UNet-Dense: An Efficient and Dense U-Net Model for Binary Instance Segmentation

Rutvik Gupta
University of Toronto
rutvik.gupta@mail.utoronto.ca

Kartik Sangwan
University of Toronto
kartik.sangwan@mail.utoronto.ca

Tanmay Gupta
University of Toronto
tanmay.gupta@mail.utoronto.ca

Abstract

We propose UNet-Dense, a simple and efficient instance segmentation network for the COCO dataset. The architecture is based on the original UNet with encoder and decoder modules, using HeNormal initialization, batch normalization, and skip connections. Our model is trained using the Adam optimizer and binary cross-entropy loss function. To evaluate the model, we conducted two experiments. In the first experiment, we compared UNet-Dense with UNet-Simple and UNet-SuperDense to determine the optimal model complexity for the task of image segmentation. We found that UNet-Dense achieved the best balance between the number of parameters, dataset size, and task complexity, resulting in better training and validation accuracy. In the second experiment, we compared UNet-Dense with state-of-the-art models such as TernausNet, InceptionV3, and DeepLabV3Plus. While our model's performance was slightly lower than that of the state-of-the-art DeepLabV3Plus, it demonstrated efficient use of resources and shorter training time, making it a valuable option for researchers and developers facing resource constraints. With further refinement and exploration of pre-training and transfer learning techniques, the UNet-Dense model has the potential to achieve even better results in image segmentation tasks. The code for this work can be found at https://github.com/RutvikGupta/CSC413-Final-Project.

Introduction

Image segmentation is a rapidly growing area in computer vision and image understanding, with applications in industrial inspection, medical image diagnosis, autonomous driving, satellite image processing, and human body parsing. Despite advances in deep convolutional neural networks (CNNs), image segmentation remains a challenging problem due to limited segmentation labels and time-consuming annotation tasks. UNet models, originally developed for biomedical image segmentation, are a class of CNNs that exhibit a U-shaped architecture, comprising an encoder-decoder structure [1] . The encoder captures contextual information by downsampling input images through a series of convolutional and pooling layers, while the decoder reconstructs the segmented output using transposed convolutions and upscaling. This powerful architecture has been widely adopted in various fields beyond biomedical imaging, such as satellite imagery, autonomous driving, and computer vision. UNet models have shown remarkable performance in these domains, providing accurate and detailed segmentation results. In this paper, we will implement a simple UNet model to perform a binary image segmentation of humans on Microsoft's Common Objects in Context (COCO) dataset [2]. We will then increase the model's complexity by adding more layers to compare how they perform relative to each other. Lastly, we will select the best-performing model, tune its hyperparameters and then compare it with state-of-the-art segmentation models such as TernausNet [3], Google's InceptionV3 [4], and DeepLabV3Plus [5].

Related Work

Research on image segmentation has extensively explored the use of UNet models. Our work takes inspiration from several key contributions in this field. Ronneberger et al. [2015] [1] present a modified fully convolutional network architecture that functions well with few training images and delivers precise segmentations. By replacing pooling operators with upsampling operators, they increase the output resolution. They combine high-resolution features with upsampled output for localization, while a successive convolution layer learns to generate a more accurate output based on this data. The expansive path closely mirrors the contracting path, forming a U-shaped architecture without fully connected layers, allowing for seamless segmentation of large images using an overlap-tile strategy [1].

Iglovikov et al. [2018] [3] demonstrate the enhancement of UNet's performance by employing pre-trained weights. They apply this technique to the Aerial Image Labeling Dataset, which consists of high-resolution aerospace images from various cities. These images have pixels categorized into two distinct classes: "building" and "non-building." By leveraging pre-trained weights, the UNet model achieves superior results in this application, proving the effectiveness of their proposed approach [3]. Chen et al. [2018] [5] introduce a new encoder-decoder structure that utilizes DeepLabV3Plus as a robust encoder and an efficient decoder module. This structure allows for adjustable resolution in the encoder features using atrous convolution, offering a balance between precision and runtime not feasible in existing models. By adopting the XceptionNet model for segmentation and applying depthwise separable convolution to both ASPP and decoder modules, they create a faster and more powerful encoder-decoder network [4].

In our paper, we aim to reduce the complexity of the model by limiting the number of encoder-decoder layers and avoiding the use of any pre-trained weights. Instead, we retain the UNet structure combined with a batch-normalization approach and employ alternative weight initialization techniques to improve segmentation performance.

Data

The Microsoft COCO [2] dataset is used for instance image segmentation to detect humans. COCO [2] comprises a large database containing approximately 90 classes of various objects in diverse real-world environments. In this paper, the COCO [2] 2017 training and validation datasets are employed. Initially, the training images are centrally and randomly cropped to a size of 224x224 pixels with three color channels. To prepare the training and validation datasets, we shuffle the images, associate the images with their corresponding masks, replicate the dataset for a predetermined number of epochs, and batch the selected images together [1].

