Semi-Dense U-Net: A Novel U-Net Architecture for Face Detection

Ganesh Pai[1*], Sharmila Kumari M[2]

Department of Computer Science and Engineering-Nitte (Deemed to be University), NMAM Institute of Technology, Nitte-574110, Karnataka, India[1]
Department of Computer Science and Engineering, P. A. College of Engineering-Affiliated to VTU,
Mangalore-574153, Karnataka, India[1, 2]

***Abstract*—Face detection and localization has been a major field of study in facial analysis and computer vision. Several convolutional neural network-based architectures have been proposed in the literature such as cascaded approach, single-stage and two-stage architectures. Using image segmentation based technique for object/face detection and recognition have been an alternative approach recently being employed. In this paper, we propose detection of faces by using U-net segmentation architectures. Motivated from DenseNet, a variant of U-net, called Semi-Dense U-Net, is designed in order to improve the binary masks generated by the segmentation model and further post-processed to detect faces. The proposed U-Net model have been trained and tested on FDDB, Wider face and Open Image dataset and compared with state-of-the-art algorithms. We could successfully achieve dice coefficient of 95.68% and average precision of 91.60% on a set of test data from OpenImage dataset.**

Keywords—Semi-Dense U-Net; face detection; segmentation; U-Net

I. Introduction

Face detection deals with the localization of face in a given image. At the outset, detection process takes an input image containing one or more faces, applies a detection and localization model and produces a confidence score and a set of bounding-box parameters containing the coordinates of the face and its dimension. Face detection being the first phase in face analysis, the detected face is subjected to facial analysis process for machine learning and computer vision applications. Over decades, several algorithms have been proposed using a variety of approaches to detect faces in the image for diverse applications addressing uncontrolled illumination, scale variance, rotation in plane, occlusion, low quality image, large and tiny faces, masked faces, faces with makeup etc. These algorithms are designed to address a subset of these issues but no algorithm can address all the issues.

A variety of face detection techniques for computer vision applications can be found in the literature. Several new techniques and approaches have been explored at a great extent in the literature, each approach trying at its maximum to address a selected subset of the issues in face detection using the available dataset. Over time, the complexity of the face dataset has widened considerably covering very high- and low-resolution images and with facial features that has laid challenges in achieving good detection rate and further promoting development of new algorithms to address the issues. Early work on face detection dates back to 1992 in [1], that used artificial neural networks. However due to the limited computational and storage resources, it did not gain considerable attention. In contrast to traditional algorithms, capability of convolutional neural networks (CNN) to learn features from its input has led to a number of recent advancements in the field of face detection using CNN. With improved computing power through GPU and now TPU's, there is more scope for research promoting construction of complex models for AI based applications. CNN architectures have made it possible today to learn complex features from large and complex datasets. Several novel architectures such as AlexNet [2], VGGNet [3], GoogLeNet [4], ResNet [5], DenseNet [6], DarkNet [7] and its variants have been used as backbone network for feature extraction. This has improved the performance of face detection frameworks over time.

Figure 1 shows a face detection process used in our work. The input color image of any scale is subjected to preprocessing, that scales down the input image to a standard size of 256×256 or 512×512. In our work, three U-Net architecture variants are used, each trained with three standard datasets. The outcome of feature extraction is a binary feature map/mask representing the segmented image, as shown in the figure. As the network output does not always produce a very fine and sharp segments, it is further refined to generate sharp rectangular regions suitable for detection of bounding box in the final step.

The remainder of the paper is organized in the following manner. Section II presents some of the prominent face detection architectures, U-net segmentation architectures and its variants used for various applications, Section III presents the proposed architectures, Section IV highlights on implementation details with the experimental results and Section V summarizes with conclusion.

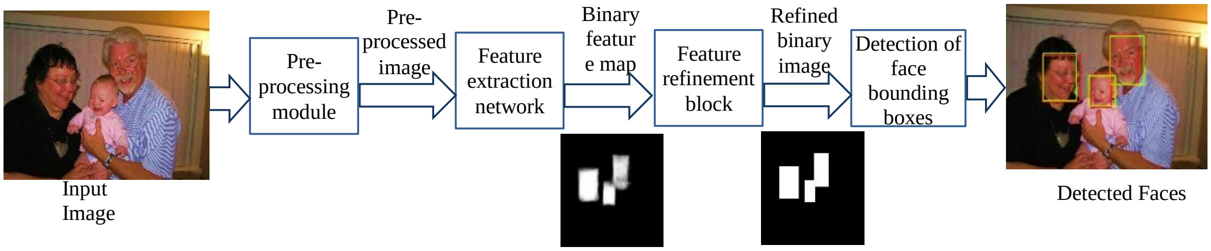


Fig. 1. General model of proposed face detection architecture. The first binary image (at bottom) is the model output and second is output of post-processing module.

