IMAGE SEGMENTATION USING U-NET MODEL

Purvi Ghidora
National Institute of Technology, Raipur

Swati Mishra
Kalinga Institute of Industrial Technology

Abstract

Image processing is a wide area of research with various important real-life applications. It is used as a way to make images easily interpretable by humans. Many times, it is important to segment an image into different subgroups to reduce the complexity of the image and make the analysis simpler. In this project, we have implemented this concept of image segmentation using deep learning on the Cityscapes dataset. The neural network used for this project is the U-net model in PyTorch. We also incorporated metrics to evaluate the segmentation quality of the model and trained it on the dataset to gain qualitative and quantitative results.

Keywords: segmentation, cityscapes, U-Net, convolution, image.

1. Introduction

An image is a visible representation of something and carries a lot of beneficial information. Analyzing the image and obtaining information from it in a way that does not influence the image's other features in order to get some tasks done is one of the important applications of digital image technology. In various sectors and real-world applications such as military, medical, astronomy, etc., pattern recognition, image analysis, and image disciplines are the most important subjects in computer science and computer engineering. Object detection, classification, and segmentation are the major concepts that are used in the development of applications and software that incorporate computer vision techniques. The concept of segmentation is used to divide components of an image based on their similarities and differences. Selecting a dataset containing relevant attributes can be used on the segmentation model to obtain results based on the application. Artificial neural networks (ANNs) are useful in image classification, pattern recognition, clustering, and prediction in various applications.

Nowadays, ingenious technologies are developing in the domain of image processing, particularly in the image segmentation field. Image segmentation is an important and difficult process. The image segmentation process is the most significant phase of image analysis. The technique of splitting an image into homogenous parts based on particular criteria and, ideally, corresponding to actual things in the scene is known as image segmentation. U-Net is one of the deep learning architectures used for this purpose.

The U-Net model is fully convolutional since it only contains convolutional layers. It consists of two parts: an encoder and a decoder. The encoder contains a 3×3 convolution layer followed by a ReLU activation unit and a 2×2 max pooling layer for downsampling, that is, for reducing the size of the input image. In the decoder part, skip connections are used to concatenate the feature maps of the encoder to the output of transposed convolutions of the same level. Upsampling is used to bring the condensed information back to its original size at every level.

The Cityscapes dataset consists of diverse urban street scenes from different cities under different weather and conditions. Our model is trained on this dataset and then successfully tested by assigning each pixel a particular label category and visualising them by mapping each label category to different colours.

2. Literature review

This literature review contains a comprehensive review of various technical papers referred to during the implementation of this project, containing a wide range of pioneering works on semantic and instance segmentation, deep learning neural architecture, encoder-decoder models, datasets, and applications of image segmentation.

2.1 Image Segmentation

Image segmentation is the process of clustering similar parts of an image together, dividing the overall image into smaller subgroups, to reduce the complexity of the image and enable further analysis of each segment. Image segmentation uses machine learning algorithms to obtain the desired output. The goal of this process is to highlight the images in the foreground to make it more efficient for the algorithm to evaluate the information provided in the given image. These individual pixel-segmented categories can then be fed to the machine learning algorithm, as recognizing these specific image segments in a photograph is easier than recognizing a single element within an image that contains other objects.

