Vector Machine (SVM)

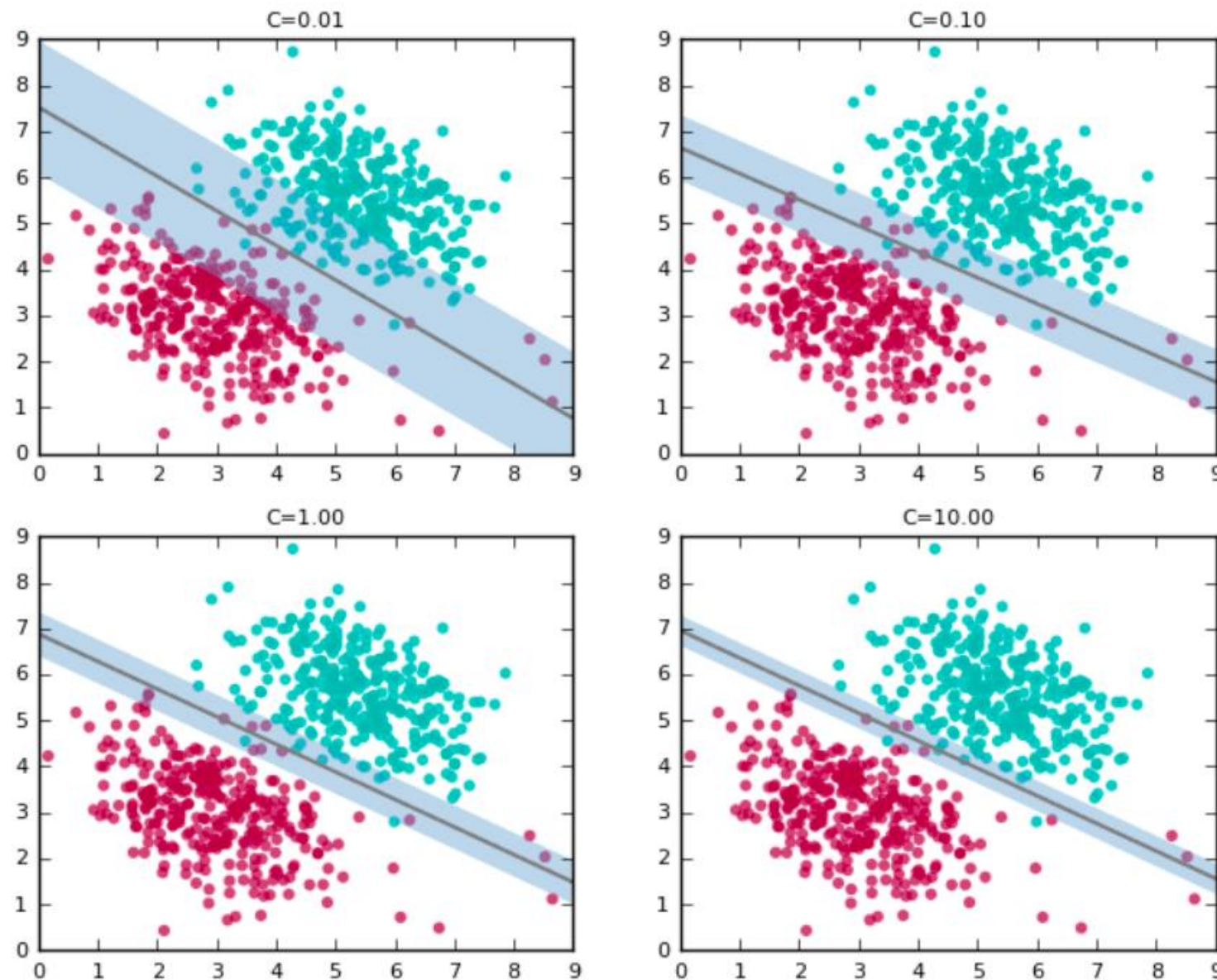
Support Vector Machine (SVM) phát triển bởi **Vladimir N.Vapnik** và đồng nghiệp vào những năm 1990. SVM chủ yếu dùng để phân loại, được gọi là Support Vector Classifier (SVC).

