Image segmentation of soft-body object using UNet-style architectures

Abstract

The tongue is a complex muscular organ that performs intricate and rapid movements, making it difficult to study using conventional methods. Tracking tongue movements in neuroscience research is important to understand the neural basis of complex motor actions and sensorimotor integration, and to develop treatments for speech and swallowing disorders. Advancements in imaging technologies and deep learning neural networks have enabled high definition and real-time tracking of the tongue, which can be used to study tongue movements in mouse models for sensorimotor integration. This project aimed to compare existing segmentation tools, such as UNet, UNet++, and DeepLabV3, to find the best neural network architecture for segmenting tongue in static images of the mouse's face captured from a bottom view camera. Results showed that a small UNet-style nerual network achieved the best inference time while still maintaining comparable segmentation performance as the other models. Furthermore, UNet models achieved a high test accuracy of over 90% measured with multiple evaluation metrics commonly used for image segmentation. The project thus demonstrated the feasibility of small UNet-style architectures for tongue segmentation in real-time applications.

1 Introduction

Tracking tongue movements is crucial for studying the neural control of complex motor actions such as eating, drinking, and speaking (1; 2). However, the tongue is a complex muscular organ that performs intricate and rapid movements, making it difficult to study using conventional methods. Understanding the neural control of tongue movements can provide valuable insights into how the brain controls complex motor actions. Such insights can assist in the development of treatments for speech and swallowing disorders that are prevalent in conditions such as stroke and Parkinson's disease (3; 4). Overall, tracking tongue movements is valuable for neuroscience research and has the potential to advance our understanding of the neural basis of complex motor actions and sensorimotor integration.

In recent years, advancements in imaging technologies have made it possible to record tongue movements in high definition and in real-time. Unlike other body parts, tracking the movement of soft-body objects like the tongue requires semantic segmentation rather than keypoint tracking. Therefore, combining high-speed imaging with deep learning neural networks can enable the analysis of tongue movements in mouse models for studying sensorimotor integration. Some of the commonly used tools for semantic segmentation include UNet, UNet++ and DeepLabV3 (5; 6; 7). Semantic segmentation tools can thus be valuable for tracking tongue in images of mouse face since the segmented mask can be subsequently used to extract/estimate variables quantifying different tongue movements.

This project aimed to leverage data collected from high-speed imaging cameras and utilize advanced image segmentation techniques to identify and track a soft-body object, specifically a mouse tongue, in static images of the mouse's face captured from a bottom view. The primary objective was to compare existing segmentation tools and find a neural network architecture that can generalize best on new/test data for this specific image segmentation task. In addition, the project compared the feasibility of the different models for use in closed-loop behavioral experiments to study tongue movements in real-time.

2 Related work

Neuroscience research labs use mouse models to study complex motor actions performed by tongue and how the brain controls them (8; 9). Recent work extracted and tracked only some features of the tongue, such as the tongue tip and base. A recent paper employed a UNet for segmenting mouse tongue using images recorded from different camera views and subsequently used the segmented masks to reconstruct a 3D view of the tongue (9). However, the research did not quantify and compare performance of the model with existing state-of-the-art segmentation tools. Furthermore, the inference time of the model limits its usage to segmentation during offline analyses only. Hence, the aim of this project was to find the best neural network architecture that can be employed for online analyses and can generalize to different experimental setups/variations.

A number of existing semantic segmentation tools employ different features in a neural network. UNet is a convolutional neural network architecture developed specifically for biomedical image segmentation. It consists of an encoder and a decoder, with skip connections that allow the network to preserve spatial information throughout the segmentation process (5). DeepLabV3 is a state-of-the-art semantic segmentation model that uses atrous convolution (also known as dilated convolution) to increase the receptive field of the network without increasing the number of parameters. DeepLabV3 also uses a series of parallel atrous convolution layers with different dilation rates to capture multi-scale contextual information (7). Hence, the project compared these existing neural network architectures to identify the best model for tongue segmentation and also implemented a smaller UNet model, which this project proposes as the best model for the image segmentation task.

