2021

Image Segmentation

SANJAY JARAS
https://sanjayjaras.github.io/

BELLEVUE UNIVERSITY

Abstract

As a human, we are very good at analyzing the scene/image we see. We can easily recognize the different things and objects we see. Our brain is very good at differentiating one object from another for example we can easily differentiate persons available in one photograph. But it was very difficult for machines to do these things some years back. With the advancements in computer vision, machines have become good at analyzing images. Now we can build machine learning models that can detect different objects in images. The same technology is being used for self-driving cars, different tumor detections in the human body. It can also help in differentiating malignant and non-malignant tumors. Image segmentation is considered the most important medical imaging process as it isolates the region of interest (ROI). Image segmentation is the process of partitioning or dividing the digital image into different segments depending on the objects or types of objects present in the image. Image segmentation simplifies the image by removing the unwanted data or data that is not of interest from the image. This simplification then helps in analyzing the region of interest. With this project, I will be implementing a machine learning model by using neural nets to segment different objects present in images.

Intro/background of the problem

If we want to cross a road then we look on both sides of the road and analyze what vehicles coming towards us. We quickly decided what kind of vehicles are coming and we decide to stop or cross the road accordingly. Similarly, if we are driving a car, we analyze every scene and analyze what kind of objects are in front of us. We can do these things very easily, as our brain learned to differentiate objects by their shapes, colors, distance from one another, and other object characteristics. If we want to give the ability to differentiate objects available in a digital image, we can use image segmentation models to train a machine. Image segmentation is the process of dividing a digital image into multiple segments. A segment is a group of pixels also termed an object. Image segmentation image helps in splitting area/region of interest from the image. For example, it can help in isolating the tumor region from a digital image of a brain. It can isolate different objects like persons, animals, vehicles, etc. from a digital image. Then these identified regions can be used for different applications like brain tumor classification and self-driving cars. Once we know the accurate type of tumor classification, it can help in surgery planning and treatment. Similarly, in self-driving cars application(traffic signal/light detection, pedestrian detection, etc.) this helps in making different decisions. There are many applications for image segmentation machine vision, face detection, face recognition, fingerprint recognition, Iris recognition, content-based image retrieval, Virtual surgery simulation, Intra-surgery navigation, Locating objects in satellite images, etc. With this project, I trained a machine learning model with the image dataset made available for competition in 2012 with 20 different objects. I used the U-Net model for training. U-Net is a convolutional neural network that was developed for biomedical image segmentation at the Computer Science Department of the University of Freiburg. With this model, we will be predicting the object masks(pixels owned by an object) of different objects and borders.

Methods

When we want to do image classification, each image should have one or more labels assigned to it and the prediction model needs to predict a label for each image. However, when we want to know the shape of an object present in an image, we need a label for each pixel owned by that object. The prediction model needs to predict a label for each pixel, this process is called segmentation. For this project, I have used the dataset published as part of Visual Object Classes Challenge 2012 (VOC2012). The dataset is downloaded from the VOC challenge website. The dataset contains 1464 images under the training dataset and 1449 images under the validation dataset. The images are of different shapes. There are two types of images for the segmentation dataset

1. Input images as a jpeg image
2. Labels(masks) as a png image

The jpeg images are with 3 channels and png images are with one channel. The png image contains the pixel color value as the label value of the object. The following are label values for objects 1=aeroplane, 2=bicycle, 3=bird, 4=boat, 5=bottle, 6=bus, 7=car , 8=cat, 9=chair, 10=cow, 11=diningtable, 12=dog, 13=horse, 14=motorbike, 15=person, 16=potted plant, 17=sheep, 18=sofa, 19=train, 20=tv/monitor. Label value 0 is used for background and 255 is used for void or unlabelled pixels. The label images define the outlined mask of the segments or the objects present in each image by the different labels. The dataset is then divided into the training set(2164 images), the test set (300 images), and the validation set(449) images.

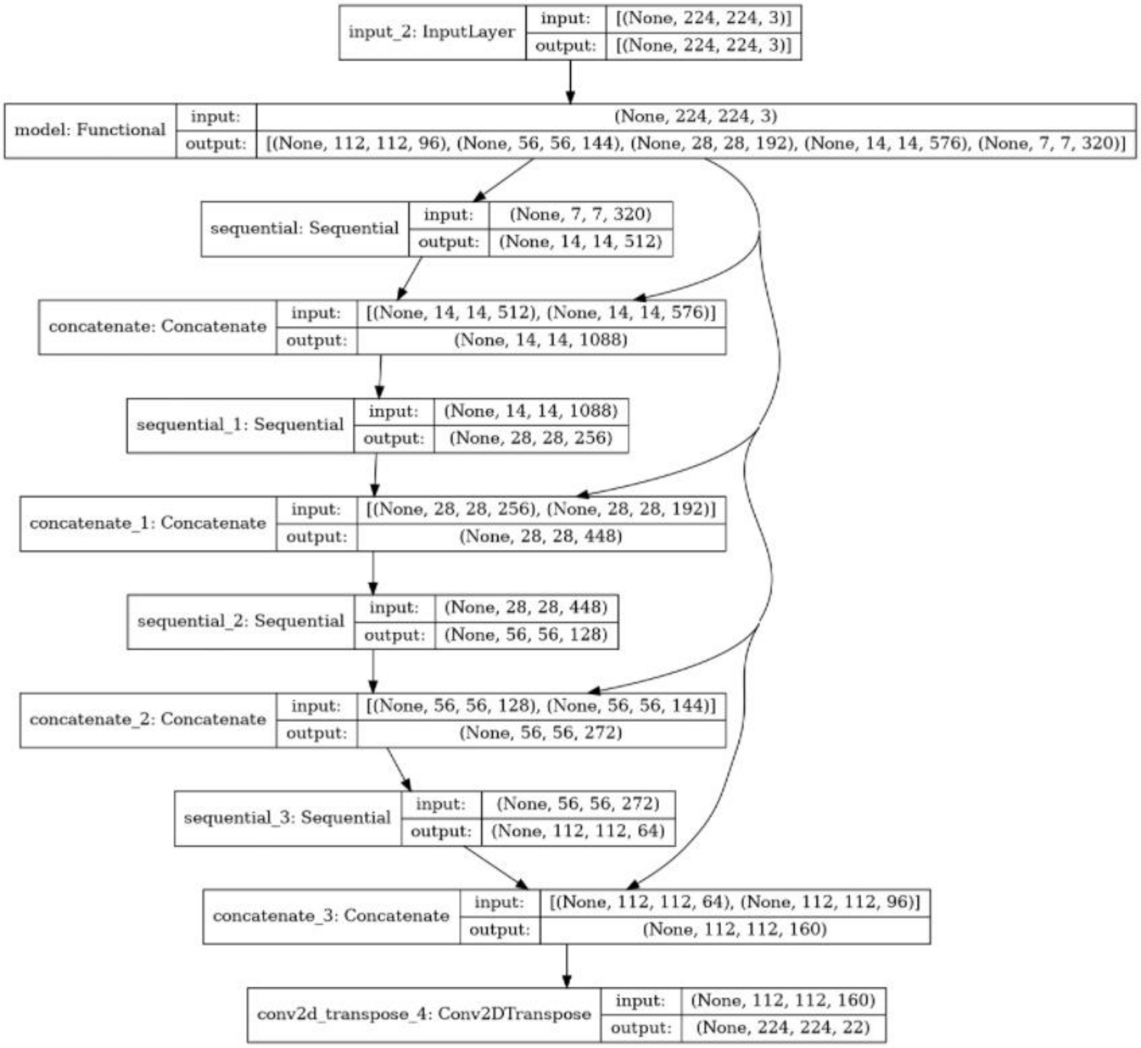
While loading images, I normalized to have pixel values from 0-1 for input images. For masked images, I replaced the unknown label value to 21 instead of 255 to keep labels in sequence from 0-21. After loading images, images are resized to 224x224 shape to keep all images of the same shape. As we have only 2164 images in the training dataset, I have decided to augment the images in the training dataset and add them into the training dataset to avoid overfitting. While augmenting, we need to use the same seed for the input image and labeled image to keep the augmentation the same for the image and the mask image. I have used vertical and horizontal flips for augmentation for each training image.

For training and validations, we have created a batched dataset with a batch size of 32. Each batch will pick random 32 images for training and validations.

