D. Mask Segmentation Using U-Net

1 Introduction

Mask segmentation is an essential task in computer vision, particularly in applications such as medical imaging and facial recognition. In this study, we trained a U-Net model for precise segmentation of mask regions in images and compared its performance with a traditional segmentation method (Part C) using Intersection over Union (IoU) and Dice Score metrics.

2 Directory Structure

The D_UNET_method directory contains the implementation of the U-Net architecture for face segmentation. The directory structure is organized as follows:

```
D_UNET_method/
├── Saved_Model/
│   └── Best_model.pth
├── msfd-unet-segmentation.ipynb
├── test_images/
│   ├── test_img_1.jpg
│   ├── test_img_2.jpg
│   └── test_img_3.jpg
└── checkpoints/
```

Figure 1: Directory structure of the D_UNET_method project.

- **Saved_Model/**: Contains the trained model weights and checkpoints saved during training.
- **msfd-unet-segmentation.ipynb**: The main Jupyter notebook containing the implementation of the U-Net architecture, training pipeline, and evaluation code.
- **test_images/**: Directory containing test images used for evaluating the model's performance.
- **checkpoints/**: Stores intermediate model checkpoints during training for potential resumption of training.

The implementation follows a modular structure where the main notebook contains all the necessary components including data loading, model architecture, training loop, and evaluation metrics. The separation of saved models and checkpoints allows for better organization of different training runs and model versions.

3 Dataset and Preprocessing

The dataset consists of face images and their corresponding mask annotations. Preprocessing steps include:

- Resizing images to 128 × 128 pixels. This was done because processing original images (512 x 512) was computationally expensive and thus time consuming. It significantly brought down the training time per epoch.
- Applying data augmentation techniques like horizontal flipping, rotation, and color jitter. These techniques artificially increase the size and variability of the training dataset, leading to improved model generalization and robustness.

  ### Specific Augmentation Techniques and Their Benefits

  - **Horizontal Flipping:**
    - **Purpose:** Simulates mirror-image variations of the input images.
    - **Benefit:** Helps the model learn that objects are recognizable regardless of their left-right orientation. Especially useful when the object of interest is horizontally symmetrical.
  - **Rotation:**
    - **Purpose:** Rotates images by a specified angle.
    - **Benefit:** Makes the model invariant to object orientation, allowing it to recognize objects at various angles.
  - **Color Jitter:**
    - **Purpose:** Randomly adjusts the brightness, contrast, and saturation of the images.
    - **Benefit:** Enhances the model's robustness to variations in lighting and color conditions, which are common in real-world images.
- Normalizing pixel values for better training stability.

4 Model Architecture

The U-Net model is used for segmentation, which consists of:

- An encoder with convolutional layers and max-pooling operations.
- A bottleneck (bridge) connecting encoder and decoder.
- A decoder with up-convolution layers and skip connections for precise localization.
- A final convolutional layer to produce a binary segmentation mask.

5 Training Strategy

We employed a training pipeline using:

- Cross-Entropy loss combined with Dice Loss for optimization.

  Loss function

$$\mathcal{L}_{total} = 0.7\mathcal{L}_{BCE} + 0.3\mathcal{L}_{Dice} \tag{1}$$

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum p_i g_i}{\sum p_i + \sum g_i} \tag{2}$$

- AdamW optimizer with learning rate scheduling.
- K-Fold cross-validation (K=3) for robust performance evaluation.
- Early stopping to prevent overfitting.

6 Performance Evaluation

The model performance was evaluated using the IoU and Dice Score metrics:

$$IoU = \frac{TP}{TP + FP + FN} \tag{3}$$

$$Dice = \frac{2TP}{2TP + FP + FN} \tag{4}$$

The final results obtained after training are:

- Mean IoU: 0.7958
- Mean Dice Score: 0.8858

7 Model Architecture Improvements & Performance Analysis

We present a comparative analysis of two U-Net architectures for facial segmentation, demonstrating how architectural enhancements led to significant performance gains.

