Image Segmentation

Sheetal Munjewar

Bellevue University

Course Number: DSC-680 Applied Data Science

Amirfarrokh Iranitalab

01/26/2025

Abstract

Humans excel at analyzing visual scenes and identifying objects effortlessly. Our brain can differentiate between objects like distinguishing people in a photograph. However, achieving similar capabilities in machines was a challenge until recent advancements in computer vision. Today, machine learning models can accurately detect and differentiate objects in images, enabling technologies like self-driving cars and medical imaging for tumor detection. Specifically, image segmentation plays a critical role in medical imaging by isolating regions of interest (ROI), such as separating malignant from non-malignant tumors.

Image segmentation involves partitioning a digital image into meaningful segments based on objects or object types. By removing irrelevant data, this process simplifies analysis of the ROI. This project aims to implement a machine learning model using neural networks for segmenting objects in images.

Business Problem

Accurate image segmentation has critical applications across industries, including:

- **Healthcare:** Identifying tumors or lesions in medical imaging.
- **Autonomous Driving:** Detecting road signs, vehicles, and pedestrians.
- **Satellite Imaging:** Analyzing land use and urban development.

This project aims to develop a scalable, efficient image segmentation model that performs well on constrained devices while maintaining high accuracy.

Data Explanation

Dataset: Visual Object Classes Challenge 2012 (VOC2012)

- **Training Data:** 1,464 images
- **Validation Data:** 1,449 images

Attributes:

Input images (JPEG format, 3 channels)

Mask images (PNG format, 1 channel, indicating object labels)

Classes: Includes objects like airplanes, cars, people, etc. Background pixels are labeled as 0, and unlabeled pixels are 255.

Preprocessing:

Images resized to 224x224 pixels for uniformity.

Data augmentation applied (horizontal/vertical flipping).

Sequential label adjustment to ensure consistent mapping.

Introduction/Background

When crossing a road, humans analyze the scene, identify approaching vehicles, and decide to proceed or wait. Similarly, while driving, we recognize objects like pedestrians or traffic signals to make decisions. These abilities stem from our brain's capacity to differentiate objects by shape, color, and distance. To enable machines to perform similar tasks, we use image segmentation models.

Image segmentation divides a digital image into multiple segments, where each segment represents an object or group of pixels with shared properties. Applications include isolating tumors in medical scans, detecting traffic signals for autonomous vehicles, and locating objects in satellite images. For example, segmenting a brain scan can assist in identifying tumor regions, aiding in treatment planning.

This project trains a machine learning model using a 2012 competition dataset containing 20 object classes. The model employs U-Net, a convolutional neural network designed for biomedical image segmentation, to predict object masks and borders.

Methods

Image segmentation assigns a label to each pixel based on the object it represents. The dataset used for this project is from the Visual Object Classes Challenge 2012 (VOC2012), containing 1,464 training and 1,449 validation images. Each image has:

Input images (JPEG format, 3 channels)

Mask images (PNG format, 1 channel), where pixel values indicate object labels.

Objects include classes such as airplane, car, person, and more, with 0 for background and 255 for unlabeled pixels. Images were resized to 224x224 pixels for uniformity, and label values were adjusted to maintain sequential ordering.

To address the limited training dataset (2,164 images), data augmentation techniques like horizontal and vertical flipping were applied. Augmentation ensured consistency between input and mask images. A batch size of 32 was used for training and validation, with early stopping based on validation loss.

The U-Net model utilized MobileNetV2 as its encoder for feature extraction and pix2pix up sampling for high-resolution mask generation. The model was trained for a maximum of 30 epochs with early stopping to optimize performance.

U-Net: A convolutional neural network designed for segmentation tasks, consisting of:

Down-sampling: Extracts features by reducing resolution while retaining essential information.

Up-sampling: Reconstructs high-resolution images from low-resolution masks.

Encoder: MobileNetV2 (pre-trained on ImageNet)

Lightweight and efficient, using depthwise separable convolutions to reduce computation costs.

