COL780/JRL780 : COMPUTER VISION

Assignment 3: Object Detection and Recognition

March 14, 2024

1 Introduction

In this assignment, we will focus on detecting and recognizing hands in images using the Histogram of Oriented Gradients (HOG) descriptor. The objective is to develop an algorithm to accurately identify whether a detected hand is open or closed.

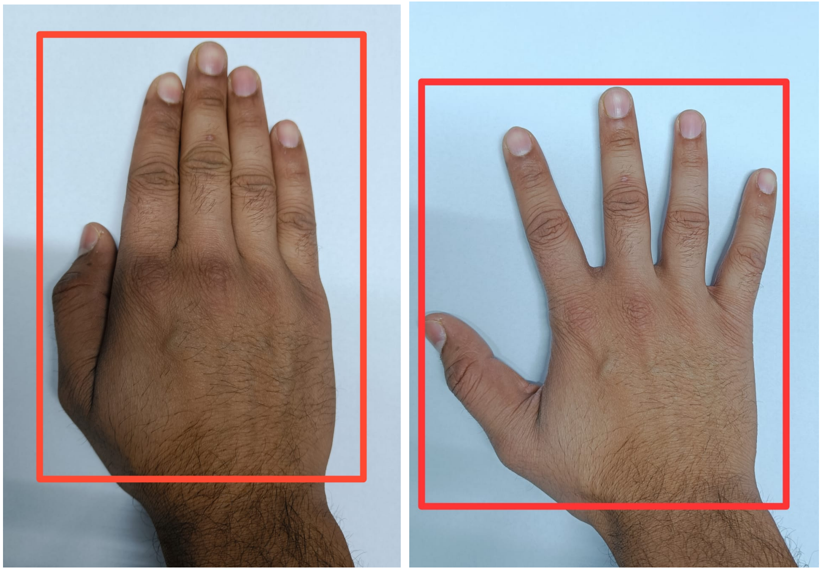

Figure 1: Example: Open and Closed hand

Hand detection and recognition have various applications in computer vision and human-computer interaction. Some of the key applications include:

- **Gesture Recognition:** Hand gestures can be used as a natural interface for controlling devices or interacting with virtual environments.
- **Sign Language Recognition:** Recognizing hand signs is important for enabling communication with individuals who use sign language as their primary mode of communication.
- **Human-Robot Interaction:** Detecting and understanding hand gestures can enhance human-robot interaction, enabling robots to interpret human commands and gestures more effectively.
- **Biometric Identification:** Hand recognition can be used for biometric identification and authentication purposes, such as access control and identity verification.
- **Virtual Reality (VR) and Augmented Reality (AR):** Hand detection and recognition are crucial for creating immersive VR and AR experiences, allowing users to interact with virtual objects using natural hand gestures.
- **Healthcare:** Hand detection and recognition can be used in healthcare settings for monitoring patient movements, rehabilitation exercises, and assisting surgeons during minimally invasive procedures.

- **Automotive:** Hand gesture recognition can be integrated into automotive systems for controlling infotainment systems, adjusting settings, and enhancing driver safety by detecting distracted or fatigued driving behavior.

2 Implementation Tasks

2.1 Task-1: Hand Detection

Detecting hands in images is a fundamental step towards hand recognition. For this task, you are required to implement a hand detection algorithm using the HOG descriptor.

- **Preprocessing:** Preprocess the input image as necessary for optimal hand detection. This may include grayscale conversion, resizing, and noise reduction.
- **Feature Extraction:** Compute the Histogram of Oriented Gradients (HOG) descriptor for each region of interest (ROI) in the image. Define appropriate ROIs where hands are likely to be present.
- **Training:** Train a classifier using the extracted HOG features. You can use machine learning algorithms such as Support Vector Machines (SVM) for this purpose. Annotated datasets for training may be provided or can be collected from publicly available hand datasets.
- **Detection:** Apply the trained classifier to detect hands in the test images. Use sliding windows or similar techniques for scanning the image at different scales and positions to detect hands efficiently.
  Once hands are detected, you can enclose the desired regions in rectangular boxes by drawing rectangles around them. This can be achieved using the coordinates of the detected regions provided by the detection algorithm.

2.2 Task-2: Hand Pose Recognition

Once hands are detected, the next task is to recognize whether a detected hand is open or closed. For this task, you are required to implement a hand recognition algorithm using the detected hand regions.

- **Feature Extraction:** Extract relevant features from the detected hand regions. These features may include texture, shape, or any other discriminative characteristics.
- **Classification:** Train a classifier to classify the detected hand regions into open or closed categories based on the extracted features. Again, SVM or similar classifiers can be used for this task.
- **Testing:** Apply the trained classifier to the detected hand regions to recognize whether they are open or closed.

2.3 Task-3-Bonus: Controlling Real-World Application through Hand gestures

For this you can create your own dataset with more hand-poses such as fist. Then add any real world application, such as forward, backward (using position of hand in the image with respect to image center), close etc using hand gestures.

3 Dataset and Evaluation

Dataset: Use the following dataset to train your model and report the test accuracy on the images in the test folder:

Here is a link to the dataset.

Evaluation: Your algorithm will be evaluated based on its accuracy in detecting hands and recognizing their states (open or closed) on a held-out test dataset. The evaluation criteria will include precision, recall, and F1-score for both hand detection and recognition tasks. Also plot ROC curves for your algorithm and report AUROC.

4 Submission Guidelines

You are required to submit the following:

- Python code implementing the hand detection and recognition algorithms.
- A main.py file that can be executed to run the algorithms on a given set of test images.
- A detailed report documenting your approach, experimentation details, and results.

Submission should be zipped with the file name ENTRY NUMBER.zip (e.g., 2019CS50768.zip) and uploaded to Moodle.

5 Note

- You are not allowed to use any pre-built object detection libraries or frameworks for this assignment.
- You are allowed to use opencv for SVM training, but feature generation should be from scratch.
- Ensure that your code follows good coding practices and is well-commented for clarity.
- Plagiarism will result in a zero score for the assignment and may lead to further consequences.
- Ensure that your implementation meets the required deadline.

6 Grading Rubric

The assignment will be graded based on the following criteria:

Task	**Marks**
Task-1 (Hand Detection)	40
Task-2 (Hand Recognition)	40
Task-3 - Bonus (Real World use)	10
Report	20

Table 1: Evaluation Criteria

7 Acknowledgments

This assignment is an individual effort. Any discussions or collaborations with peers should be acknowledged in the report.