For training and prediction, I have selected the U-Net model. The U-Net was developed for biomedical image segmentation by the University of Freiburg Germany. For U-Net's contraction path, down-sampling, or encoder, I am using the MobileNetV2 model. From MobileNetV2 I am using four functional layers. For expansion path, up-sampling or decoder, I am using pix2pix upsampler. MobileNetV2 is used to extract features from an image by reducing the size but keeping information as it is. Pix2pix up-sampler will create a high-resolution image of a mask from a low-resolution image.

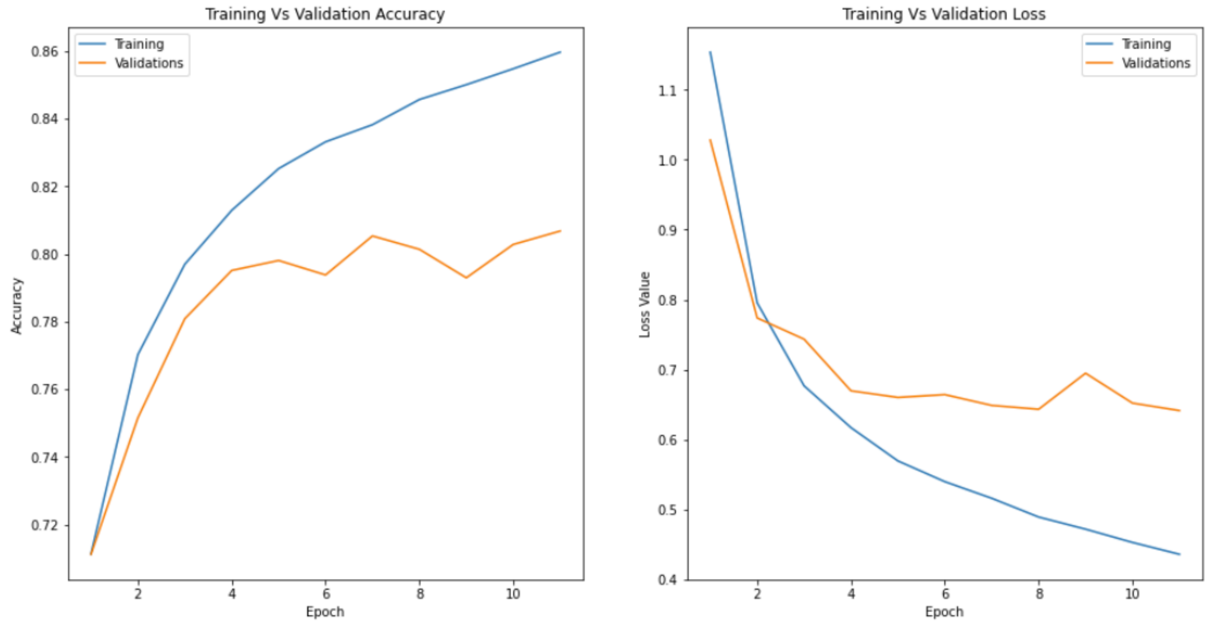
The U-Net model is trained on training images by using a batch size of 32 and validation splits of 5. The model has trained for a max of 30 epochs with early exit true with a stopping condition depending on validation loss.

Out[18]:


input_2: InputLayer | input: | [(None, 224, 224, 3)] | output: | [(None, 224, 224, 3)]
model: Functional | input: | (None, 224, 224, 3) | output: | [(None, 112, 112, 96), (None, 56, 56, 144), (None, 28, 28, 192), (None, 14, 14, 576), (None, 7, 7, 320)]
sequential: Sequential | input: | (None, 7, 7, 320) | output: | (None, 14, 14, 512)
concatenate: Concatenate | input: | [(None, 14, 14, 512), (None, 14, 14, 576)] | output: | (None, 14, 14, 1088)
sequential_1: Sequential | input: | (None, 14, 14, 1088) | output: | (None, 28, 28, 256)
concatenate_1: Concatenate | input: | [(None, 28, 28, 256), (None, 28, 28, 192)] | output: | (None, 28, 28, 448)
sequential_2: Sequential | input: | (None, 28, 28, 448) | output: | (None, 56, 56, 128)
concatenate_2: Concatenate | input: | [(None, 56, 56, 128), (None, 56, 56, 144)] | output: | (None, 56, 56, 272)
sequential_3: Sequential | input: | (None, 56, 56, 272) | output: | (None, 112, 112, 64)
concatenate_3: Concatenate | input: | [(None, 112, 112, 64), (None, 112, 112, 96)] | output: | (None, 112, 112, 160)
conv2d_transpose_4: Conv2DTranspose | input: | (None, 112, 112, 160) | output: | (None, 224, 224, 22)