II. Related Work

CNN segmentation architectures are being used for several applications, such as, for medical images [8][9], object and instance segmentation such as Mask-RCNN [10], video object segmentation [11] to name a few. It however has been used to segment the entire region of the object and can be extended to recognition applications. U-Net [8] is one such neural network architecture designed for biomedical image segmentation. Its U-shaped network consists of an encoder CNN that gradually reduces the spatial size of the input image while increasing the number of feature maps to retains high-level contextual information and a decoder section that reconstruct the segmentation map by gradually increasing the spatial size of the features and concatenating them with the corresponding features from the encoder part using skip connections for precise localization information. The study [12] proposed a simplified UNet architecture for medical image segmentation. UNet++ [9] is a U-Net based architecture using nested and dense skip connections designed to improve the accuracy of medical image segmentation. It improved U-Net by adding skip pathways that connect the two sub-networks and uses deep supervision. The deep supervision module enables the model to operate in an accurate mode, where the outputs from all segmentation branches are averaged; and fast mode that selects the final segmentation map from the segmentation branches. This choice determines the extent of model pruning and speed gain. Motivated by DenseNet architecture [6], Li et al. [13] proposed H-denseunet segmentation architecture for liver and liver tumor segmentation. The study [10] is another such approach to detect faces using a segmentation architecture. Using improved Mask R-CNN, Lin et al. proposed G-Mask [14] for detection and segmentation of face that incorporates both into one framework. It used ResNet-101 for feature extraction, RPN to generate RoIs, and fully convolutional network to generate binary mask. A most recent work on U-Net can be found in [15] that segments lungs in the chest radiography images. With several recent segmentation architectures and its improvements thereof, we find them being applied in a diverse domain meaningfully able to elaborate on the semantic aspects of the problem domain to the solution space. Each architecture has tried to extract and extend the prominent features of the base architecture and inherit prominent features of other architectures to address the drawbacks in the base architectures. In this paper, our proposed architecture inherits the features of U-Net, DenseNet and ResNet to build an improved segmentation architecture that produces more accurate segmentation output and successfully applied it on a face detection problem. The outputs observed are on par with some of the standard convolutional face detection architectures.

III. Proposed Architectures

U-Net based segmentation architectures have been widely used on medical images for disease detection and localization. In this paper, U-Net architecture is used as a base to develop an improved U-Net architecture to improve the accuracy of the binary mask generated that will be postprocessed to detect faces. Here, we use three U-Net based architectures for face detection application. The details of the architectures used are as follows:

A. *Using U-net*

Our experiment uses U-Net architecture comprising of six blocks at encoder with two convolutional layers in each with the kernel size of 3, same padding, he_normal kernel initializer, relu activation, max pooling of size 2 and a dropout of 0.2. Input to the architecture is a single channel image of dimension 512, with the corresponding training output being its binary image with masks at the face regions. Size of the feature map converges at the encoder generating 512 feature maps of size 16 and decoder upsamples it to a single binary feature map image of size 512 generating a segmented binary image for the given input. The segmented regions represent the faces predicted. The prediction may not have always a clear and sharp edge. These variations are addressed by a post-processing module that refines the prediction of segmented regions. Bounding box is then computed over the refined image.

B. *Using VGG16-Unet*

Transfer learning today speeds up the training time by using pre-trained weights of a backbone network. This is experimented by using a pre-trained VGG-16 backbone network at the encoder side of U-net, configured with ImageNet dataset weights. The weights at the encoder side were configured to be non-trainable and the decoder part to be trained with the input dataset. The network is expected to learn faster as the encoder is already in possession of valuable weights. The model takes a colored image of size 512 and produces a binary segmented image of same size. As mentioned in the above model, the output is further refining using a post-processing module to improve the segmented regions. Bounding box is then computed over the refined image.

C. *Using Semi-Dense U-Net*

In a general CNN, each layer produces a set of features and is forwarded to the next layer for deep feature extraction. ResNets [5] introduced a concept of skip-connections where an output can bypass the normal flow of non-linear transformations with an identity function and get combined with a layer down the network. With transition function H_l for the l^{th} layer, we can represent the result of skipped layer as:

$$x_l = H_l(x_{l-1}) + x_{l-1} \tag{1}$$

U-Net uses this skip-connection to connect between encoder and decoder. The research [6] further improved it by adding a dense connectivity between layers where each layer will be learning features from its all-previous layers. Hence the transition function will have feature maps $x_0, \dots, x_{l-1}$,, as input. This can be formulated as

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \tag{2}$$

This generates a strong feature map through which each layer will be encapsulating low- to high-level feature. This architecture is referred as DenseNets. Motivated from ResNets and DenseNets, a modified U-net architecture is proposed by adding skip-connections at the encoder side that connect to layers within the encoder and dense connections at the decoder side with links from various layers at the encoder side scaled down at respective levels at the decoder side, and we name it *Semi-Dense U-Net*. Our proposed architecture uses seven blocks at the encoder side and six blocks at the decoder, as shown in Fig. 2. Feature maps are increased progressively from 16 at the first block, B1, to 512 at the seventh block, B7.

1) Layer structure: Each layer is built using two convolution layers with kernel size of three, same padding, he_normal kernel initializer with relu activation, batch normalization, max pooling of size two and a dropout of 0.2. At the encoder side, the normalized output at laver l is concatenated with feature map from layer $l - 1$ followed by max pooling. At the decoder side, output of the dense feature scale module is a 512×512 color image scaled down to 8x8 with 256 channels followed by decoder network up-sampling the features back to single channel of size 512×512. In U-Net and VGG16-Unet model, dropout of layers B1, B2, B10, B11 is 0.1, B3, B4, B8, B9 is 0.2 and B5, B6, B7 is 0.3. Semi-Dense U-Net uses dropout of 0.1 at layers B1, B2, B12, B13 and 0.2 at all others.

2) Dense feature scaling module: Dense feature scaling module scales down feature maps of each upper layer to 16 feature maps using max pooling. Hence, at layer l, we get $(l - 1) \times 16$ feature vectors. This will be later concatenated with the feature maps generated at level l along with the up-sampled feature map from level $l + 1$ at the decoder side. More semantic information are obtained from the feature maps at deeper layers [16]. Our feature map progressively encapsulates high to low range features as we go deeper. Hence, semantic information from the dense features is extracted by using a 1×1 convolution layer and is normalized.