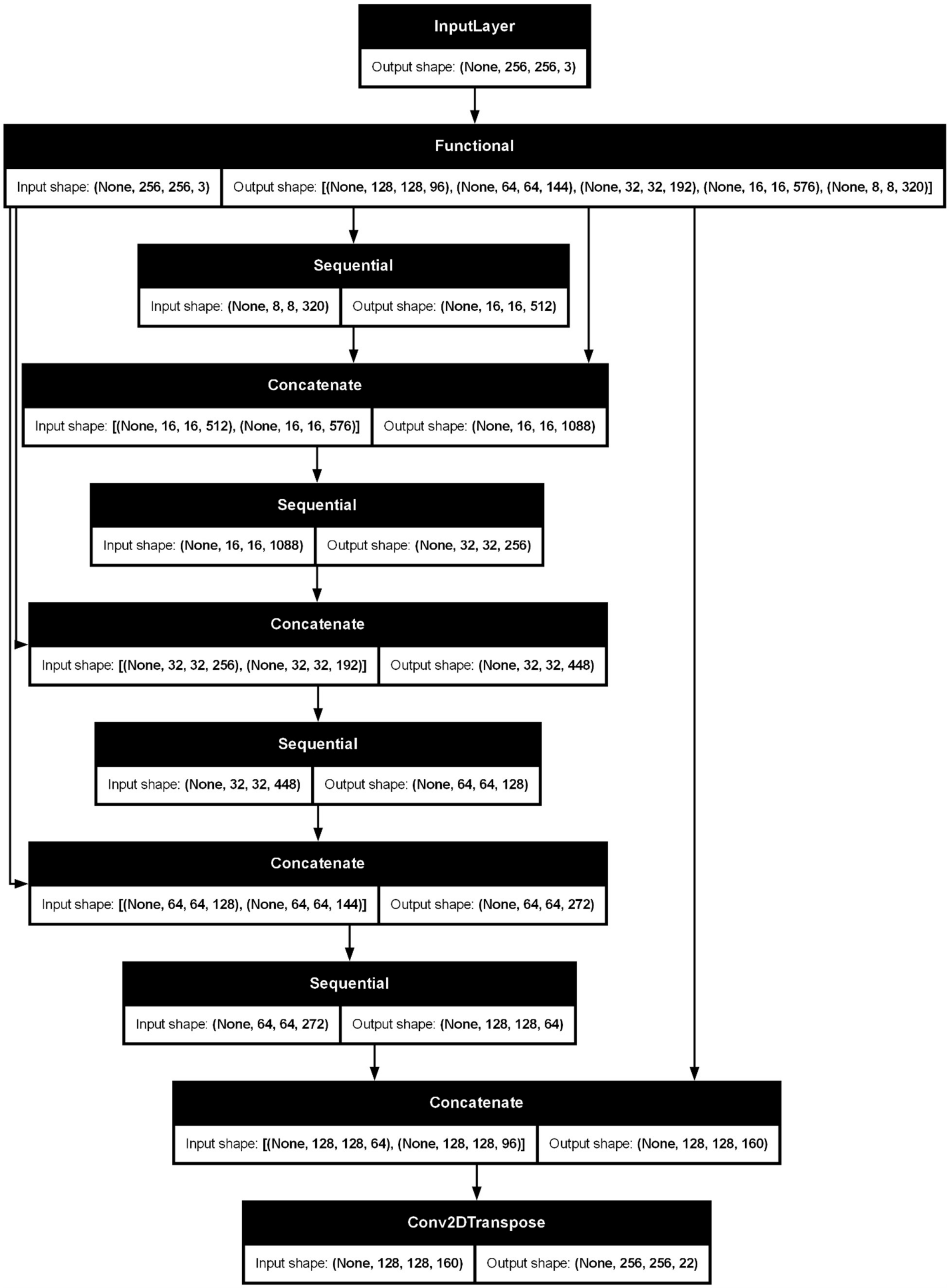
Optimized for resource-constrained environments.

Training Configuration:

Batch Size: 32

Epochs: 30 (with early stopping based on validation loss)

