Received: 3 July 2025 | Revised: 4 January 2026 | Accepted: 22 January 2026 | Published online: 1 February 2026

RESEARCH ARTICLE

Artificial Intelligence and Applications
2026, Vol. 00(00) 1–15
DOI: 10.47852/bonviewAIA62026690

BON VIEW PUBLISHING

An Encoder–Decoder-Based Deep Learning Model for Segmenting Occlusion in the Lower Part of the Face

Mrinmoy Sadhukhan[1,*], Indrajit Bhattacharya[2], Paramartha Dutta[1], and Kaushik Roy[3]

[1] *Department of Computer & System Sciences, Visva-Bharati University, India*

[2] *Department of Computer Applications, Kalyani Government Engineering College, India*

[3] *Department of Computer Science, West Bengal State University, India*

Abstract: This paper presents a deep learning-based model for accurate segmentation of facial occlusions from the lower facial region. Unlike many existing image segmentation methods that rely heavily on bounding box annotated datasets, the proposed model eliminates the need for such supervision, thereby improving generalization to unseen data. The system generates a binary mask to identify occluded facial areas caused by masks, incorrectly worn masks, and niqabs (where only the forehead and eye regions remain visible). Segmenting such occlusions becomes particularly difficult in datasets with varied types of valid and invalid masks or other similar obstructions that were not present during training. To tackle this, the architecture of the UNet model is modified with the addition of self-attention block. The model is trained using an augmented dataset and validated on multiple benchmark datasets. To evaluate its practical deployment potential, the model is tested on edge devices within a simulated environment. The proposed model exhibits enhanced performance when compared to current state-of-the-art approaches, achieving a remarkable 99.62% training accuracy and 99.48% validation accuracy with a minimal training and validation loss of 0.01 and 0.0172, respectively. Additionally, the model accurately segments diverse facial masks, enabling the identification of individuals attempting to conceal their identities. This capability also supports facial reconstruction by restoring occluded regions, thereby enhancing security applications. Its design balances high accuracy with broad applicability, making it a robust solution for facial occlusion handling and recognition.

Keywords: face occlusion, segmentation, UNet, self-attention

1. Introduction

In recent times, intruder detection has gained significant importance in both public and private spaces such as malls, offices, hospitals, and similar environments, particularly in the aftermath of COVID-19, as a large number of people now wear face masks for their safety. The widespread adoption of face masks in public is driven by their dual advantages: they not only offer protection against dust, pollution, and the transmission of viruses through droplets and aerosols but also conceal facial expressions in public. Unfortunately, this growing trend of mask usage for identity concealment has facilitated illicit activities in public areas, including streets, tourist attractions, and religious sites. This emerging issue underscores the urgent need for a precise pixel-level occlusion segmentation system capable of efficiently segmenting masks from faces in minimal time and under diverse mask type. The segmented regions can later be utilized to reconstruct occluded facial areas using the generated binary masks, which might be useful for facial identification.

This work proposes a segmentation model based on the UNet architecture for precise segmentation of occlusions from lower part of the face. Different layers of the UNet model are customized, keeping the main architecture intact to meet the requirements of the present work. An augmented dataset of masked faces is generated by superimposing diverse face masks onto the existing images in the CelebA [1] dataset. Upon training with this augmented dataset, the model demonstrates robust performance across diverse facial postures not limited to frontal views and accurately segments both properly worn masks and irregular coverings (e.g., handkerchief, towel, niqab). Optimized for edge deployment, the system enables occlusion segmentation in IoT environments, with performance validated through simulated testing.

The main contribution of this work is as follows:

1) Development of customized UNet architecture that can segment the occluded area (e.g., a facial mask) from the lower part of the face.
2) Incorporation of attention blocks into the deeper layers of the model to enhance its ability to effectively manage different scenarios, such as properly and improperly worn face masks.
3) Segmentation of masks (of varying sizes, shapes, and types, including unconventional face coverings objects like handkerchiefs and niqabs) across diverse face postures.

For further elaborations the article is divided into distinct sections. Section 1 introduces the problem along with the study's contribution. Section 2 reviews related works. Section 3 outlines the proposed methodology and its implementation. Section 4 provides experimental details. Section 5 illustrates the limitation of the proposed system. Finally, Section 6 concludes the paper with a summary of the findings.

[*] **Corresponding author:** Mrinmoy Sadhukhan, Department of Computer & System Sciences, Visva-Bharati University, India. Email: 03333342201@visva-bharati.ac.in

2. Related Work

Segmenting occlusions from face images is crucial in facial part regeneration. This process has two main components: first, detecting the occluding object within the face image, and second, accurately segmenting it from the rest of the image. In recent years, computer vision has made great strides with advanced machine learning and deep learning algorithms, enabling rapid and precise segmentation of occlusions or masks from facial images pixel-by-pixel with remarkable accuracy. The following sections delve into the details of these leading algorithms, explaining how they address the challenging task of segmenting occlusions in facial images.

2.1. Machine learning-based approach

Saeed et al. [2] proposed a hybrid k-nearest neighbors (k-NN) model for MRI tumor segmentation, combining GrabCut, support vector machines (SVM), and a hidden Markov model (HMM) with K-Means clustering. Chen et al. [3] proposed an adaptive histogram-based granulation and reciprocal rough entropy thresholding method for image segmentation. The approach dynamically adjusts both granule size and threshold values based on grayscale intensity distributions, enabling robust separation of foreground and background regions without manual parameter tuning. By integrating variable precision rough set theory, the method improves segmentation accuracy under noise and complex intensity variations. Mittal et al. [4] reviewed clustering-based image segmentation techniques, providing a detailed overview of their pros and cons.

2.2. Deep learning-based approach

Pixel-level classification involves the segmentation of an image by grouping the pixels belonging to the same object class by dividing an image into multiple segments or objects. With the emergence of deep neural networks (DNNs), image segmentation has made tremendous progress. The deep learning-based image segmentation task can be divided into semantic segmentation and instance segmentation. Several studies have surveyed convolutional neural network (CNN) segmentation models for semantic and instance-based methods [5, 6]. In the following sections, some of the state-of-the-art models are discussed.

2.2.1. Semantic segmentation-based approach

Semantic segmentation consists of assigning each pixel in an image a class label that matches the object that has been identified and draws a bounding box around it. One of the most popular methods to achieve this task is the fully convolutional network (FCN) [7], which replaces the fully connected layer with a 1×1 convolution and uses an upsampling method to obtain a pixel-wise output (label map) of the image. Another widely used technique is the UNet [8, 9] architecture, an encoder–decoder-based FCN that captures image features using a series of convolutions with max pooling layers, followed by upsampling of encoded transformations using a transposed convolution network. The feature maps from the encoder layer are concatenated with the output of the previous upsampling layer to incorporate better contextual information. UNet-based semantic segmentation has been used in various applications, such as facial mask detection [10] and medical instrument segmentation [11]. Meenpal et al. [10] used UNet-based semantic segmentation for facial mask detection, a modified UNet version was used with VGG16 (Visual Geometry Group)-based FCN as a backbone, and achieved 94.682% accuracy. Kurmann et al. [11] proposed a shared encoder–decoder-based UNet architecture, where two decoders were fed with one encoder network. One decoder produced the mask of medical instruments, while another decoder created the offsets of the mask. Subsequently, the two outputs were combined to achieve an accurate mask. In the final step, every single instance of the instrument was classified with different mask colors. Teliti et al. [12] reviewed different semantic segmentation models and showed a landmark-guided semantic segmentation model where CNN first predicted the landmarks of a face; this information was then used to segment the face from the whole image. Mahmoud et al. [13] proposed an autoencoder model, GANMasker, to segment the facial mask from face images and regenerate the occluded part based on the segmentation mask. They used the convolutional block attention module (CBAM) block as an attention mechanism between the connection from the encoder layer to the decoder layer to enhance feature representation. This CBAM block consists of channel attention and spatial attention mechanisms. This proposed segmentation model achieved 99% accuracy. Ye et al. [14] proposed a transformer based face reconstruction and face mask segmentation model, DeMaskGAN. The proposed model has a transformer reconstruction head (TRH) that restores the features of the masked face. It also uses the transformer segmentation head to help the TRH focus on the masked face area and rebuild the face to an unmasked state while keeping the identity information. They used the MobileFaceNet model to check the accuracy of the reconstructed facial parts and achieved 97% accuracy.

2.2.2. Instance segmentation-based approach

Instance segmentation is a specialized form of image segmentation that involves detecting object instances and separating them by their boundaries. This type of segmentation is beneficial in scenarios where multiple objects of similar types need to be identified individually. One example is the UNet-based instance segmentation model proposed by Wagner et al. [15], which extracts buildings from satellite images using three identical UNet (UNet-id) models that produce different color levels for the border, segmentation map, and inner image. The model is trained with WorldView-3 satellite RGB images and secured 97.67% accuracy. Another approach is the Mask Transfiner network proposed by Ke et al. [16], which decomposes image regions into a quadtree and uses a transformer-based method to correct error-prone nodes, resulting in highly accurate masks at a low cost. The extended Mask R-CNN framework called MaskPlus [17] uses a Faster R-CNN model with a feature pyramid network (FPN) structure as the backbone and employs several techniques such as contextual fusion, deconvolutional pyramid module, improved boundary refinement, quasi-multitask learning, and biased training to improve segmentation performance. Chen et al. [18] introduced a self-attention-based UNet architecture for reconstruction bias in optical remote sensing. They have customized the UNet architecture using a self-attention block as a transformer block to learn channel and position weights from input images and sit between the input image and feature extraction module. They achieved an Intersection over Union (IoU) score of 73.49%, with an impressive 84.72% F1 score. Varghese et al. [19] proposed the YOLOv8 model for object detection, which can be used for segmentation. YOLOv8 model with 640×640 image dimension achieves 43.4 mAP for segmentation mask creation and 53.4 mAP for segmentation box detection. Recent YOLO variants have progressively enhanced real-time segmentation capabilities through architectural and training improvements. YOLOv9 [20] introduces Programmable Gradient Information (PGI) to strengthen feature learning and stabilize training, leading to improved boundary localization and multiscale feature representation for segmentation tasks. YOLOv10 [21] simplifies the detection pipeline by adopting an end-to-end optimization strategy that removes non-maximum suppression, resulting in faster inference and more consistent segmentation outputs with integrated segmentation heads. YOLOv11 [22] further refines the architecture with improved backbone design and enhanced feature fusion strategies, enabling more accurate pixel-level predictions while maintaining real-time efficiency across detection and segmentation applications. Kirillov et al. [23] introduced the Segment Anything model (SAM) from meta-researchers. It has been trained on a dataset of 11 million images and 1.1 billion masks, with strong zero-shot performance on a wide range of tasks and datasets. SAM can be used for various applications, such as object detection, instance segmentation, semantic segmentation, panoptic segmentation, and interactive segmentation. SAM v2 [24] extends the original Segment Anything framework by supporting both image and video segmentation with improved temporal consistency and multi-modal prompts, enabling more robust and flexible pixel-level segmentation. The researcher released the SAM and corresponding dataset (SA-1B) of one billion masks and 11 million images to foster research for further development of foundation models for computer vision.

