VR Mini-Project 1

Ishan Singh(IMT2022124), Vansh Sinha (IMT2022122)

March 25, 2025

1 Introduction

The goal of this project is to develop a computer vision solution to detect and segment face masks in images. The project involves two primary tasks: binary classification (mask/no mask) and segmentation of mask regions in images. The tasks are approached using both traditional machine learning techniques and deep learning methods, including Convolutional Neural Networks (CNN) and U-Net for segmentation.

2 Dataset

The dataset used in this project consists of images of individuals with and without face masks, which are used for classification tasks. For the segmentation task, a separate dataset of cropped face images with ground truth masks is used. These datasets can be accessed from the following sources:

- Face Mask Detection Dataset: GitHub Link
- Masked Face Segmentation Dataset: GitHub Link

3 Methodology

3.1 Binary Classification Using Handcrafted Features and Machine Learning Classifiers

For the classification task, handcrafted features such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and color histograms were extracted from the images. Three machine learning classifiers were trained to classify the images as either "with mask" or "without mask":

- **Support Vector Machine (SVM)**: A linear kernel was used with default hyperparameters.
- **Neural Network (MLP)**: A Multi-layer Perceptron (MLP) classifier was trained with a hidden layer architecture.

- **XGBoost**: A gradient boosting classifier was used to improve prediction accuracy.

The models were evaluated using F1-score, precision, recall, and accuracy on the validation set.

3.2 Binary Classification Using Convolutional Neural Networks (CNN)

For the CNN approach, a Convolutional Neural Network was designed to perform binary classification. Several hyperparameter variations, including learning rate, batch size, optimizer, and activation functions, were experimented with to optimize the model performance. The CNN model performed feature learning directly from the raw image data, which led to a significant improvement over the handcrafted feature-based models.

3.3 Region Segmentation Using Traditional Techniques

Traditional image segmentation methods were applied to segment the mask regions in images of people wearing masks. The methods include:

- **Thresholding**: Binary thresholding was applied to detect mask regions based on pixel intensity.
- **Otsu's Thresholding**: An automatic thresholding technique was used to improve the segmentation results.
- **Edge Detection**: The Canny edge detection algorithm was employed to identify the edges of the mask.
- **Watershed Segmentation**: A more advanced segmentation technique that uses distance transform and marker-based region growing.

Each method was evaluated based on its Intersection over Union (IoU) and Dice score, comparing the predicted mask to the ground truth.

3.4 Mask Segmentation Using U-Net

U-Net, a deep learning model specifically designed for image segmentation tasks, was used for precise mask segmentation. The U-Net architecture consists of an encoder-decoder structure with skip connections, which is particularly effective in handling segmentation tasks with high spatial resolution. The model was trained using the binary cross-entropy loss function and optimized with the Adam optimizer. The performance was compared with traditional segmentation methods using metrics such as IoU and Dice score.

4 Results

4.1 Classification Performance

The performance of the machine learning classifiers (SVM, Neural Network, and XGBoost) was evaluated on the validation set. The Neural Network achieved the highest F1-score, with a test F1-score of 0.9372 among the ML classifiers. The CNN model, when compared to the machine learning classifiers, outperformed them significantly in terms of both accuracy and F1-score achieving 0.967 F1 Score.

4.2 Segmentation Performance

The traditional segmentation methods were evaluated based on the IoU and Dice scores. The results showed that Otsu's thresholding method had the highest average IoU and Dice score among all the traditional methods. Traditional methods were visualised for a sample image whose results can be seen in the Figure 1.

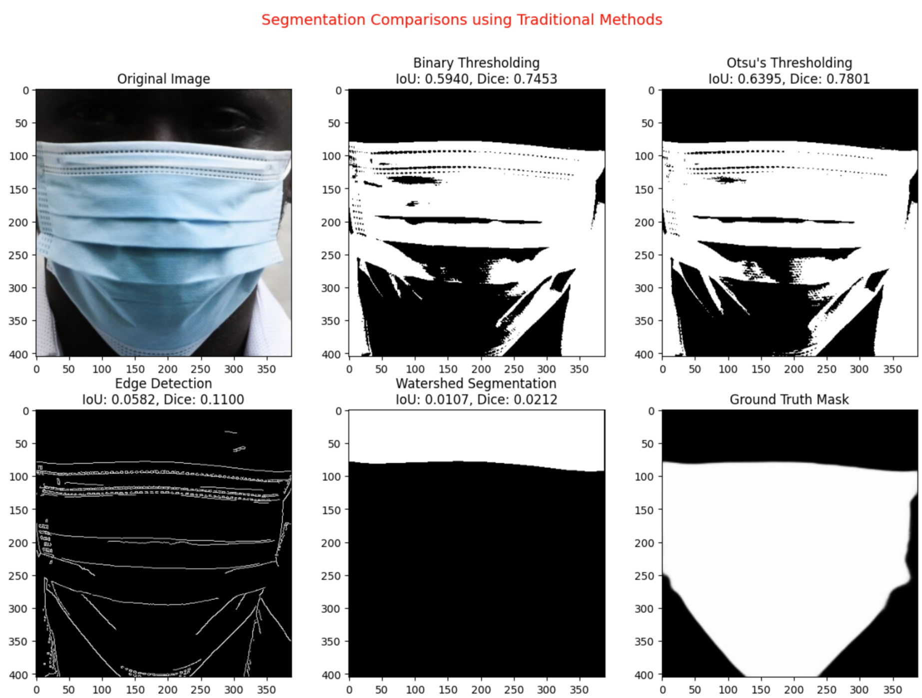

Figure 1: Image Segmentation with traditional methods

But U-Net outperformed all the traditional methods and provided the most accurate results, with significantly higher performance. The U-Net results are as follows:

- **U-Net IoU**: 0.5655
- **U-Net Dice Score**: 0.7219

5 Observations and Analysis

5.1 Challenges

The primary challenge in the classification task was the handling of images with varying lighting conditions and facial orientations, which sometimes led to inaccurate predictions. In the segmentation task, traditional methods struggled with complex mask boundaries, and their performance was greatly influenced by the quality of the input image.

5.2 Insights

Deep learning models, particularly CNN and U-Net, demonstrated superior performance in both classification and segmentation tasks. The ability of CNNs to learn features directly from the data gave them a distinct advantage over traditional feature extraction techniques. U-Net, with its encoder-decoder structure, significantly outperformed traditional methods, particularly in handling intricate mask regions.

6 How to Run the Code

To execute the code for this project:

- Install the required Python libraries, including `tensorflow`, `opencv-python`, `scikit-learn`, and `xgboost`.
- Clone the project from the provided GitHub link.
- Download the datasets from the github links.
    1. For classification, download the repo, select the dataset folder from the repo and rename it to classification_dataset and put it in the same directory as the notebooks.
    2. For segmentation, download the MFSD dataset from github link and rename MFSD to segmentation_dataset and put it in the same directory as the notebooks.
- Execute the Python scripts for classification and segmentation tasks (Run all cells in both classification.ipynb and segmentation.ipynb) .
- For segmentation tasks, a GPU is recommended to accelerate the training of the U-Net model.