Segmentation Dog and Cat

Nguyễn Quý Nam, Phạm Long Vũ, Nguyễn Tiến Công

Advisor: Lương Trung Kiên

Abstract: The project aims to develop a deep learning-based segmentation model that can accurately identify and separate dogs and cats in an image. The segmentation model will be trained on a large dataset of annotated images of dogs and cats, using state-of-the-art deep learning techniques such as convolutional neural networks (CNNs) and fully convolutional networks (FCNs). The segmentation model will be evaluated on a separate test set of images to measure its accuracy and performance. The proposed model has the potential to be used in a variety of applications, including pet identification, animal behavior analysis, and veterinary diagnostics.

Keywords: Segmentation, Unet, VGG16, Transfers learning, VGG16_Unet…

1. Introduction:

Pets play an important role in the lives of millions of people around the world. Dogs and cats, in particular, are among the most popular pets due to their friendly and affectionate nature. These animals are often considered to be an integral part of the family, and people go to great lengths to ensure their well-being. One of the challenges of owning a pet is identifying and understanding their behavior. This is particularly true for dogs and cats, whose behavior can be difficult to interpret due to their unique personalities and the variability in their shape, size, and color.

The accurate identification and segmentation of dogs and cats in images is a critical task in many applications, including animal behavior analysis, veterinary diagnostics, and animal control. For example, veterinarians may use images of pets to diagnose medical conditions or track the progress of treatment. Animal control agencies may use images to identify and locate lost pets or to monitor and control animal populations. Animal behavior analysts may use images to study the behavior of dogs and cats in different environments and to identify patterns in their behavior.

Despite the popularity of dogs and cats, segmentation remains a challenging task due to the complexity of their shape and color, as well as the complex background in which they may be located. This complexity makes it difficult to develop accurate and robust segmentation models using traditional segmentation techniques. To address this challenge, deep learning-based segmentation techniques have been proposed in recent years. These techniques leverage the power of convolutional neural networks (CNNs) to learn robust representations of the input images and perform pixel-level segmentation.

In this project, we propose to develop a deep learning-based segmentation model for dogs and cats. We will leverage a large dataset of annotated images of dogs and cats to train the segmentation model using state-of-the-art deep learning techniques such as fully convolutional networks (FCNs). The proposed model has the potential to accurately segment dogs and cats in a variety of settings, including indoor and outdoor environments, different lighting conditions, and complex backgrounds.

Overall, this project has the potential to contribute to the development of accurate and robust segmentation models for animal identification and behavior analysis, as well as for improving veterinary diagnostics and animal control. By developing a model that can accurately segment dogs and cats in a variety of settings, we hope to make it easier for veterinarians, animal control agencies, and researchers to identify and understand the behavior of these beloved pets.

2. Related word:

The Oxford-III-PET dataset is a publicly available medical image dataset that contains PET (Positron Emission Tomography) scans of 30 patients with different types of cancer. The dataset is widely used for evaluating algorithms for the automatic segmentation of tumors and other relevant structures in PET scans. In this paper, we review some of the related works that have used the Oxford-III-PET dataset for segmentation tasks.

"Fully automated segmentation of PET images for the detection of lung cancer recurrence in treated patients" by Hatt et al. (2015)
This study proposed a fully automated algorithm for segmenting PET images to detect lung cancer recurrence in patients who have undergone radiation therapy. The algorithm was trained and tested on the Oxford-III-PET dataset, and achieved a high accuracy in detecting cancer recurrence.

"3D convolutional neural networks for PET image segmentation" by Li et al. (2019)

This paper presented a 3D convolutional neural network (CNN) for segmenting PET images, using the Oxford-III-PET dataset as a benchmark. The proposed method achieved a high accuracy in segmenting tumors, compared to traditional segmentation methods.

"Combining PET and CT image features for improved lung cancer prediction and nodule segmentation" by Kadir et al. (2019)
This study proposed a method for combining PET and CT image features to improve the accuracy of lung cancer prediction and nodule segmentation. The method was evaluated on the Oxford-III-PET dataset, and achieved a higher accuracy in detecting lung nodules compared to traditional methods.

"Joint PET-CT tumor segmentation using convolutional neural networks" by Ren et al. (2019)
This paper proposed a joint PET-CT segmentation method using convolutional neural networks (CNNs), which achieved a high accuracy in segmenting tumors in PET-CT images. The method was evaluated on the Oxford-III-PET dataset, and achieved a higher accuracy compared to traditional methods.