Mục tiêu của SVM là tìm một siêu phẳng sao cho “**khoảng cách từ siêu phẳng đến các điểm dữ liệu gần nhất**” là **lớn nhất**. Khoảng cách đó gọi là lề (margin)



Support Vector Machine (SVM)

Soft Margin SVM



Đối với dữ liệu không phân biệt tuyến tính (**non linear separable**), có thể dùng **Soft Margin SVM**. Cho phép SVM mắc một số lỗi nhất định, cân bằng giữa việc tối thiểu phân loại sai và tối đa hóa lề, giúp giảm thiểu overfitting và giúp SVM linh hoạt hơn.

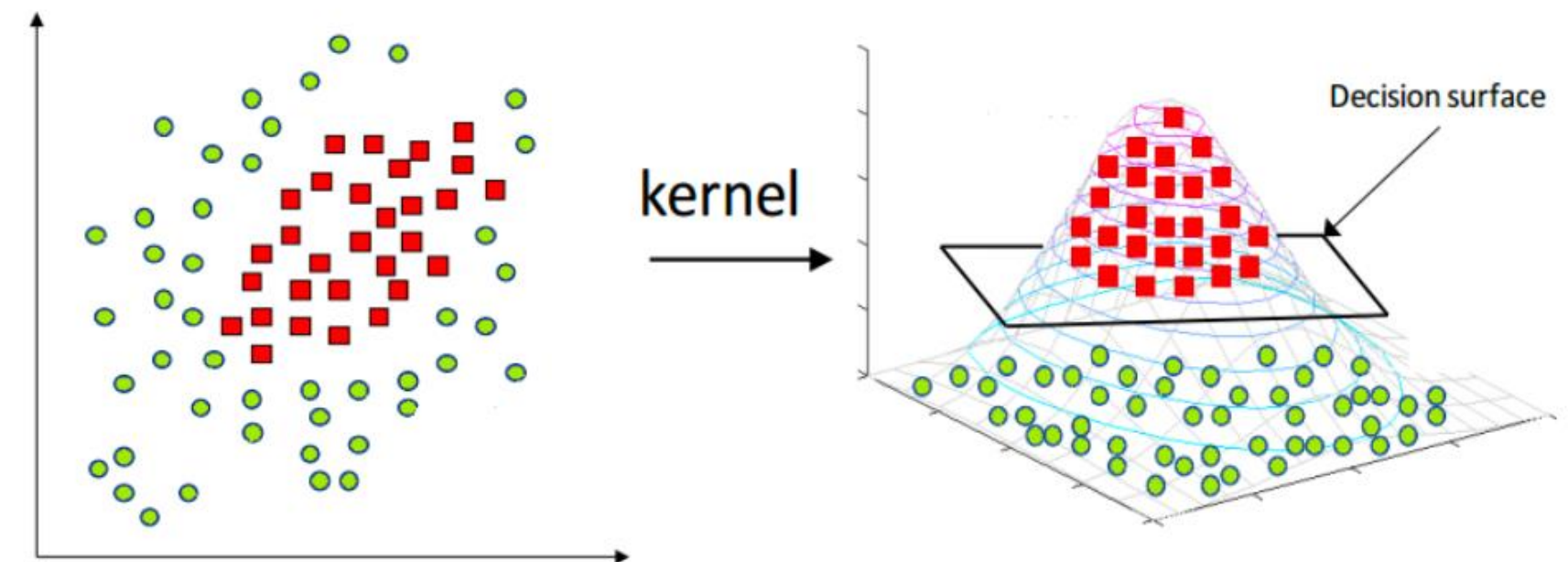
Tham số C (là mức độ chấp nhận lỗi) để kiểm soát mức độ sai sót của Soft Margin SVM. C **càng lớn** thì SVM bị **phạt càng nặng** khi phân loại sai và kết quả sẽ cho ra 1 lề (margin) hẹp và ngược lại.

Support Vector Machine (SVM)

Kernel SVM

Kernel SVM xử lý dữ liệu không tuyến tính bằng cách dùng hàm nhân (Kernel function), ánh xạ dữ liệu vào không gian khác, sau đó tìm siêu phẳng phân chia tuyến tính trong không gian mới đó.