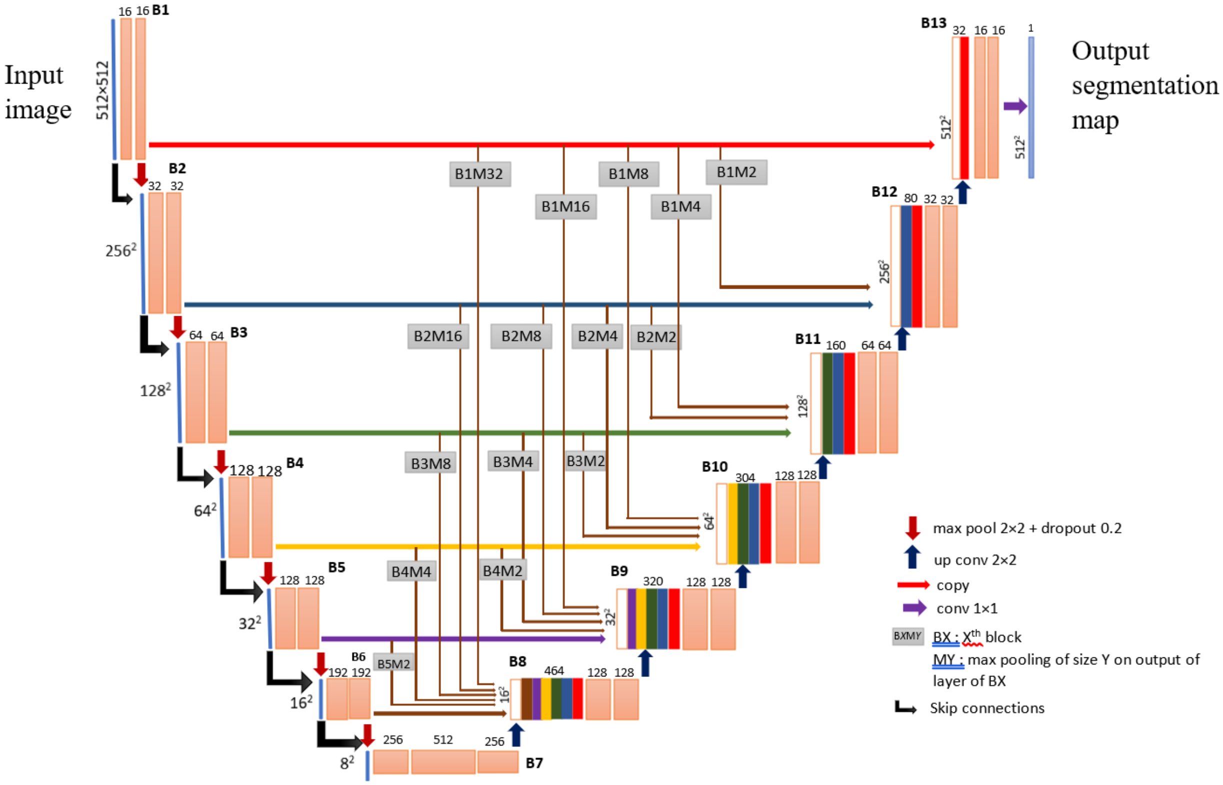


Fig. 2. Semi-Dense U-Net architecture. Each pink block represents multi-channel feature map and colored blocks at the decoder, the feature maps scaled down from upper layers using BXMY module.

3) Growth rate: At the encoder, if the function H_l at layer l produces k feature maps, it follows that layer l has input feature-maps from layer $(l-1)$ and $(l-2)$, where $l \geq 0$ and $l = -1$ representing channels from the input vector. This can be visualized in Fig. 3. It can also be observed from the Fig. 2 that block 4 and 5 carries forward a constant number of 128 feature vectors. This effectively controls the expanding parameters of the network. At the decoder side, feature maps of encoder are scaled and squeezed to a fixed number of 16 feature maps at each level, as discussed above. By limiting it to a constant value, the number of parameters of the network can be kept small. Hence, if a layer l generates k feature maps, the decoder concatenates $2k + 16(l-1)$ feature maps at each layer.

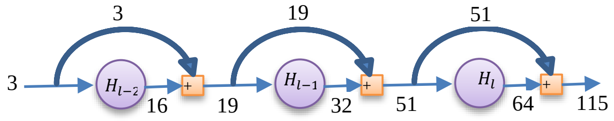


Fig. 3. Growth rate at the encoder side of the network.

D. Post-processing Module

Output of the network is an image that may contain traces of black within white predicted regions (Fig. 4 (a)). The network does not always generate a clear rectangular region. Further, there can be certain regions in the output with light traces of white pixels that are in fact false predictions. Hence the output image certainly should be subjected to post-processing to refine the predictions and eliminate false predictions. This module uses image enhancement techniques to produce a clear and sharp image, as shown in Fig. 4 (b), suitable for computing the bounding box of the predicted regions.

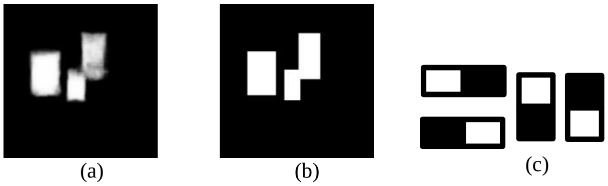


Fig. 4. Post-processing result. a) Model output b) Post-processing output c) Haar-like edge features.

Refinement has been experimented using two approaches: thresholding and grid-based region growing approach. With thresholding, image is first enhanced using Otsu threshold and further refined using opening morphological operation with kernel of 3×2px. Image is further contoured to extract boundaries of the segments. Overlapped segments are further processed to extract distinct rectangular regions out of the segmented regions and their bounding boxes. The post-processed result is shown in Fig. 4(b).

