CMC UNIVERSITY

FACULTY OF COMPUTER SCIENCE

CMC UNIVERSITY

Aspire to Inspire the Digital World

Project Report:

Develop an application for face detection based on machine learning models

Members:	Lê Xuân Thành
	Hà Khánh
	Nguyễn Mạnh Thắng
Class:	23CS - GM

Team Role

Number	Team member	Task
1	**Lê Xuân Thành**	**ALL**
2	**Hà Khánh**	**ALL**
3	**Nguyễn Mạnh Thắng**	**ALL**

Part I: Introduction

1. Background and Significance of the Problem

Face detection technology plays a crucial role in various real-world applications, including security, authentication, and user interaction. In security systems, accurate facial detection helps enhance surveillance, access control, and identity verification. In the field of artificial intelligence, it contributes to advancements in biometric authentication, ensuring secure and efficient user identification.

For students like us, developing a face detection application offers an opportunity to apply theoretical knowledge to a practical problem. This project not only emphasizes the significance of biometric technologies but also allows us to explore the challenges and methodologies involved in facial recognition systems.

2. Research Objectives

As a team working on a final course project, our objective is to investigate and implement fundamental techniques for face detection and recognition. We aim to understand the basic principles behind facial recognition, utilize publicly available datasets, and apply machine learning methods such as image processing

The purpose of this study is to develop a simple yet effective approach that demonstrates how facial data can be processed and analyzed. Through this project, we hope to strengthen our research and technical skills while creating a practical application for face detection.

3. Scope and Limitation

a. Scope:

We want to input a picture into the model and see if the result show the model detect any faces or not.The image must be split into multiple sub-images

b. Limitations:

The model can not detect multiple faces in one single image

PART II: Overview of related technology and concept

1. Histogram of Oriented Gradients (HOG)

The Histogram of Oriented Gradients (HOG) is a popular feature descriptor technique in computer vision and image processing. It analyzes the distribution of edge orientations within an object to describe its shape and appearance. The HOG method involves computing the gradient magnitude and orientation for each pixel in an image and then dividing the image into small cells.

**) Steps to process

Step 1: Calculating Gradients (direction x and y)

Step 2: Calculate the Magnitude and Orientation with formula:

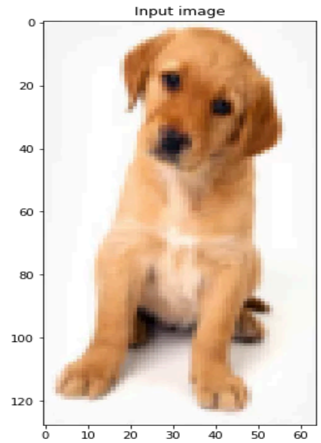
Magnitude = $\sqrt{[(G_x)^2+(G_y)^2]}$

Orientation : $\Phi = atan(Gy / Gx)$

Step 3: Calculate Histogram of Gradients in 1 cells

Step 4: Normalize gradients in 1 block:

Ex:



2. Support Vector Machine (SVM)

-) Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. While it can handle regression problems, SVM is particularly well-suited for classification tasks.

-) SVM aims to find the optimal hyperplane in an N-dimensional space to separate data points into different classes. The algorithm maximizes the margin between the closest points of different classes.

Some common applications of SVM are:

- Face detection – SVMc classify parts of the image as a face and non-face and create a square boundary around the face.
- Text and hypertext categorization – SVMs allow Text and hypertext categorization for both inductive and transductive models. They use training data to classify documents into different categories. It categorizes on the basis of the score generated and then compares with the threshold value.
- Classification of images – Use of SVMs provides better search accuracy for image classification. It provides better accuracy in comparison to the traditional query-based searching techniques.
- Bioinformatics – It includes protein classification and cancer classification. We use SVM for identifying the classification of genes, patients on the basis of genes and other biological problems.
- Protein fold and remote homology detection – Apply SVM algorithms for protein remote homology detection.
- Handwriting recognition – We use SVMs to recognize handwritten characters used widely.