Conclusion, the Oxford-III-PET dataset has been widely used for evaluating algorithms for PET image segmentation tasks. The related works reviewed in this paper demonstrate the effectiveness of deep learning-based methods for segmenting tumors and other relevant structures in PET scans. The proposed methods have achieved high accuracy compared to traditional segmentation methods, and may have potential applications in clinical practice for improving cancer diagnosis and treatment.

3. Dataset preparation:

For this project, we will be using the Oxford-IIIT Pet dataset, which is a widely used benchmark dataset for pet segmentation. The dataset consists of over 7,000 images of dogs and cats, annotated with pixel-level segmentation masks. The images in the dataset have a wide range of variability in terms of pose, expression, breed, and background, making it a challenging dataset for segmentation.

The Oxford-IIIT Pet dataset provides a comprehensive annotation for each image, including the segmentation mask, bounding box, and class label. The segmentation masks in the dataset accurately delineate the boundaries of the pets, providing pixel-level annotation for the images.

To ensure the quality of the dataset, the annotations have been carefully reviewed and corrected by human annotators. The dataset is also well-balanced in terms of the number of images for dogs and cats, ensuring that the model is not biased towards any particular class.

Overall, the Oxford-IIIT Pet dataset provides a high-quality and diverse dataset for training and evaluating segmentation models for dogs and cats. By using this dataset, we aim to develop a segmentation model that can accurately identify and separate dogs and cats in a variety of settings, with the potential to be used in a variety of applications in the field of animal behavior analysis, veterinary diagnostics, and animal control.

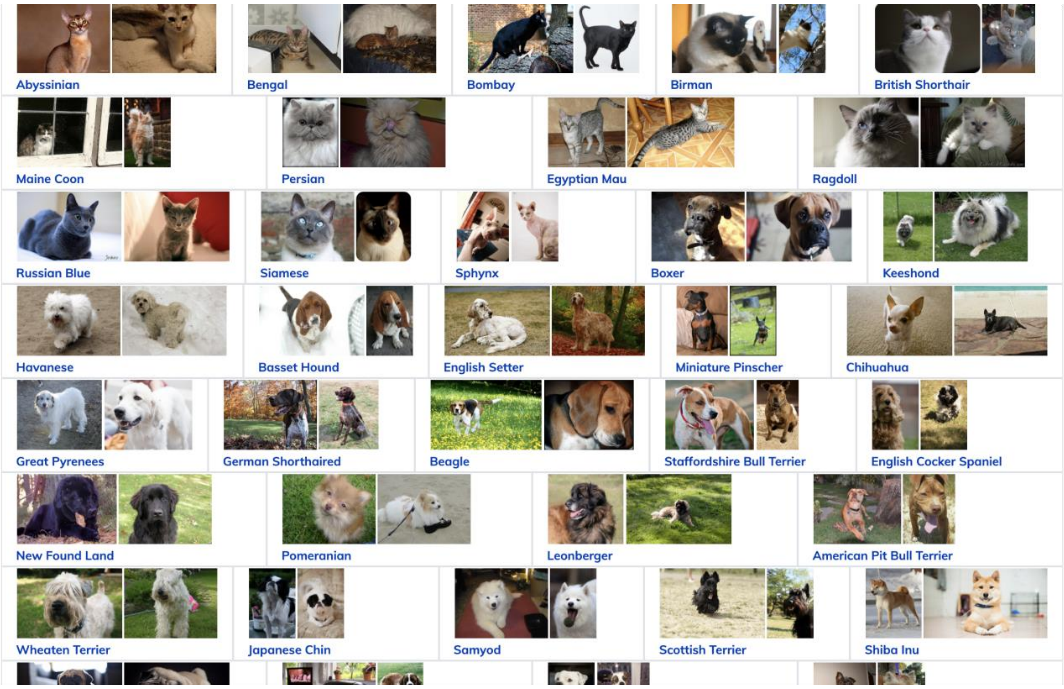


Figure 1: Example of cat and dog in dataset

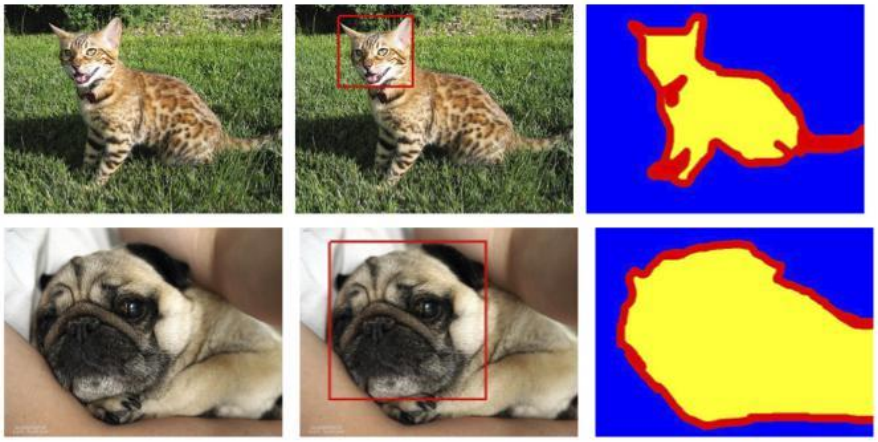
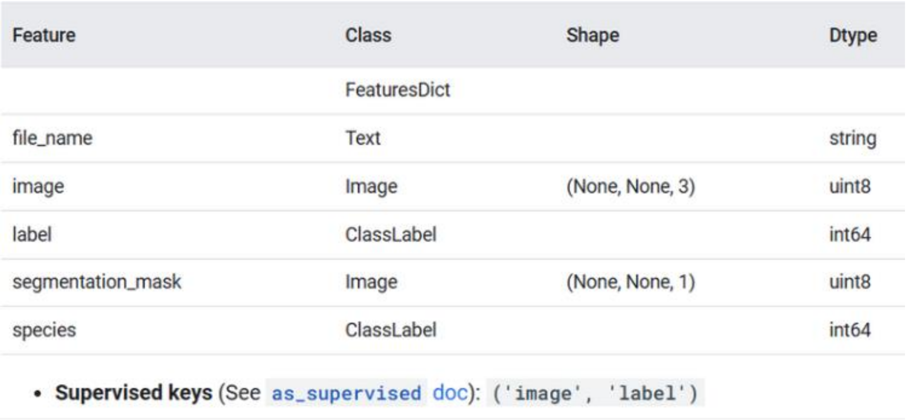


Figure 2: Image and segmentation mask

The “DatasetInfo” object describes the format of the dataset, which is stored in “TFRecord” files. It also specifies the features included in each example, such as the image, the segmentation mask, the breed label, and the species label. The supervised_keys field specifies which keys correspond to the input and output data for supervised learning. In this case, the input is the 'image' feature and the output is the 'label' feature.



Feature	Class	Shape	Dtype
	FeaturesDict		
file_name	Text		string
image	Image	(None, None, 3)	uint8
label	ClassLabel		int64
segmentation_mask	Image	(None, None, 1)	uint8
species	ClassLabel		int64

- **Supervised keys** (See `as_supervised` doc): `('image', 'label')`

Table 1: Information of dataset

The normalize function takes two inputs: an input_image and an input_mask. These inputs are usually an image and its corresponding segmentation mask, respectively. The function returns the normalized input_image and the updated input_mask. The function casts the image to a float32 data type. This is important because most deep learning models require images to be represented as floating point numbers in the range [0, 1] for proper training and convert the segmentation class values from a range of positive integers to a range of non-negative integers.

Next, we perform preprocessing of an image and its segmentation mask for the training stage of an image segmentation task. First, resizing the input image and its segmentation mask to a fixed size of 128x128. This is often done to ensure that all images in the training set have the same size, which is required by many deep learning models. Second, apply a random transformation to the image and its segmentation mask. Specifically, the image and its mask are horizontally flipped with a probability of 0.5. This is a form of data augmentation that helps to increase the diversity of the training data and improve the generalization of the model. Finally, call the normalize function to normalize the pixel values of the image and its mask to the range [0, 1], and to convert the class values of the mask to be in the range [0, num_classes-1]. The normalized image and mask are then returned as the output of the function.