In the grid-based region growing approach, the image of 512×512 is divided into grid of size 4×4. Intensity of the grid is evaluated using $G_{i,j} = \frac{1}{16\, I_{max}} \sum I_{p,q}$, where I_{max} is the maximum intensity of the image, $I_{p,q}$ is the p, q^{th} pixel intensity of grid cell $G_{i,j}$. Based on the value of $G_{i,j}$, the cell is initially classified as, full-white (FW), full-black (FB) or fuzzy (FZ). Cells with $G_{i,j} \geq 70$ are labeled as FW and $G_{i,j} \leq 30$ are labelled as FB. Remaining are considered as fuzzy cells. FZ are further processed base on adjacency positions and haar-like features [17]–[19] to relabel them as FB/FW. Haar-like edge features (Fig. 4 (c)) are analyzed in each cell. Based on the pixel intensity proportion in the adjacent bands and its adjacency to the FW/FB, cells are labelled as FW or FB. This is then followed by contouring, extraction of boundaries and bounding box, as discussed for thresholding approach.

IV. Experiments

The model is trained using TensorFlow deep learning API's on a Nvidia Tesla P100-PCIE (12 GB) GPU. For the U-net architecture, the image is initially converted to grayscale before feeding into the network. Inputs to the other two networks are color images. Feature refinement module partially uses OpenCV library for image enhancement and morphological operations. Image is preprocessed by scaling down to 512×512.

A. Training Datasets

Each model is trained and tested on FDDB, Wider face and OpenImage dataset. FDDB dataset contains 5,171 faces in 2,845 images. As FDDB dataset represents faces using ellipses, our models are trained to generate elliptical segments, as shown in Fig. 5. The post processing module extracts the elliptical regions coordinates, angle, major and minor axis. Model is tested using 10-fold cross validation and accuracies averaged. Wider face dataset contains faces with a high degree of variability in scale, pose and occlusion with images organized based on 61 event classes. Dataset randomly select 40%/10%/50% data as training, validation and testing. It contains 3,93,703 annotated faces in 32,203 images. We have used 12,880 training images to train our model and 3,226 validation images to test our model. Faces are classified as easy, medium and hard based on the face size of less than 50px, 50px to 300px and above 300px respectively. OpenImage-v6 face dataset contains 3,44,043 annotated images with 10,60,312 faces. It is classified into 3,31,627 training images (1,037,710 faces), 3,124 validation images (5,594 faces) and 9,292 test images (17,008 faces). Due to hardware resource limitations, we use 10,000 annotated training images as our dataset with 80%:20% for training and validation/testing. Model is trained to produce rectangular segments for the detected faces in OpenImage and Wider face datasets as provided in the dataset.

All models have been trained for 150 epochs with early stopping where validation loss is monitored with the patience of 10.

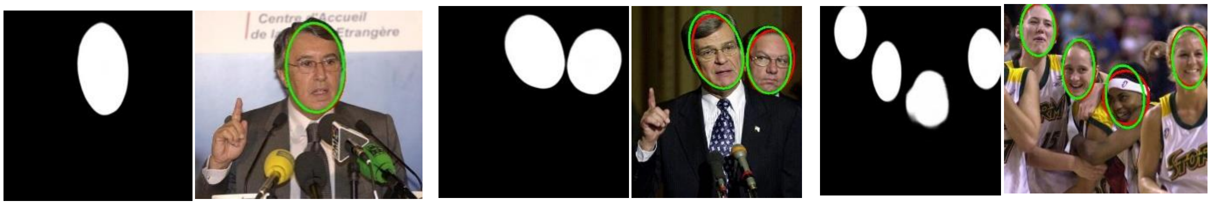


Fig. 5. Predicted binary mask image by the model and its corresponding image with face ellipses. Green ellipse represents ground truth and red represents prediction by the model for samples from FDDB dataset.

B. Model Hyperparameters

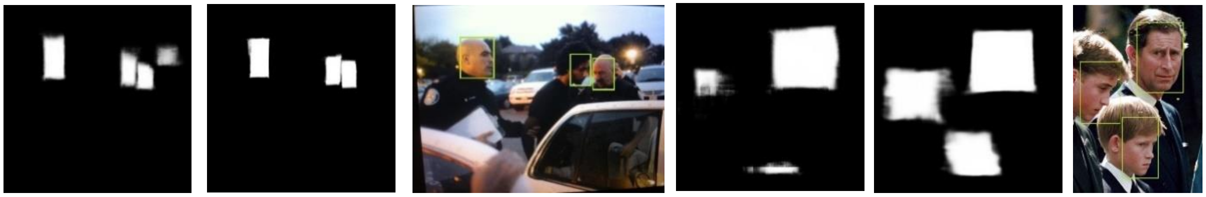
Model has been trained with adam optimizer with a constant learning rate of 0.001. Each layer is appended with batch normalization (BN) to optimize the segmentation accuracy. All the convolutional layers have been configured with relu activation function, he_normal kernel initializer and sigmoid function at the last layer. Models have been trained with binary cross-entropy loss function for 150 epochs and batch size of 16 for U-Net and 8 for VGG16-Unet and Semi-Dense U-Net architectures. U-Net model used contains 7.7M parameters and VGG16-unet contains a total of 25.8M parameters with 11.3M trainable parameters. The proposed Semi-Dense U-Net, is optimized with only 5.8M parameters.

C. Effect of Batch Normalization

Models have been experimented with and without BN. Fig. 6 shows the results on two sample images drawn from wider face dataset. Fig. 6 (a) and 6(d) are the model output without BN whereas 6(b) and 6(e) are with BN. With BN, we can observe a comparatively better and sharper approximation than without BN. Further we can observe certain cloudy region at the rightmost side of the image in 6(a) leading to false positives. With BN, such regions have been eliminated in 6(b). In 6(d), it can be observed that leftmost two segments are not sufficiently predicted as a facial region, leading to false negative. With BN, we get a better approximation of the region that is suitable to come under true positive even with an iou of 80%. Hence, BN has normalized the covariate values of the dataset and has given a better prediction accuracy. Fig. 6(c) and 6(f) show the original image with face bounding boxes obtained with BN.