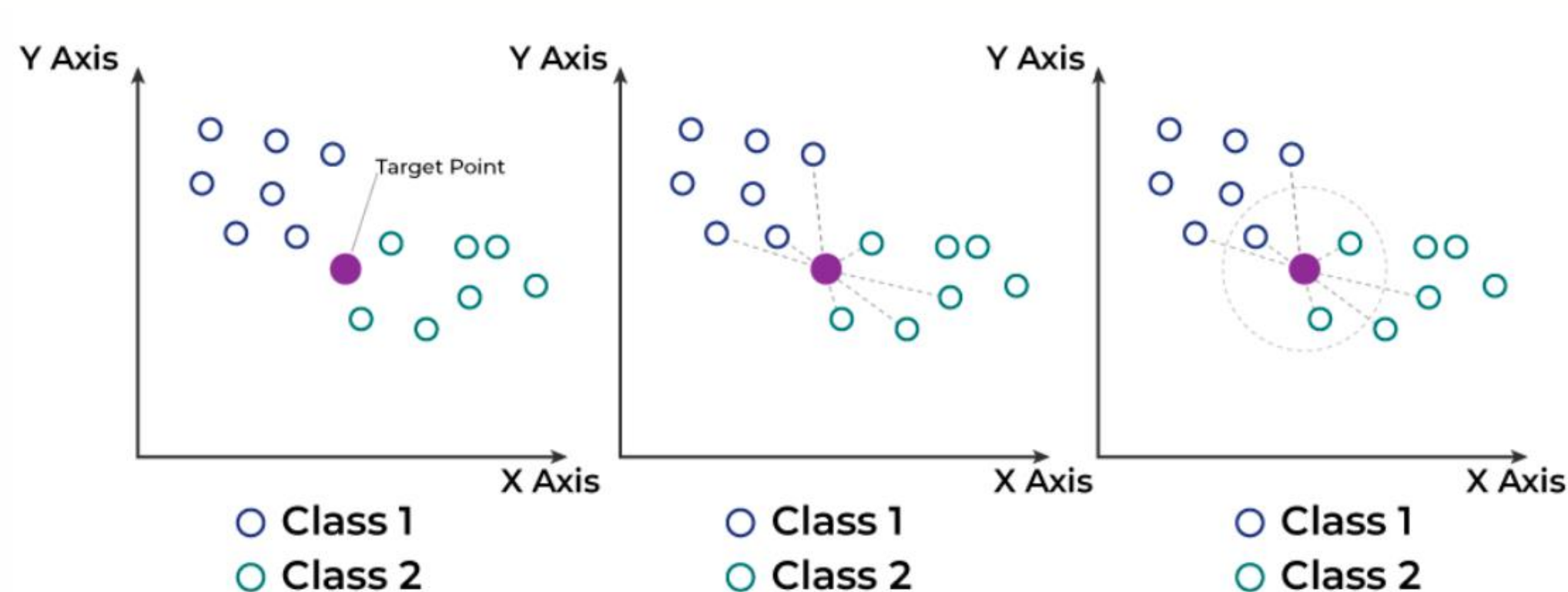
Tên	Công thức
linear	$\mathbf{x}^T \mathbf{z}$
polynomial	$(r + \gamma \mathbf{x}^T \mathbf{z})^d$
sigmoid	$\tanh(\gamma \mathbf{x}^T \mathbf{z} + r)$
rbf	$\exp(-\gamma \ \mathbf{x} - \mathbf{z}\ _2^2)$



k-Nearest Neighbors (KNN)

k-Nearest Neighbors (KNN) được công bố lần đầu bởi Eve-lyn Fix và Joseph Hodges (1951) và được mở rộng bởi Thomas Cover (1967). Label của một điểm dữ liệu mới được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong tập dữ liệu huấn luyện.