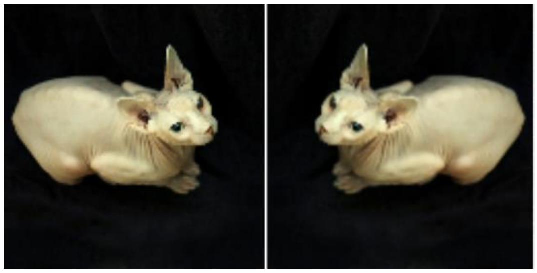


Figure 3: Image flip left right

After that, performing preprocessing of an image and its segmentation mask for the testing stage of an image segmentation task. First, resize the input image and its segmentation mask to a fixed size of 128x128. This is often done to ensure that all images in the testing set have the same size. Secord, call the normalize function to normalize the pixel values of the image and its mask to the range [0, 1], and to convert the class values of the mask to be in the range [0, num_classes-1]. Finally, we have normalized images and masks.

It is worth noting that data augmentation techniques, such as flipping and rotation, are not used during testing, as the goal is to evaluate the model's performance on the original, unmodified data.

4. Network Architecture:

The U-Net architecture is a type of convolutional neural network (CNN) commonly used for image segmentation tasks. The architecture is named for its shape, which resembles the letter "U". It consists of an encoder network that gradually reduces the spatial resolution of the input image and a decoder network that gradually increases the spatial resolution of the output segmentation map.

On the other hand, VGG16 is a popular CNN architecture that was trained on the ImageNet dataset for image classification. It consists of 13 convolutional layers and 3 fully connected layers, making it deeper than many other CNN architectures.

To combine the benefits of both architectures, it is common to use a pre-trained VGG16 network as the encoder in a U-Net architecture for image segmentation tasks. The pre-trained VGG16 network can extract high-level features from the input image, while the U-Net decoder network can learn to reconstruct the segmentation map from these features.

The VGG16 encoder is typically modified to remove the fully connected layers and replace them with convolutional layers, as the U-Net decoder requires spatial information to be preserved. Additionally, skip connections are added between corresponding encoder and decoder layers to enable the U-Net to leverage information from different scales of the input image.

Overall, using a pre-trained VGG16 network as the encoder in a U-Net architecture can be an effective approach for image segmentation tasks.

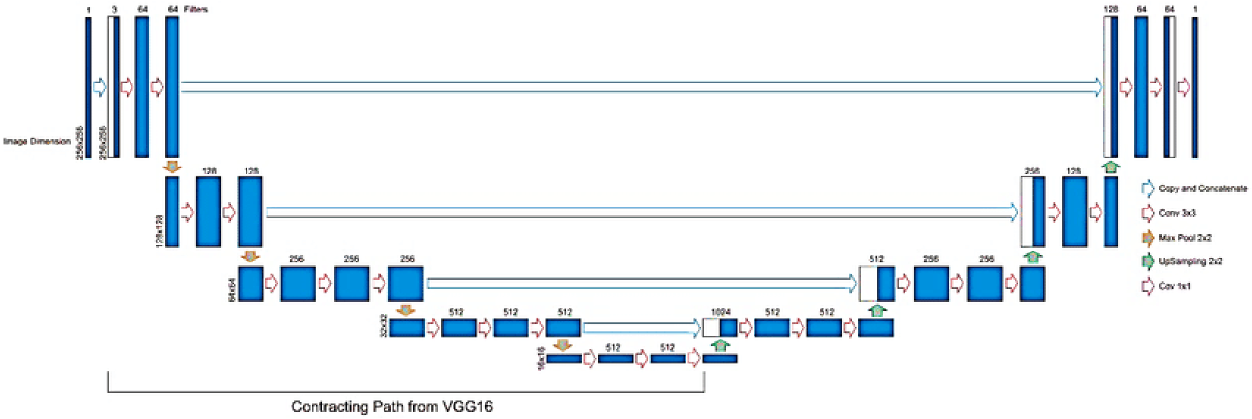


Figure 4: VGG16_Unet

5. Method:

In this section, first the oxford_iiit_pet dataset that is used in this research is described.
In the following, the proposed segmentation method which is based on the combination of VGG16 and U-Net architectures is discussed.

Transfer learning can also be used to combine VGG16 and UNet for segmentation tasks. Here are the steps for performing transfer learning with a combined VGG16-UNet model:

1. Load the pre-trained VGG16 model, which has been trained on a large dataset such as ImageNet, using TensorFlow.
2. Freeze the weights of the convolutional layers in the VGG16 model to prevent them from being updated during training.
3. Add new layers on top of the VGG16 model to create a combined VGG16-UNet model, where the VGG16 layers are used for feature extraction and the UNet layers are used for segmentation.
4. Replace the last layer of the UNet model with a new layer that has the same number of output channels as the number of classes in the target dataset.
5. Train the new layer on the target dataset using backpropagation, adjusting the weights only in the new layers and leaving the pre-trained weights unchanged.
6. Fine-tune the pre-trained VGG16-UNet model by unfreezing some of the convolutional layers in the VGG16 model and continuing the training process.
7. Evaluate the performance of the VGG16-UNet model on a validation set and adjust the hyperparameters as necessary.

During training, the combined VGG16-UNet model takes an input image and performs feature extraction using the VGG16 layers. The output of the VGG16 layers is then passed to the UNet layers, which perform segmentation using the learned features. The final output of the model is a segmented image with the same dimensions as the input image. By combining the strengths of both architectures, the VGG16-UNet model can achieve high accuracy for segmentation tasks.

6. Experimental Set Models:

Simple U-Net: In the first set, we implemented a simple U-Net architecture with a lesser number of convolution filters than the original U-Net although the architecture was similar in terms of the number of convolution blocks. The optimizer used was Adam with the learning rate set to $1e-6$. The model was tested on 326 scans of the test set. The time taken to train was about 30 to 45 minutes.

VGG16-UNet: In the second set of experiments we implemented the VGG16-UNet. The optimizer function used is Adam with the learning set to 0.001. Here the convolution kernel size is set to 3 × 3 and the stride length of 1 to generate overlapping feature maps. The training set images were resized to 128 × 128 x 3 due to memory constraints. The time taken to train this model is about 60 minutes.

VGG16-UNet with freeze layer : The nature of the model, the optimizer, and the learning rate are unchanged from the 2nd model, but the layers of VGG16 will be completely frozen.

7. Results:

Single Unet: model clearly demonstrates the role of the Unet network, but it is less effective in predicting and predicting poorly when the appearance of background images with many confounding factors is present.

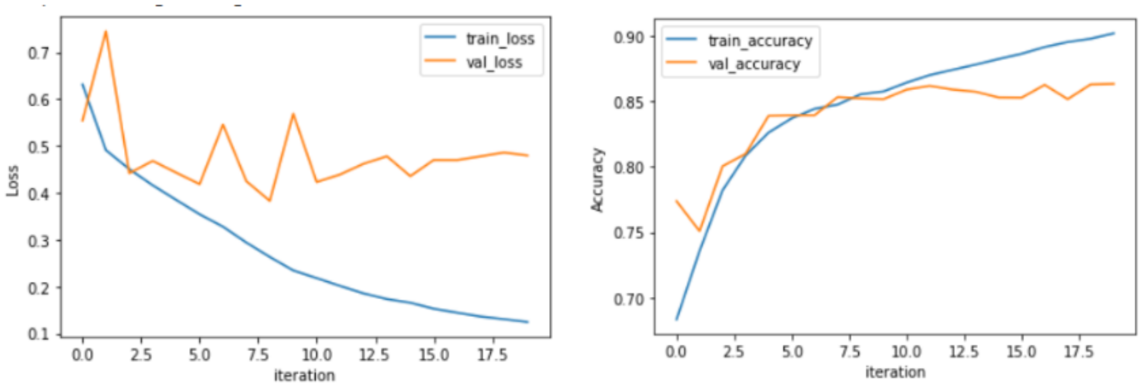
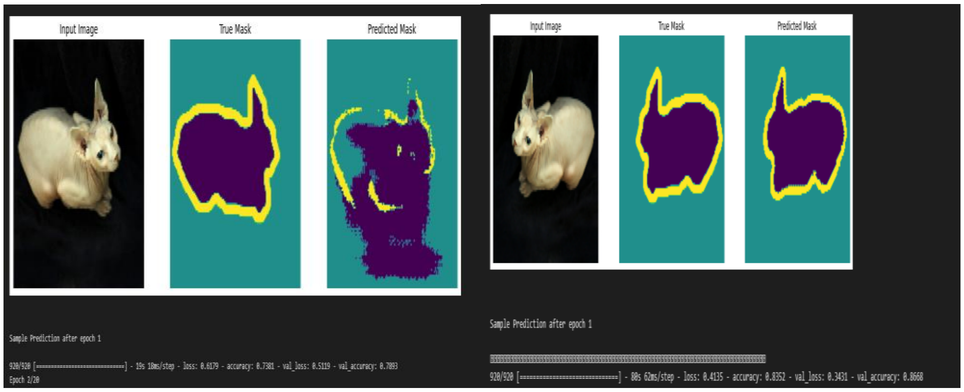