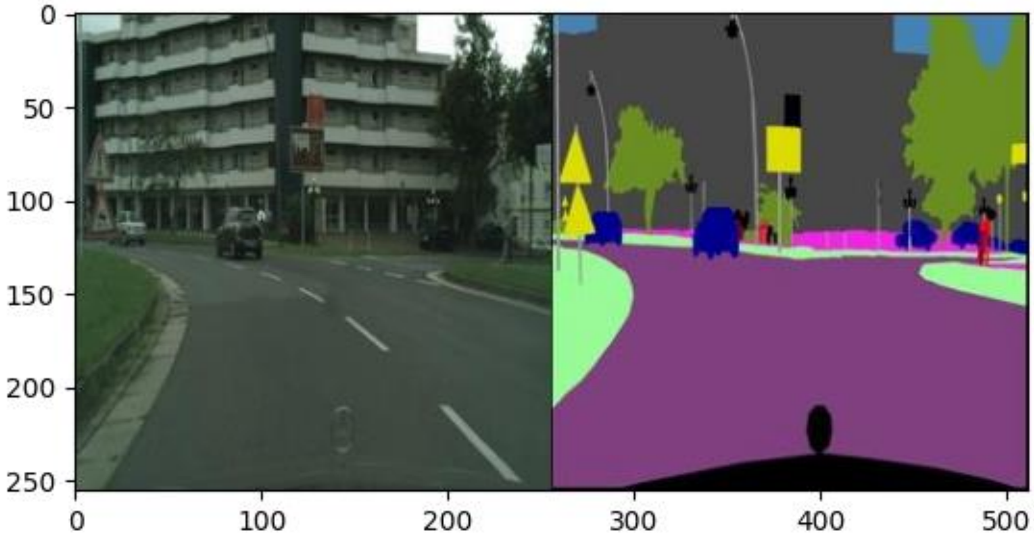


Figure: Semantic Segmentation

Similarity and discontinuity are the two common approaches in segmentation. The similarity approach involves recognizing similarity in the pixels of the image to form an individual segment, along with a specific threshold given beforehand. Alternatively, the discontinuity approach deals with identifying the pixel density values present in the image.

Segmentation models have achieved the highest accuracy using deep learning models, which showcase a remarkable improvement in the performance of various applications. Deep learning image classification mainly involves Convolutional Neural Networks (CNN). Mask R-CNN algorithms generate three different outputs for each object in an image: the object mask of the image, the class of the image, and bounding box coordinates.

Image segmentation plays a crucial role in the development of computer vision software programs. To illustrate this, examples of video surveillance systems use image segmentation to identify people, cars, street lights, and other miscellaneous objects within video recordings. On the other hand, medical professionals use image segmentation while using medical imaging software, as these programs must be able to identify specific features within the human body.

2.2 Semantic and Instance segmentation

Semantic segmentation refers to labelling the image pixels with a set of object categories (e.g., human, car, tree, sky) for all image pixels, therefore it is more challenging than image classification, which predicts a single label for the image. Instance segmentation marks the boundaries of an object and identifies instances. (e.g., partitioning of a single person) Some examples are Cityscapes, PASCAL VOC and ADE20. Usually, we evaluate the models with the Mean Intersection-Over-Union (Mean IoU) and Pixel Accuracy metrics.

Semantic Segmentation follows three steps:

- Classifying: Classifying a certain object in the image.
- Localizing: Finding the object and drawing a bounding box around it.
- Segmentation: Grouping the pixels in a localized image by creating a segmentation mask.

Semantic segmentation frequently necessitates the extraction of features and representations that can derive meaningful correlations from the input image, thereby removing noise.

The convolutional neural network (CNN) performs this task and is frequently used in most computer vision tasks. The following section explores different semantic segmentation methods that use CNN as the core architecture. The architecture can be modified by adding extra layers and features, or by changing the architecture and its design altogether.

2.3 UNET

Convolutional networks are used for the purpose of classification, i.e., we obtain the output image as a single class label. But in practical examples like biomedical image processing, a class label must be assigned to each pixel (localization), and thousands of images are really challenging to emulate for biomedical tasks. To overcome this, Ciresan et al trained a network in a sliding window setup to define the class label for all the pixels and provide the pixel with a local region (patch) as input. To facilitate this, the U-Net architecture is "U" shaped, and symmetric.

The model can be divided into two major parts: the left part, generally known as the "contracting path", which carries forward the general convolutional process, and the right path, the "expansive part", which is made up of transposed 2-d convolutional layers. One significant thing in this architecture is the addition of more feature channels to the upsampling portion, which enables the network to transport context information to higher resolution layers. So as a result, the expansive approach results in a u-shaped design that is roughly symmetric to the contracting path. The segmentation map only comprises the pixels for which the whole context is present in the input image, as the network has no fully linked layers and only uses the valid part of each convolution. By using an overlap-tile technique, this solution enables the smooth segmentation of arbitrary huge images. The missing context is extrapolated by mirroring the input image in order to forecast the pixels in the border region of the image.