Previous studies have shown significant progress in segmentation techniques, demonstrating high effectiveness. However, these methods often demand large volumes of training data and substantial computational resources, making them costly and less practical, especially when dealing with the images containing facial mask or occlusion. Many researchers have proposed a segmentation model using autoencoder or UNet architecture with some modifications to the CNN block and have achieved impressively up to 99% accuracy (pixel accuracy) in generating the segmentation map, but their performance in other segmentation metrics such as SSIM, meanIoU, and Dice coefficient was comparatively lower, as detailed in quantitative comparisons. While models may perform well in terms of training and validation accuracy, this performance alone does not conclusively establish their overall effectiveness. For segmentation tasks, comprehensive performance evaluation requires additional metrics such as SSIM, meanIoU, and Dice coefficient where the model should perform well. Moreover, prior research does not consider the segmentation of incorrectly worn face masks or unfamiliar occlusions, which are not used during the time of training the neural network. To address this issue, a modified UNet based segmentation network is designed to focus specifically on mask segmentation.

3. Methodology

The general layout of the proposed system is shown in Figure 1. It comprises two main components: 1) preprocessing of images and 2) map module. The specific details of each module are elaborated in the following sections.

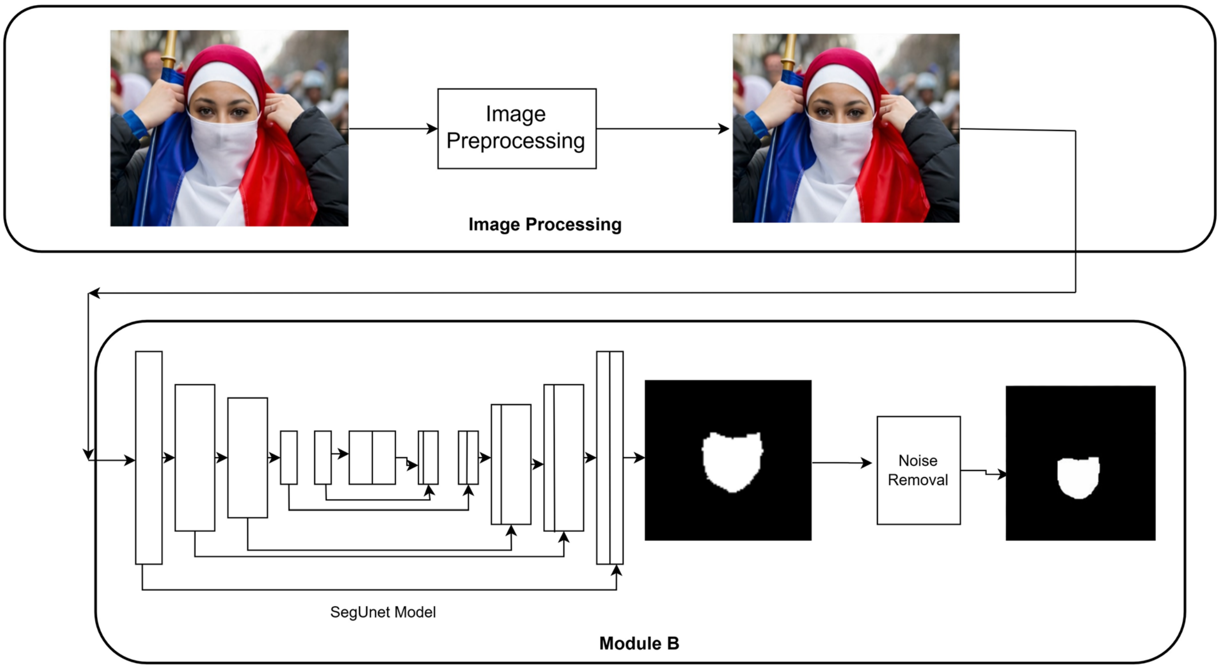
3.1. Preprocessing of images

The preprocessing stage is fundamental in the development and training of any CNN architectures. Determining the input image dimensions at the input layer is necessary, as it affects the model's performance, structural complexity, and computational efficiency. The face regions captured from video streams are generally small, mainly due to the camera's limited resolution and the distance between the subject and the lens. Consequently, processing high-resolution images is not essential. Smaller image dimensions are preferred to achieve faster inference and optimize resource usage without significantly affecting segmentation accuracy. Considering these factors, an input size of 128×128 is chosen for the proposed model. Prior to training, all images were resized to this specified dimension to ensure a smooth and efficient training process.

3.2. Map module

The map module processes an input face image containing a mask and produces a binary segmentation map. In the generated map, regions marked in black hold a pixel value of 0, and white regions hold a pixel value of 1. The white regions accurately indicate the masked areas on the face, while the black regions represent the remaining facial part. The map module utilizes a modified UNet neural network with a CNN based encoder–decoder architecture; its design is illustrated in Figure 2. In the proposed neural network architecture, some well-known techniques are used like batch instance normalization [25], spectral normalization [26], and self-attention block [27]. Spectral normalization is used in the proposed model to stabilize the training process by regulating weight magnitudes and maintaining steady gradient propagation. Batch instance normalization is used to dynamically balance the benefits of batch and instance normalization, making it especially useful for tasks that require both content structure preservation and style invariance. This normalization method improves model adaptability, stability, and

Figure 1
Proposed architecture of the model



generalization across diverse data distributions. Additionally, leaky ReLu activation is applied after the batch instance normalization layer to introduce nonlinearity, addressing the vanishing gradient problem. The integration of a self-attention block into the modified UNet architecture, enables precise feature learning from masked face images while effectively preserving the boundaries of the mask regions. This technique also enables the model to produce a segmentation map of the face mask region even when the type of mask is unknown to the system. Although the self-attention block has some computational overhead, using it with lower image dimensions does not hinder the performance of the model in training and inferencing.

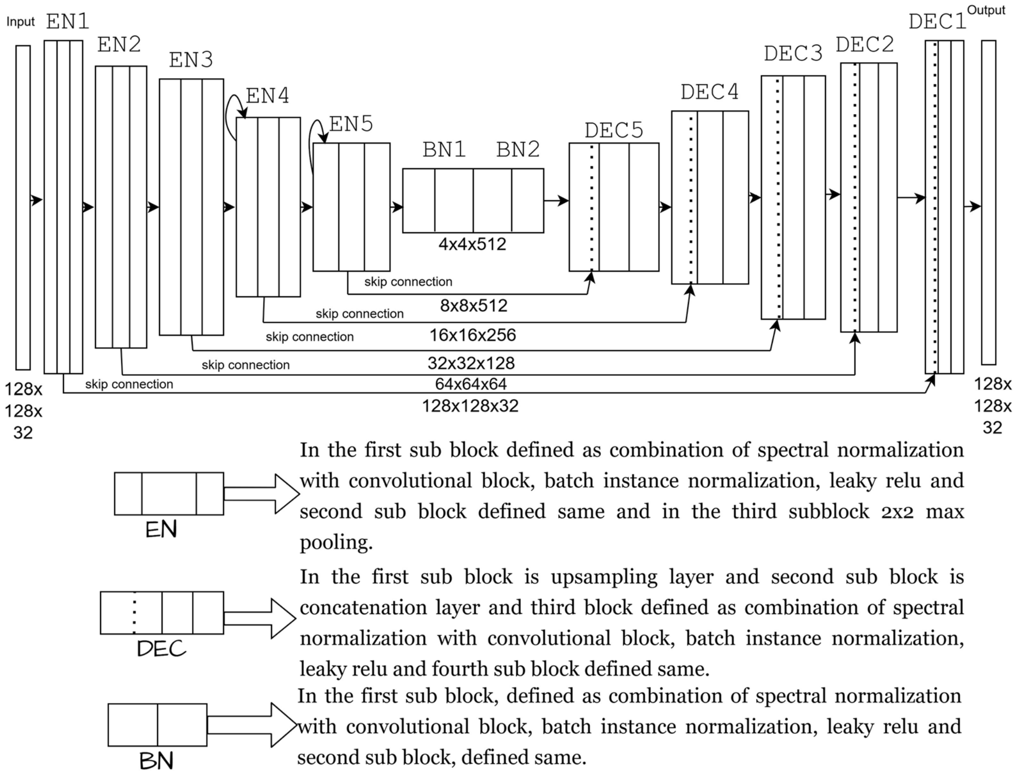
The modified UNet architecture in Figure 2, consists of a contracting path (encoder) and an expanding path (decoder) composed of several blocks, each consisting of several sub-blocks. In face mask segmentation scenarios, as the mask portion is significant and different complex information related to the face mask is to be fetched for accurate segmentation, the proposed modified UNet architecture incorporates five encoder blocks, five decoder blocks, and two bottleneck blocks. In the contracting path, each encoder block, referred to as "EN," typically comprises two convolutional layers. These are followed by a downsampling operation, such as max pooling, which reduces the spatial dimensions of the feature maps. Each convolutional block is paired with spectral normalization, batch instance normalization, and leaky ReLu.