Tham số **n_neighbors** là số hàng xóm gần nhất quyết định nhãn của dữ liệu đầu vào
Tham số **weight** là trọng số của các điểm hàng xóm lân cận



Mục lục

1. Giới thiệu đề tài

- 1.1. Lý do chọn đề tài
- 1.2. Phát biểu bài toán
- 1.3. Nội dung thực hiện

2. Cơ sở lý thuyết

- 2.1. Histogram Of Oriented Gradient - HOG
- 2.2. Principal Component Analysis - PCA
- 2.3. Support Vector Machine - SVM
- 2.4 k-Nearest Neighbors - KNN
- 2.5. Random Forest - RF

3. Thực nghiệm

- 3.1. Dataset
- 3.2. Cài đặt
- 3.3. Đánh giá

4. Demo



Dataset

<https://www.kaggle.com/datasets/antobenediti/animals>



lion



horse



elephant



dog



cat

Inf

```
--cat.jpg  
--dog.jpg  
-- elephant.jpg  
__horse.jpg  
__lion.jpg
```

Train

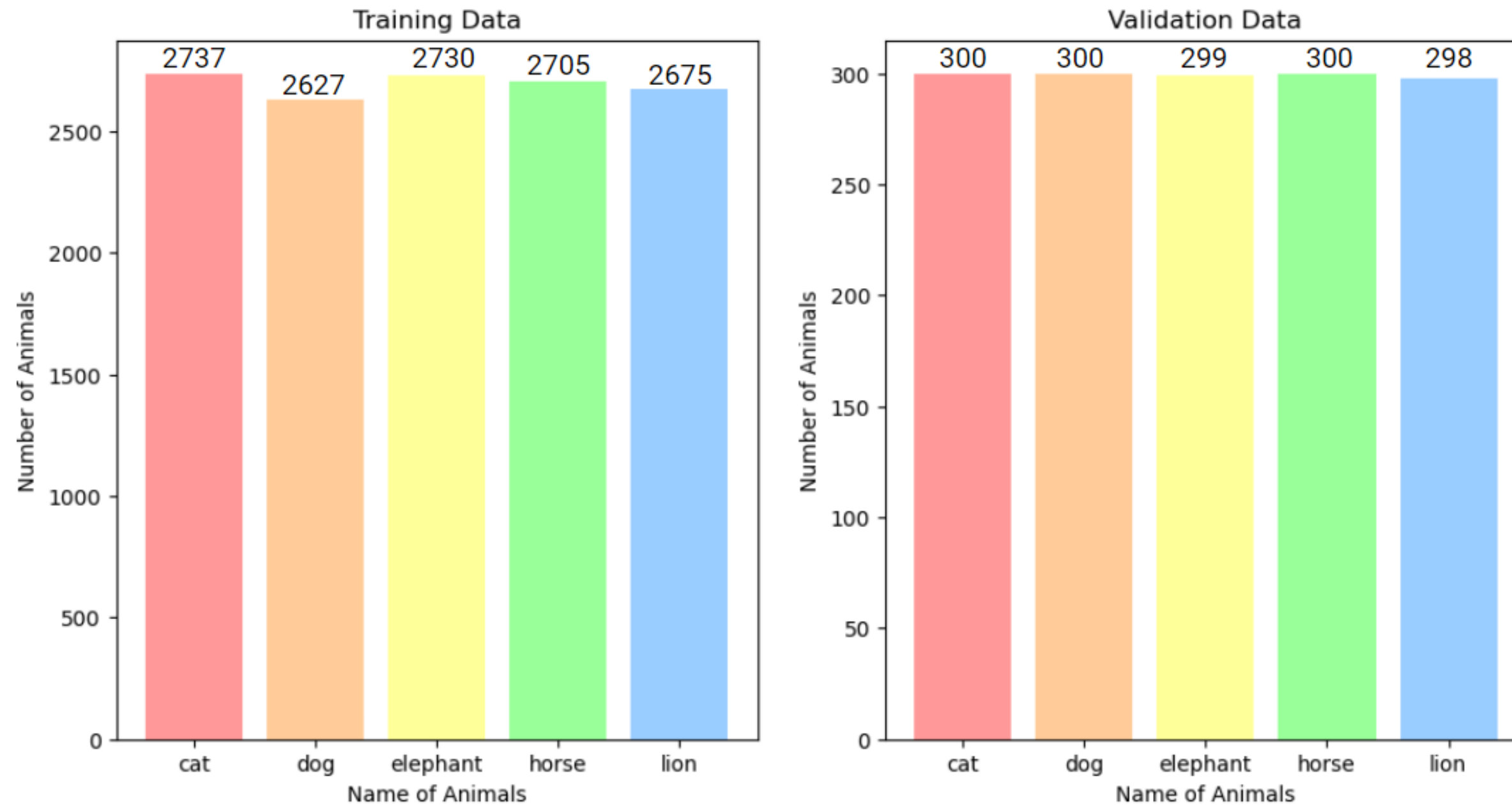
```
--lion  
|  --lion1.jpg  
|  --lion2.jpg  
|  -- ...  
|  __lion2683.jpg  
--horse  
--elephant  
--dog  
--cat
```