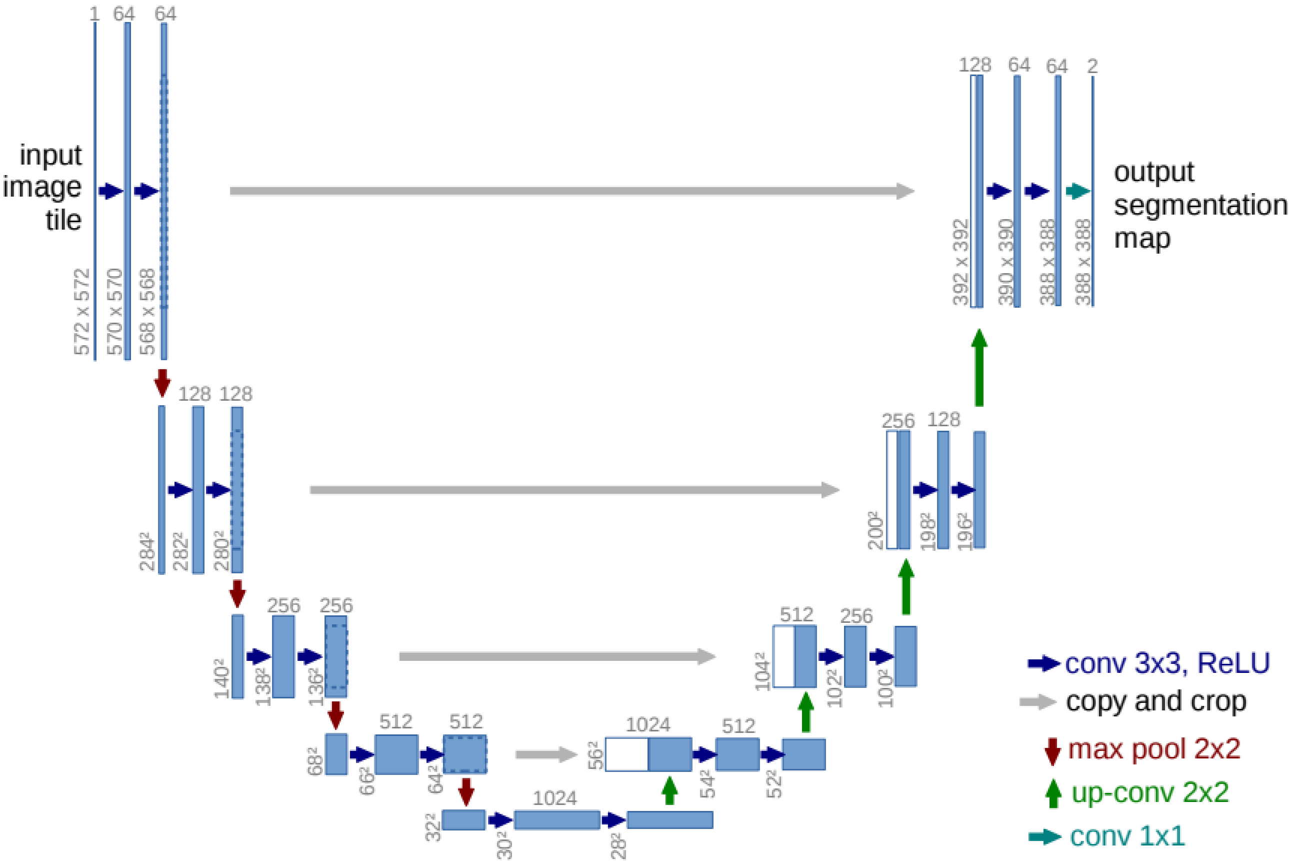


Figure: UNET architecture

(Image taken from "U-Net: Convolutional Networks for Biomedical Image Segmentation" by Olaf Ronneberger, Philipp Fischer, and Thomas Brox)

<u>Contracting Path</u>

The contracting path follows the formula:

conv_layer1 -> conv_layer2 -> max_pooling -> dropout(optional)

In this method, each process is made up of two convolutional layers, and there is an increase in the number of channels to increase the depth of the image.

Expansive Path

Expansive path is a process where the image is upscaled to its original size. The formula is given below:

conv_2d_transpose -> concatenate -> conv_layer1 -> conv_layer2

The method of transposed convolutions expands the size of the image, and is an upscaling technique.