D. Evaluation on Datasets

Table I tabulates the training and validation accuracy of the three models on FDDB, Wider face and OpenImage datasets. The accuracies observed are on par with the standard datasets. We can observe that Semi-Dense U-Net comparatively gives better accuracy than the other two. The accuracies mentioned in the table for FDDB dataset are the average accuracy over 10-fold cross validation. Fig. 7 to Fig. 9 show training and validation accuracy curves of U-Net, VGG16-Unet and Semi-Dense U-Net model on all three datasets. Training accuracy curves on Wider face dataset are projected for easy, medium and hard samples. In wider face dataset, the performance is found to fluctuate frequently for hard faces. This can be possibly due to the tiny faces in the samples. Most of the time, tiny faces are part of crowd images. Wider face dataset contains several classes of images that has crowded people. Often such faces are blur in nature, making it hard to extract detailed features.

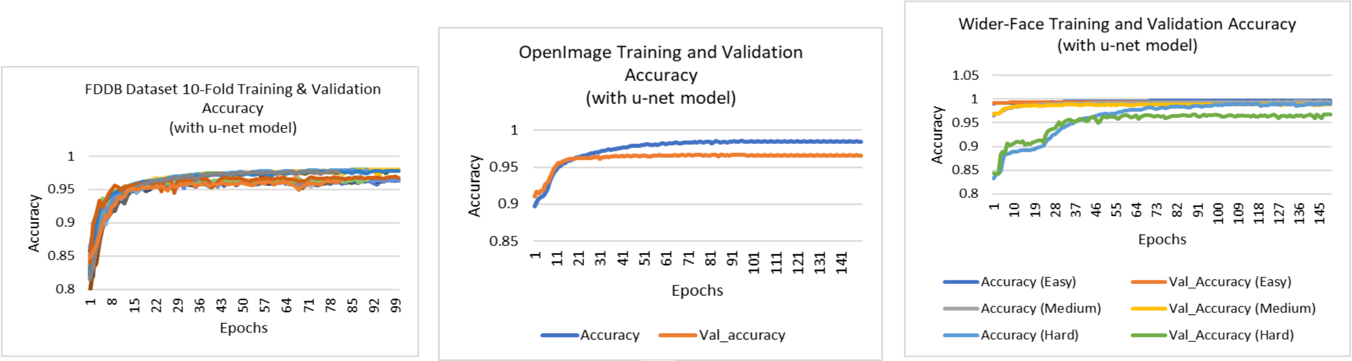


Without BN (a) With BN (b) Image with bounding boxes (c) Without BN (d) With BN (e) Image with bounding (f)

Fig. 6. Sample images projecting effect of introducing batch normalization at the end of each layer. (a), (c) without BN; (b), (e) with BN; (c), (f) Original image with face bounding box using outcome with BN.

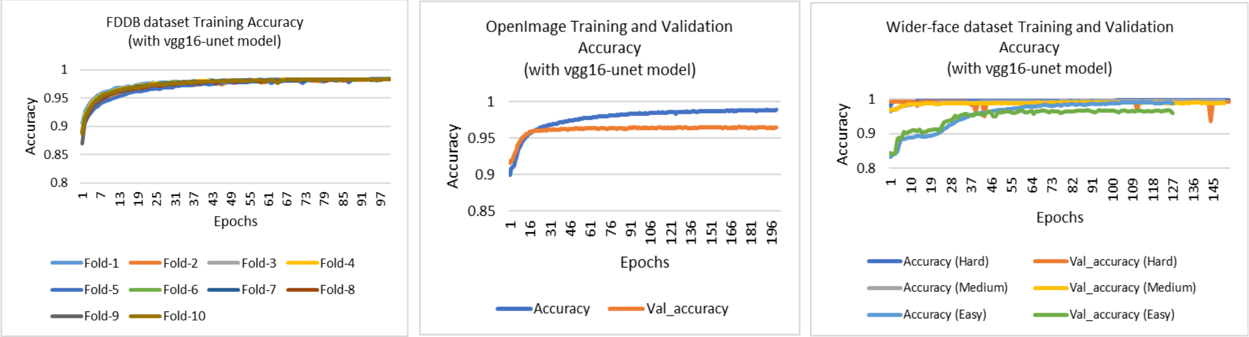
TABLE I. TRAINING AND VALIDATION ACCURACIES OF FDDB, WIDER FACE AND OPENIMAGE DATASET

	FDDB Dataset (Accuracy in %)		**Wider Face Dataset (Accuracy in %)**						**Open Image dataset (Accuracy in %)**	
			Easy		**Medium**		**Hard**			
	Train	**Val**	**Train**	**Val**	**Train**	**Val**	**Train**	**Val**	**Train**	**Val**
U-net	98.22	96.83	99.68	99.35	99.44	98.88	99.00	96.75	98.52	96.72
VGG16-Unet	98.37	96.92	99.58	99.28	98.97	98.71	99.78	96.36	99.64	96.74
Semi-Dense U-Net	98.54	96.98	99.73	99.37	99.72	98.90	99.32	96.70	99.40	96.97