Val

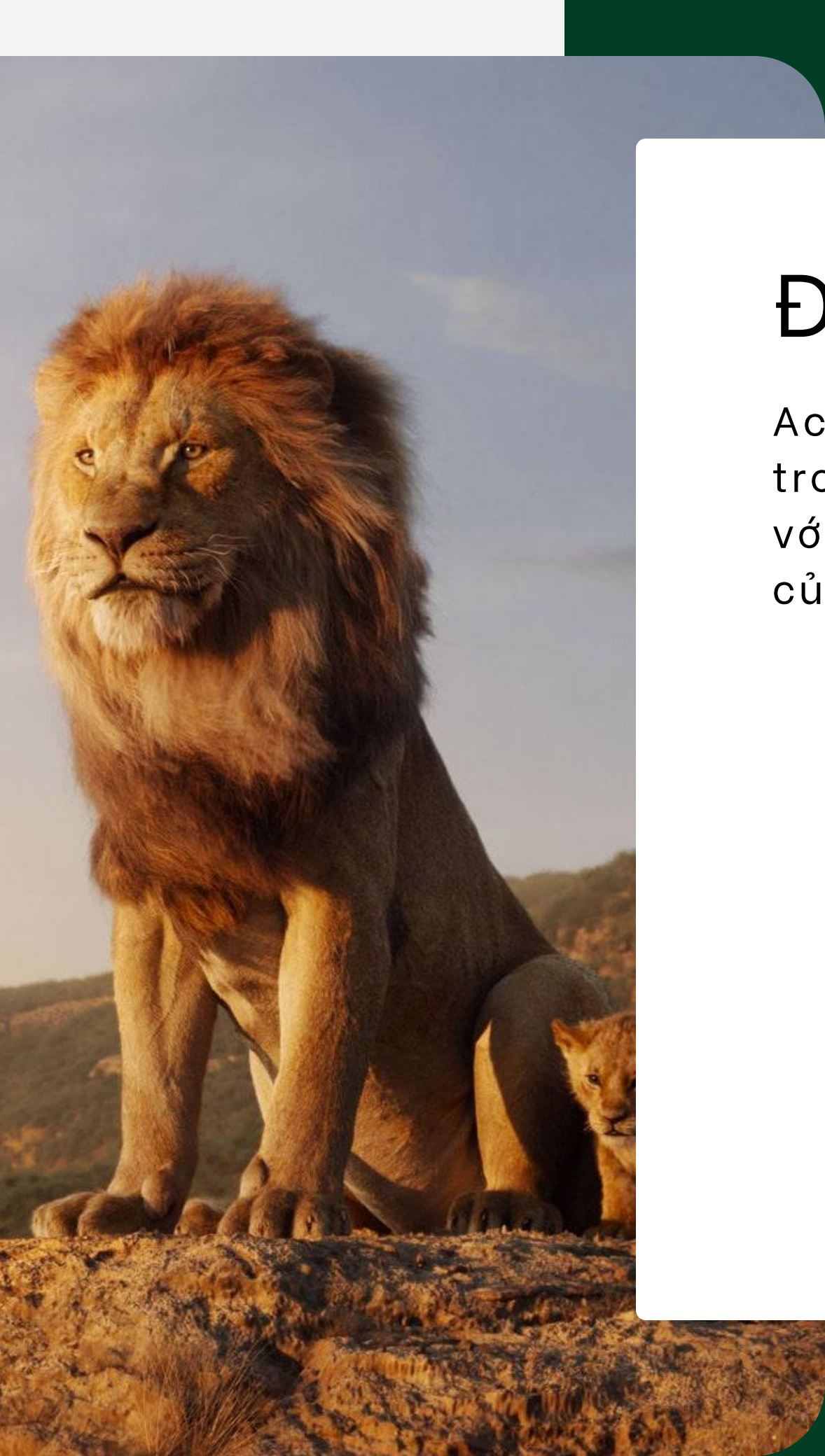
```
--lion  
|  --lion1.jpg  
|  --lion2.jpg  
|  -- ...  
|  __lion2683.jpg  
--horse  
--elephant  
--dog  
--cat
```