Results

The trained model learned to predict segmentation masks for a total of 22 classes as mentioned above. The trained model can predict segmentation in images with 85% percent accuracy on training data and 81 percent accuracy on validation images. When I tested the model on test data it yield 85% accuracy. As accuracy on training and test data is almost the same we can say the model is not over-fitted and the augmentation approach played a key role in this. The following plots show the accuracy and loss for training and validation images while training the model.



Discussion/conclusion – Next steps

As this model is based on U-Net it can be trained with fewer images. We are using MobileNetV2 as an encoder, we can run this model on mobile devices or machines with low resources. Because of U-Net and MobileNet2 performance improves drastically however it reduces the accuracy by a very small amount. If we want to use this model in applications that require high accuracy we need to use some other encoder with dense layers. We can further improve this model by using the MobileNetV3 and more images.

Acknowledgments

The authors of this project have referred to VOC and Kaggle for the data set. I referred to some notebooks available on Kaggle and GitHub. There is a significant number of solutions available on the Internet for the problem. We have referred to some of those available solutions and have designed our solution. We have also referred to multiple websites(not limited to the below references) including data science articles from various authors, machine learning websites, mediam.com, towardsdatascience.com, and many others for the basic machine learning and statistics concepts and practical examples.

References

1. [1] Computer Vision Tutorial: A Step-by-Step Introduction to Image Segmentation Techniques – Pulkit Sharma - https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/
2. [2] Image Segmentation - https://en.wikipedia.org/wiki/Image_segmentation
3. [3] Neutrosophic sets in dermoscopic medical image segmentation - Yanhui Guo, Amira S. Ashour https://www.sciencedirect.com/topics/engineering/medical-image-segmentation
4. [4] Image segmentation - https://www.tensorflow.org/tutorials/images/segmentation

5. [5] Image Segmentation in 2021: Architectures, Losses, Datasets, and Frameworks - Derrick Mwiti, Katherine (Yi) Li – Aug 2021 - https://neptune.ai/blog/image-segmentation-in-2020

6. [6] U-net segmentation – Yuanfan You - https://www.kaggle.com/yuanfanyou/u-net-segmentation

7. [7] Visual Object Classes Challenge 2012 (VOC2012) – Pascal 2 - http://host.robots.ox.ac.uk/pascal/VOC/voc2012/#data

8. [8] What Is Image Segmentation? - https://www.mathworks.com/discovery/image-segmentation.html

9. [9] Tutorial 3: Image Segmentation - https://ai.stanford.edu/~syyeung/cvweb/tutorial3.html

10. [10] Image Segmentation With 5 Lines 0f Code - Ayoola Olafenwa – May 2020 - https://towardsdatascience.com/image-segmentation-with-six-lines-0f-code-acb870a462e8