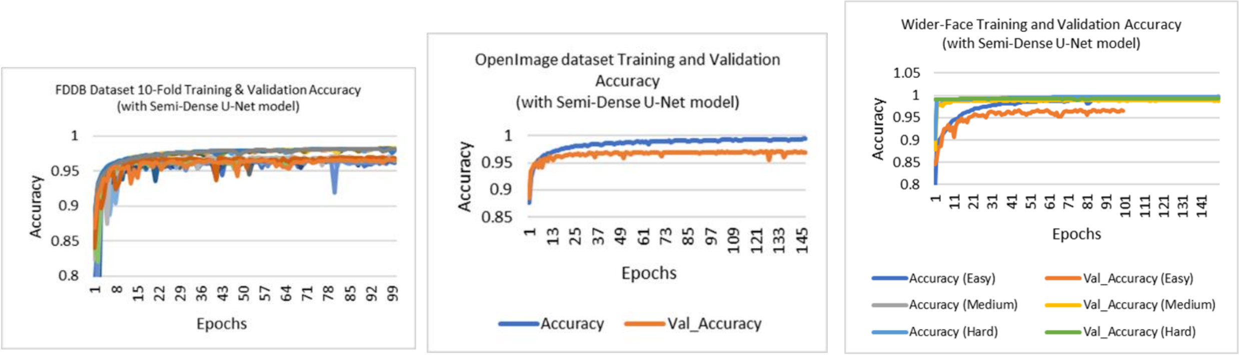
a b c

Fig. 7. Training and validation accuracy of U-Net Model on a) FDDB dataset b) OpenImage datasets c) Wider face dataset.



a b c

Fig. 8. Training and validation accuracy of VGG16-UNet Model on a) FDDB dataset b) OpenImage datasets c) Wider face dataset.



a b c

Fig. 9. Training and validation accuracy of Semi-Dense U-Net Model on a) FDDB dataset b) OpenImage datasets c) Wider face dataset.

E. *Results and Discussion*

The core architecture used in this paper is based on U-Net and as discussed earlier, it is a segmentation architecture that produces image segments for regions of interest with matched features. The image segments are first extracted in post-processing module and then bounding boxes or elliptical parameters are formulated. Commonly used metrics for evaluating the performance of the segmented images are Jaccard coefficient to measures similarity between finite sample sets and is represented as $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$ indicating the ratio of intersection over union, and dice coefficient $DC = \frac{2TP}{2TP+FP+FN}$, a parameter to measure the test accuracy, were TP, FP and FN represents true positive, false positive and false negative respectively and is commonly called as F_1 score. Metrics precision, recall and F_1 scores are computed at IoU of 0.4. Further we compare our performance with state-of-the-art face detection algorithms. Table II shows precision, recall and F_1 score of the various models on the three standard datasets.

Table III tabulates the average precision of the models on FDDB and OpenImage dataset at iou's 0.4, 0.6 and 0.8. The results tabulated for FDDB are average of 10-fold cross validation. It can be observed that, at AP@40 U-Net achieves 86.5%, vgg16-unet achieves 52.8% and Semi-Dense U-Net achieves 98.97%. In contrast to the standard MTCNN that produced AP of 98.8%, Semi-Dense U-Net performed better than MTCNN for AP@40. At higher iou, the performance of Semi-Dense U-Net is comparatively less than MTCNN but is observed to be on par with MTCNN. Comparing the performance of Semi-Dense U-Net with U-Net and vgg16-unet, U-Net and Semi-Dense U-Net performs much better than vgg16-unet. On the contrary, Semi-Dense U-Net performs much better due to better feature learning from skip and dense connections. A similar instance can be observed with OpenImage dataset where performance of Semi-Dense U-Net is on par with MTCNN and better than U-Net and VGG16-Unet. Table IV shows the average precision of the models on Wider face dataset. Comparing the results of easy, medium and hard, the proposed model performs well for easy and medium datasets but observed to perform very poorly on hard dataset. It was observed from the predictions that the model under performs for accurate prediction of tiny faces but manages to predict faces greater than 50px. A similar result can be seen in U-Net and vgg16-unet. A deeper analysis of the prediction reveals that the output of the Semi-Dense U-Net model is able to accurately predict frontal face of medium and large size and lightly blur faces but accuracy of predicting heavily blur and occluded faces are not as expected. This has led to the degradation of the performance at various stages. Comparing the results of easy and medium, we observe model to perform better on medium than easy. The reason is due to the limited number of large faces (>300px) in the dataset compared to the medium size face samples. Hence, the model possibly had far a smaller number of images to extract diverse large size features to accurately tune the model parameters. The performance is expected to improve by training the model with a greater number of large sized face samples. Table V projects performance of state-of-the-art CNN algorithms for face detections on wider face dataset. Comparing the results of U-Net and Semi-Dense, we can infer from the results that the performance is close to each other but Semi-Dense U-Net performs better than U-Net. This is due to the better feature learning from the previous layers and better representation of the binary mask features at the model output. This enabled accurate detection of facial regions and its corresponding bounding box during post-processing at various scales making it an improvement of standard U-Net architecture.

TABLE II. PRECISION, RECALL AND F1 SCORE OF THE THREE MODELS

	FDDB dataset			OpenImage dataset			Wider face (Easy)			Wider face (Medium)			Wider face (Hard)		
	P %	R %	F_1	P %	R %	F_1	P %	R %	F_1	P %	R %	F_1	P %	R %	F_1
U-Net	84.13	78.25	81.09	94.24	89.22	91.66	60.40	83.56	70.12	90.47	72.93	80.76	70.64	27.24	39.31
VGG-16 U-Net	48.67	63.88	55.25	78.78	83.08	80.88	46.60	84.38	60.04	83.86	70.30	76.48	47.91	28.65	35.86
Semi-Dense U-Net	89.94	82.52	85.60	97.92	93.72	95.68	86.67	89.04	87.84	92.25	69.64	79.37	58.10	33.57	42.55