Dataset

<https://www.kaggle.com/antobenedetti/animals>



Thống kê số lượng các loại động vật trong tập Training và tập Validation



Độ đo – **Accuracy**

Accuracy là một phép đo đánh giá hiệu suất của một mô hình dự đoán trong machine learning. Nó đo lường tỷ lệ các dự đoán chính xác so với tổng số lượng các dự đoán. Accuracy càng cao thì độ chính xác của mô hình càng tốt.

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}}$$

Các tham số chạy thực nghiệm

- **HOG** orientations = [9, 18, 36]
- **PCA** n_components = [50, 100, 200, 500, None]
- **SVM** Kernel = ['linear', 'poly', 'rbf', 'sigmoid']
 C = [0.01, 0.1, 1, 10]
- **KNN** k = (0,30)
- **RF** default

=> Mỗi thuật toán máy học chạy hết tất cả các tổ hợp tham số.

Kết quả thực nghiệm

HOG + PCA

Validation Set

<div>components</div> <div>orientations</div>	50	100	200	500	NoPCA
9	0.6955	0.688	0.6716	0.6506	0.6445
18	0.706	0.6935	0.6769	0.6566	0.6424
36	0.7064	0.6903	0.6715	0.6616	0.6321

Accuracy trung bình của tổ hợp tham số orientations (HOG) và n_components (PCA)

Kết quả thực nghiệm

SVM

Validation Set

C \ kernel	linear	poly (degree=3)	rbf	sigmoid
C=0.01	0.7142	0.4426	0.6037	0.6848
C=0.1	0.7115	0.6710	0.7341	0.7053
C=1	0.7119	0.7573	0.7712	0.6456
C=10	0.7119	0.7516	0.7705	0.6060