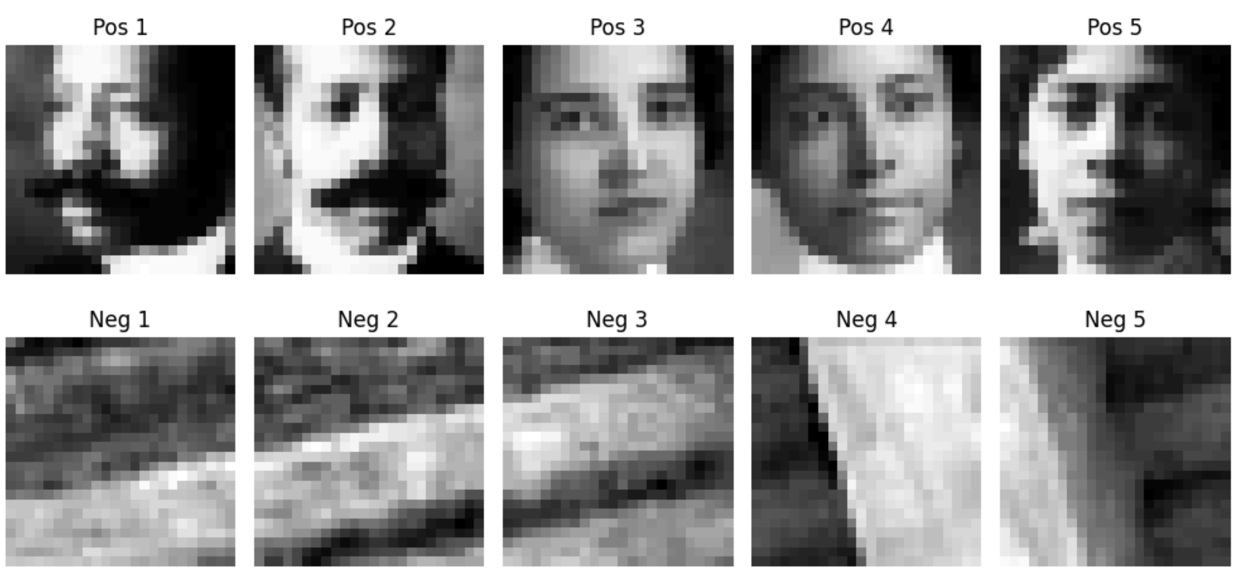
PART III: Detailed solution content

1. Dataset

The files contain grayscale image samples, likely used for training a face detection model.

- **possamples.mat stores** positive samples (images containing faces) in **24x24 pixel format**.
- **negsamples.mat includes** negative samples (images without faces) in **24x24 pixel format**.

Ex:



2. Pre-processing

a. Undersampling

```
data_pos = loadmat('possamples.mat')
data_neg = loadmat('negsamples.mat')
possamples = data_pos['possamples']  # 4000 samples
reduced_negsamples = data_neg['negsamples'][:, :, :8000]  # Limit to 8000 samples
```

The **negsamples** have 17000 pictures but **possamples** have only 4000 pictures so we limit **negsamples** to 8000 pictures which helps the model not to be biased towards the class with a larger number of samples.
Shapes of training data:

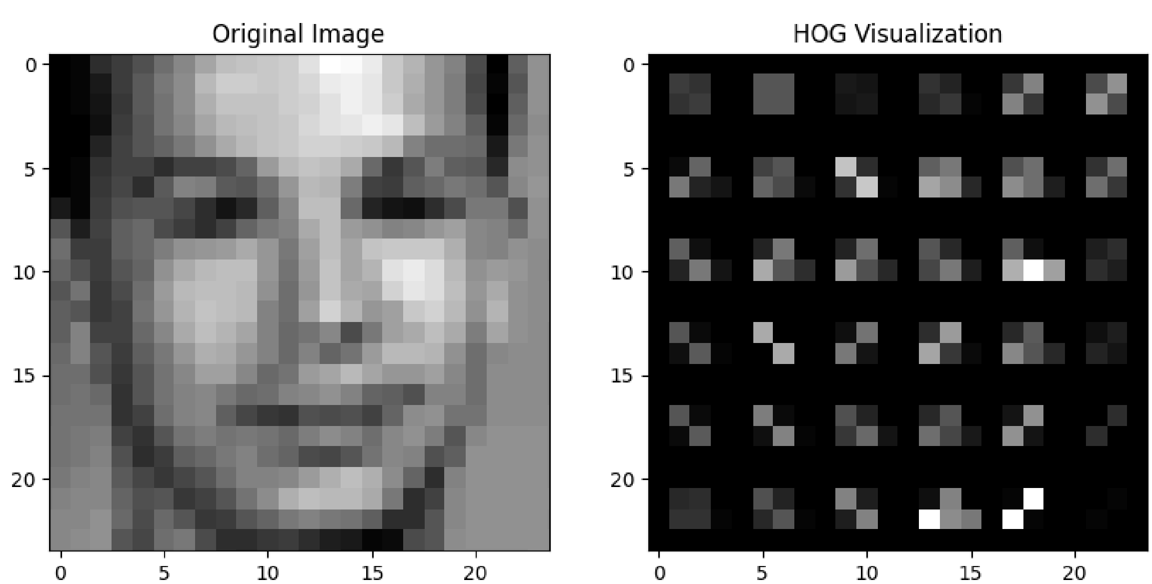
X_train: (24, 24, 12000) y_train: (12000,)

b. Apply hog

Apply **hog** for each pictures with parameter: 4x4 pixel per cell, 3x3 cell per block, 9 features we get

Shape of HOG features - Train: (12000, 1296)

Example:



3. Model: Support Vector Machine (SVM)

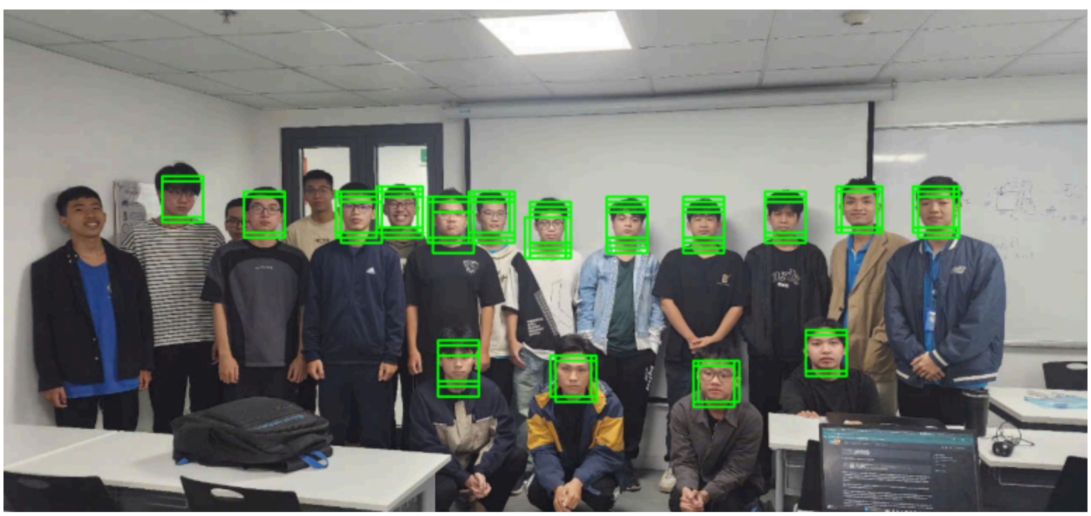
```
svm = SVC()

grid_search = GridSearchCV(svm, param_grid, cv=5, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train_hog, y_train)
```

- Use GridSearchCV to find best parameter
- Here, SVC() initializes an **SVM** model from sklearn.
- The model has no specific hyperparameters (C, kernel, gamma, etc.) because **GridSearchCV** will optimize them.
- Uses **5-fold cross-validation** to evaluate each hyperparameter combination.
- Uses **accuracy** as the evaluation metric to find the best model.

4. Post-processing:

Implement Non-Maximum Suppression (NMS) to filter out overlapping detections and reduce false positives

We only keep images identify as human face with >= 90% confidence.



5. Evaluation of results

a. Evaluation

Best model parameters: {'C': 1, 'gamma': 'scale', 'kernel': 'poly'}

Best cross-validation score: 0.994

- **C = 1:**

This parameter controls the trade-off between achieving a low error on the training data and keeping the model as simple as possible. A value of 1 means the model balances well between fitting the training data and generalizing to new data.

- **gamma = 'scale':**

The gamma parameter affects the influence of a single training example. When set to 'scale', gamma is automatically computed as 1/(number of features * variance of X), which adapts to the data's characteristics.

- **kernel = 'poly':**

This specifies that the SVM is using a polynomial kernel, which allows the model to learn non-linear relationships in the data by mapping it to a higher-dimensional space.

- **Validation Accuracy =** 0.9940 **(or 99.4%):**

This means that about 99.4% of the examples in the validation set were correctly classified by the model using these parameters, indicating very high performance.

b. Result

- **Face Detection Quality:**
  - The combination of SVM and HOG successfully detected most of the faces in the image.
  - Faces are enclosed within green bounding boxes, indicating that the model performs quite accurately.
- **Strengths:**
  - The model detects faces well, even when they are slightly tilted or partially obscured.
  - HOG is a powerful feature extraction method, providing stable face detection even under imperfect lighting conditions.
  - SVM works well with HOG for classifying and recognizing faces in the image.
- **Limitations and Errors:**
  - Some faces may not be detected due to poor lighting or challenging angles.
  - If the image has low resolution or the faces are too small, SVM + HOG may struggle to detect them accurately.
  - HOG may have difficulty handling extreme variations in pose or facial expressions.

PART IV: Conclusion and future directions

1. Conclusion

In conclusion, combining SVM with HOG provides a straightforward yet effective method for face detection. While HOG efficiently extracts features like edges and gradients, SVM accurately classifies these features to identify faces. However, factors such as poor lighting, scale variations, and pose differences can challenge the system's performance. By applying preprocessing techniques, using multi-scale detection, and enhancing the training dataset, these issues can be mitigated. Overall, this approach offers a practical balance between simplicity and accuracy in many real-world applications.

2. How to improve

a. Low-Light or Poor Lighting Conditions

Problem:

Faces in poorly lit conditions can have low contrast and noisy features, making it hard for HOG to capture clear edge or gradient information.

Solutions:

- **Preprocessing Enhancements:**
    - **Histogram Equalization or Adaptive Histogram Equalization (CLAHE):** Enhance the contrast of the image to make facial features stand out more.
    - **Gamma Correction:** Adjust brightness levels to improve image clarity before feature extraction.
- **Noise Reduction:**
    - **Denoising Filters:** Applying a Gaussian or bilateral filter can help reduce noise while preserving important edges.
- **Training Data Augmentation:**
    - Include images with various lighting conditions during training. Simulating low-light conditions in your training data can help the classifier generalize better.

b. Scale Variations and Face Positioning

Problem:
Faces appearing at different scales (large vs. small) or off-center (e.g., in the corners) can be missed if the detector isn't scale-invariant.

Solutions:

- **Multi-Scale Detection:**
  - **Image Pyramids:** Process the image at multiple scales so that both small and large faces can be detected effectively.
  - **Sliding Window at Multiple Resolutions:** Ensure that your detection window can cover faces of varying sizes.

---

c. Pose Variations and Occlusions:

- **Augment the Training Dataset:**
  - Include faces with various poses (frontal, profile, tilted) and partial occlusions so the model learns to generalize.
- **Alternative Features or Models:**
  - Consider combining HOG with other descriptors such as LBP (Local Binary Patterns) which might capture texture details that are less sensitive to pose changes.
- Investigate more modern deep learning methods (e.g., CNN-based detectors like MTCNN) that inherently handle pose variations and occlusions better.

3. Reference

[HOG Feature Descriptor: Feature Engineering for Images]
Support Vector Machine | Face Detection