3 Methods

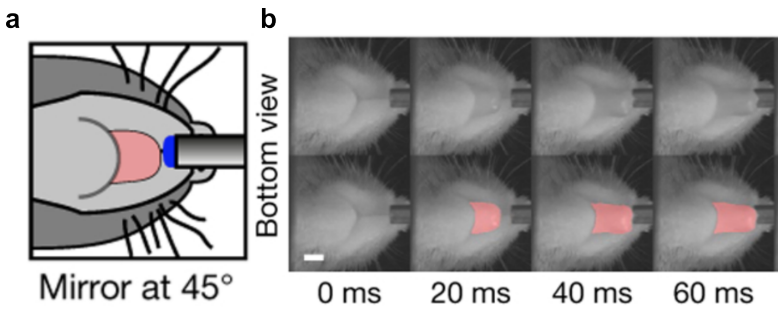

Figure 1: **Recordings of mouse tongue movement from bottom view camera. a**. Bottom view illustration **b**. Bottom view camera recordings at different time points showing raw frames [top] and labelled frames by human annotator [bottom]. Images obtained from (9).

3.1 Dataset

The dataset used for this project was obtained from two labs. The dataset consisted of the bottom view of the mouse face (9) recorded using a high-speed imaging camera placed in front of a mirror angled at 45° below the mouth of the mouse (Figure 1). Videos were acquired with a resolution of 192×200 pixels at $1,000$ frames per second (fps). The dataset from one of the lab consisted of frames pseudo randomly selected from a dataset of $25,258,017$ frames from 12 mice. The other dataset obtained (8) was recorded using a high-speed video camera (400Hz, $32\mu m$ per pixel, 400×320 pixels) providing bottom view of the mouth region. The entire dataset consisted of a total of $16,772$ frames. The dataset was split into a training set consisting of $13,687$ frames and a validation set consisting of $1,521$ frames, which was used to evaluate the performance of the model during training. The dataset split is summarized in Table 2. The dataset consisted of balanced number of frames with and without the tongue visible in frames (Table 3). The size of images varied from 192×200 to 400×320 pixels. During pre-processing step, all samples were down-sampled or up-sampled to image size of 256×256 pixels and converted to grayscale.

3.2 Models

A number of different neural network architectures were used for comparison. The backbone and details of the model architectures is illustrated in Figure 3 (appendix). Following subsections summarize the key features of the different models.

3.2.1 UNet

UNet is a U-shaped convolutional neural network architecture commonly used for image segmentation tasks. It consists of an encoder network that reduces the spatial dimensions of an input image and consists of convolutional layers with pooling operations. A decoder network follows the encoder network and consists of upsampling layers and convolutional layers to finally produce a segmentation mask as its target/output. In addition, the network uses skip connections between corresponding layers in the encoder and decoder networks to preserve spatial information from the input image. The different features of the network have proven it useful for segmentation tasks, such as in medical imaging (5).

3.2.2 UNet - Small

The project implemented a modified UNet architecture called UNet - Small to test a smaller network for the image segmentation task. UNet - Small uses fewer number of filters (feature maps) and the number of upsampling steps are reduced as well. The objective of this network modification was to implement a network similar to UNet, but make it smaller in terms of the number of parameters to potentially reduce the inference time while preserving the key features of a UNet-style architecture. This is useful for implementation in closed-loop behavioral experiments where high frame-rate videos have to be processed in real-time. Further details of the UNet - Small architecture are shown in Table 4 and the backbone of the model is illustrated in Figure 3 (appendix).

3.2.3 UNet++

UNet++ is similar to a UNet designed to address some of its limitations in capturing fine details and context information in segmentation tasks. UNet++ includes a nested, recursive approach to combine feature maps at each stage of the decoder network, rather than using skip connections as in U-Net. Specifically, UNet++ includes a series of nested U-shaped architectures, where each U-Net block receives input from the previous block and combines feature maps using a concatenation operation. In addition, the model includes attention gates to focus on important features. This modified approach thus allows the network to capture context information at different scales and improve the segmentation accuracy (6).