TABLE III. TABULATION OF MODEL PERFORMANCE ON FDDB AND OPENIMAGE DATASET

	FDDB			OpenImage		
Model used	iou@0.4	iou@0.6	iou@0.8	iou@0.4	iou@0.6	iou@0.8
MTCNN	0.9884	0.9688	0.9077	0.9175	0.8845	0.8278
U-Net	0.8653	0.7697	0.3504	0.8226	0.7270	0.3077
VGG-16 U-Net	0.5285	0.3553	0.1058	0.6664	0.4932	0.1436
Semi-Dense U-Net (proposed)	0.9897	0.9578	0.8846	0.9160	0.8633	0.7173

TABLE IV. TABULATION OF MODEL PERFORMANCE ON WIDER FACE DATASET (EASY, MEDIUM AND HARD)

	Easy			Medium			Hard		
Model used	iou@0.4	iou@0.6	iou@0.8	iou@0.4	iou@0.6	iou@0.8	iou@0.4	iou@0.6	iou@0.8
U-Net	0.4477	0.3666	0.2223	0.6667	0.5855	0.1679	0.1973	0.0952	0.0009
Vgg-16 U-Net	0.3494	0.2854	0.1527	0.5802	0.4278	0.0522	0.1482	0.0639	0.0017
Semi-Dense U-Net (proposed)	0.8320	0.7920	0.5538	0.7963	0.7118	0.2972	0.2018	0.0838	0.0009

TABLE V. PERFORMANCE OF STATE-OF-THE-ART FACE DETECTION METHODS ON WIDER FACE DATASET

Method	Easy	Medium	Hard
Faceness [20]	0.713	0.664	0.424
ScaleFace [21]	0.821	0.818	0.701
MTCNN [22]	0.851	0.820	0.607
G-Mask [14]	0.902	0.854	0.662

Table VI (a) and (b) shows the model output for all three models and (c), (d), its corresponding detected faces for two samples drawn from FDDB dataset. We use elliptical annotations as used by the dataset. It can be inferred from the "Predictor Output" column that segments generated by U-Net are blur in nature and fails to detect some faces accurately leading to false negatives. On the other hand, VGG16-unet generates a sharper representation but fails to predict a face in (b) of the predictor output. Further, the rightmost segment of (b) has incomplete face regions. In the "Detection Results" column, face circled green are ground truth annotations and the one in red are detection by the model for the respective images in Predictor Output column. Comparatively, Semi-Dense U-Net is observed to have better accuracy in terms of prediction of face regions as well as sharpness of the segments. Several samples are observed to possess cloudy regions in the predicted image at several places in the U-Net and VGG16-Unet but are eliminated in the Semi-Dense U-net architecture.

In Table VI (e) to (h) and Table VII we observe similar outcomes for samples from OpenImage and Wider face dataset respectively. Semi-Dense U-net produces better segmentation results compared to the other two. By producing better and sharper segmented results, we can reduce the time of post-processing as it will eliminate the need for image enhancement to predict the bounding boxes. It can easily be computed over the predicted image. This proportionately will increase the overall detection speed. The average prediction time is observed to be approximately 30ms per image and post processing time is approximately 15ms. At this performance, we will be able to process around 22.22 images per second. By improving the prediction accuracy, the need for complex post-processing can be eliminated, thereby improving the computation time per image. While the prediction time is independent of the number of faces, postprocessing time varies based on the number of faces detected in the binary mask image.

TABLE VI. OUTPUT OF SAMPLE IMAGES FROM (A) TO (D) FDDB DATASET, (E) TO (H) OPENIMAGE DATASET. (A), (B), (E), (F) ARE MODEL OUTPUT. (C), (D), (G), (H) ARE ORIGINAL IMAGES WITH PREDICTIONS IN RED ELLIPSES/BOUNDING BOXES AND GROUND TRUTH IN GREEN ELLIPSES/BOUNDING BOXES

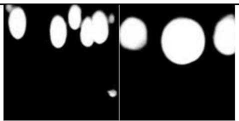
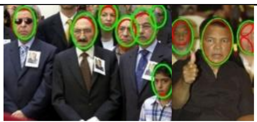
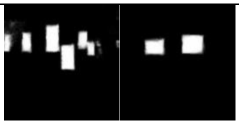
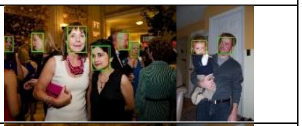
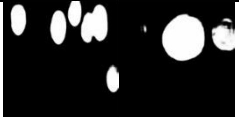
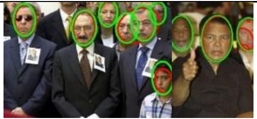
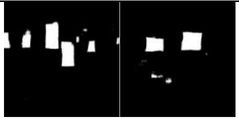
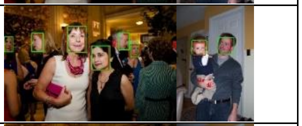
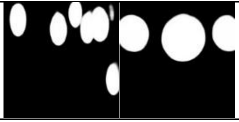
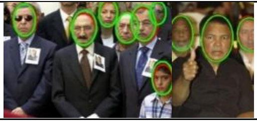
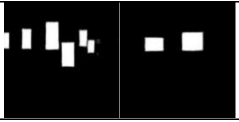
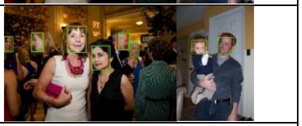
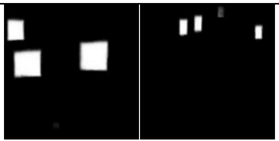
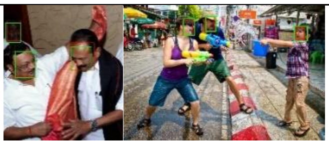
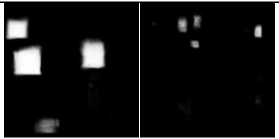
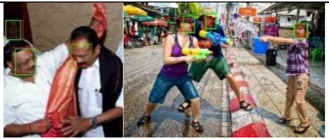
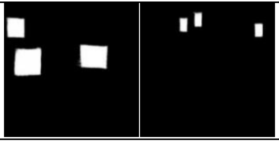
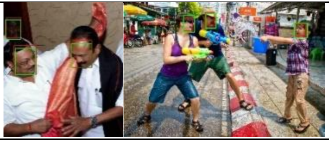
Model	*Predictor Output*	*Detection Results*	*Predictor Output*	*Detection Results*
Unet				
VGG16-Unet				
Semi-Dense U-Net (proposed)				
	a *b*	*c* *d*	*e* *f*	*g* *h*