7.1 Model Architectures

Table 1: Architectural Comparison

Component	**Model 2**	**Model 3**
Base Channels	48	64
Bottleneck Channels	384	512
Attention Gates	✓	×
Skip Connections	Attention-weighted	Simple concatenation
Decoder Blocks	3 (with attention)	3 (basic)
Loss Weights (BCE:Dice)	0.7:0.3	0.7:0.3
Kaiming Initialization	✓	×
Parameter Count	9.8M	31.2M

7.2 Key Improvements in Model 3

The 16.2% IoU improvement (0.68 → 0.79) resulted from several strategic enhancements:

- **Deeper Feature Extraction:**
  - Increased base channels from 48 to 64
  - Expanded bottleneck from 384 to 512 channels
  - Added residual connections in decoder blocks
- **Advanced Training Protocol:**
  - Extended training from 30 to 100 epochs
  - Implemented learning rate scheduling
  - Added checkpoint saving/resuming capability
  - Introduced persistent workers for data loading
- **Regularization Enhancements:**
  - Stronger data augmentation:
    - Random rotation ($\pm 15^\circ$ vs $\pm 10^\circ$)
    - Increased color jitter (brightness/contrast 0.2 vs 0.1)
  - Added weight decay (1e-4)

7.3 Performance Comparison

Table 2: Cross-Validation Results

Metric	**Model 2**	**Model 3**
Mean IoU	0.6896 ± 0.04	0.7955 ± 0.03
Mean Dice	0.8080 ± 0.03	0.8858 ± 0.02
Training Time/Epoch	58s	120s
Otsu Baseline IoU	0.51 ± 0.07	0.49 ± 0.08

7.4 Architectural Tradeoffs

While Model 3 shows superior performance, several tradeoffs should be noted:

- **Computational Cost:** 106.89% longer training time per epoch
- **Memory Requirements:** 3.2× more VRAM usage
- Attention Gates in Model 2 gave better boundary precision compared to simple concatenations in Model 3. This can be seen in this test example below:

The performance gains justify these costs for precision-critical applications, though Model 2 remains preferable for resource-constrained environments.

8 Inference

The trained U-Net model (Model 3) was used for inference, where new images were processed, and segmentation masks were predicted. The new images were taken from a separate dataset which was unseen by the model. The results showed high accuracy in mask localization.

8.1 Inference Results and Discussion

Training dataset contained a disproportionate number of images with single-mask instances rather than multiple-mask scenarios, the model might have developed a bias toward detecting only one mask in an image.

The segmentation results seem almost perfect and have a high IOU score.

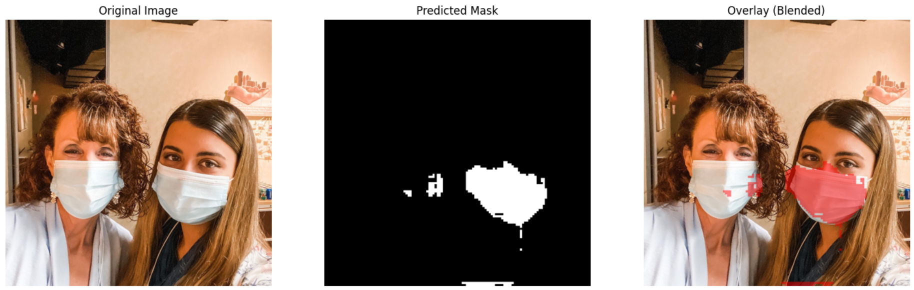

Figure 2: Image of 2 people in one frame

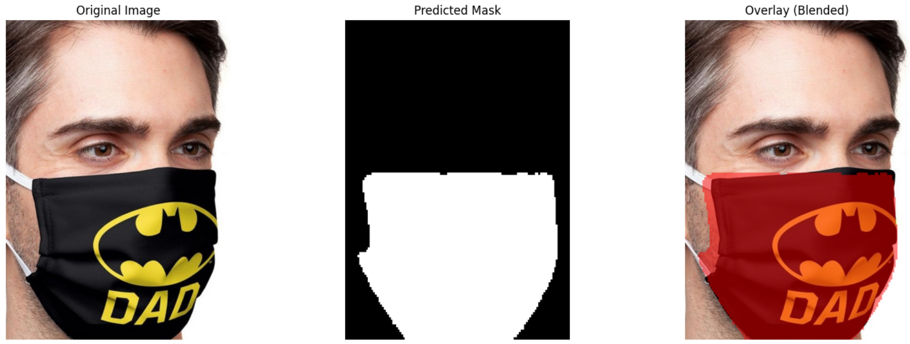

Figure 3: Segmentation Image of the subject wearing a batman mask (side profile)