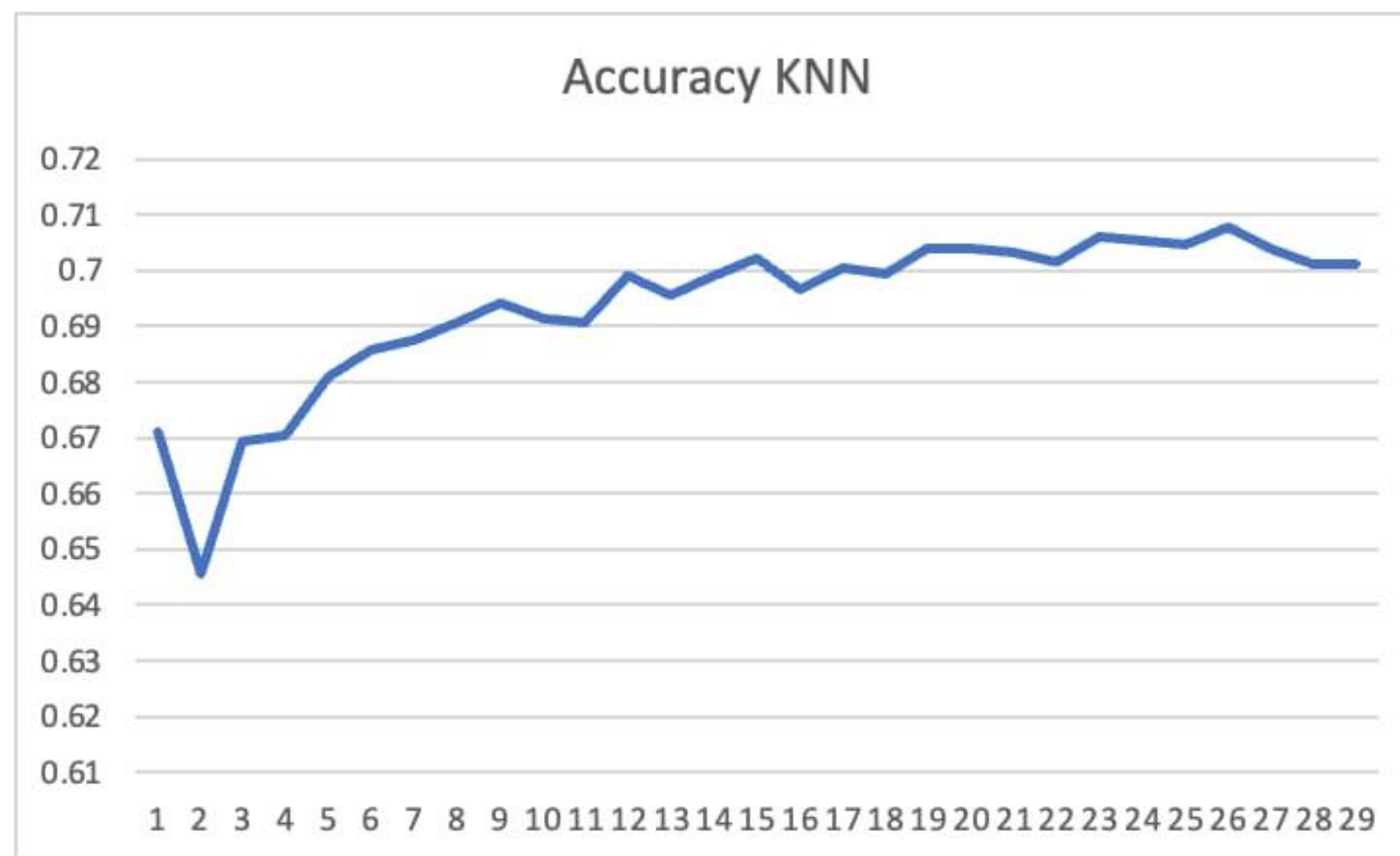
Accuracy của tổ hợp tham số kernel và C trong SVM

(Với đặc trưng sử dụng orientations=36 và n_components=50)

Kết quả thực nghiệm

KNN

Validation Set



Accuracy của KNN theo k

(Với đặc trưng sử dụng orientations=36 và
n_components=50)

Kết quả thực nghiệm

SVM & KNN & RF

Validation Set

	SVM (kernel='rbf',C=1)	KNN (k=26)	RF (n_estimator = 100)
Accuracy	77.12%	70.78%	71.33%
Precision	76.99%	70.77%	71.11%
Recall	77.12%	70.78%	71.33%
F1	77.03%	69.98%	71.09%

Kết quả thực nghiệm

SVM & KNN & RF

Test Set

	SVM (kernel='rbf',C=1)	KNN (k=26)	RF (n_estimator = 100)
Accuracy	80.09%	72.75%	73.48%
Precision	80.02%	73.41%	73.14%
Recall	80.09%	72.75%	73.48%
F1	79.96%	72.06%	73.07

Mục lục

1. Giới thiệu đề tài

- 1.1. Lý do chọn đề tài
- 1.2. Phát biểu bài toán
- 1.3. Nội dung thực hiện

2. Cơ sở lý thuyết

- 2.1. Histogram Of Oriented Gradient – HOG
- 2.2. Principal Component Analysis – PCA
- 2.3. Support Vector Machine – SVM
- 2.4 k-Nearest Neighbors – KNN
- 2.5. Random Forest – RF