Optimization: Adam optimizer, learning rate = 0.001



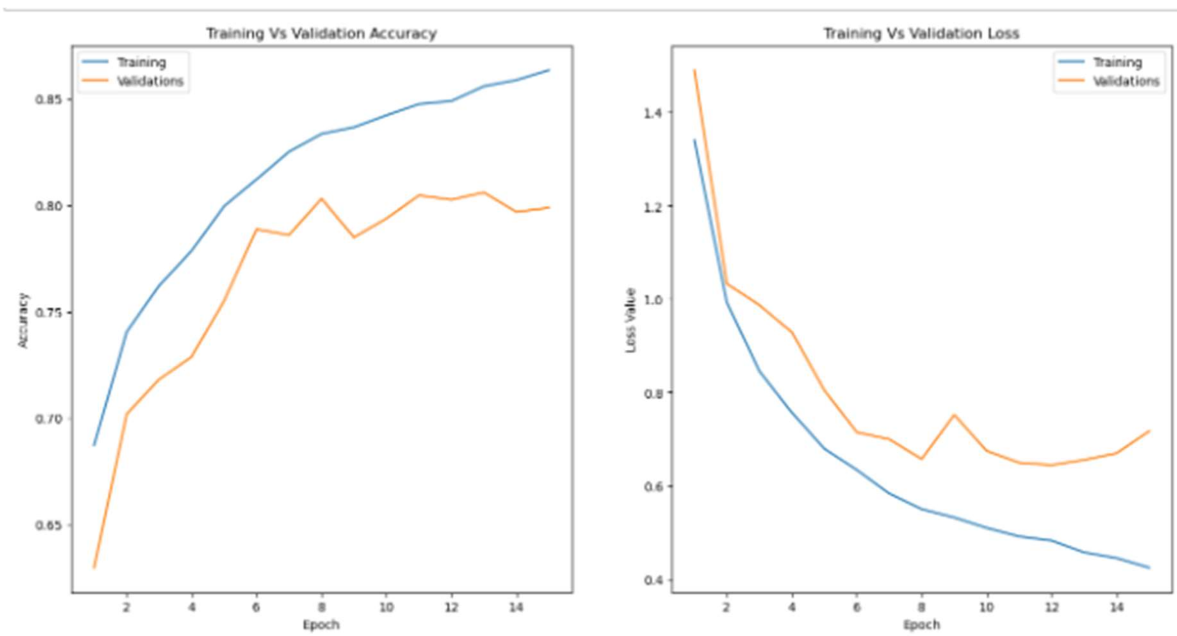
InputLayer
Output shape: (None, 256, 256, 3)
Functional
Input shape: (None, 256, 256, 3)
Output shape: [(None, 128, 128, 96), (None, 64, 64, 144), (None, 32, 32, 192), (None, 16, 16, 576), (None, 8, 8, 320)]
Sequential
Input shape: (None, 8, 8, 320)
Output shape: (None, 16, 16, 512)
Concatenate
Input shape: [(None, 16, 16, 512), (None, 16, 16, 576)]
Output shape: (None, 16, 16, 1088)
Sequential
Input shape: (None, 16, 16, 1088)
Output shape: (None, 32, 32, 256)
Concatenate
Input shape: [(None, 32, 32, 256), (None, 32, 32, 192)]
Output shape: (None, 32, 32, 448)
Sequential
Input shape: (None, 32, 32, 448)
Output shape: (None, 64, 64, 128)
Concatenate
Input shape: [(None, 64, 64, 128), (None, 64, 64, 144)]
Output shape: (None, 64, 64, 272)
Sequential
Input shape: (None, 64, 64, 272)
Output shape: (None, 128, 128, 64)
Concatenate
Input shape: [(None, 128, 128, 64), (None, 128, 128, 96)]
Output shape: (None, 128, 128, 160)
Conv2DTranspose
Input shape: (None, 128, 128, 160)
Output shape: (None, 256, 256, 22)

Results

The model achieved:

- **Training Accuracy:** 85%
- **Validation Accuracy:** 81%

Training and validation loss curves indicated stable convergence without overfitting, demonstrating the effectiveness of data augmentation.



Conclusion

The U-Net model, with a MobileNetV2 encoder, provides efficient segmentation performance on limited resources. While slightly less accurate than heavier models, it is suitable for mobile applications and devices with limited computational power.

Assumptions

The dataset annotations are accurate and represent real-world conditions.

Data augmentation compensates for the limited dataset size.

Limitations

The model was trained on a relatively small dataset (2,164 images).

Performance might degrade for unseen object classes or larger image dimensions.

Challenges

Balancing computational efficiency and model accuracy.

Handling class imbalance in the dataset.

Future Uses/Additional Applications

Expanding the dataset to include diverse objects and scenarios.

Exploring MobileNetV3 or other encoders for improved performance.

Applying the model to real-time applications, such as video segmentation.

Recommendations

Use larger and more diverse datasets to enhance generalizability.

Incorporate advanced data augmentation techniques like color jitter and Gaussian blur.

Evaluate the model with alternative loss functions to handle imbalanced classes better.

Implementation Plan

Data Preparation: Preprocess data and augment for robustness.

Model Training: Optimize hyperparameters and monitor validation performance.

Deployment: Implement the model on resource-constrained platforms (e.g., mobile devices).

Monitoring: Continuously evaluate model performance in production and update as needed.

Ethical Assessment

Data Privacy: Ensure dataset compliance with privacy regulations.

Bias Mitigation: Avoid model biases by incorporating diverse datasets.

Transparency: Clearly communicate model limitations and intended applications.

Discussion/Conclusion – Next Steps

The U-Net model, enhanced with MobileNetV2, is efficient and can run on resource-constrained devices, albeit with a slight reduction in accuracy. For applications requiring higher precision, alternative encoders with dense layers could be explored. Future improvements include leveraging MobileNetV3 and incorporating larger datasets for better accuracy.

Acknowledgments

This project references datasets and solutions from VOC2012, Kaggle, GitHub, and numerous online resources, including analytics and machine learning tutorials.