Table 1 shows the details of each encoder block where, through all encoder blocks, a 3×3 convolutional channel is used with stride 1 in all convolutional operations. Blocks EN4 and EN5 are modified, which contain self-attention based convolutional layers instead of the first convolutional layer. In the expanding path, each decoder block, referred to as "DEC," begins with an upsampling operation that doubles the spatial dimensions of the previous layer's output. To facilitate the concatenation of feature maps between corresponding blocks in the contracting and expanding paths, the number of decoder blocks should match the number of encoder blocks. The concatenation operation merges the output of the upsampling layer with the skip connection, effectively combining global and local features within a unified representation. This concatenation layer is followed by two convolutional layers, each paired with spectral normalization, batch-instance normalization, and leaky ReLu activation. Table 1 also presents

Table 1
Different blocks of proposed model

	Name of components
Encoder block	
EN1 EN2 EN3	Convolutional layer with spectral normalization, batch instance normalization, leaky ReLu, and max pooling layer
EN4 EN5	Self-attention block, convolutional layer with spectral normalization, batch instance, normalization, leakyReLU, max pooling layer
Bottleneck block	
BN1 BN2	Convolutional layer with spectral normalization, batch instance normalization, and leaky ReLu
Decoder block	
DEC5 DEC4 DEC3 DEC2 DEC1	Upsampling layer, concatenation layer, convolutional layer with spectral normalization, batch instance normalization, and leaky ReLu

Figure 2
Modified UNet structure



In the first sub block defined as combination of spectral normalization with convolutional block, batch instance normalization, leaky relu and second sub block defined same and in the third subblock 2x2 max pooling.

In the first sub block is upsampling layer and second sub block is concatenation layer and third block defined as combination of spectral normalization with convolutional block, batch instance normalization, leaky relu and fourth sub block defined same.

In the first sub block, defined as combination of spectral normalization with convolutional block, batch instance normalization, leaky relu and second sub block, defined same.

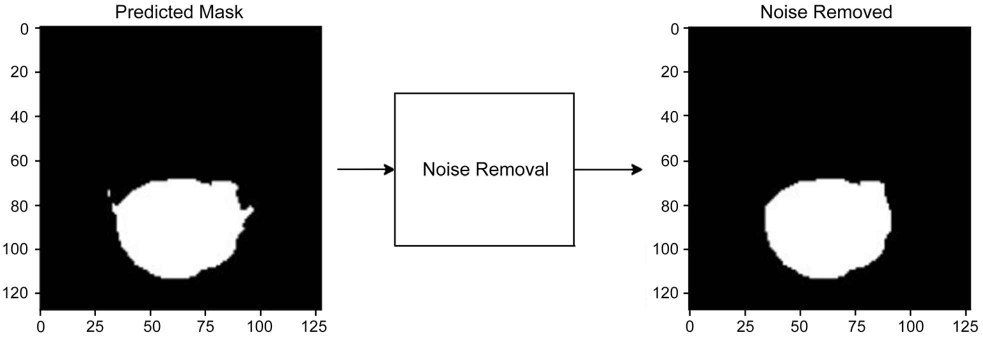
a detailed overview of the decoder block, where all the convolutional operations in the decoder blocks contain a 2×2 convolutional channel with a stride of 2. The bottleneck block (BN) is a specialized block in the architecture that connects the encoder and decoder paths. It captures the most salient features of the input image and facilitates the segmentation process. The bottleneck block enables the network to capture more global context while preserving local details, which is essential for accurate image segmentation. The bottleneck block typically consists of two convolutional layers, each paired with spectral normalization, batch instance normalization, and leaky ReLu. Table 1 is a detailed overview of the bottleneck block. In the bottleneck block a 3×3 convolutional channel is used with stride 1 in all convolutional operations. The output layer uses one convolutional layer with a sigmoid activation function and one color channel to map the mask position in the binary segmentation result. The main reason for using the sigmoid as an activation function is that it produces output values between 0 and 1 for each class independently. It is especially suitable for tasks like binary segmentation, where the classification of each pixel is independent of others. The model employs binary cross-entropy loss, which is particularly suitable for binary segmentation. This function computes per-pixel probability differences for both the predicted masks (class 1) and backgrounds (class 0) against the ground truth as a loss value.

The training of the proposed model is a complex problem because it has 17,657,825 trainable parameters that need to be trained using 14,000 images. This computationally intensive training process necessitates preliminary hyperparameter optimization via GridSearchCV, which is focused on learning rate, kernel dimensions, and weight initialization methods. Algorithm 1 presents the outline of a step-by-step process for model building, parameter initialization, and the training procedure for this model. In Figure 3, it has been presented that noise processing is applied on the predicted mask to get a clean mask. Here, an erosion operation with 7×7 kernel size, is initially applied to shrink the white areas in the binary mask, effectively eliminating small noise. This is followed by dilation, which restores the size of the white regions. This sequence of morphological operations, erosion followed by dilation, helps to remove noise while preserving the overall shape. It is applied during training as well as in validation of the proposed model. The model is trained on the augmented CelebA dataset to obtain superior results. The improvements of the proposed

Algorithm 1 Algo_model_Training

```
Input: augmented CelebA dataset with binary mask of the face mask region.
Output: Predicted Binary mask of the face mask region.
1: begin
2: Preprocess the input images to bring down the image dimension to 128×128.
3: Define the proposed model architecture with appropriate layers and activation function.
4: Split the whole dataset in 70% training, 10% validation set and 20% testing set.
5: Initialize the epoch size=250, batch size=8, learning rate=2e-4, beta=0.5 and
metrics=accuracy.
6: Initialize adam optimizer with previously defined learning rate.
7: Initialize Binary cross-entropy as loss function.
8: Initialize the counter i.
9: For i in the range of epoch
        a: The neural network fits with small batches of the training dataset and processes them
        entirely.
        b: Loss value between predicted and original segmentation mask is then calculated based
        on binary cross-entropy.
        c: Gradient is then back-propagated with loss value, learning rate, and adam optimizer
        to recalculate the weight of different trainable parameters for minimizing the loss
        between predicted and original output in the next iteration.
        d: The model is then tested on images from both the training and validation datasets, and
        its accuracy and loss are calculated based on these evaluations using the accuracy metric.
10: End of for loop.
11: Model after training is ready for segmenting the face mask from face images.
12: end
```

Figure 3
Noise processing

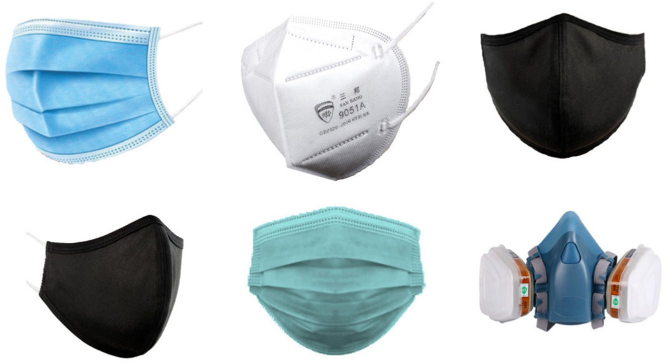


model over others, with some additional information related to model training, are further described in the Experiment and Results section.

4. Experiment and Results

This section covers the following topics: the method used to develop an augmented dataset, the training details of the proposed model, visualizations of the proposed model, comparisons with other state-of-the-art models, and evaluations of the proposed model's accuracy, binary mask generation capabilities for both correctly and incorrectly worn face masks.

Figure 4
Examples of synthetic mask variations



4.1. Augmented dataset generation

Currently, different publicly available masked face datasets exist on platforms like GitHub and Kaggle. They often contain multiple faces per image, making them incompatible with the proposed model, which requires single-face images with corresponding binary masks. To overcome this limitation, an augmented dataset is built using the publicly available CelebA dataset [1]. The augmented dataset comprises 20,000 high-quality masked face images, providing sufficient data to train a complex neural network effectively. The augmentation process involved overlaying 8 distinct mask types with 30 color/texture variations and 3 tilt positions onto faces. Representative mask samples are illustrated in Figure 4. MaskTheFace library [28] is employed to detect facial landmarks, estimate mask keypoints, compute face tilt, and dynamically place user-selected masks onto faces. Additionally, a modification is made in the library to generate binary masks for the augmented images. The dataset was systematically organized into training, testing, and validation subsets. Figure 5 showcases sample images from the augmented CelebA dataset.