The image we obtain after this will be concatenated with the corresponding image from the contraction. This is done in order to combine the information from the previous layers so that we get a more precise prediction. After the contracting and expansive paths, we have reached the uppermost layer of the architecture. Lastly, we reshape our image in order to satisfy our prediction requirements. Therefore, U-net has the ability to perform image localisation by conducting pixel-to-pixel prediction, and this convolutional neural network is capable enough to provide useful outputs based off of a very small number of datasets by using data augmentation techniques.

2.4 Convolutional Neural Networks (CNN)

Computer programs called artificial neural networks (ANN) are modelled after the human nervous system. They have a large number of connected computational nodes that use input to learn how to optimise output. CNNs are the most popular and effective deep learning neural networks for computer vision tasks. A CNN's neurons learn to optimise themselves. Each neuron receives a weighted input of a raw image to perform an action on and outputs a weighted scoring function. CNNs are primarily used in pattern recognition, allowing us to incorporate feature-specific images into architecture while reducing the number of model parameters needed. A convolutional layer, a pooling layer, and a non-linear activation function make up the architecture. For semantic segmentation, the layers are not necessarily fully connected, because in semantic segmentation the end goal is not to predict the class label of an image but to obtain the features before using them to separate the image into segments. However, the issue with convolutional networks is that the size of the image is reduced as it passes through the network because of the max-pooling layers. To efficiently separate the image into multiple segments, we need to upsample it using an interpolation technique, which is achieved using deconvolutional layers. The main computational advantage of CNNs is that all the receptive fields in a layer share weights, resulting in a significantly smaller number of parameters than fully-connected neural networks.

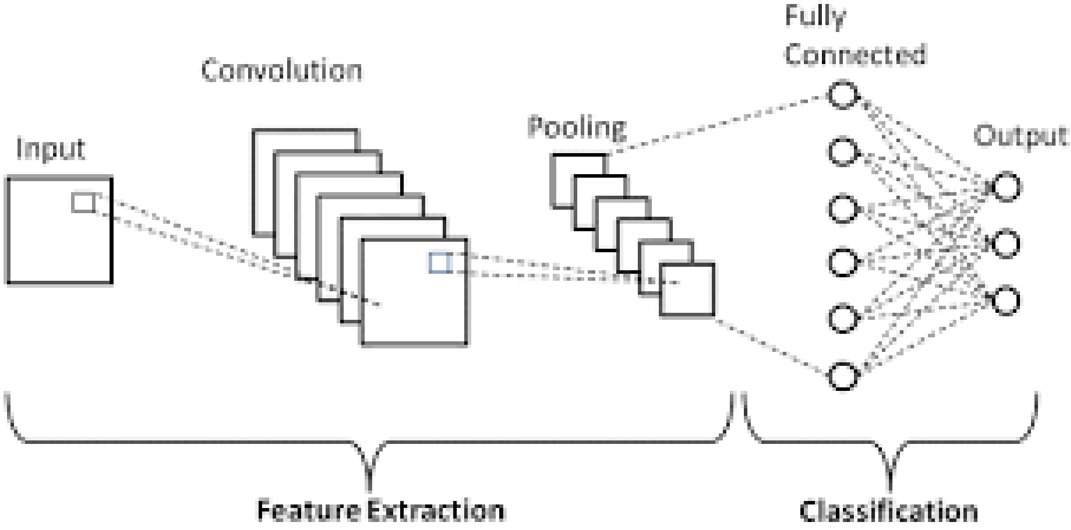


Figure: CNN Architecture

The basic functionality of the example CNN above can be broken down into four key areas.

1. The input layer will hold the pixel values of the image.
2. The convolutional layer will determine the output of neurons, some of which are connected to local regions of the input, through the calculation of the scalar product between their weights and the region connected to the input volume. The rectified linear unit (abbreviated ReLu) attempts to apply an "elementwise" activation function, such as a sigmoid, to the output of the previous layer's activation.
3. Then, the pooling layer will just do downsampling along the spatial dimension of the given input, which will reduce the number of parameters in that activation even more.

The layer's parameters focus on the use of learnable kernels. These kernels are usually small in spatial dimensionality, but they spread along the entirety of the depth of the input. When the data hits a convolutional layer, the layer convolves each filter across the spatial dimensionality of the input to produce a 2D activation map. These activation maps can be visualized. As we glide through the input, the scalar product is calculated for each value in that kernel. From this, the network will learn kernels that ’fire’ when they see a specific feature at a given spatial position of the input. These are commonly known as activations.