References

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/

Image Segmentation - https://en.wikipedia.org/wiki/Image_segmentation

Neutrosophic sets in dermoscopic medical image segmentation - Yanhui Guo, Amira S. Ashour https://www.sciencedirect.com/topics/engineering/medical-image segmentation

Image segmentation - https://www.tensorflow.org/tutorials/images/segmentation

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

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

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

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

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

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

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

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

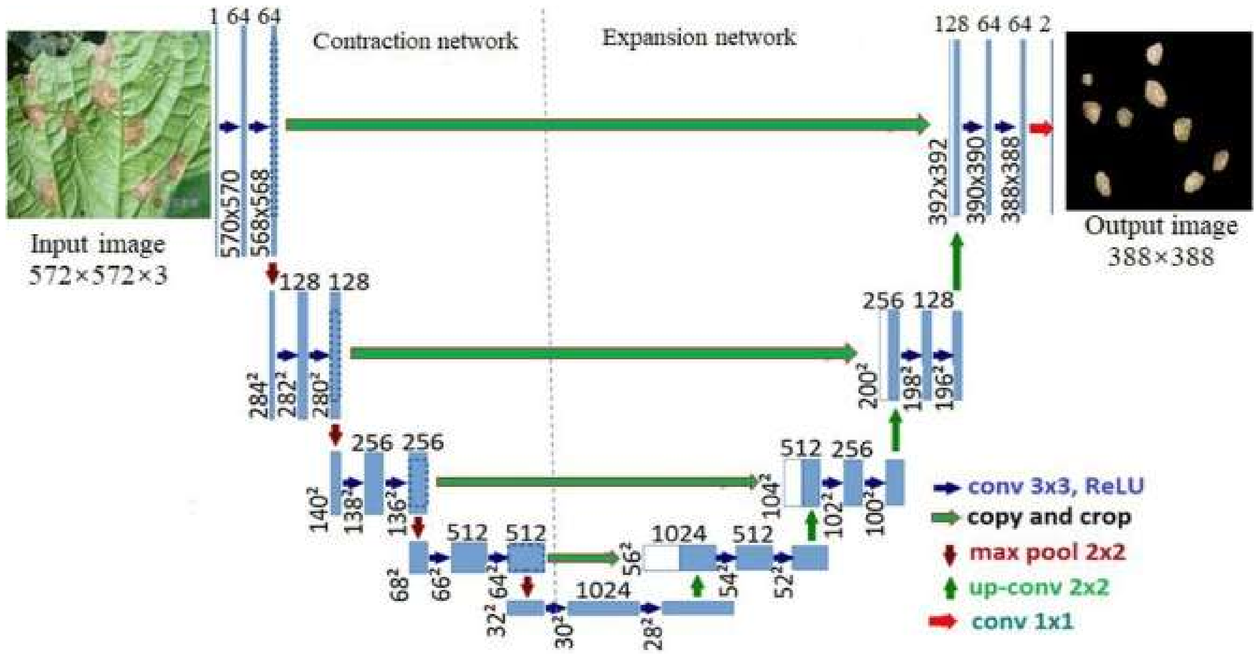
Understanding Semantic Segmentation with UNET – Harshall Lamba – Feb 2019 https://towardsdatascience.com/understanding-semantic-segmentation-with-unet 6be4f42d4b47

Appendix

The U-Net is an end-to-end fully convolutional network designed specifically for fast and accurate image segmentation. Unlike traditional networks, it contains only convolutional layers and excludes dense layers, making it highly efficient. U-Net's architecture outperforms conventional methods like the sliding-window convolution network and requires fewer training images to achieve high accuracy. The network is composed of two primary components: the down-sampling and up-sampling paths. The down-sampling path is responsible for feature extraction, reducing the resolution while retaining essential information. Conversely, the up-sampling path reconstructs high-resolution images from low-resolution masks, enabling precise segmentation.

For this project, I utilized a pre-trained MobileNetV2 as the encoder. MobileNetV2, trained on the ImageNet dataset, is readily available within TensorFlow as a pre-configured encoder for various tasks, including image classification, detection, and segmentation. This model employs depthwise separable convolutions, significantly reducing computational requirements compared to traditional layers—by nearly a factor of k2k^2. With k=3k = 3 (3 × 3 depthwise separable convolutions), MobileNetV2 achieves an 8- to 9-fold reduction in computational cost while maintaining only a minor trade-off in accuracy. Its lightweight design ensures compatibility with mobile devices and other low-resource environments.

Below is the U-Net architecture for reference.



Input image
572×572×3
Output image
388×388
Contraction network
Expansion network
1 64 64
570x570
568x568
128 128
284²
282²
280²
256 256
140²
138²
136²
512 512
68²
66²
64²
1024
32²
30²
28²
1024 512
56²
54²
52²
512 256
104²
102²
100²
256 128
200²
198²
196²
128 64 64 2
392x392
390x390
388x388
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1