4.2. Training details

In order to train the map module, the model accepts an input image and produces a segmented binary map that closely resembles the target binary map. The training workflow is illustrated in Figure 6. The process involved using 14,000 images for training, 2000 images for validation, and 4000 images for testing, all sourced from the augmented

Figure 5
Image samples derived from augmented CelebA dataset

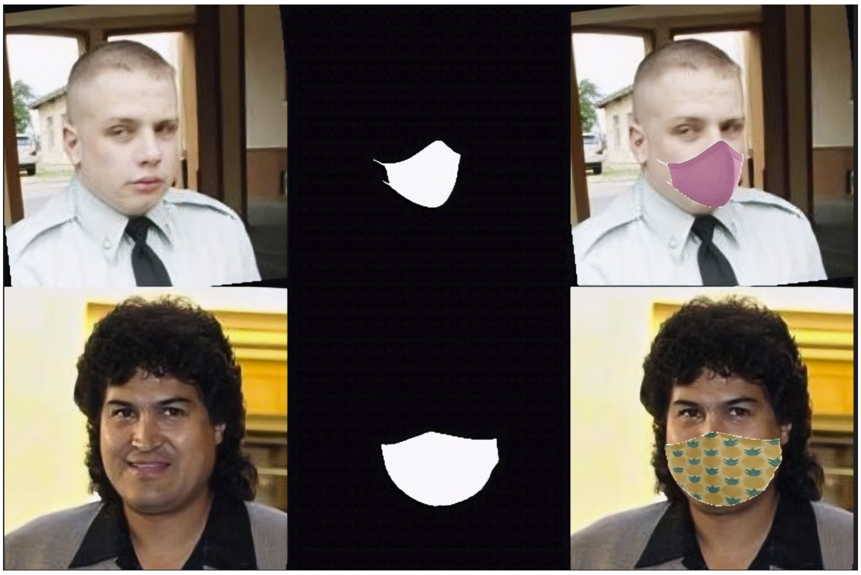
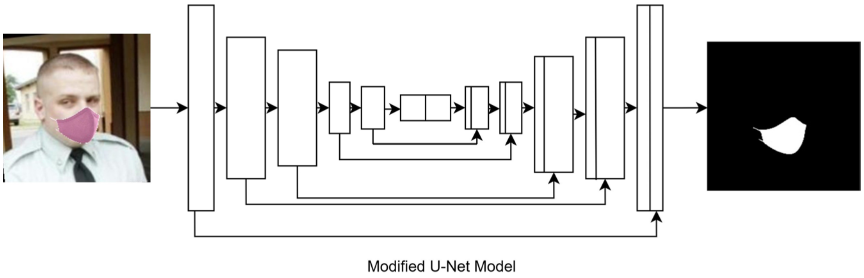


Figure 6
Architecture of training process



CelebA dataset. During training, images were resized to 128×128 pixels, a batch size of 8 was selected due to hardware constraints and adam optimizer was employed for faster convergence. Here, a relatively small image size is used to reduce computational complexity, which makes the training faster and less memory-intensive. The proposed UNet network was trained for 250 iterations using NVIDIA Jetson Xavier Nx with 8GB integrated GPU and CPU RAM.

Figure 7 shows the graph of model's training and validation accuracy versus number of epochs, while Figure 8 shows the model's training and validation loss. Alongside training the base UNet model on the same dataset, the corresponding accuracy and loss graphs are shown in Figures 7 and 8, respectively, for comparison purposes. The blue line represents the training results of the modified UNet, while the yellow line indicates its validation performance. The green line represents the training results of the basic UNet, and the red line corresponds to its validation results. The result presented in Figure 7 and Figure 8 shows that customized UNet demonstrates exceptional performance, achieving a consistent 99.62% training accuracy and maintaining a 99.48% validation accuracy with minimal validation loss of 0.0172.

Throughout training, it sustains a stable training loss of 0.01, confirming robust optimization. A minor performance dip to 96% occurs after epoch 120, attributable to challenges in generalizing across diverse mask types, dataset noise, normalization effects, and inherent overfitting tendencies. These limitations are common in deep learning architectures. Despite this transient fluctuation, the model maintains significant superiority in segmentation quality (Figure 9). In contrast, the base UNet achieves only 98% training accuracy and exhibits critical instability. Its accuracy repeatedly drops below 92% after 78 epochs, failing to sustain target performance levels. This inconsistency is mirrored in its loss profile, where the model struggles to reach a 0.01 loss value in training until 80 epochs and cannot maintain this threshold thereafter. The substantial training and validation loss directly correlates with erratic behaviour, resulting in significantly more erroneous segmentation maps during testing compared to the proposed model. Figure 9 presents a comparison of segmentation maps produced at different epoch intervals, where the first row represents model performance by customized UNet and the second row by base UNet, offering a comprehensive view of how customized model outperforms over base model in segmentation task.

Figure 7
Performance analysis based on accuracy values for both the basic and modified UNet models

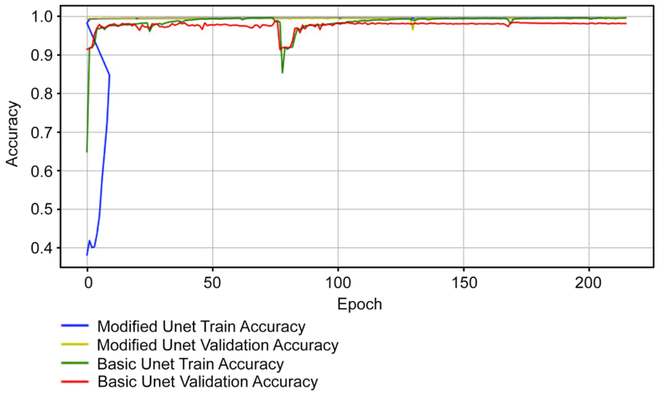
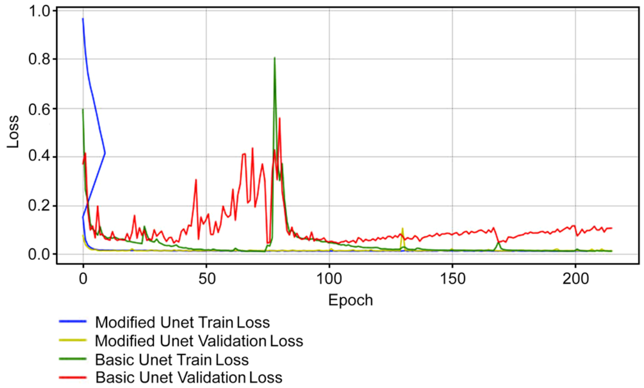


Figure 8
Performance analysis based on loss values for both the basic and modified UNet models



In Figure 8, it is evident that the customized UNet initially struggles to generate an accurate segmentation map primarily due to the model's increased complexity, which requires more time to effectively adjust its parameters and learn meaningful feature representations. As training progresses, the model gradually overcomes this slow learning phase and begins to predict the segmented binary mask with higher precision. This enhanced performance is achieved through the incorporation of a self-attention block in the EN4 and EN5 blocks, along with spectral normalization and batch instance normalization with leaky ReLu. The self-attention block proves instrumental in extracting precise segmentation maps from input images after a few iterations. This approach ultimately yields superior results.

4.3. Comparison

This section is dedicated to examining the outcomes produced by the proposed model when compared to other cutting-edge models. Different models have been chosen for comparison such as GAN based network [29] where comparison is made with map module only: Base UNet [8], Facial Mask Detector [10], Maskthenclassify [11], UNet-Id [15], MaskPlus [17], GANMasker [13] segmentation module, and YOLOv8-seg [19, 30].

4.3.1. Qualitative comparison

In this section, the performance of the proposed model is evaluated based on visual inspection, patterns, or subjective assessment. Augmented CelebA, MAFA, FMLD, and RMFD datasets were compared. The augmented CelebA dataset was the training dataset, and the rest of them were used for testing purposes. Figure 10 presents some images from MAFA dataset with face masks and their corresponding predicted binary map, in the shape of 128×128 image dimension. In Figure 11 and Figure 12, many face images are presented with different mask colors and postures, taken from testing portion of the augmented CelebA dataset. It can be observed that the model performs well in different scenarios irrespective of color, postures, and mask types. To check the validity of the proposed model over other datasets, the model was tested with MAFA, FMLD, and RMFD datasets and achieved an average 98.90% accuracy (Table 2).

To infer the model performance in real images, Dlib's HOG and Linear SVM-based frontal face and landmark detector is used to crop facial images. This detector can provide us with a total of 68 coordinates located on the mouth, right eyebrow, left eyebrow, right eye, left eye, nose, and jaw of any face. By visualizing the right eyebrow, left eyebrow, right eye, and left eye, all 68 coordinates of any face can be easily found. With the aid of these key points, the face can be cropped [12] from the original image. After cropping the face, a white background is added to make it fit for the map module as input to get a segmented mask. To analyze augmented CelebA, MAFA, FMLD, and RMFD datasets, the Kolmogorov–Smirnov test is applied.

Figure 9
Comparison of segmentation mask of proposed model (row 1) and base UNet model (row 2) in different epoch intervals taken from augmented CelebA dataset