Convolutional layers are also able to significantly reduce the complexity of the model through the optimization of its output. These are optimized through three hyperparameters: the depth, the stride, and setting zero-padding.

RNN: RNNs are typically problematic with long sequences as they cannot capture long-term dependencies in many real-world applications. However, a type of RNNs called Long Short Term Memory (LSTM) is designed to avoid these issues. The LSTM architecture includes three gates (input gate, output gate, and forget gate), which regulate the flow of information into and out of a memory cell, which stores values over arbitrary time intervals.

In general AI terminology, the convolutional network that is used to extract features is called an encoder. The encoder also downsamples the image, while the convolutional network that is used for upsampling is called a decoder.

2.5 Encoder-Decoder Models

Encoder

Encoding means converting data into a required format. In the context of machine learning, we convert a sequence of words into a two-dimensional vector; this two-dimensional vector is also known as the hidden state. The encoder is built by stacking recurrent neural network (RNN). We use this type of layer because its structure allows the model to understand the context and temporal dependencies of the sequences. The output of the encoder, the hidden state, is the state of the last RNN timestep.

Hidden State (Sketch)

The output of the encoder, a two-dimensional vector that encapsulates the whole meaning of the input sequence. The length of the vector depends on the number of cells in the RNN.

Decoder

To decode means to convert a coded message into intelligible language. In the machine learning model, the role of the decoder will be to convert the two-dimensional vector into the output sequence, the English sentence. It is also built with RNN layers and a dense layer to predict the English word.

One of the major advantages of this model is that the length of the input and output sequences may differ. This opens the door for very interesting applications such as video captioning or question-and-answer.

The major limitation of this simple encoder-decoder model is that all the information needs to be summarized in a one-dimensional vector, and for long input sequences, that can be extremely difficult to achieve.

2.6 Image segmentation Datasets

For a fair analysis of an image segmentation algorithm, a dataset is used as the key. The performance of segmentation methods is evaluated in comparison to the difficulties presented by the benchmark datasets. A benchmark dataset typically consists of a selection of photos with varying degrees of detail. Illumination variation, intra-class variance, and background complexity are a few of the frequent issues that segmentation algorithms must address.

Data augmentation is a process of artificially increasing the amount of data by generating new data points from existing data. This includes adding minor alterations to data or using machine learning models to generate new data points in the latent space of original data to amplify the dataset. Data augmentation has proven to improve the performance of the models, especially when learning from limited datasets, such as those in medical image analysis. It can also be beneficial in yielding faster convergence, decreasing the chance of over-fitting, and enhancing generalization. For some small datasets, data augmentation has been shown to boost model performance more than 20%.

We group these datasets into 3 categories—2D images, 2.5D RGB-D (color+depth) images, and 3D images.

(i) 2D DATASETS

The majority of image segmentation research has focused on 2D images; therefore, many 2D image segmentation datasets are available. Some commonly used datasets are:

Cityscapes is a large-scale database with a focus on semantic understanding of urban street scenes. It contains a diverse set of stereo video sequences recorded in street scenes from 50 cities, with high quality pixel-level annotation of 5k frames, in addition to a set of 20k weakly annotated frames. It includes semantic and dense pixel annotations of 30 classes, grouped into 8 categories—flat surfaces, humans, vehicles, constructions, objects, nature, sky, and void.

The Cityscapes Dataset is intended for:

1. assessing the performance of vision algorithms for major tasks of semantic urban scene understanding: pixel-level, instance-level, and panoptic semantic labeling;

2. supporting research that aims to exploit large volumes of (weakly) annotated data, e.g. for training deep neural networks.

PASCAL visual object class (VOC)

For the segmentation task, there are 21 classes of object labels—vehicles, household, animals, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, TV/monitor, bird, cat, cow, dog, horse, sheep, and person (pixel are labeled as background if they do not belong to any of these classes). This dataset is divided into two sets, training and validation, with 1,464 and 1,449 images, respectively.