3.2.4 DeepLabv3

DeepLabv3 is a popular deep neural network for semantic segmentation, which has achieved state-of-the-art performance on several benchmark datasets. The key features of DeepLabv3 include a multi-scale feature extractor, dilated convolutional network, spatial pyramid pooling module and a decoder network with skip connections. DeepLabv3 consists of different architectures that can be used as its backbone, which include ResNet50, ResNet101 and mobilenet. The backbone extracts features at multiple scales to capture both fine and coarse details in the input image. A dilated convolutional network is then used to increase the receptive field of the network while preserving spatial resolution. The spatial pyramid pooling module captures multi-scale context information by pooling features at different scales and resolutions. A decoder network with skip connections follows the pooling module to combine low-level and high-level features and produce a more precise segmentation mask. Finally, a refinement module is used, which uses atrous convolution and batch normalization to further refine the segmentation mask and remove spurious predictions. Hence, DeepLabv3 is a large and complex architecture used for semantic segmentation tasks. The various key features allow the network to capture multi-scale context information, refine object boundaries, and remove spurious predictions. (7)

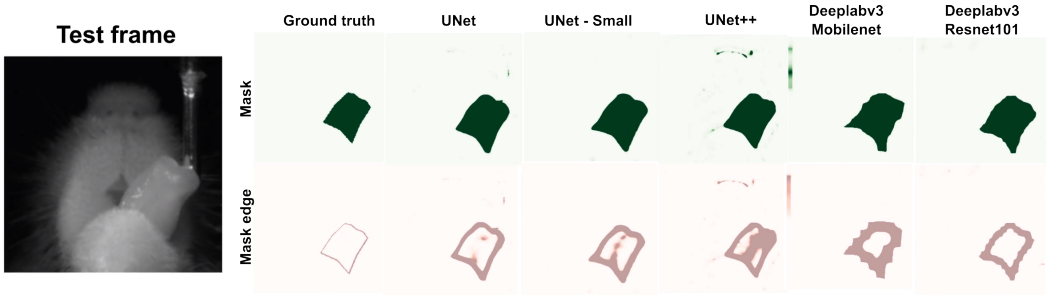

Figure 2: **Performance of different neural network architectures on test data.** Test frame recording of the bottom view of mouse face [left]. Ground truth masks labelled by human annotator and mask edge computed using the mask. Predicted masks and mask edges shown in the different columns for each of the different neural networks in Table 1.

3.3 Training

The different models were implemented and trained using PyTorch. Each model was trained for varying number of epochs, with a maximum of 150 epochs, using a batch size of 8. During training, an initial learning rate of 0.001 was used for all models. Adam optimizer, a stochastic gradient descent algorithm, was used to optimize the model parameters. To prevent the model from overfitting, a learning rate scheduler was employed. This scheduler lowered the learning rate on plateau by a factor of 0.5 following 5 consecutive epochs after the minimum loss was obtained. Gradually reducing the learning rate as the loss function reaches a plateau helps to prevent the model from getting stuck in a local minimum. Overall, these techniques helped to optimize the different models and improve performance.

A number of custom data augmentation methods were used to improve the performance of the models. Rotation by a small angle upto 5 degrees was used to better generalize to different orientations and angles of the same object. Another method used was horizontal flipping to learn features that are invariant to horizontal flips and also increase the size of the training set. Contrast adjustment was used to help the model to better differentiate between different regions of an image that have different contrast levels. Gaussian blur was used to better recognize features that are invariant to small variations in the image and reduce the impact of noise in the data. These techniques were thus used to create a larger and more diverse training dataset.

A number of model outputs/targets were explored in the project. This included masks (heatmaps) which contained a binary (0/1) label indicating the probability of the tongue being visible at each pixel. Subsequently, mask edges were computed using the masks to obtain the boundaries of the tongue. Additionally, a distance to boundary mask was computed to indicate how far each pixel was from the edge/boundary of the tongue. However, a number of preliminary experiments showed that only the masks and mask edges were helpful in training and achieved higher test accuracy. Hence, only mask and mask edges were used in the objective or loss function for all models tested in this project. The final loss function computed the binary cross-entropy loss for both the predicted mask and mask edges, such that the loss on mask edges was weighted by an arbitrary value. Finally, the same training protocol was used for all models tested. Loss and accuracy of the different models was monitored for training and validation data (Appendix: Figure 4).