TABLE VII. OUTPUT OF SAMPLE IMAGES FROM WIDER FACE DATASET. (A), (B) ARE MODEL OUTPUT. (C), (D) ARE ORIGINAL IMAGES WITH PREDICTIONS IN RED ELLIPSES/BOUNDING BOXES AND GROUND TRUTH IN GREEN ELLIPSES/BOUNDING BOXES

Model	*Predictor Output*	*Detection Results*
Unet		
Vgg16-unet		
Semi-Dense unet (proposed)		
	a *b*	*c* *d*

V. Conclusion

Segmentation architectures are gradually increasing in numbers as does its applications. Performance of the architecture is still of concern even today. In this paper, we proposed to improve standard U-Net segmentation architecture commonly used in medical image segmentation and applied it to face segmentation and human faces detection. Our proposed architecture, Semi-Dense U-Net, produces improved results compared to standard U-Net architecture. Here, feature learning is improved by introducing skip connections and dense connections at various levels. While it produced considerably good prediction results for medium and large face, the model may not be suitable for application with tiny face detection requirements. In the future work, the architecture will be further improved to detect tiny faces and will be fine-tuned to predict occluded and heavily blur faces.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Conceptualization, methodology, training and validation, writing original draft preparation and editing, Ganesh Pai; writing review and editing, supervision, Sharmila Kumari M.

References

[1] M. Propp and A. Samal, “Artificial Neural Network architectures for human face detection,” in Intelligent Engineering Systems Through Artificial Neural Networks, 1992, vol. 2, pp. 535–540.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in Proceedings - Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[3] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.

[4] C. Szegedy et al., “Going deeper with convolutions,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jun. 2015, vol. 07-12-June, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jun. 2016, vol. 2016-Decem, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Jul. 2017, vol. 2017-Janua, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[7] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” Apr. 2018, [Online]. Available: http://arxiv.org/abs/1804.02767.

[8] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9351, no. Cvd, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[9] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: A Nested U-Net Architecture for Medical Image Segmentation BT - Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support,” Miccai, vol. 11045, no. 2018, pp. 3–11, 2018, doi: 10.1007/978-3-030-00889-5.

[10] O. Cakiroglu, C. Ozer, and B. Gunsel, “Design of a deep face detector by mask R-CNN,” 27th Signal Process. Commun. Appl. Conf. SIU 2019, no. April, pp. 1–4, 2019, doi: 10.1109/SIU.2019.8806447.

[11] H. Wang, X. Jiang, H. Ren, Y. Hu, and S. Bai, “SwiftNet: Real-time Video Object Segmentation,” in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2021, pp. 1296–1305, doi: 10.1109/CVPR46437.2021.00135.

[12] H. Lu, Y. She, J. Tie, and S. Xu, “Half-UNet: A Simplified U-Net Architecture for Medical Image Segmentation,” Front. Neuroinform., vol. 16, no. June, pp. 1–10, 2022, doi: 10.3389/fninf.2022.911679.

[13] X. Li, H. Chen, X. Qi, Q. Dou, C. W. Fu, and P. A. Heng, “H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation from CT Volumes,” IEEE Trans. Med. Imaging, vol. 37, no. 12, pp. 2663–2674, 2018, doi: 10.1109/TMI.2018.2845918.

[14] K. Lin et al., “Face Detection and Segmentation Based on Improved Mask R-CNN,” Discret. Dyn. Nat. Soc., vol. 2020, 2020, doi: 10.1155/2020/9242917.

[15] T. Agrawal and P. Choudhary, “ReSE-Net: Enhanced UNet architecture for lung segmentation in chest radiography images,” Comput. Intell., Apr. 2023, doi: 10.1111/coin.12575.

[16] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 936–944, 2017, doi: 10.1109/CVPR.2017.106.

[17] S. G. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 11, no. 7, pp. 674–693, Jul. 1989, doi: 10.1109/34.192463.

[18] E. J. Stollnitz, A. D. DeRose, and D. H. Salesin, “Wavelets for computer graphics: a primer.1,” IEEE Comput. Graph. Appl., vol. 15, no. 3, pp. 76–84, May 1995, doi: 10.1109/38.376616.

[19] P. Viola and M. J. Jones, “Robust Real-Time Face Detection,” Int. J. Comput. Vis., vol. 57, no. 2, pp. 137–154, 2004, [Online]. Available: https://link.springer.com/content/pdf/10.1023/B:VISI.0000013087.49260.fb.pdf.

[20] S. Yang, P. Luo, C. C. Loy, and X. Tang, “Faceness-Net: Face Detection through Deep Facial Part Responses,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 8, pp. 1845–1859, Aug. 2018, doi: 10.1109/TPAMI.2017.2738644.

[21] S. Yang, Y. Xiong, C. C. Loy, and X. Tang, “Face detection through scale-friendly deep convolutional networks,” arXiv, 2017, [Online]. Available: http://arxiv.org/abs/1706.02863.

[22] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks,” IEEE Signal Process. Lett., vol. 23, no. 10, pp. 1499–1503, 2016, doi: 10.1109/LSP.2016.2603342.