(ii) 2.5D DATASETS

Visual 2.5D perception involves understanding the semantics and geometry of a scene through reasoning about object relationships with respect to the viewer in an environment. However, existing works in visual recognition primarily focus on the semantics. To bridge this gap, we study 2.5D visual relationship detection (2.5VRD), in which the goal is to jointly detect objects and predict their relative depth and occlusion relationships.

(iii) 3D DATASETS

3D object segmentation is a fundamental and challenging problem in computer vision with applications in autonomous driving, robotics, augmented reality and medical image analysis. It has received significant attention from the computer vision, graphics and machine learning communities. Traditionally, 3D segmentation was performed with hand-crafted features and engineered methods which failed to achieve acceptable accuracy and could not generalize to large-scale data. Driven by their great success in 2D computer vision, deep learning techniques have recently become the tool of choice for 3D segmentation tasks as well.

2.7 Metrics to evaluate segmentation quality

- Measuring the percentage of pixels in an image that were correctly identified is a different way to assess the correctness of a semantic segmentation. The pixel accuracy is frequently given both generally and individually for each class. A true positive represents a pixel that is correctly predicted to belong to the given class (according to the target mask), whereas a true negative represents a pixel that is correctly identified as not belonging to the given class. This evaluation of the per-class pixel accuracy is essentially a binary mask evaluation. When the class representation in the image is minimal, this metric may occasionally produce inaccurate findings since it will overstate your ability to recognize negative cases (where the class is not present).

- Intersection over union: IoU is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth.

- F1 Score: Precision and recall are defined for each class, as well as the aggregate level. It consists of a true positive fraction, as well as a false negative fraction. We usually use a combined version of precision and recall rates. F1 score is the harmonic mean of the precision score.

2.8 Loss function

The loss function is a function that is used to determine the efficiency of a model. If the loss function comes out to be a large value, then we conclude that the network doesn’t really perform well, and our aim will be to minimize the function.

2.9 Training a deep learning model

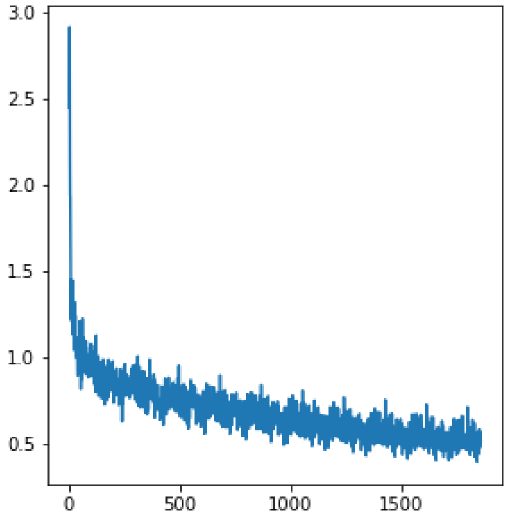
When we train a model, we start with one that does not perform very well and end up with a high performing model. We minimize the loss function output by adjusting the weights (weighted inputs). The size of the batch to use for estimating the derivatives is another hyper-parameter that we consider. Usually, we choose a large batch size, as much as our memory can handle.

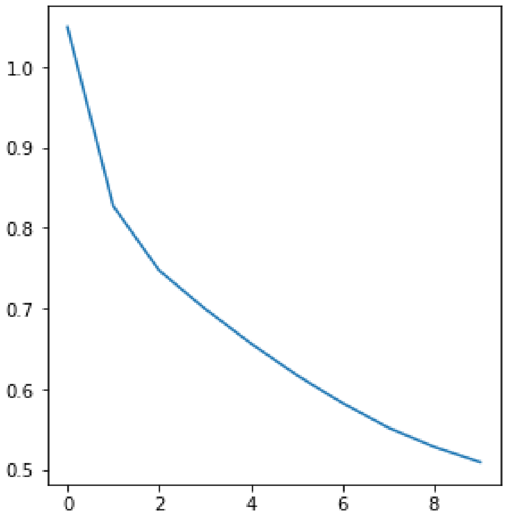
The extreme version of mini-batch gradient descent with batch size equal to 1 is called Stochastic Gradient Descent. When we refer to Stochastic Gradient Descent (SGD), we actually refer to mini-batch gradient descent. Most deep learning frameworks will let us choose the batch size for SGD.

3. