

Phân loại cảm xúc khôn mặt

Thuyết trình bởi

Lê Cường Thịnh - 22521409

Nguyễn Duy Thịnh - 22521414

1. Lí do lựa chọn đề tài

2. Phát biểu bài toán

3. Trích xuất đặc trưng

4. Phương pháp sử dụng

5. Dataset

6. Thực nghiệm

7. Demo

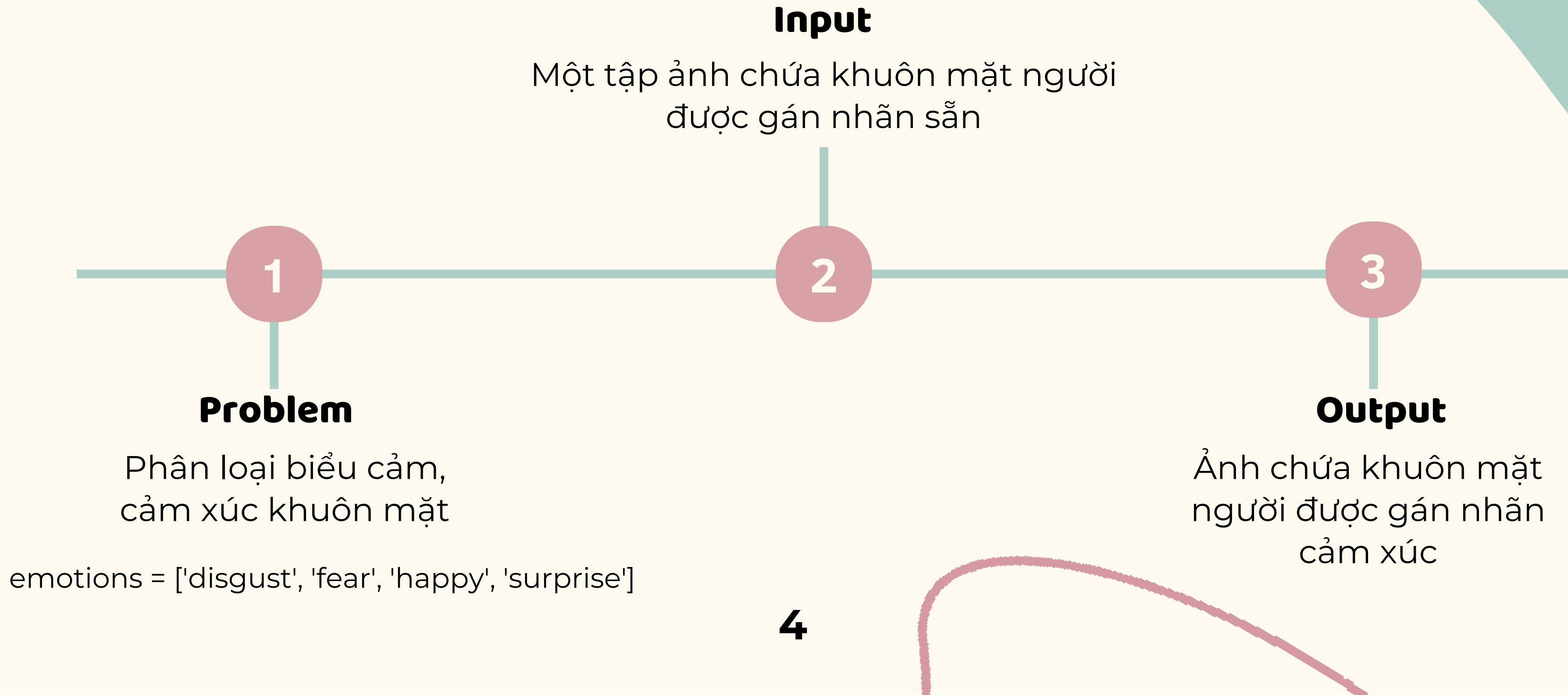
Nội dung

Lí do lựa chọn đề tài

Tầm quan trọng của phân loại cảm xúc

- Tăng cường giao tiếp và hiểu nhau hơn
- Hỗ trợ các công tác dịch vụ chăm sóc khách hàng: y tế, du lịch, mua bán...
- Ứng dụng trong công nghệ và trí tuệ nhân tạo

Phát biểu bài toán



Trích xuất đặc trưng

Histogram of Oriented Gradient (HOG)

- HOG là một loại “feature descriptor”. Mục đích của “feature descriptor” là trừu tượng hóa đối tượng bằng cách trích xuất ra những đặc trưng của đối tượng đó và bỏ đi những thông tin không hữu ích.
- Áp dụng cho bài toán:

```
orientations = 7,  
pixels_per_cell = (8, 8),  
cells_per_block = (4, 4),  
block_norm = 'L2-Hys',  
transform_sqrt = False
```



Phương pháp sử dụng

Support Vector Machine - SVM

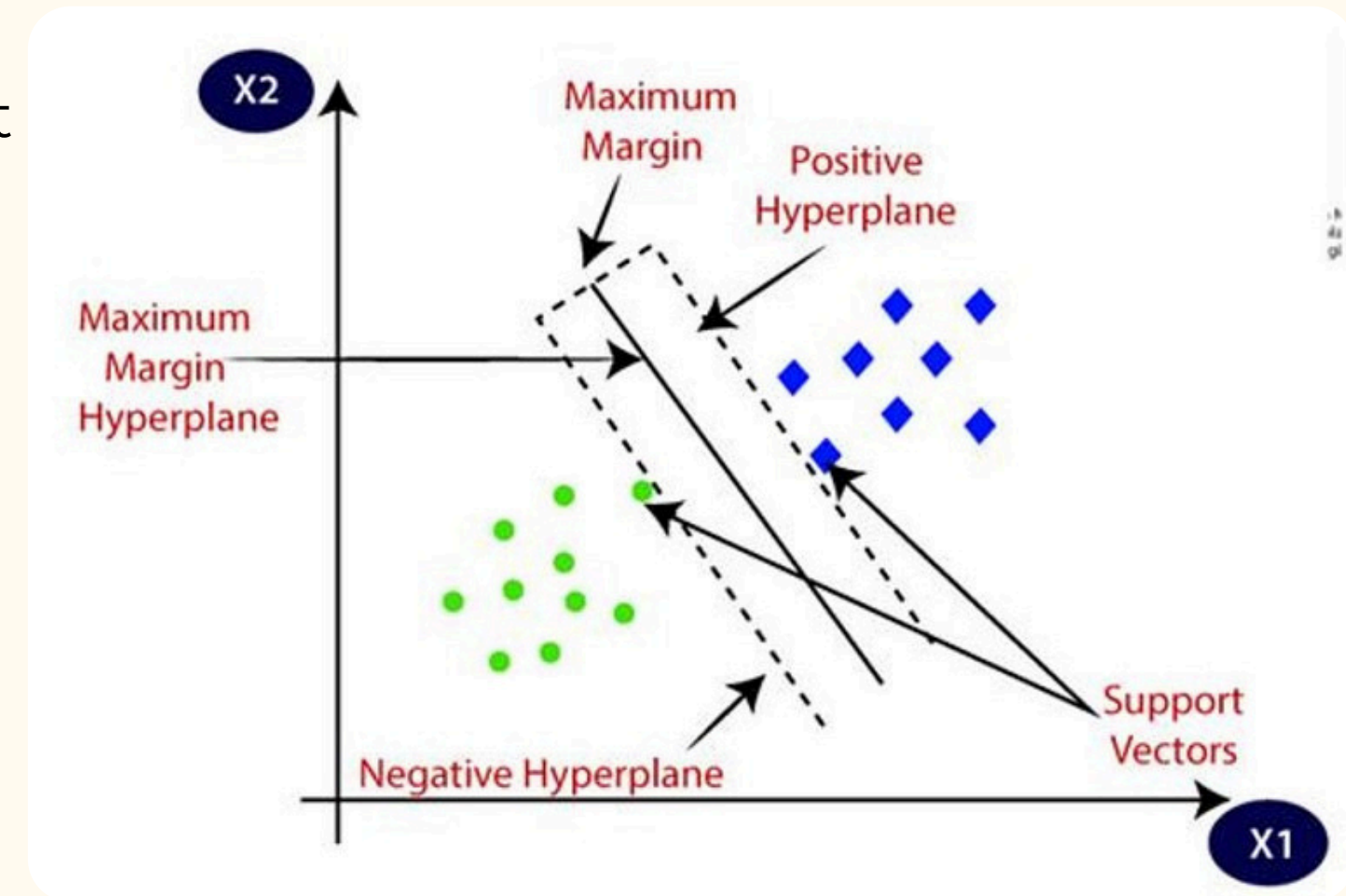
K-Nearest Neighbors - KNN

SVM

Mục đích: SVM là một thuật toán học máy có giám sát, SVM tìm kiếm một siêu phẳng (hyperplane) tốt nhất để phân tách các lớp trong không gian đặc trưng.

Siêu phẳng: Là một đường hoặc mặt phẳng quyết định giúp phân tách các dữ liệu thuộc các lớp khác nhau.

Vector hỗ trợ (Support Vectors): Là các điểm dữ liệu gần nhất với siêu phẳng và có ảnh hưởng đến vị trí và định hướng của siêu phẳng. Những điểm này rất quan trọng vì chúng "hỗ trợ" siêu phẳng tối ưu.



Biên (Margin): Là khoảng cách giữa siêu phẳng và các điểm dữ liệu gần nhất từ mỗi lớp. SVM tìm cách tối đa hóa biên này để tăng cường khả năng phân biệt giữa các lớp.

Kernel Trick: Cho phép SVM hoạt động hiệu quả trong không gian đặc trưng cao bằng cách ánh xạ dữ liệu vào một không gian có chiều cao hơn, từ đó có thể tìm được siêu phẳng phân tách phi tuyến tính.

Áp dụng vào bài toán

```
# Sử dụng Grid Search để tìm tham số tốt nhất
param_grid = {'C': [0.1, 1], 'kernel': ['linear', 'rbf']}
grid_search = GridSearchCV(SVC(), param_grid, cv=3)
grid_search.fit(X_train, y_train)

# In ra tham số tốt nhất
print("Best parameters found: ", grid_search.best_params_)
```

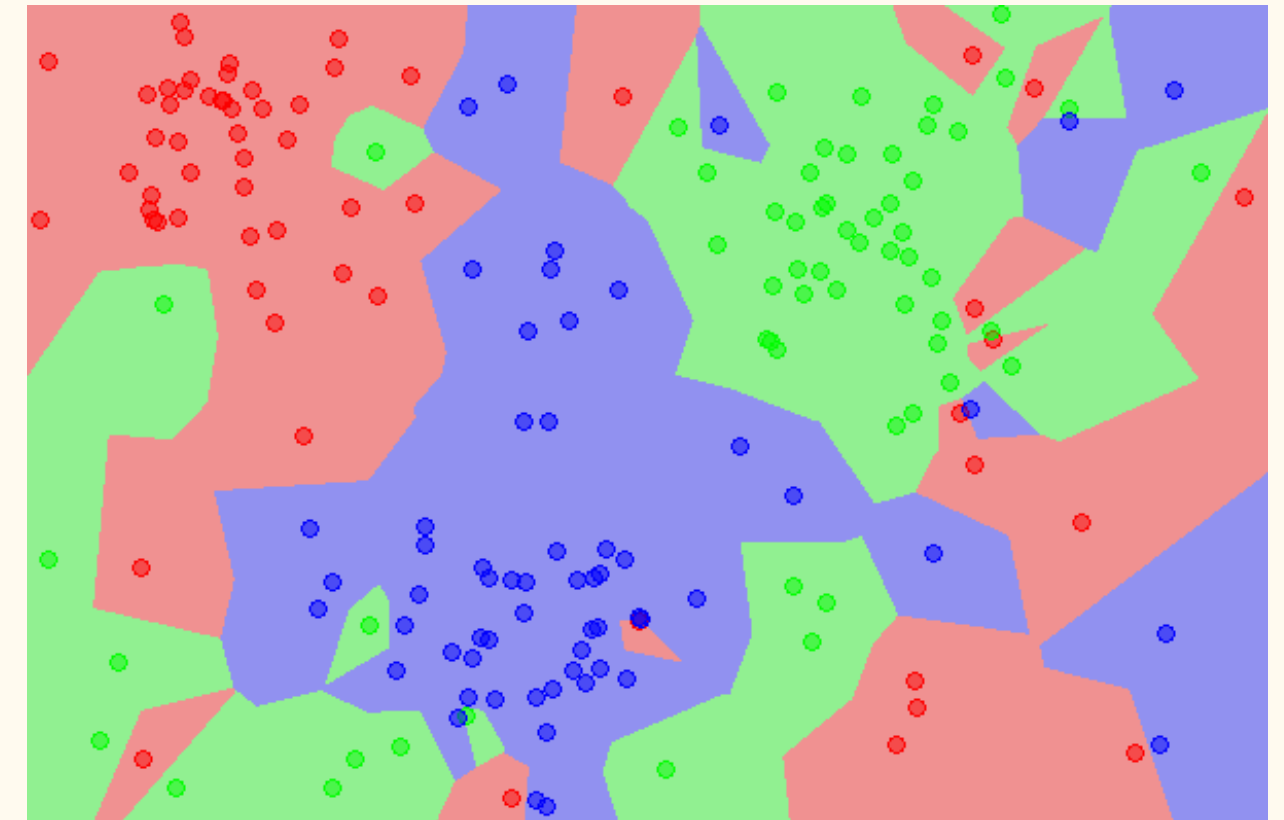
Sử dụng GridSearch để tìm tham số tốt nhất

- C: 1
- Kernel: *rbf*
- Giá trị gamma sẽ là mặc định là 'scale'

KNN

KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong training set gần nó nhất (K-lân cận), không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này là nhiều.

Thuật toán **k -nearest Neighbor** (k -NN) là một phương pháp học có giám sát phi tham số được phát triển lần đầu tiên bởi **Evelyn Fix** và **Joseph Hodges** vào năm **1951**, và sau đó được **Thomas Cover** mở rộng . Nó được sử dụng để phân loại và hồi quy.



Bản đồ của 1NN

Áp dụng vào bài toán

Sử dụng GridSearch để tìm tham số tốt nhất

```
#Sử dụng Grid Search để tìm tham số tốt nhất  
param_grid = {'n_neighbors': np.arange(1, 25),  
              'weights': ['uniform', 'distance'],  
              'metric' : ['manhattan', 'euclidean', 'cosine', 'minkowski']}
```

```
Best parameters found: {'metric': 'manhattan', 'n_neighbors': 3, 'weights': 'distance'}
```

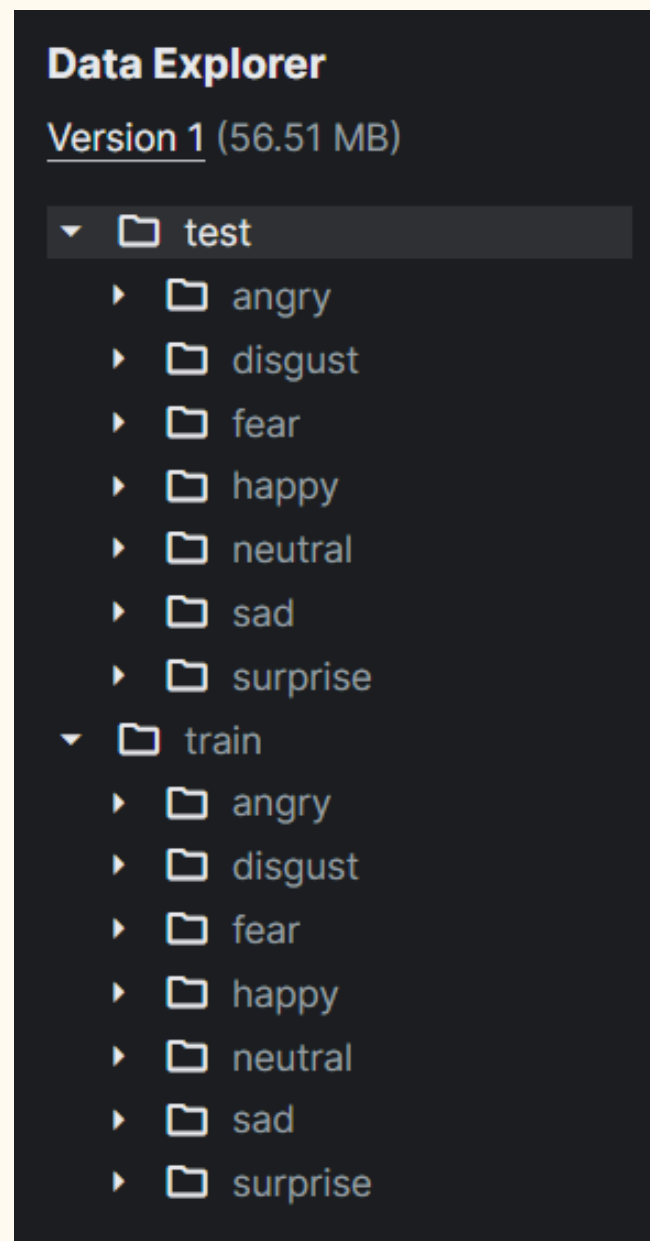
Bộ tham số tốt nhất là:

- n_neighbors: 3
- weights: 'distance'
- metric: 'manhattan'

Fer - 2013

MANAS SAMBARE

Dữ liệu bao gồm các hình ảnh khuôn mặt có thang độ xám 48x48 pixel.



Dataset

**Tập
train**



80 % train

20% validaion

emotions = ['disgust', 'fear', 'happy', 'surprise']

Thực nghiệm

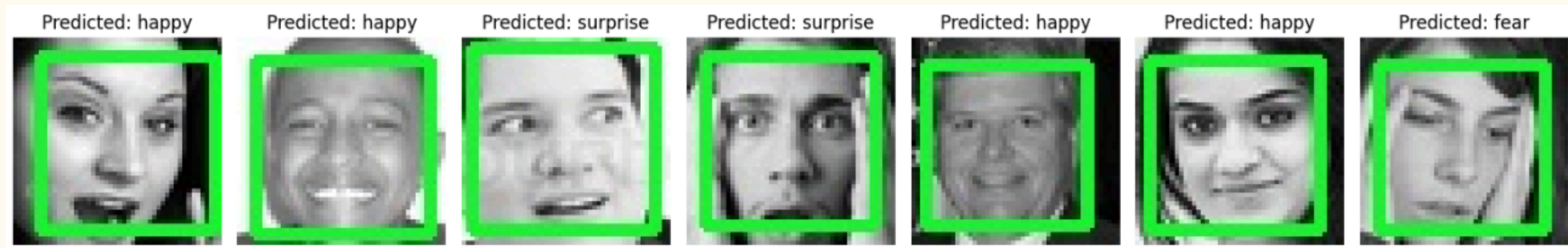
Độ đo đánh giá: accuracy

	Validation	Test	
SVM	80.16%	80.93%	
KNN	76.88%	78.60%	

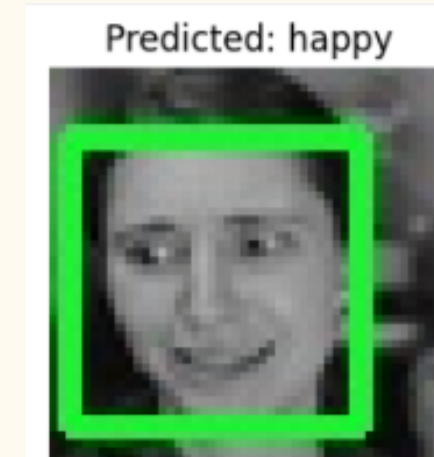
Kết quả

Dự đoán bằng SVM

Dự đoán đúng



Dự đoán sai





Demo



Xin cảm ơn!

Cảm ơn Thầy và các bạn đã lắng nghe!