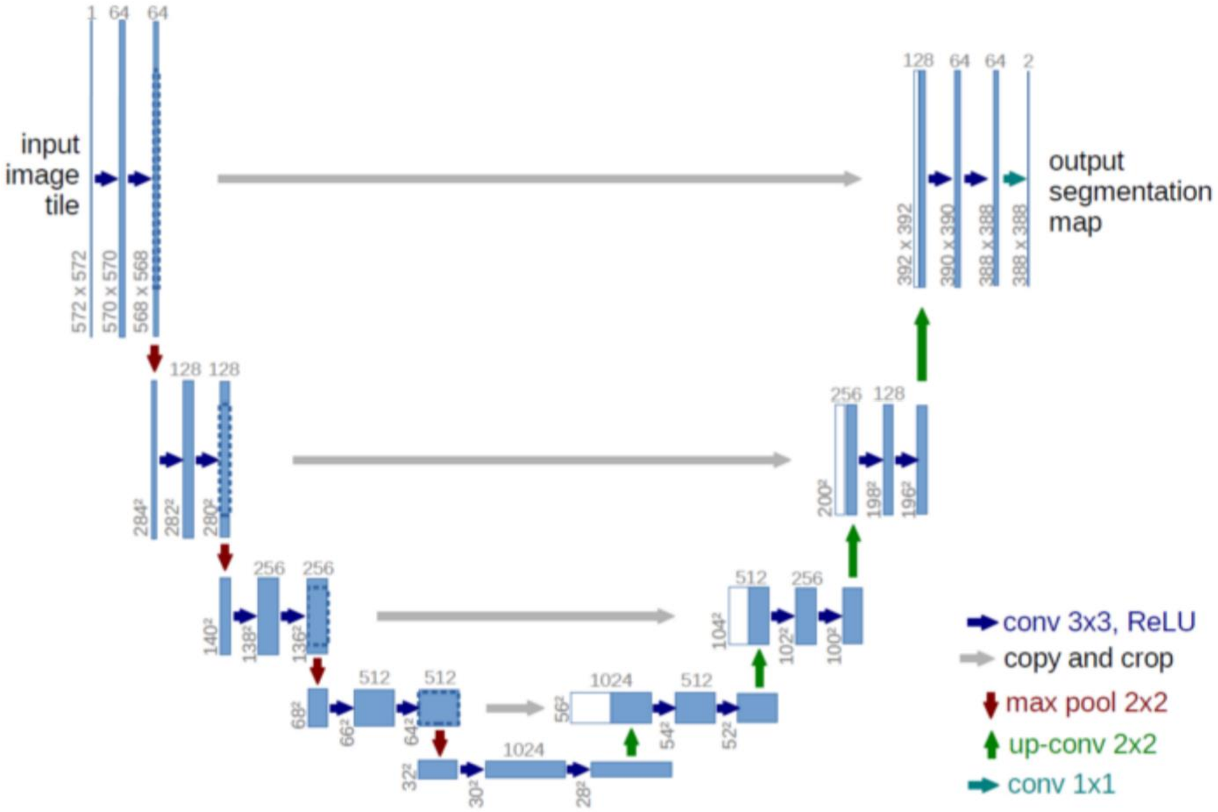
11. [11] Image Segmentation: Part 1 - Mrinal Tyagi - Jul 2018 - https://towardsdatascience.com/image-segmentation-part-1-9f3db1ac1c50

12. [12] MobileNetV2: The Next Generation of On-Device Computer Vision Networks – Mark Sandler and Andrew Howard – April 2018 - https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html

13. [13] Understanding Semantic Segmentation with UNET – Harshall Lamba – Feb 2019 https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47

Appendix

U-Net is an end-to-end fully convolution network. It only contains convolution layers and does not contain any dense layer. The U-Net is a convolutional network architecture for fast and precise segmentation of images. The accuracy of this model is better than the usually used methods like sliding-window convolution network. This network can work with fewer training images. U-Net consists of two parts down-sampling and up-sampling. The down-sampling is used for extracting the features from an image by reducing the resolution but keeping the information intact. The up-sampling is used to convert low-resolution masks to high-resolution images. I have selected a pre-trained MobileNetV2 as an encoder. The MobileNettV2 model is trained on the imagenet dataset. Tensorflow includes the pre-trained MobileNetV2 encoder that is prepared and ready to use. MobileNetV2 can be used with image classification, detection, and segmentation. Effectively depthwise separable convolution reduces computation compared to traditional layers by almost a factor of k^{21} MobileNetV2 uses k = 3 (3 × 3 depthwise separable convolutions) so the computational cost is 8 to 9 times smaller than that of standard convolutions at only a small reduction in the accuracy. MobileNetV2 is very lightweight and can be run on mobile devices. Please find below the U-Net architecture.


1
64
64
input
image
tile
572 x 572
570 x 570
568 x 568
128
128
284^2
282^2
280^2
256
256
140^2
138^2
136^2
512
512
68^2
66^2
64^2
1024
32^2
30^2
28^2
1024
512
56^2
54^2
52^2
512
256
104^2
102^2
100^2
256
128
200^2
198^2
196^2
128
64
64
2
392 x 392
390 x 390
388 x 388
388 x 388
output
segmentation
map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1