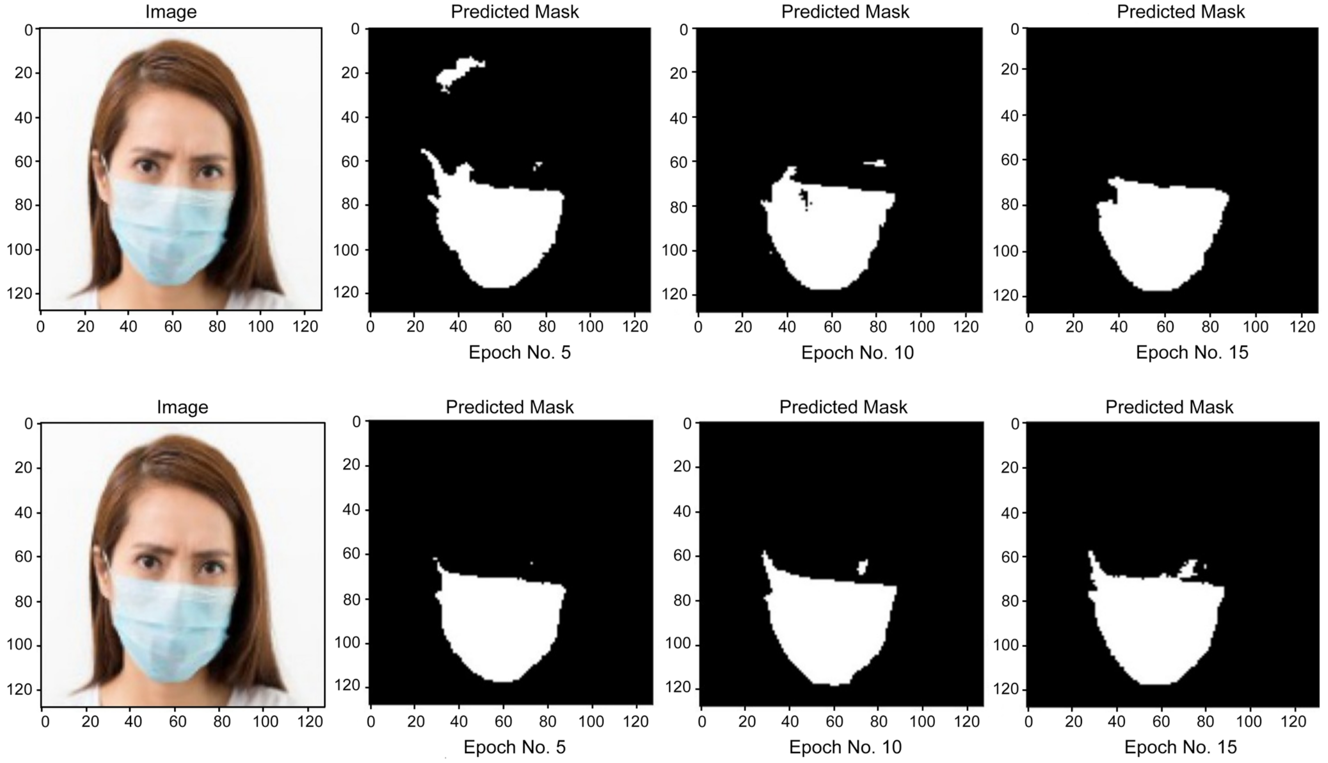
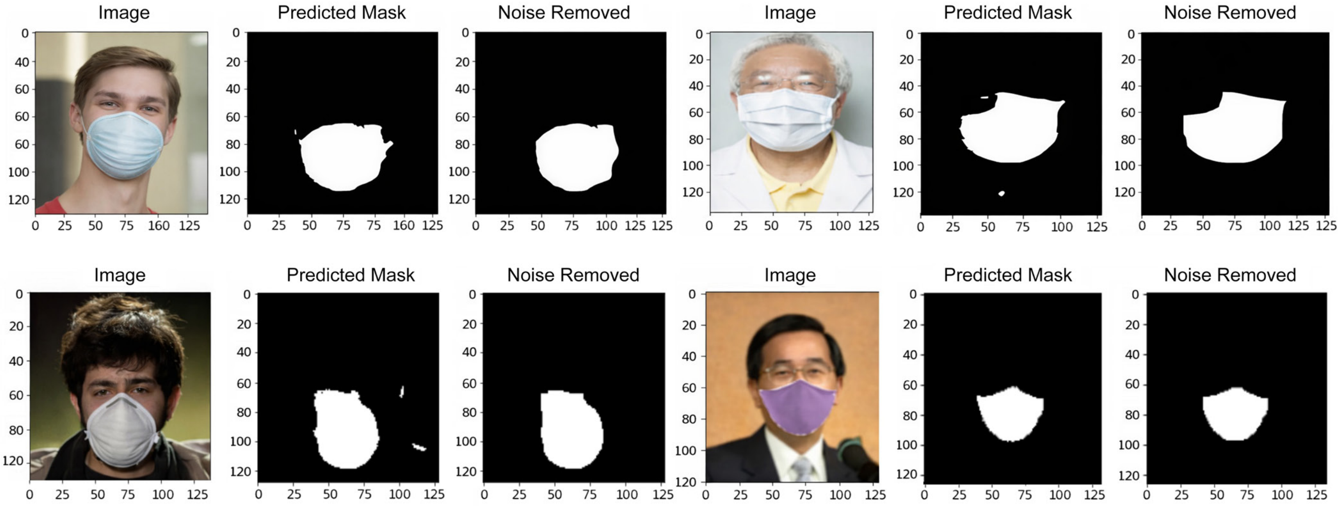


Figure 10
Segmentation output from the proposed model on MAFA test dataset



$$D = \max_{1 \leq i \leq N} \left(F(Y_i) - \frac{i-1}{N}, \frac{i}{N} - F(Y_i) \right) \tag{1}$$

In statistics, the Kolmogorov–Smirnov test (K–S test or KS test) is a non-parametric test, which helps us to determine if the probability distribution of these two datasets is different or the same. This further helps us to know if the datasets are a subset of a large dataset. In Equation 1, the Kolmogorov–Smirnov test function is given, where F represents the theoretical cumulative distribution of the distributions being tested. The dataset distribution must be continuous and cannot be a discrete distribution, such as the binomial or Poisson. Here, SciPy library was used to conduct the Kolmogorov–Smirnov test. It returns two parameters: D statistics (the maximum vertical distance between the empirical cumulative distribution functions (CDFs) of two samples) and P value (quantifies the strength of evidence against the null hypothesis). Here, a lower D statistic indicates that the two datasets are more similar and likely originate from the same underlying distribution, whereas a higher D statistic suggests greater dissimilarity between the datasets and a higher p-value suggests that the two datasets are more likely to come from the same source, while

Figure 11
Segmentation of face mask of different color taken from test portion of augmented CelebA dataset

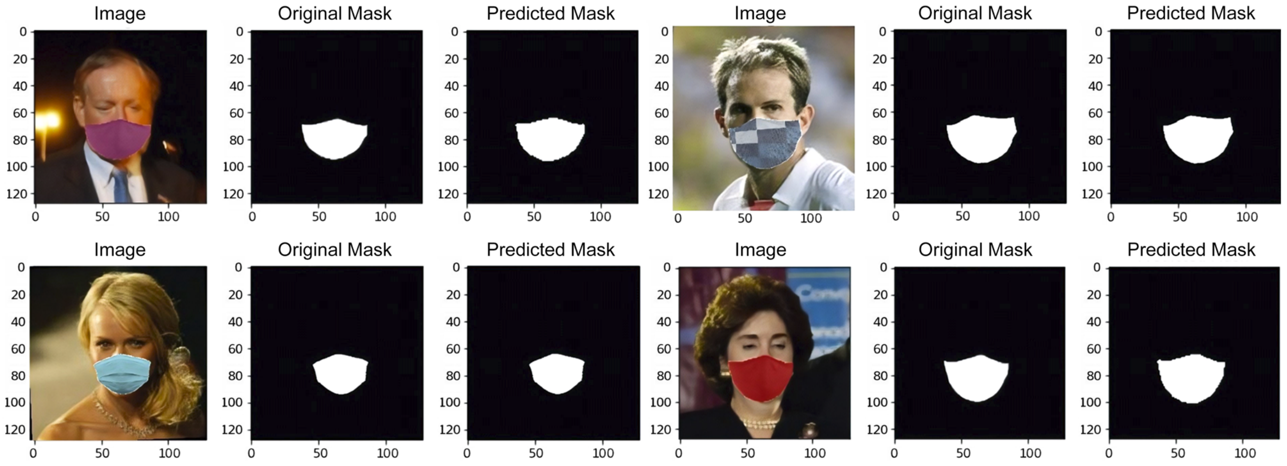
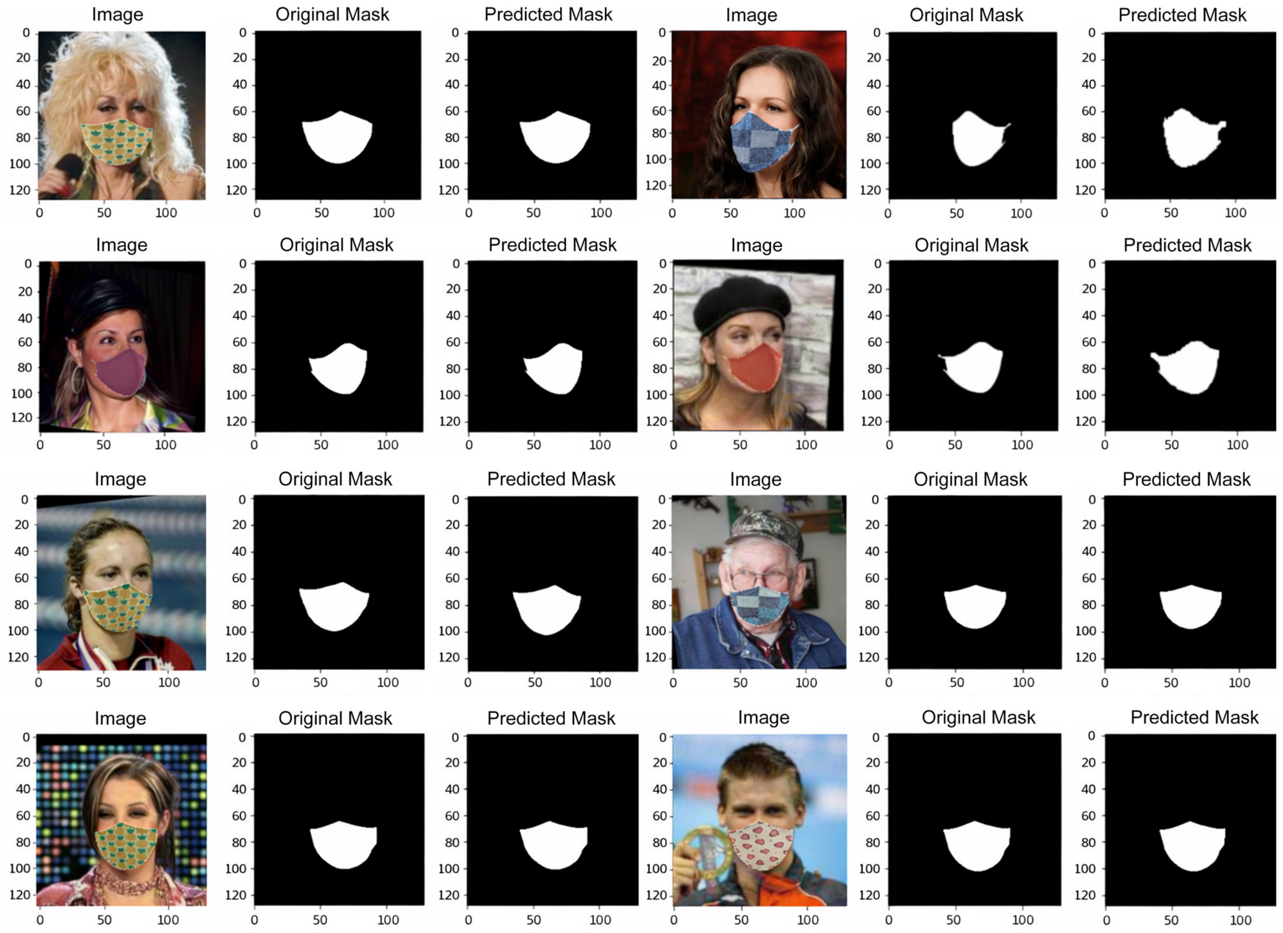


Figure 12
Segmentation of face mask from different face postures taken from test portion of augmented CelebA dataset



a lower p-value indicates a significant difference between them. To perform the KS test, the feature vector was first extracted from images using the local binary pattern (LBP) method with 24 sampling points and a radius of 8 pixels. After that, the array was flattened to make it one-dimensional so that it can comfortably fit with the KS-test function. Then, the KS test was applied to determine the D statistic value.

