

Project: Face Mask Detection, Classification, and Segmentation

Total Marks: 15 Marks

Deadline: 24th March

Objective

Develop a computer vision solution to classify and segment face masks in images. The project involves using handcrafted features with machine learning classifiers and deep learning techniques to perform classification and segmentation.

Dataset

- A labeled dataset containing images of people with and without face masks can be accessed here: <https://github.com/chandrikadeb7/Face-Mask-Detection/tree/master/dataset>
- A Masked Face Segmentation Dataset with ground truth face masks can be accessed here: <https://github.com/sadjadrz/MFSD>

Tasks

- a. Binary Classification Using Handcrafted Features and ML Classifiers (4 Marks)
 - i. Extract handcrafted features from the dataset.
 - ii. Train and evaluate at least two machine learning classifiers (e.g., SVM, Neural network) to classify faces as "with mask" or "without mask."
 - iii. Report and compare the accuracy of the classifiers.
- b. Binary Classification Using CNN (3 Marks)
 - i. Design and train a Convolutional Neural Network (CNN) to perform binary classification on the same dataset.
 - ii. Try a few hyper-parameter variations (e.g., learning rate, batch size, optimizer, activation function in the classification layer) and report the results.
 - iii. Compare the CNN's performance with the ML classifiers.
- c. Region Segmentation Using Traditional Techniques (3 Marks)
 - i. Implement a region-based segmentation method (e.g., thresholding, edge detection) to segment the mask regions for faces identified as "with mask."
 - ii. Visualize and evaluate the segmentation results.
- d. Mask Segmentation Using U-Net (5 Marks)
 - i. Train a U-Net model for precise segmentation of mask regions in the images.
 - ii. Compare the performance of U-Net with the traditional segmentation method using metrics like IoU or Dice score.

Submission Guidelines

- a. Submit your work as a GitHub repository link: VR_Project1_[YourName]_[YourRollNo]
- b. All code must be written in Python and submitted as:
 - i. A single script or notebook for binary classification tasks (handcrafted features and CNN).
 - ii. A separate script or notebook for segmentation tasks (traditional techniques and U-Net).

- c. README file:

The README file must serve as a detailed report for the project. It should include:

- i. Introduction: Briefly describe the objectives of the project.
 - ii. Dataset: Mention the dataset details, including its source and structure.
 - iii. Methodology: Clearly explain the steps taken for each task, including feature extraction, model training, and segmentation techniques.
 - iv. Hyperparameters and Experiments: For CNN and U-Net models, describe the hyperparameters used and variations tried, along with their results.
 - v. Results: Present the evaluation metrics (accuracy, IoU, Dice score, etc.) for all tasks, comparing different approaches.
 - vi. Observations and Analysis: Summarize insights gained from the results, challenges faced, and how they were addressed.
 - vii. How to Run the Code: Provide step-by-step instructions to execute the scripts.
- d. Ensure the code runs without additional intervention, and all outputs are clearly labeled.