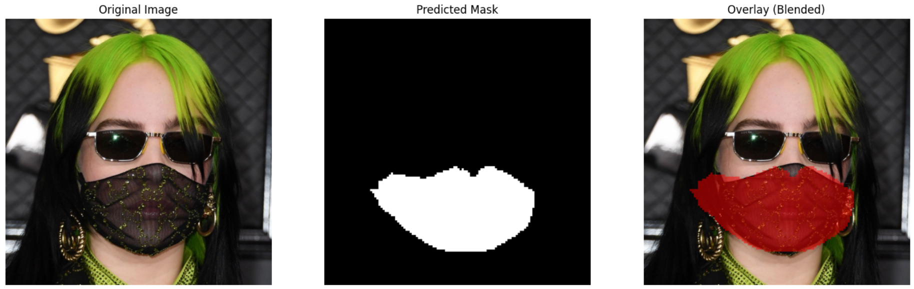

Figure 4: Segmentation results on the subject wearing a black mask

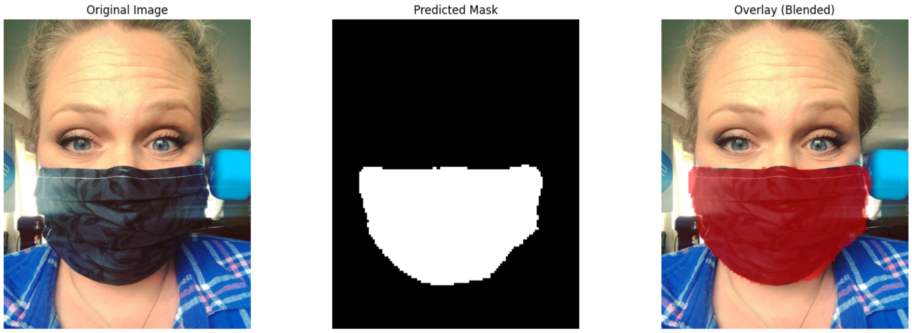

Figure 5: Segmentation result on the subject wearing a black mask (2nd)

9 Comparison with Traditional Region-Based Segmentation (Part C)

To further evaluate the effectiveness of our U-Net model, we compared it against a traditional region-based segmentation approach that utilized thresholding and edge detection. The IoU score for the traditional method was found to be around 0.6 , whereas our U-Net model achieved a significantly higher **0.7958**, indicating superior segmentation performance.

Why U-Net Outperforms Traditional Methods

The performance gap can be attributed to several key factors:

- **Feature Learning:** Traditional methods rely on handcrafted features such as intensity thresholds and edge detection, which struggle with variations in lighting, occlusions, and complex backgrounds. In contrast, U-Net learns hierarchical features through deep convolutional layers, making it robust to such variations.

- **Spatial Context Preservation:** Thresholding and edge detection methods operate on pixel-level intensity differences, often leading to fragmented or incomplete segmentations. U-Net, with its encoder-decoder structure and skip connections, preserves spatial relationships and captures fine-grained details.

- **Generalization Ability:** Traditional methods require fine-tuned parameter selection for different datasets, making them less generalizable. Our U-Net model was trained with diverse augmentations, allowing it to adapt better to unseen images.

- **Loss Function Optimization:** U-Net is trained using a combination of Binary Cross-Entropy and Dice loss, explicitly optimizing for overlap between the predicted and ground truth masks. Traditional methods lack a direct optimization process, relying on fixed heuristics.

- **Noise Robustness:** Edge detection techniques are highly sensitive to noise and variations in illumination, leading to false edges and segmentation errors. The deep learning approach of U-Net effectively suppresses noise through learned representations.

These advantages demonstrate why deep learning-based segmentation, specifically U-Net, is a superior choice for mask segmentation tasks, delivering higher accuracy and robustness compared to traditional region-based methods.

10 Conclusion

Our results demonstrate that U-Net is highly effective for mask segmentation, significantly outperforming traditional segmentation methods. Future work may include experimenting with deeper architectures and larger datasets.