In Table 3, a comparison of different datasets is depicted to show the different D statistics and P values. When the KS test is applied to the training and validation datasets, it produces a very low D statistic value, which means that the datasets are drawn from the same source. In other cases, such as comparison with augmented CelebA dataset with MAFA, FMLD, or RMFD, they produce very high D statistic values, which means the datasets are different.

Table 2
Accuracy of proposed model over different dataset

Sl. No.	Dataset name	Accuracy (%)
1.	Augmented CelebA dataset	99.48
2.	MAFA (MAsked FAces)	99.30
3.	FMLD (Face Mask Label Dataset)	98.47
4.	RMFD (Real Masked Face Dataset)	98.86

Table 3
Statistical comparison of different dataset

Dataset 1	Dataset 2	D statistics	P
Training dataset created from augmented dataset	Validation dataset created from augmented dataset	0.00258	0.864
Augmented CelebA dataset	MAFA dataset	0.1082	0.001
Augmented CelebA dataset	FMLD	0.0952	0.002
Augmented CelebA dataset	RMFD	0.1634	0.001

4.3.2. Quantitative comparison

Here, the comparison of the proposed model with other SOTA models is presented with the help of different quantitative matrices like accuracy, Structural Similarity Index (SSIM), meanIoU, and Dice. The SSIM metrics aid in quantifying the visibility errors between the original image and the predicted image by calculating the degradation of structural information between the two images after extracting the structural information from the image. The SSIM quality assessment index is based on three factors: luminance, contrast, and structure of an image. Equation 2 is the formula for calculating SSIM:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_x y+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \tag{2}$$

where μ_x, μ_y, σ_x, σ_y, and σ_{xy} are the local means, standard deviations, and cross-covariance for images x and y, respectively. SSIM values range from 0 to 1, where 1 means a perfect match between the original image and the predicted image. Besides the SSIM, another metric, meanIoU, is used, which is one of the most fundamental methods used in computer vision and machine learning for the evaluation of image segmentation-related tasks. Equation 3 presents the formula for IoU, which measures how closely two sets of images overlap. MeanIoU calculates the ratio of the area overlapped by two images to the area of their union. IoU is calculated for every class and the average of this IoU is used to calculate the meanIoU of the image.

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

The Dice score, which is also called the Dice Similarity Coefficient, shows how similar two sets of data are to each other. They are usually shown as binary arrays. For example, in image segmentation, the Dice score can be used to see how similar a predicted segmentation mask is to the ground truth segmentation mask. The formula for the dice metric is presented in Equation 4:

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

Table 4 provides a quantitative comparison of the proposed model with several other networks, including the GAN based network [29], UNet [8], Facial Mask Detector [10], Maskthenclassify [11], UNet-Id [15], and MaskPlus [17]. Additionally, a comparison was made with the YOLOv8 seg [19, 30] model, an instance segmentation model originally designed for object detection and later adapted for segmentation using a modified version of UNet. During training, the YOLO models handle both detection and segmentation tasks simultaneously, which results in longer training times and requires being fed with both segmentation and bounding box annotations data that is not always available for any specific kind of task. In such a scenario, UNet like segmentation models help us. To ensure consistent comparison across models, each model is trained on the augmented version of the CelebA dataset. From the comparison table (Table 4), it can be shown that the proposed model achieves the highest quantitative scores in accuracy, SSIM, and Dice metrics. Notably, the GAN based network and GANMasker achieve a 99% accuracy, nearly matching the proposed model's accuracy. However, the proposed model outperforms other existing systems in terms of SSIM and meanIoU metrics. The model achieves an SSIM of 0.92, indicating 8% structural dissimilarity with the original image, reflecting a high degree of structural similarity. While the YOLO based segmentation model scores high in meanIoU, it lags behind in accuracy and other metrics compared to the proposed model. The modified UNet model achieves a compact size of 71.4 MB, enabled by its reduced number of trainable parameters and higher proportion of non-trainable parameters. This lightweight design makes it suitable for deployment on IoT devices for inference. During testing on an Intel i5 7th-generation system with 8GB RAM and an SSD, the model demonstrated an inference speed of 55 ms, which includes the time required by network's forward pass. Further details on IoT implementation and inference performance are provided in Section 4.3.5.

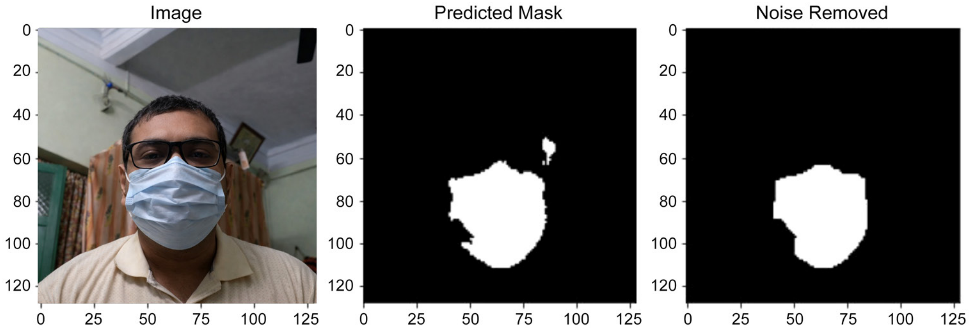
Table 4
Performance comparison of proposed and different state-of-the-art model

Method	Accuracy	SSIM	MeanIoU	Dice
GAN based [29] (map module)	99.02	0.80	0.89	0.92
Base UNet [8]	98.02	0.85	0.86	0.90
Facial mask detector [10]	93.88	0.80	0.85	0.87
Maskthenclassify [11]	94	0.85	0.852	0.86
UNet-Id [15]	97.67	0.75	0.694	0.75
MaskPlus [17]	89.25	0.65	0.569	0.62
YOLOv8-seg [19]	97.23	0.84	0.9526	0.94
GANMasker [13] (segmentation module)	99	0.90	0.92	0.99
Proposed model	99.62	0.92	0.94	0.99

4.3.3. Model testing with captured images

An evaluation was performed using a sample image captured during model inference on an edge device with a Logitech C270 webcam (1280×720 resolution) while a face mask was worn (Figure 13). The model produced a binary mask of the input image, initially displaying some distortion, which was later reduced through noise filtering to obtain a refined mask. Importantly, this test image was not part of the augmented CelebA dataset. Despite this, the model delivered a strong performance, achieving an accuracy of 0.98 and an SSIM of 0.89, demonstrating its ability to generalize well to unseen data. These results suggest that the proposed model is effective when applied to previously unencountered datasets.

Figure 13
Model output on masked face images



4.3.4. Additional testing results

Figure 14 presents the qualitative performance of the proposed segmentation model on real-world masked face images from the RMFD dataset. The model effectively produces accurate binary segmentation masks under real-life conditions, demonstrating robustness to variations in illumination and background clutter. Figure 15 illustrates model outputs on RMFD images exhibiting different head pose variations under different lighting conditions, further highlighting the robustness of the proposed approach to pose and illumination changes. From Figure 16, it can be seen that the face image covered with a niqab cannot be segmented by the base UNet model or other segmentation model. However, when the proposed segmentation model is applied to a cropped face image (Figure 17), which is generated by the proposed testing architecture, given in Section 4.3.1, the model can accurately generate the binary mask. In Figure 18 and Figure 19, the model is trained enough to predict segmented binary masks of incorrect face masks, such as handkerchiefs or any other objects that are used by people to hide themselves. An additional observation can be found from Figure 10 that the proposed model can accurately segment facial masks from facial parts when a person is wearing a spectacle.

4.3.5. Edge device implementation

The inference of the model is implemented on the Nvidia Jetson Xavier NX board. The board has WiFi and LAN connectivity for connection to the internet. It has an active SSH connection to visualize the results on the computer screen. A Linux operating system (Ubuntu 18.04) with the necessary CUDA driver was loaded onto the board via a pre-booted MicroSD card. NVMe M.2 SSD storage was used to infer the model on the board. A Logitech C270 web camera was used to capture the face images for inference. In Figure 13, a face image is given with its binary segmentation mask, which was captured during inference of the proposed model in the edge device. Figure 20 shows the performance of the board, where the proposed model took 60 ms to produce a binary mask from an occluded face image, which is slightly longer than the time taken by the laptop.