3.4 Evaluation

A number of evaluation metrics commonly used in literature were used to assess the performance of the models. First, intersection over union (IoU) commonly used for semantic segmentation was used to measure the overlap between the prediction obtained from the model and ground truth. Higher values indicate better agreement between the predicted and ground truth regions. Second, dice coefficient was used which provides a measure of the segmentation accuracy that is sensitive to both false positives and false negatives. Dice coefficient is defined as twice the intersection of the predicted and ground truth masks divided by the sum of their areas. The two metrics were used for mask and mask edges. Some other metrics used were inference time and the number of parameters to measure

the efficiency and computational complexity of the models. Inference time, measured in milliseconds, recorded the amount of time required to process a single input image or batch of images through each model and generate the two outputs. All models were tested on the same GPU. Lower inference time indicates faster processing which is useful for real-time analysis. Lastly, the number of parameters were compared to measure the computational complexity and memory requirements of each model. Smaller models are preferable for implementation in machines with limited resources. Hence, the various metrics compared the accuracy, speed and complexity of the different models.

4 Results

The evaluation metrics used to compare the performance of the different models are summarized in Table 1. The table shows a quantitative comparison of the different segmentation models using six evaluation metrics: mask IoU (%), mask edges IoU (%), mask dice coefficient, mask edges dice coefficient, inference time (ms), and the number of parameters. The results show that UNet - Small achieved the best inference time and parameter efficiency while still maintaining comparable segmentation performance with the other models. UNet achieved the best performance in terms of mask IoU, mask edges IoU, mask dice coefficient, and mask edges dice coefficient, while DeepLabV3 (MobileNetV3) achieved the lowest performance in all metrics except for the number of parameters. The results demonstrate the trade-offs between segmentation performance and computational efficiency and highlight the importance of considering both metrics when selecting a segmentation model for a given application.

Table 1: Quantitative comparison of various segmentation models using Mask IoU (%), Mask Edges IoU (%), Mask Dice Coefficient, Mask Edges Dice Coefficient, Inference time (ms) and the number of parameters. UNet-Small achieved the best inference time and parameter efficiency while still maintaining comparable segmentation performance with the other models.

Model	Mask IoU (%)	Mask Edges IoU (%)	Mask Dice Coefficient	Mask Edges Dice Coefficient	Inference time (ms)	# Parameters
UNet	**90.58**	**38.93**	**0.9363**	**0.5492**	7.03	31M
UNet - Small	90.42	38.48	0.9335	0.5437	**4.43**	**1.6M**
UNet++	90.55	37.25	0.9355	0.5322	7.77	9.2M
DeepLabV3 (MobileNetV3)	89.81	3.84	0.9329	0.0722	8.19	11M
DeepLabV3 (ResNet50)	89.98	20.35	0.9326	0.3330	8.38	39.6M
DeepLabV3 (ResNet101)	90.13	19.65	0.9347	0.3234	13.04	58.6M

A qualitative comparison of the different models was performed by visualizing the predicted masks and mask edges. Figure 2 shows a sample test frame and the ground truth mask and mask edges. Each column shows the prediction from the different models for the test frame selected. UNet and UNet - Small predicted similar maps with little to no false positive. Whereas, UNet++ included some outlier regions indicating higher false positives where the nose of the mouse was labelled as the tongue. DeepLabv3 models predicted maps with fewest false positives, however the predicted maps were pixelated showing a lower resolution prediction which lost the finer features or boundaries of the tongue. More details and visualization of the test video labelled by the different models can be found in supplementary video (See appendix). The qualitative analysis thus provides some insight into understanding the strengths and weaknesses of each model.

5 Discussion

This project compared various segmentation models for real-time tracking of mouse tongue movement. The classic UNet model achieved the best performance while still being reasonably fast. The UNet - Small model was suitable for applications that require smaller and faster models as it was roughly twice as fast as the other models while achieving a high test accuracy of over 90%. The DeepLabv3 models did not meet similar performance in terms of accuracy and had higher model complexity as measured by the number of parameters. However, DeepLabv3 models could be better utilized for use-cases involving more training data or more complex segmentation problems/objective. The larger ResNet-based DeepLabv3 models had fewer false positives in mask edge predictions, which could be useful in certain contexts where accurate spatial localization of tongue is important.

Although UNet and DeepLabv3 demonstrated promising results for tongue segmentation, they demonstrated some limitations depending on the application. UNet-style models had higher false positives in test videos. Additionally, UNet-style models performance could be improved by using a more balanced dataset since the network tends to prioritize learning the dominant class at the expense of the minority class. DeepLabv3 had highest computational and memory requirements due to its use of deep residual networks, which have a large number of parameters. This can make it difficult to train and deploy on machines with limited resources. Additionally, the pixel-wise prediction nature of DeepLabv3 led to pixelated and less smooth predicted masks resulting in less accurate tongue boundaries. Thus, the important features of the different networks led to different qualitative performance. These features could be combined in a smaller model to design a task-specific segmentation model that can achieve a high test accuracy and still be feasible for real-time applications.

The segmentations obtained from the model can be used to transform the mask information into tongue directions. The mask information can be used to estimate the volume of the tongue and identify time points when the tongue is retracting or protruding. Further training data from different camera views of the mouse face can be used to derive a 3D reconstruction of the tongue which can be used to estimate the position of the tongue tip and 3D volume. The resulting information can be used in combination with neural data to model and understand how the sensorimotor regions of the brain control different movement parameters of the tongue.

Overall, this project demonstrates great promise for real-time segmentation and tracking of mouse tongue movement. Future research could explore other experimental setups with more data variation, such as different camera perspectives and alternative data collection environments. Transfer learning, such as pre-training on imagenet, could also be applied to improve the performance of the UNet models, whereas the DeepLabv3 backbones were already pre-trained on imagenet. These future directions can help to further improve the performance and practical application of the proposed method.

6 Conclusions

In conclusion, this work demonstrated promising results for real-time segmentation and tracking of the mouse tongue movements. The project compared various neural network architectures and UNet-style neural networks worked best in terms of test accuracy measured using intersection over union and dice coefficient. This work thus proposes a small and fast neural network architecture that can be employed in closed-loop behavioral experiments while achieving high accuracy for the tongue segmentation task.

Code

The project codebase can be found on Github: `https://github.com/Atika-Syeda/TongueSegmentation`

References

[1] Chartier J, Anumanchipalli GK, Johnson K, Chang EF. Encoding of articulatory kinematic trajectories in human speech sensorimotor cortex. Neuron. 2018;98(5):1042-54.

[2] Kier WM, Smith KK. Tongues, tentacles and trunks: the biomechanics of movement in muscular-hydrostats. Zoological journal of the Linnean Society. 1985;83(4):307-24.

[3] Remesso GC, Fukujima MM, Chiappetta ALdML, Oda AL, Aguiar AS, Oliveira AdSB, et al. Swallowing disorders after ischemic stroke. Arquivos de Neuro-psiquiatria. 2011;69:785-9.

[4] Argolo N, Sampaio M, Pinho P, Melo A, Nóbrega AC. Swallowing disorders in Parkinson's disease: impact of lingual pumping. International Journal of Language & Communication Disorders. 2015;50(5):659-64.

[5] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer; 2015. p. 234-41.

[6] Zhou Z, Rahman Siddiquee MM, Tajbakhsh N, Liang J. Unet++: A nested u-net architecture for medical image segmentation. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4. Springer; 2018. p. 3-11.

[7] Chen LC, Papandreou G, Schroff F, Adam H. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:170605587. 2017.

[8] Xu D, Dong M, Chen Y, Delgado AM, Hughes NC, Zhang L, et al. Cortical processing of flexible and context-dependent sensorimotor sequences. Nature. 2022;603(7901):464-9.

[9] Bollu T, Ito BS, Whitehead SC, Kardon B, Redd J, Liu MH, et al. Cortex-dependent corrections as the tongue reaches for and misses targets. Nature. 2021;594(7861):82-7.

A Appendix

Table 2: Dataset used for the study, which was collected from two different labs across multiple sessions and different animales (mice).

	#Training images	#Validation images	#Test images
Dataset	13,687	1,521	1,564

Table 3: Proportion of training and test images with and without tongue visible in frame. The training set used for the study was balanced, and almost half of the frames contained visible portions of the tongue.

	Proportion w/ tongue	Proportion w/o tongue
Train	0.40	0.60
Test	0.37	0.63

Table 4: Model architecture details for UNet - Small

Layers	Channels_in	Channels_out	W x H	Activation	BatchNorm
Input	1	16	128 x 128	-	-
Conv_1	16	16	64 x 64	-	-
Conv_2	16	32	64 x 64	ReLU	True
Conv_3	32	32	32 x 32	ReLU	True
Conv_4	32	64	32 x 32	ReLU	True
Conv_5	64	64	16 x 16	ReLU	True
Conv_6	64	128	16 x 16	ReLU	True
Conv_7	128	128	8 x 8	ReLU	True
Conv_8	128	200	8 x 8	ReLU	True
Conv_9	200	200	8 x 8	ReLU	True
Up_conv	200	200	16 x 16	-	-
Conv_10	328	128	16 x 16	ReLU	True
Conv_11	128	128	16 x 16	ReLU	True
Up_conv	128	128	32 x 32	-	-
Conv_12	192	64	32 x 32	ReLU	True
Conv_13	64	64	32 x 32	ReLU	True
Up_conv	64	64	64 x 64	-	-
Conv_14	96	32	64 x 64	ReLU	True
Conv_15	32	32	64 x 64	ReLU	True
Up_conv	32	32	128 x 128	-	-
Conv_16	48	16	128 x 128	ReLU	True
Conv_17	16	16	128 x 128	ReLU	True

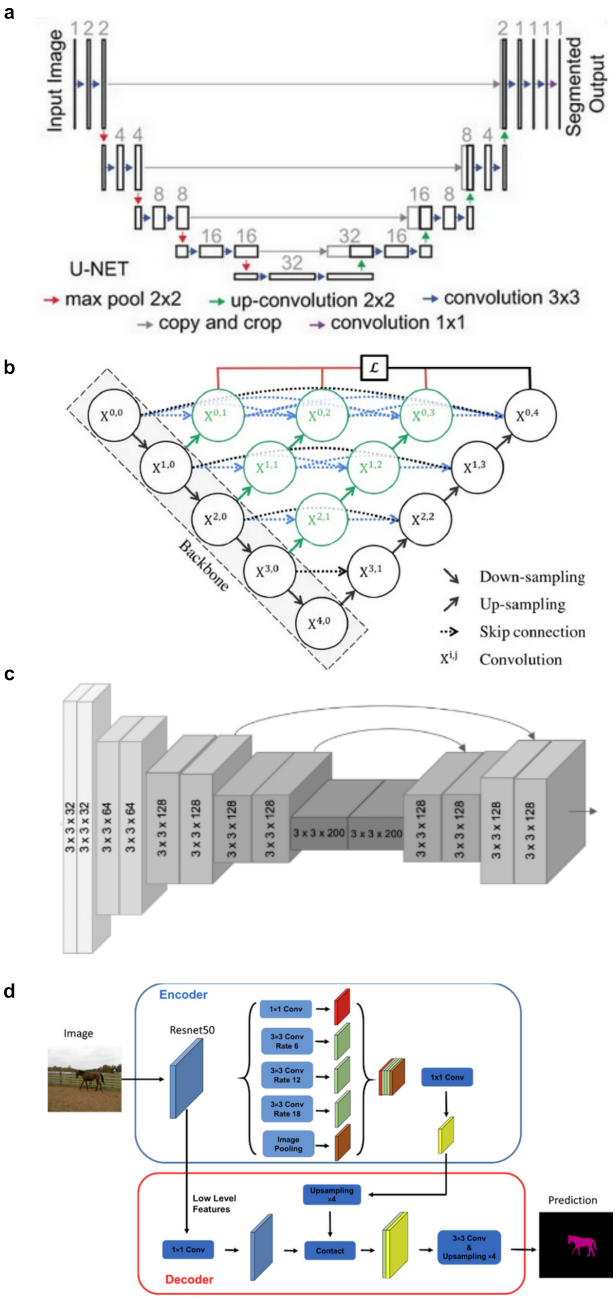

Figure 3: **Neural network architectures used for the tongue segmentation task. a**. UNet model **b**. UNet++ model **c**. UNet - Small, **d**. DeepLabv3 backbone.

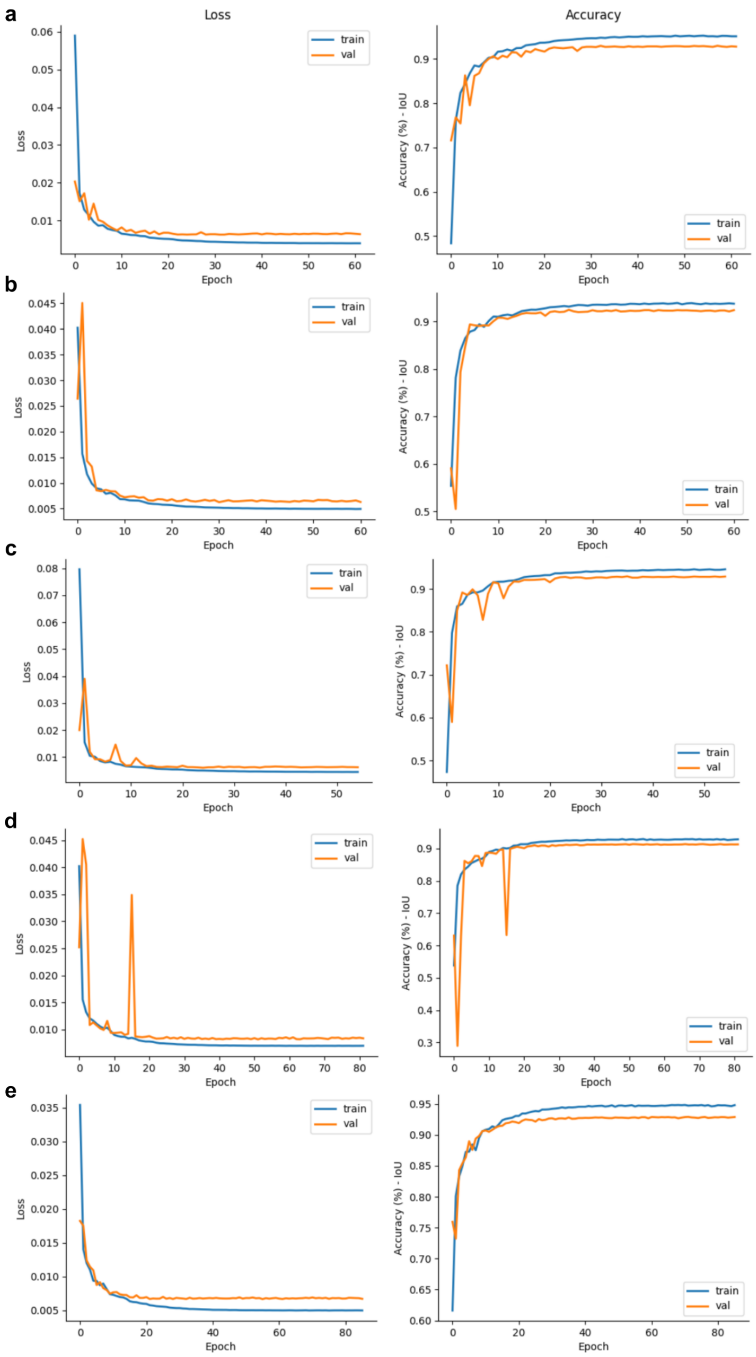

Figure 4: **Training loss and accuracy curves for different neural networks. a**. UNet loss curve [left] and accuracy curve [right] for training and validation set. Model training stopped using early stopping after no improvement in validation loss for 30 consecutive epochs. **b**. Same as in **a** for UNet - Small, **c**. Same as in **a** for UNet++, **d**. Same as in **a** for DeepLabv3 - MobileNetV3, **e**. Same as in **a** for DeepLabv3 - ResNet50

Supplementary videos: Videos visualizing predicted mask and edges for test data generated using the different models can be found at `https://livejohnshopkins-my.sharepoint.com/:f:/g/personal/asyeda1_jh_edu/ElZlFodqw8lDrOSX9QNfstEBVaF2efzL60gF82QMoMUmYg`.