Method

UNet [1] is a widely-used architecture for image segmentation tasks, comprising two primary components: an encoder and a decoder module. The encoder progressively downsamples the input image using a combination of convolution and max-pooling layers. This process results in an efficient bottleneck that encodes the image's context in a low-dimensional space. In contrast, the decoder restores the encoded information by applying deconvolution layers, also known as transposed convolution layers. These layers upsample the feature maps to match the original size of the input image, which allows for precise object localization within the image. U-Net is commonly employed for tasks such as image segmentation, where the goal is to compute boundaries for different objects in an image.

UNet-Dense

We propose a UNet-Dense, a variation of the UNet [1] architecture designed for efficient and fast instance segmentation. It features a streamlined structure consisting of 4 encoder blocks (ConvEncoder) and 4 decoder blocks (ConvDecoder). Each ConvEncoder block contains two convolution layers with ReLU activation, HeNormal weight initialization [7], and a 3x3 filter size. To improve the model's generalization capabilities, it employs batch normalization layers and max-pooling layers. The ConvDecoder block uses Conv2DTranspose layers for image deconvolution and upsampling, in addition to a Conv2D layer to learn new features from concatenated inputs. Skip connections are implemented to retain information from previous layers, enhancing the model's data generalization. The final output is a 256x256x1 binary mask, with the model comprising approximately 8 million parameters overall. [See Figure 1]

TernausNet

TernausNet [3] is another U-Net-based model that incorporates a pre-trained VGG-11 [3] encoder and a decoder module. It features 16 convolution layers with ReLU activations and 3x3 filters, as well as 5 max-pooling layers that successively reduce input dimensions while increasing the number of channels. The central layer has 512 channels and acts as

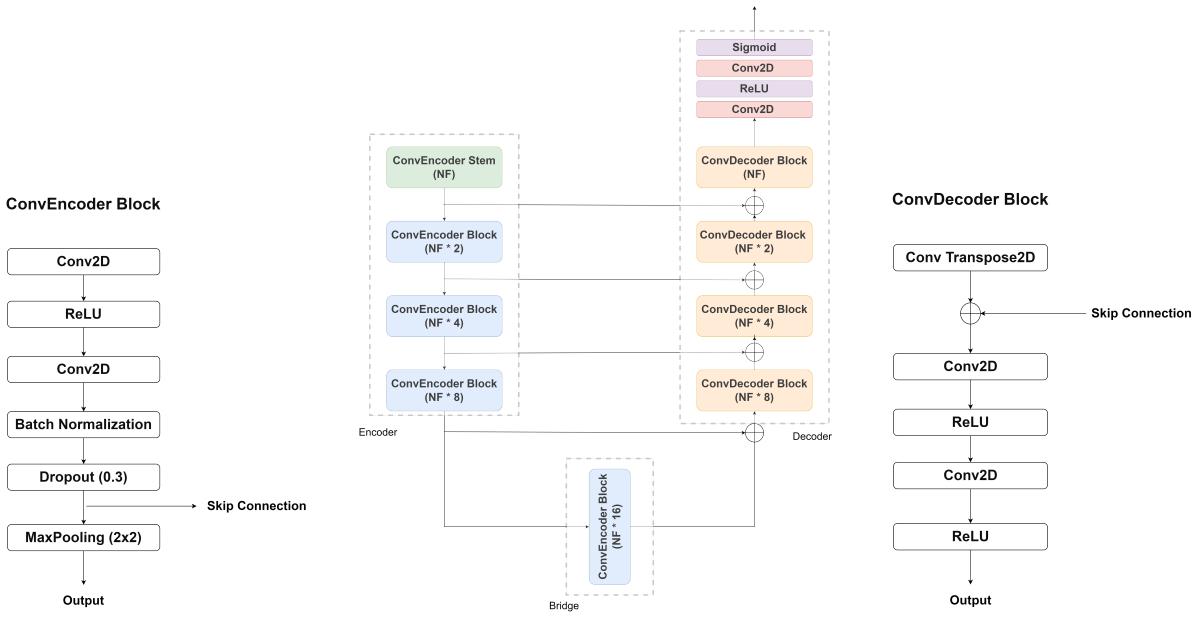

Figure 1: **Left:** The ConvEncoder Block. **Middle:** The UNet-Dense architecture. **Right:** The ConvDecoder Block. The model takes in an image of size (3, H, W) and outputs a segmentation mask of size (1, H, W). NF: No. of Filters.

a bottleneck separating the encoder from the decoder. The decoder consists of 5 deconvolution layers with varying numbers of filters and kernel sizes, which reconstruct the output image from the low-dimensional feature representation produced by the encoder. Skip connections are also used to preserve information from earlier layers, making it easier for the network to learn the relationship between the input image and output mask. The resulting output is a binary mask, and the model contains around 32 million parameters.

DeepLabV3Plus

DeepLabV3Plus [5] is a deep CNN encoder-decoder model specifically developed for semantic image segmentation tasks. Its key innovation is the inclusion of atrous/dilated spatial pyramid pooling (ASPP) in the encoder module, which is a pre-trained ResNet50 [5]. ASPP [5] layers are responsible for encoding input features at multiple dilation rates, enabling the model to capture information at different resolutions without increasing the number of parameters. This approach allows the model to process both global and local context information within the input image, refining the segmentation output across multiple scales. The concatenated output from the ASPP [5] layers is then fed into the decoder, which subsequently up-samples and converts it into a binary prediction mask. DeepLabV3Plus [5] is currently state-of-the-art in semantic image segmentation and contains approximately 12 million parameters.

Training

We experimented with multiple-weight initialization and concluded that HeNormal [7] would suit our encoder. HeNormal [7]. Specifically, the HeNormal [7] initialization method sets the initial weights of each layer's neurons in CNN to a Gaussian distribution with zero mean and a standard deviation that is dependent on the number of neurons in the previous layer. HeNormal [7] helps prevent vanishing and exploding gradient problem and also improve training speeds. Simple-Net has a very small number of parameters which further reduces the training time and amount of data required to achieve close to state-of-the-art accuracy on the image segmentation task. For training the model we used the Tesla K80 GPU with 12GB of RAM and ran the model training and corresponding experiments. Each epoch of the training required about 1-2 minutes for a batch size of 32. In training the model, we used the Adam optimizer [9] with a learning rate of 0.0001. Adam [9] was chosen as it incorporates momentum to accelerate the optimization process. Additionally, Adam [9] incorporates L2 regularization by default, which helps prevent over-fitting in very large models during training. We use the common binary cross-entropy loss function for comparing binary pixel class which indicates whether or not the image pixel belongs to the interesting image segment (humans in our case).

Binary Cross-Entropy and loss function for image segmentation can be expressed as: is given by:

$$H = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)] \tag{1}$$

$$Loss = H - \log J \tag{2}$$

where J is the intersection over union (IoU) metric that measures the similarity between the predicted segmentation mask and the ground truth mask [3]. Minimizing this loss function implies that we maximize J which is IoU and minimize H, which is Binary Cross Entropy. Both are desired in the scenario as we want to predict the class of the correct pixel and maximize the intersection over the union between the predicted and ground truth segmentation masks. Lastly, after experimentation, we concluded that batch size of 40, 40 epochs, and 0.0001 learning rate served as optimal hyper-parameters which optimized the model validation accuracy, and mean IOU for the given time and resource constraints.

Experiments

Experiment 1: Model Complexity

In this experiment, we take the UNet-Simple and gradually increase the number of encoder and decoder blocks in the model. The first configuration called UNet-Simple consists of 3 encoder and 2 decoder blocks. The second configuration is the UNet-Dense which consists of 5 encoder and 4 decoder blocks. The third model configuration called UNet-SuperDense consists of 9 encoder and 8 decoder blocks. We keep all other hyper-parameters the same and train all three models in order to determine the optimal model complexity for the task of image segmentation on the COCO dataset [2].

Experiment 2: Comparison with State of Art

In this experiment, we compare our UNet-Dense model with TernausNet [3], InceptionV3 [4], and DeepLabV3Plus [5] models to evaluate the performance relative to the current state-of-the-art models. We train all three models on the same data set, a number of epochs, batch size, learning rate, and optimizer. We evaluate and compare the train, validation loss, and accuracy across the four models. Furthermore, we perform time profiling for each of the models to determine the resource consumption in each case.

Results

Result 1: Model Complexity

The table below summarises the training and validation results for the three models:

Model	Parameters	Validation Accuracy	Training Accuracy
UNet-Simple	0.528M	79.2%	81.1%
UNet-Dense	8.164M	**85.6%**	**86.2%**
UNet-SuperDense	553M	74.9%	85.5%
CNN-Dense	120M	74.3%	79.1%

Table 1: Results for UNet Model Complexity performance on COCO dataset [2]

As the number of encoder-decoder blocks increases, the number of neurons and connections also increases. Due to its limited number of parameters, the Unet-Simple model fails to encode the image with sufficient feature encoding, resulting in poor performance and low validation and training accuracy. The model under-fits the given data. The Unet-Dense provides the correct balance between the number of parameters, dataset size, and image segmentation task complexity, resulting in better training and validation accuracy. However, the UNet-SuperDense contains a very high number of parameters for our task and dataset size. Therefore, it overfits the training dataset and performs poorly at generalization. The CNN-Dense model, which has the same architecture as UNet-Dense but without skip connections and more encoder-decoder layers performs poorly which indicates the significance of skip connections to encode previous feature encoding and improve the localization of image objects at different resolutions. In summary, selecting an appropriate number of encoder-decoder blocks and balancing the number of parameters with the dataset size and complexity of the task are crucial for achieving optimal model performance as shown in the experiment.

Result 2: Comparison with State of Art

The table below summarises the results for training accuracy, validation accuracy, binary-cross entropy loss, and timing analysis of our model in comparison with state-of-the-art techniques.

Model	Parameters	Validation Accuracy	Training Accuracy	Training Time
UNet-Dense	8.164M	85.6%	86.2%	1.6 hrs
TernausNet	32.2M	90.3%	91.3%	3.1 hrs
DeepLabV3Plus	11.85M	**93.4%**	**93.9%**	2.4 hrs
InceptionV3	21.82M	89.3%	90.6%	2.7 hrs

Table 2: Results for UNet-Dense and other State-of-the-Art Models performance on COCO dataset [2]

We can observe that our UNet-Dense model is able to achieve about 86% accuracy on the validation data set of COCO image segmentation of the human object class. DeepLabV3Plus [5] which is a state-of-the-art model reaches about 93.4% accuracy. Inceptionv3 [4] and TernausNet [3] architectures reach 90.3% and 89.3% accuracy respectively. Our model performs slightly poorly in comparison to DeepLabV3Plus [5], but the results are significant. In terms of the complexity of the model, our model contains about two-thirds the number of parameters as compared to DeepLabV3Plus [5]. The training time for our model is about half the training time required for DeepLabV3Plus [5]. Lastly, DeepLabV3Plus [5] encoder weights are pre-trained on the ImageNet [8] data set containing millions of images, making the model highly robust and generalizable. Our model did not use any pre-trained weights and is capable of running and training on a single GPU in about 1.6 hours from scratch. Overall, our UNet-Dense model optimizes the use of compute resources as it is significantly less complex and requires less training time while compromising slightly on accuracy. The training, validation curves, and predicted masks for certain images comparing all three models are provided in the appendix in Figures 2-9.

Conclusion

We developed and analyzed UNet-Dense, a simple and efficient architecture for image segmentation tasks. Through a series of experiments, we investigated the impact of model complexity on performance and compared our model with state-of-the-art techniques. Our findings highlight the importance of selecting an appropriate level of complexity for a given task and dataset size to achieve optimal results. UNet-Dense achieved a validation accuracy of 85.6% and a training accuracy of 86.2% on the COCO [2] dataset. Although it slightly underperformed compared to DeepLabV3Plus [5], the top-performing model with 93.4% validation accuracy, UNet-Dense offers a significantly less complex architecture with fewer parameters and reduced training time. The model can train on a single GPU in 1.6 hours from scratch, making it an attractive option for practitioners with resource constraints. In conclusion, our UNet-Dense model optimizes the use of compute resources while providing efficient performance in image segmentation tasks. By balancing model complexity, dataset size, and task requirements, it offers a promising approach for researchers and practitioners working under resource constraints.

Limitations and Future Work

The UNet-Dense model has the potential to achieve improved performance in image segmentation tasks with further refinement, pre-training, and the application of transfer learning techniques. However, hardware and RAM limitations during this study restricted the exploration of various hyperparameter configurations, such as batch sizes, learning rates, and optimizer settings. Tailoring model complexity for specific tasks is crucial to prevent under-fitting. Incorporating advanced data augmentation techniques could greatly benefit the model's performance by diversifying the training data and enhancing its generalization to unseen images. Some of these data augmentation techniques include random rotation, flipping, scaling, cropping, color jittering, and the addition of noise, such as Gaussian noise or salt-and-pepper noise. These techniques can help improve the model's robustness to different types of input images. Researchers can also experiment with tuning hyperparameters, applying different loss functions like weighted-cross entropy or dice loss, and modifying the architecture for multi-class image segmentation, which would be beneficial in various real-world applications, such as medical imaging, autonomous driving, and remote sensing. Additionally, future research could explore the integration of attention mechanisms, such as spatial and channel attention, to improve model performance. Chen et al. [2016] [6] propose an attention mechanism that can help the model focus on important regions or features in the input image, potentially resulting in more accurate and precise segmentation results.

Contribution

Rutvik Gupta

Researched Methods, Worked on Code, Trained and Evaluated Models, and Generated Visualizations and Results. Added Architecture Diagram and Appendix section to the Report.

Kartik Sangwan

Researched Methods and Experiments, Wrote Methods, Experiments, Results Section, and Future Work sections of the Report.

Tanmay Gupta

Trained and Evaluated Models, Wrote Abstract, Introduction, Conclusion, and Limitations sections of the Report.

References

[1] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 234-241. https://arxiv.org/abs/1505.04597

[2] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 740-755. https://arxiv.org/abs/1405.0312

[3] Iglovikov, V. I., Shvets, A. A. (2018). TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation. *arXiv preprint arXiv:1801.05746*. https://arxiv.org/abs/1801.05746

[4] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818-2826. https://arxiv.org/abs/1512.00567

[5] Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H. (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 801-818. https://arxiv.org/abs/1802.02611

[6] Chen, L. C., Yang, Y., Wang, J., Xu, W., Yuille, A. L. (2016). Attention to Scale: Scale-Aware Semantic Image Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3640-3649. https://arxiv.org/abs/1511.03339

[7] He, K., Zhang, X., Ren, S., Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1026-1034. https://arxiv.org/abs/1502.01852

[8] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248-255. http://www.image-net.org/papers/imagenet_cvpr09.pdf

[9] Kingma, D. P., Ba, J. (2014). Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1412.6980

Appendix

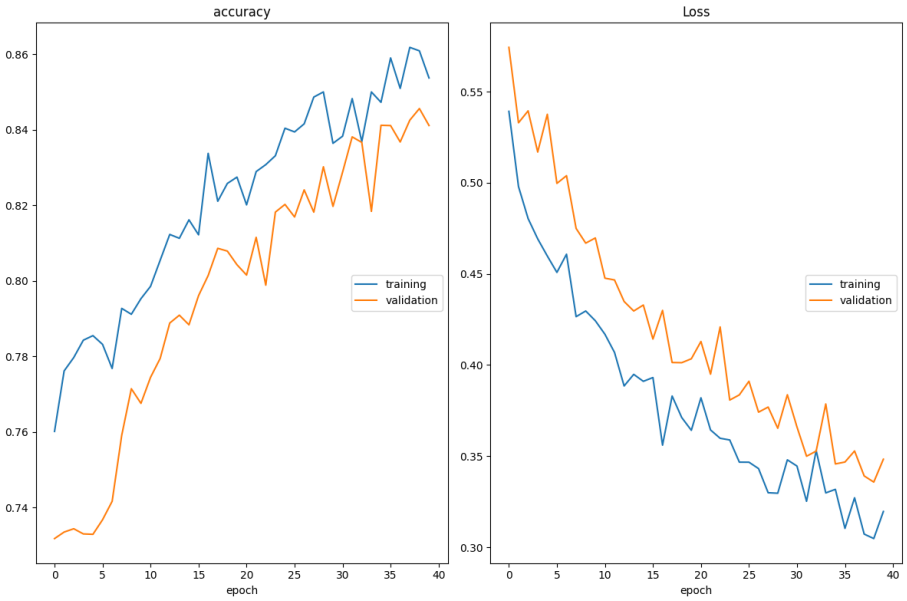

Figure 2: UNet-Dense Training and Evaluation Plots

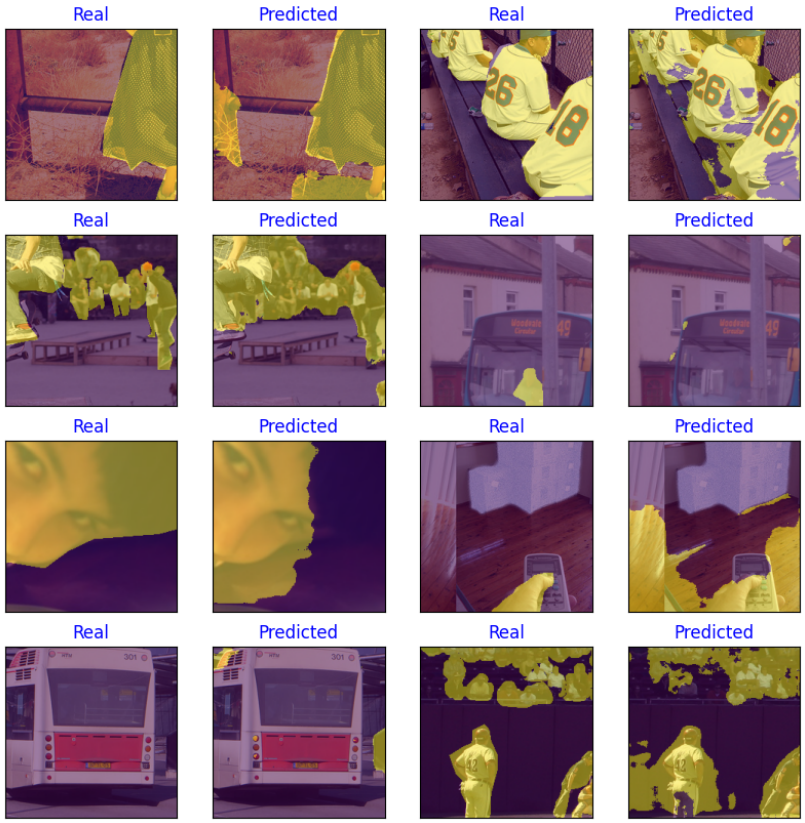

Figure 3: Binary Human Instance Segmentation on COCO dataset [1] by UNet-Dense Model