5. Limitations

There are some limitations in the model and the current structure of the system. It cannot detect any faces and does not generate a binary mask when all facial parts are covered, except for the eyes, such as

Figure 14
Model output on masked face images from RMFD dataset

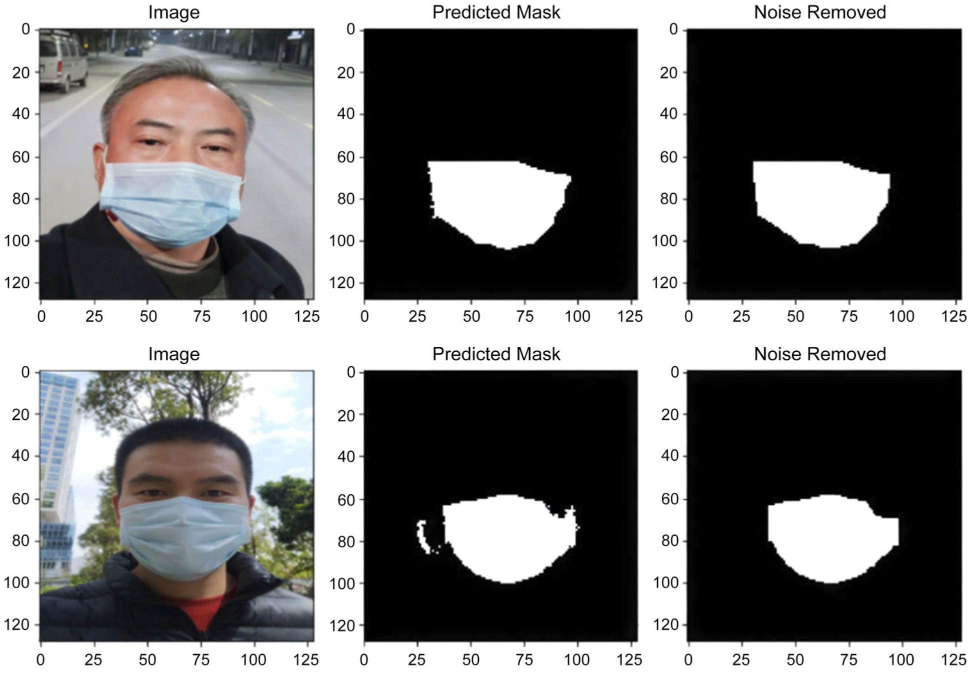


Figure 15
Model output on masked face images from RMFD dataset with different head postures

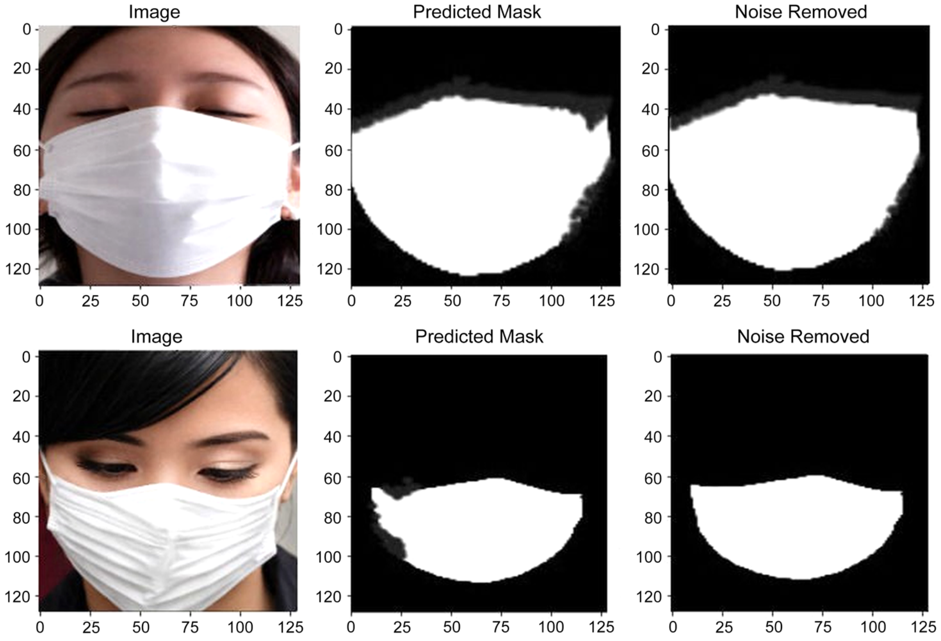


Figure 16
Face covered with niqab from facemask detector dataset

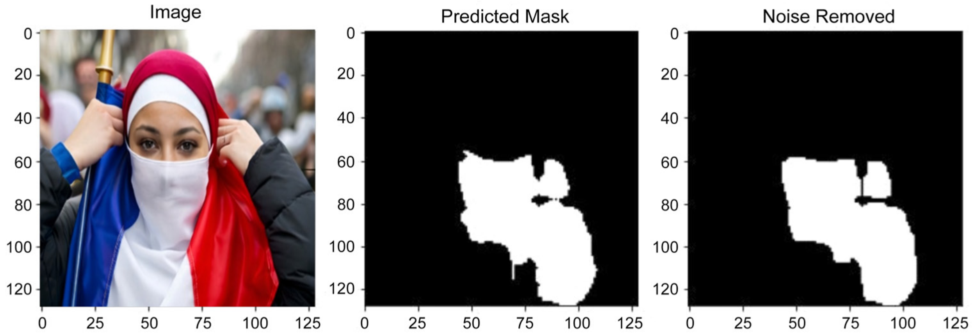
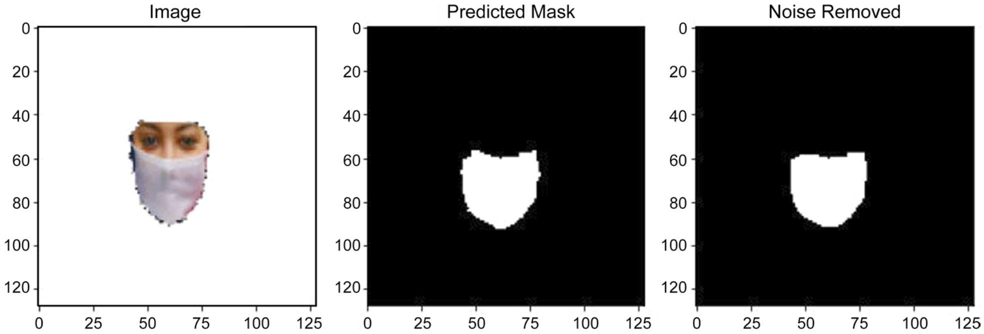


Figure 17
Prediction of binary mask for Incorrect face mask from facemask detector dataset



when ladies wear a burka, which is a cultural aspect of a religion. In the current system, segmentation of face mask from the people's faces who have worn improper face mask critically depends on Dlib face detector model, which requires at least two visible facial features (e.g., eyes, forehead) for reliable detection of faces from images. The performance of the face detector model may be hindered when most of the facial features are occluded. In addition, the proposed segmentation network operates on single-face inputs and cannot directly segment multiple faces within a single image without prior face detection and cropping. As a result, the system currently relies on Dlib to detect and isolate individual faces before segmentation. Furthermore, although the model demonstrates robustness to moderate variations in head pose,

Figure 18
Prediction of binary mask for incorrect face mask from facemask detector dataset

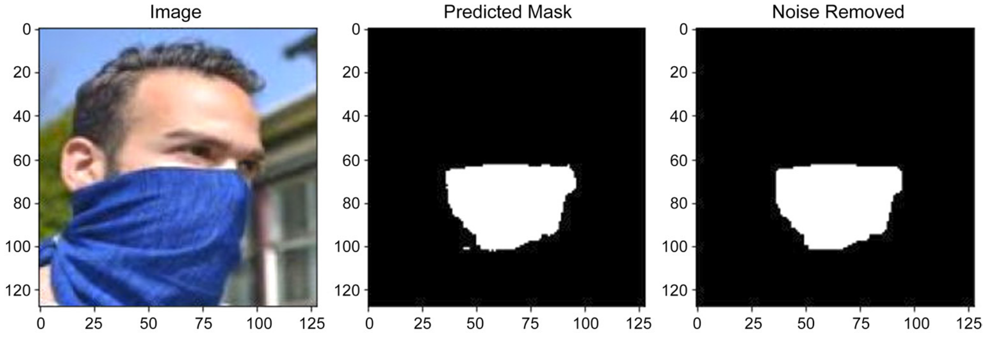


Figure 19
Prediction of binary mask for incorrect face mask from facemask detector dataset

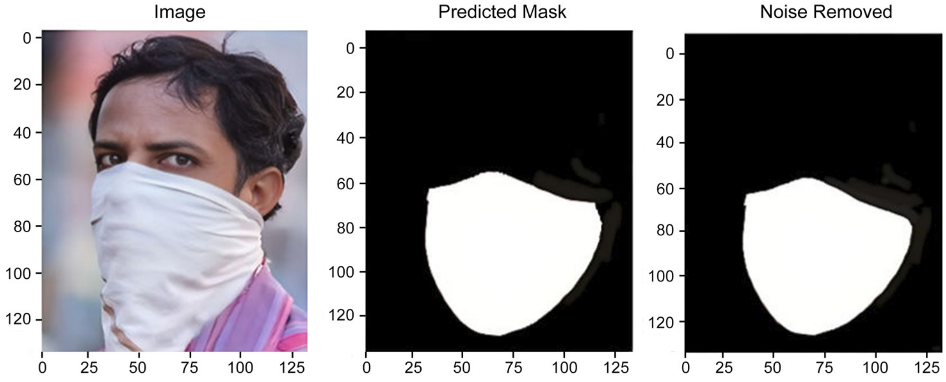
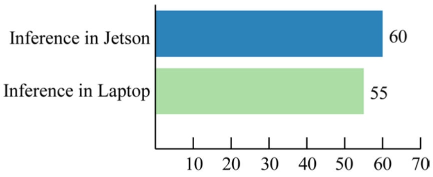


Figure 20
Comparative analysis of the proposed model's inference time (ms) in different hardware architectures



the tolerance limits with respect to yaw, pitch, and roll angles are not quantified explicitly, which is acknowledged as an additional limitation. Future work will focus on developing a unified, occlusion resistant framework potentially based on vision transformer architecture that can jointly handle face detection and mask segmentation for multiple faces in a single forward pass, thereby improving robustness under extreme occlusion, pose variation, and real-world multi-face scenarios.