3. Thực nghiệm

- 3.1. Dataset
- 3.2. Cài đặt
- 3.3. Đánh giá

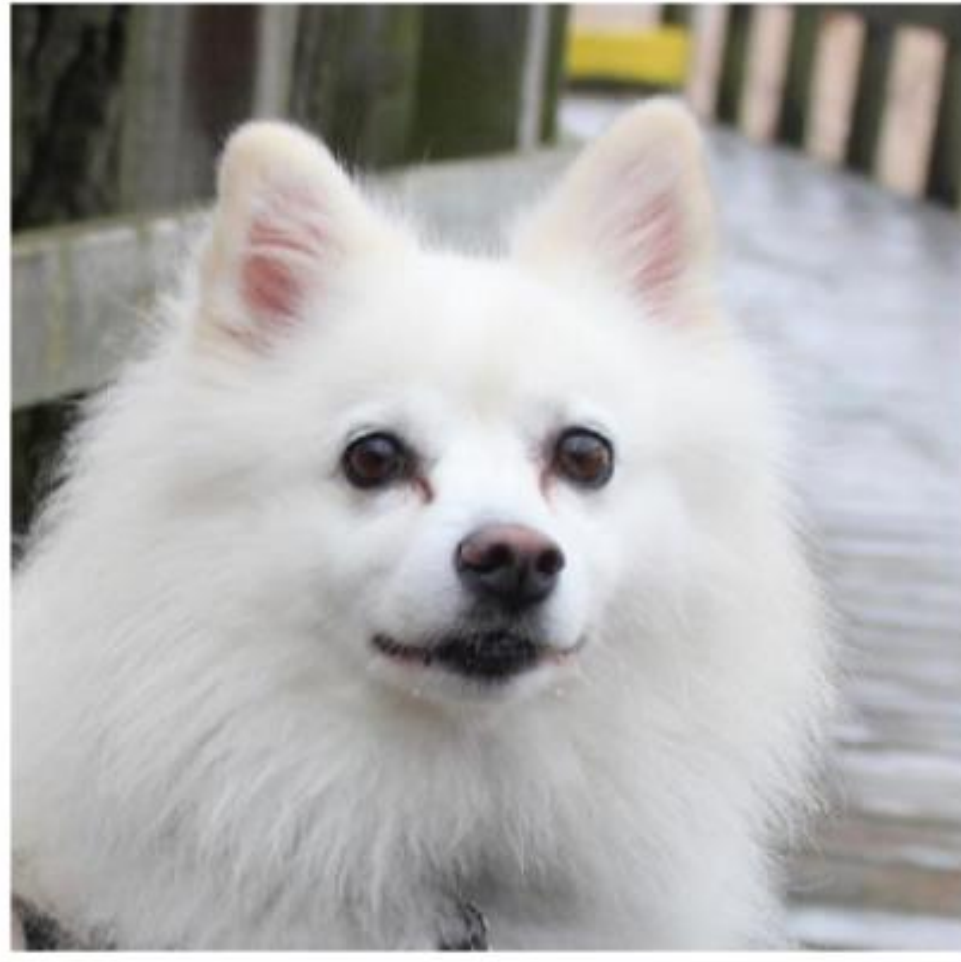
4. Demo



Demo



SVM - Predicted Label: cat
KNN - Predicted Label: cat
RF - Predicted Label: cat



SVM - Predicted Label: dog
KNN - Predicted Label: dog
RF - Predicted Label: dog



SVM - Predicted Label: horse
KNN - Predicted Label: horse
RF - Predicted Label: horse

Demo



SVM - Predicted Label: lion
KNN - Predicted Label: cat
RF - Predicted Label: lion



SVM - Predicted Label: cat
KNN - Predicted Label: cat
RF - Predicted Label: lion



SVM - Predicted Label: elephant
KNN - Predicted Label: horse
RF - Predicted Label: horse


Tài liệu tham khảo:

- [1] <https://phamdinhhkhanh.github.io/2019/11/22/HOG.html>
- [2] <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
- [3] <https://machinelearningcoban.com/2017/06/15/pca/>
- [4] <https://www.geeksforgeeks.org/k-nearest-neighbours/>
- [5] Breiman, L. Random Forests. <https://doi.org/10.1023/A:1010933404324>

 CS321.O21.KHTN

Thank You!

Contact Us

 22520004@gm.uit.edu.vn

 22520361@gm.uit.edu.vn