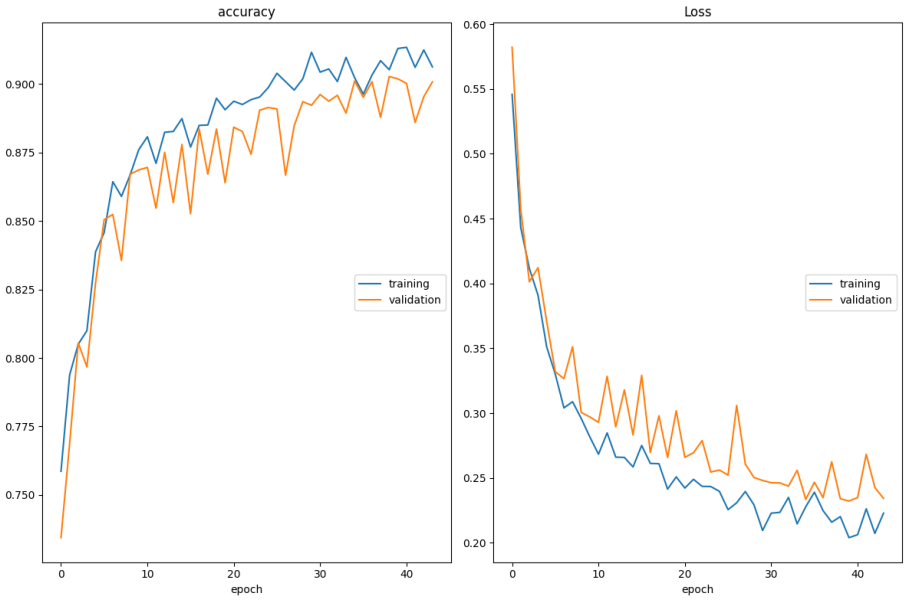

Figure 4: TernausNet Training and Evaluation Plots

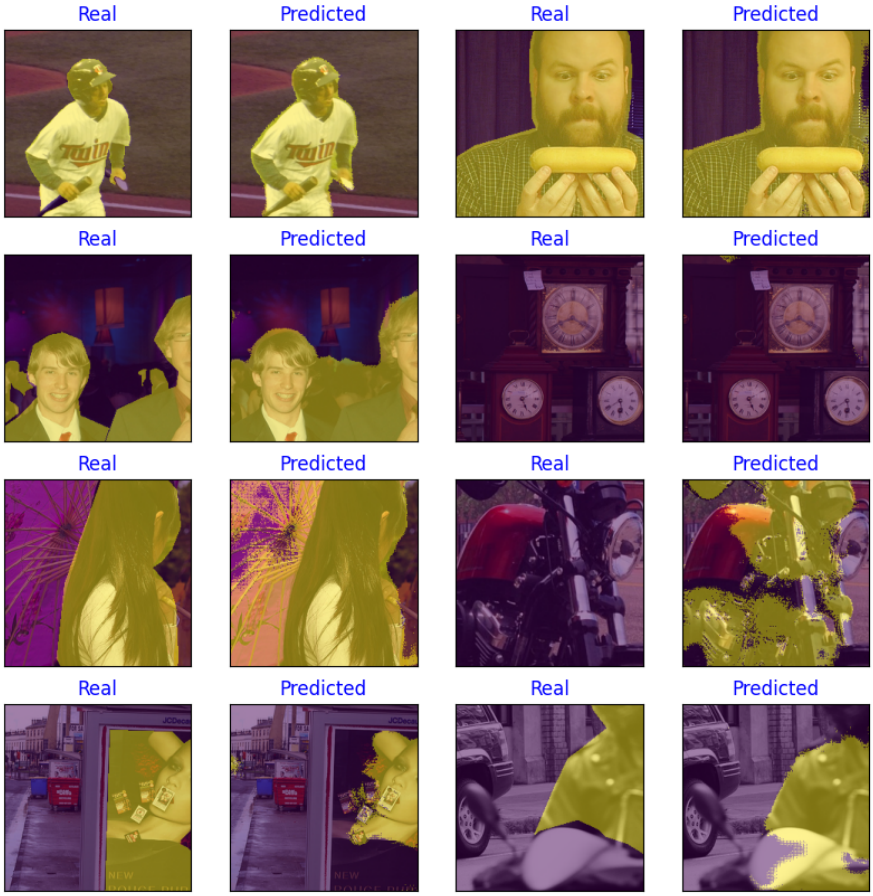

Figure 5: Binary Human Instance Segmentation on COCO dataset [1] by TernausNet Model

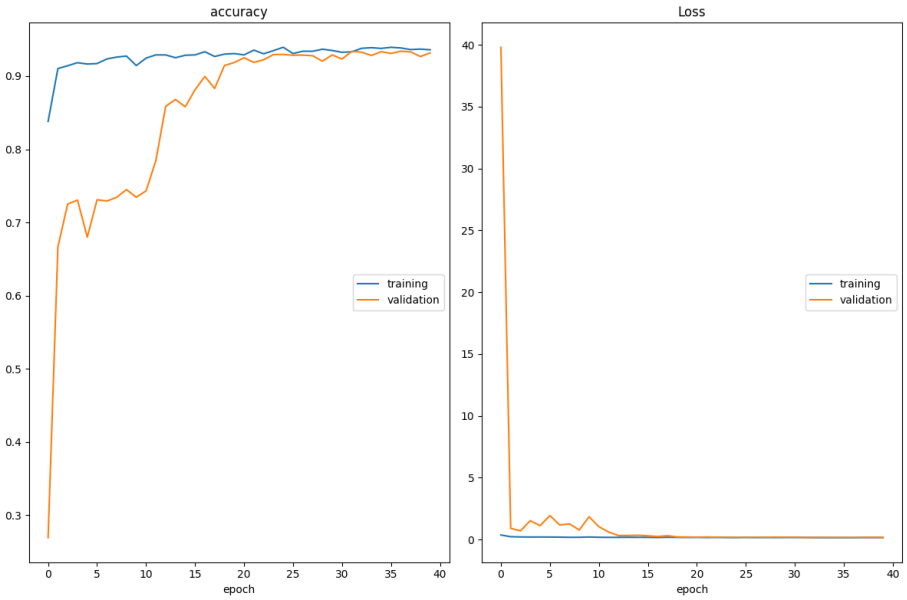

Figure 6: DeepLabV3Plus Training and Evaluation Plots

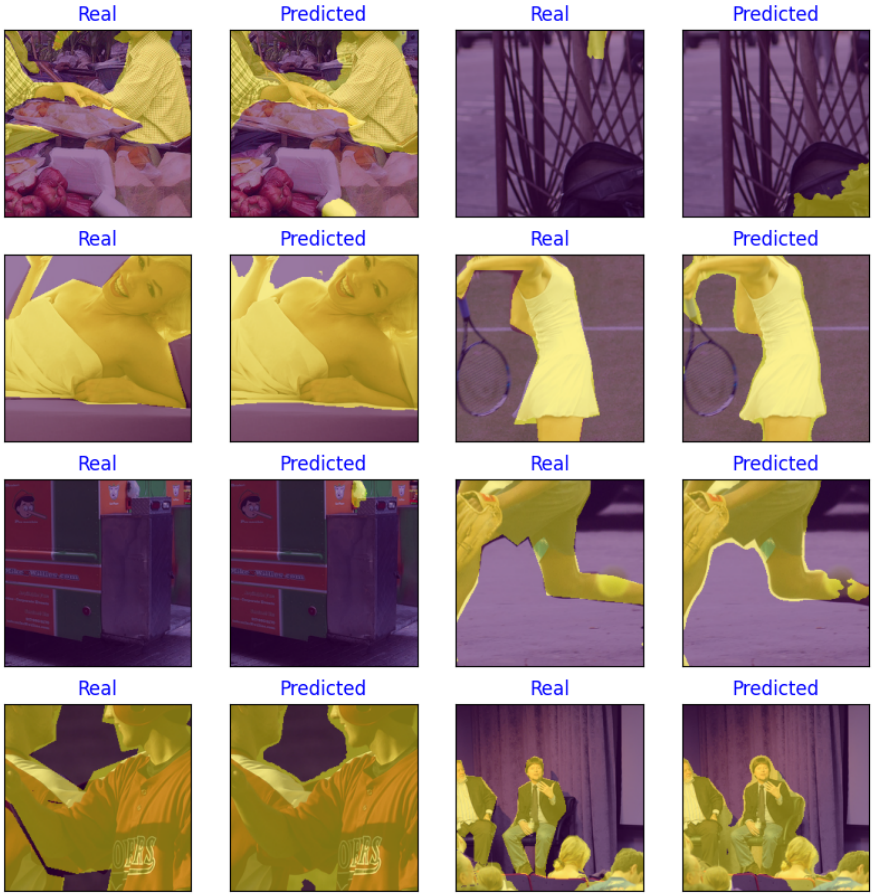

Figure 7: Binary Human Instance Segmentation on COCO dataset [1] by DeepLabV3Plus Model

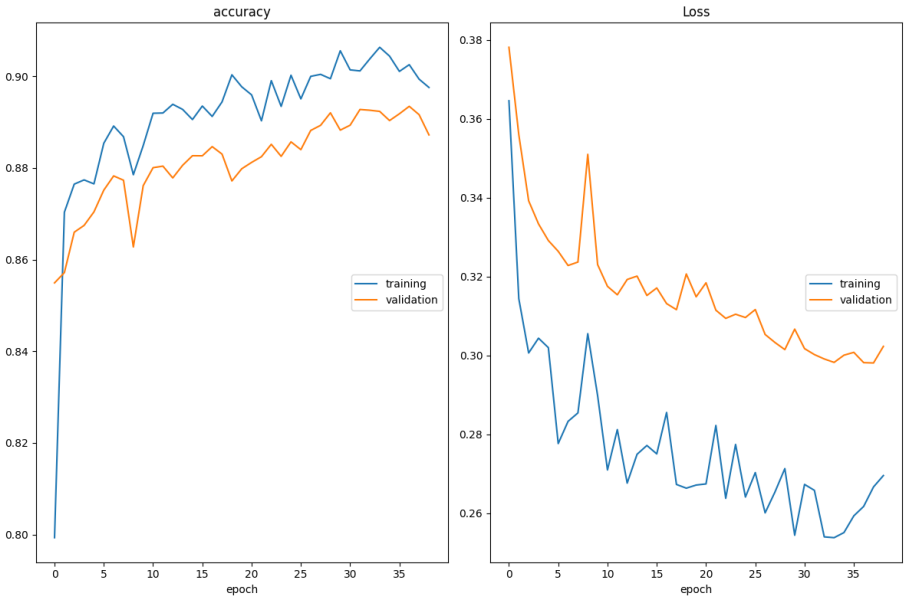

Figure 8: InceptionV3 Training and Evaluation Plots

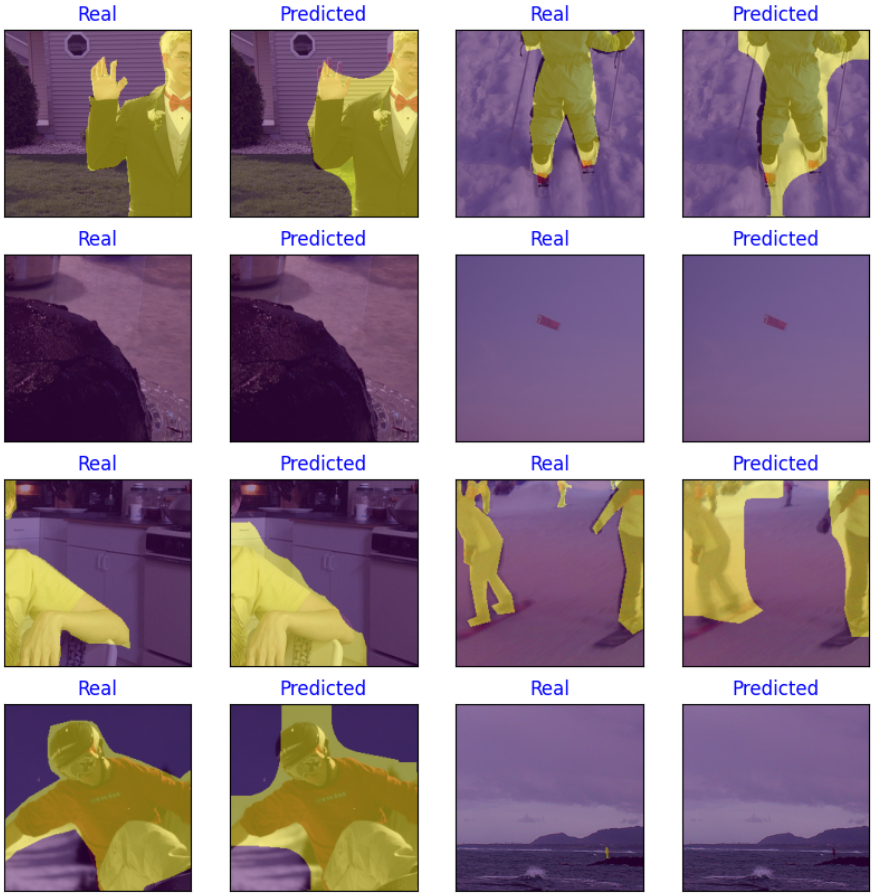

Figure 9: Binary Human Instance Segmentation on COCO dataset [1] by InceptionV3 Model