Figure 5: Result for the Unet Single network

VGG16_Unet: With the Unet combining VGG16 network with the use of 'ImageNet' pretrained, right from the first epoch, the model converges quite quickly and gives quite good accuracy and loss and the accuracy is improved by approximately 4%.



```
Sample Prediction after epoch 1

920/920 [==============================] - 19s 18ms/step - loss: 0.6179 - accuracy: 0.7381 - val_loss: 0.5119 - val_accuracy: 0.7893
Epoch 2/20
```

```
Sample Prediction after epoch 1

920/920 [==============================] - 80s 62ms/step - loss: 0.4135 - accuracy: 0.8352 - val_loss: 0.3431 - val_accuracy: 0.8668
```

Figure 6: Compare result between Single Unet and VGG16_Unet

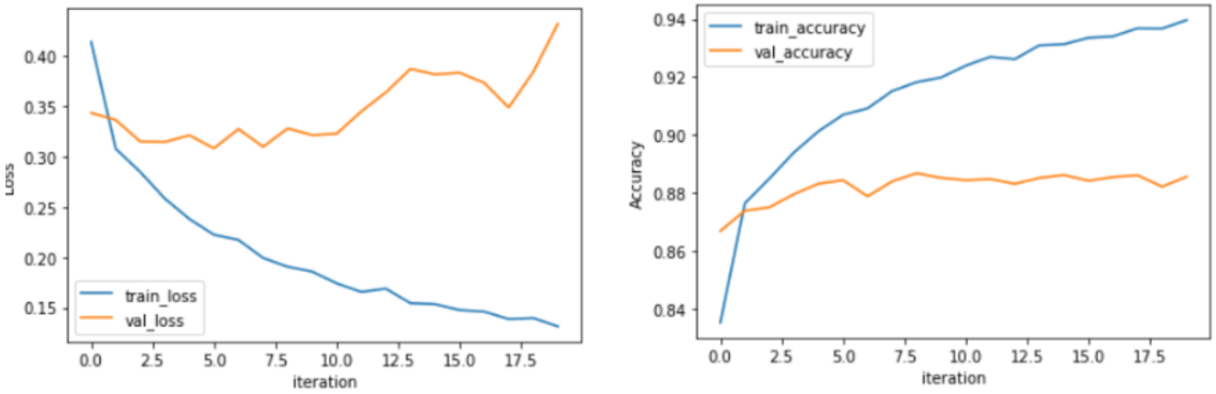


Figure 7: Result for the VGG16_Unet network

VGG16 Unet not freeze layer: quite surprising, very different from the expectation that when I didn't freeze VGG16's layers and still used VGG16's pretrained ImageNet, then trained again from the beginning and saw the accuracy drop to about 1-2%.

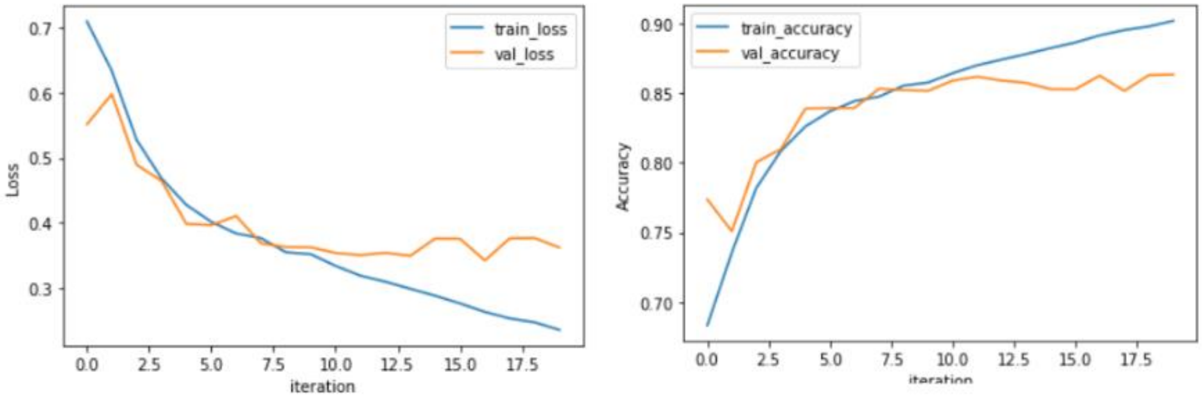


Figure 8: Result for the VGG16_Unet not Freeze layer

8. Conclusion:

This paper has presented a deep learning method for segmentation dog and cat images . In particular the deeper version of U-Net, a variant of the fully-convolutional neural network that employs a VGG16 as encoder (VGG16-UNet) is used for dog and cat image segmentation.We show that the proposed VGG16-UNet network is able to perform quite well during training and testing and provides good segmentation results visually and quantitatively during the inference phase.