6. Conclusion

The present study introduces an innovative method for segmenting large occluded areas in facial images, with a particular focus on face masks. To accomplish this, an encoder–decoder-based neural network was used to produce a precise binary segmentation mask from facial images. The model was trained using an augmented CelebA dataset, and a robust testing approach ensured an accurate segmentation of different types of facial coverings, including handkerchiefs and niqabs. The proposed model's qualitative and quantitative comparisons show that it produces high-quality segmentation masks for large occlusions in facial parts compared to other state-of-the-art models. The model achieved 99.62% training and 99.48% validation accuracy, outperforming other models. The proposed model, by producing accurate segmentation masks of occluded facial areas, can support the reconstruction of original facial images, thereby potentially assisting detective agencies, police, law enforcement agencies, and government officials in identifying individuals when their facial part is concealed by masks during a crime. Finally, an edge-device implementation of the proposed model is presented, showcasing its feasibility for deployment in resource constrained devices.

Ethical Statement

This study did not involve any new data collection from human or animal participants. All facial images used in the experiments were obtained from publicly available datasets (e.g., CelebA, MAFA), which were originally collected and released by their respective authors with appropriate consent and usage licenses for research purposes. According to the policies of Visva Bharati University, research based on exclusively publicly available, anonymized, and licensed datasets does not require formal Institutional Review Board (IRB) or ethics committee approval. The study complies with relevant data protection, privacy, and ethical guidelines, and no personally identifiable information beyond the dataset content was collected or disclosed.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in CelebFaces Attributes Dataset at https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html; in RMFD-Real-World Masked Face Dataset at https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset; in FMLD Dataset at https://arxiv.org/abs/1511.06523v1 and https://github.com/borutb-fri/FMLD; in MAFA - Masked Faces at https://www.kaggle.com/datasets/revanthrex/mafadataset; in Augmented CelebA Dataset at https://www.kaggle.com/datasets/mrinmoysadhukhan/augmented-celeba-dataset/data; and in Face Mask Detector at https://www.kaggle.com/datasets/spandanpatnaik09/face-mask-detectormask-not-mask-incorrect-mask.

Author Contribution Statement

Mrinmoy Sadhukhan: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Indrajit Bhattacharya:** Conceptualization, Validation, Formal analysis, Investigation, Resources, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **Paramartha Dutta:** Conceptualization, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Supervision. **Kaushik Roy:** Resources, Data curation, Writing – review & editing.

References

[1] Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. In *International Conference on Computer Vision*, 3730–3738. https://doi.org/10.1109/ICCV.2015.425

[2] Saeed, S., Abdullah, A., Jhanjhi, N. Z., Naqvi, M., Masud, M., & AlZain, M. A. (2022). Hybrid GrabCut hidden Markov model for segmentation. *Computers, Materials & Continua*, *72*(1), 851–869. https://doi.org/10.32604/cmc.2022.024085

[3] Chen, X., Liu, C., Xie, D., & Miao, D. (2025). Image thresholding segmentation method based on adaptive granulation and reciprocal rough entropy. *Information Sciences*, *695*, 121737. https://doi.org/10.1016/j.ins.2024.121737

[4] Mittal, H., Pandey, A.C., Saraswat, M., Kumar. S., & Pal. R. (2022). A comprehensive survey of image segmentation: Clustering methods, performance parameters, and benchmark datasets. *Multimedia Tools And Application*, *81*, 35001–35026. https://doi.org/10.1007/s11042-021-10594-9

[5] Zanaty, E. A., Abdel-Aty, P. M., & Ali, K. A.-W. (2022). Comparing U-Net convolutional network with mask R-CNN in nuclei segmentation. *International Journal of Computer Science and Network Security*, *22*(3), 273–275. https://doi.org/10.22937/IJCSNS.2022.22.3.35

[6] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827

[7] Verma, S., Saini, V., Singh, V. K., & Nethravathi, B. (2024). Advancements in semantic segmentation: A comprehensive review and comparative analysis of fully convolutional networks (FCN). *International Research Journal of Modernization in Engineering Technology and Science*, 1265–1269. https://doi.org/10.56726/IRJMETS48360

[8] Yuan, Y., & Cheng, Y. Medical image segmentation with UNet-based multi-scale context fusion. *Scientific Reports*, *14*, 15687 (2024). https://doi.org/10.1038/s41598-024-66585-x

[9] Aqthobirrobbany, A., Al-Fahsi, R. D. H., Soesanti, I., & Nugroho, H. A. (2024). Enhanced U-Net architecture with CNN backbone for accurate segmentation of skin lesions in dermoscopic images. *International Journal of Advances in Intelligent Informatics*, *10*(3), 490. https://doi.org/10.26555/ijain.v10i3.1379

[10] Meenpal, T., Balakrishnan, A., & Verma, A. (2019). Facial mask detection using semantic segmentation. In *International Conference on Computing, Communications and Security*, 1–5. https://doi.org/10.1109/CCCS.2019.8888092

[11] Kurmann, T., Márquez-Neila, P., Allan, M., Wolf, S., & Sznitman, R. (2021). Mask then classify: Multi-instance segmentation for surgical instruments. *International Journal of Computer Assisted Radiology and Surgery*, *16*, 1227–1236. https://doi.org/10.1007/s11548-021-02404-2

[12] Teliti, D., Shehu, O., Mehanović, D., Kevrić, J., & Karlik, B. (2025). Performance comparison of face detection algorithms for accurate face counting. In *International Conference on Communications, Information, Electronic and Energy Systems*, 1–6. https://doi.org/10.1109/CIEES66347.2025.11300078

[13] Mahmoud, M., & Kang, H.-S. (2023). GANMasker: A two-stage generative adversarial network for high-quality face mask removal. *Sensors*, *23*(16), 7094. https://doi.org/10.3390/s23167094

[14] Ye, Z., Zhang, H., Li, X., & Zhang, Q. (2023). DeMaskGAN: A de-masking generative adversarial network guided by semantic segmentation. *The Visual Computer*, *40*, 5605–5618. https://doi.org/10.1007/s00371-023-03125-0

[15] Wagner, F., Dalagnol, R., Tarabalka, Y., Segantine, T., Thomé, R., & Hirye, M. (2020). U-Net-Id, an instance segmentation model for building extraction from satellite images - Case study in the Joanópolis City, Brazil. *Remote Sensing*, *12*(10), 1544. https://doi.org/10.3390/rs12101544

[16] Ke, L., Danelljan, M., Li, X., Tai, Y.-W., Tang, C.-K., & Yu, F. (2022). Mask transfiner for high-quality instance segmentation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4412–4421.

[17] Xu, S., Lan, S., & Zhu, Q. (2020). MaskPlus: Improving mask generation for instance segmentation. *Winter Conference on Applications of Computer Vision*, 2019–2027. https://doi.org/10.1109/WACV45572.2020.9093379

[18] Chen, Z., Li, D., Fan, W., Guan, H., Wang, C., & Li, J. (2021). Self-attention in reconstruction bias U-Net for semantic segmentation of building rooftops in optical remote sensing images. *Remote Sensing*, *13*(13), 2524. https://doi.org/10.3390/rs13132524

[19] Varghese, R., & S. M. (2024). YOLOv8: A novel object detection algorithm with enhanced performance and robustness. *International Conference on Advances in Data Engineering and Intelligent Computing Systems*, 1–6. https://doi.org/10.1109/ADICS58448.2024.10533619

[20] Wang, C. Y., Yeh, I. H., & Mark Liao, H. Y. (2025). YOLOv9: Learning what you want to learn using programmable gradient information. In A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, & G. Varol (Eds.), Computer Vision – ECCV 2024 (pp. 1–21). Springer. https://doi.org/10.1007/978-3-031-72751-1_1

[21] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., & Han, J. (2024). Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, *37*, 107984–108011. https://dl.acm.org/doi/10.5555/3327144.3327181

[22] Khanam, R., & Hussain, M. (2024). *Yolov11: An overview of the key architectural enhancements*. arXiv. https://arxiv.org/abs/2410.17725

[23] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., ..., & Girshick, R. (2023). Segment anything.

IEEE/CVF International Conference on Computer Vision, 3992–4003. https://doi.org/10.1109/ICCV51070.2023.00371

[24] Ravi, N., Gabeur, V., Hu, Y. T., Hu, R., Ryali, C., Ma, T., ... & Feichtenhofer, C. (2024). *Sam 2: Segment anything in images and videos*. arXiv. https://arxiv.org/abs/2408.00714

[25] Nam, H., & Kim, H. E. (2018). Batch-instance normalization for adaptively style-invariant neural networks. In *International Conference on Neural Information Processing Systems*, 2563–2572). https://dl.acm.org/doi/10.5555/3327144.3327181

[26] Shu, Z., & Zhang, K. (2024). Spectral normalization for generative adversarial networks for artistic image transformation. *International Journal of Digital Multimedia Broadcasting*, *2024*(1). https://doi.org/10.1155/2024/6644706

[27] Wu, Z., Wei, C., Xia, Y., & Ji, Z. (2023). SAITI-DCGAN: Self-attention based deep convolutional generative adversarial networks for data augmentation of infrared thermal images. *Applied Sciences*, *14*(23), 11391. https://doi.org/10.3390/app142311391

[28] Aqeelanwar. (2024). MaskTheFace: Convert face dataset to masked dataset [Computer software]. https://github.com/aqeelanwar/MaskTheFace

[29] Din, N. U., Javed, K., Bae, S., & Yi, J. (2020). A novel GAN-based network for unmasking of masked face. *IEEE Access*, *8*, 44276–44287. https://doi.org/10.1109/ACCESS.2020.2977386

[30] Zhang, J., & Wang, Y. (2024). A new workflow for instance segmentation of fish with YOLO. *Journal of Marine Science and Engineering*, *12*(6), 1010. https://doi.org/10.3390/jmse12061010

How to Cite: Sadhukhan, M., Bhattacharya, I., Dutta, P., & Roy, K. (2026). An Encoder–Decoder-Based Deep Learning Model for Segmenting Occlusion in the Lower Part of the Face. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA62026690