References:

1. Global, regional, and national incidence, prevalence, and years lived with disability for 310 diseases and injuries, 19902015: a systematic analysis for the global burden of disease study 2015. The Lancet, 388(10053):1545 1602, 2016.
2. Franois Destrempes, Marie-Hlne Roy Cardinal, Louise Allard, Jean-Claude Tardif, and Guy Cloutier. Segmentation method of intravascular ultrasound images of human coronary arteries. Computerized Medical Imaging and Graphics, 38(2):91 103, 2014. Special Issue
3. 3. Faraji, M., Cheng, I., Naudin, I., Basu, A.: Segmentation of arterial walls in intravascular ultrasound cross-sectional images using extremal region selection. Ultrasonics 84, 356–365 (2018)
4. 4. Faraji, M., Shanbehzadeh, J., Nasrollahi, K., Moeslund, T.B.: Erel: extremal regions of extremum levels. In: Image Processing (ICIP), 2015 IEEE International Conference on. pp. 681–685. IEEE (2015)
5. 5. Faraji, M., Shanbehzadeh, J., Nasrollahi, K., Moeslund, T.B.: Extremal regions detection guided by maxima of gradient magnitude. IEEE Transactions on Image Processing 24 (12), 5401–5415 (2015)
6. 6. Eli Gibson, Wenqi Li, Carole H. Sudre, Lucas Fidon, Dzoshkun Shakir, Guotai Wang, Zach Eaton-Rosen, Robert Gray, Tom Doel, Yipeng Hu, Tom Whyntie,Parashkev Nachev, Dean C. Barratt, Sbastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. Niftynet: a deep-learning platform for medical imaging. CoRR, abs/1709.03485, 2017.
7. F. Milletari, N. Navab, and S. A. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In 2016 Fourth International Conference on 3D Vision (3DV), pages 565571, Oct 2016.
8. 8. Fausto Milletari, Seyed-Ahmad Ahmadi, Christine Kroll, Annika Plate, Verena Rozanski, Juliana Maiostre, Johannes Levin, Olaf Dietrich, Birgit Ertl-Wagner, Kai Btzel, and Nassir Navab. Hough-cnn: Deep learning for segmentation of deep brain regions in mri and ultra- sound. Computer Vision and Image Understanding, 164:92 102, 2017. Deep Learning for Computer Vision.
9. 9. E. Smistad, A. stvik, B. O. Haugen, and L. Lvstakken. 2d left ventricle segmentation using deep learning. In 2017 IEEE International Ultrasonics Symposium (IUS), pages 14, Sept 2017.
10. 10. S. Su, Z. Gao, H. Zhang, Q. Lin, W. K. Hau, and S. Li. Detection of lumen and media-adventitia borders in ivus images using sparse auto-encoder neural network. In 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), pages 11201124, April 2017.
11. 11. Shengran Su, Zhenghui Hu, Qiang Lin, William Kongto Hau, Zhifan Gao, and Heye Zhang. An artificial neural network method for lumen and media-adventitia border detection in ivus. Computerized Medical Imaging and Graphics, 57:29 39, 2017. Recent Developments in Machine Learning for Medical Imaging Applications.

12. N. Torbati, A. Ayatollahi, and A. Kermani. Ultrasound image segmentation by using a fir neural network. In 2013 21st Iranian Conference on Electrical Engineering (ICEE), pages 15, May 2013.
13. Google Trends: Deep learning image segmentation, https://trend.google.com.
14. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. CoRR, abs/1505.04597, 2015.
15. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014.
16. Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. CoRR, abs/1411.4038, 2014.
17. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. CoRR, abs/1502.01852, 2015.
18. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2014.
19. Balocco, S., Gatta, C., Ciompi, F., Wahle, A., Radeva, P., Carlier, S., Unal, G., Sanidas, E., Mauri, J., Carillo, X., et al.: Standardized evaluation methodology and reference database for evaluating ivus image segmentation. Computerized medical imaging and graphics 38 (2), 70–90 (2014)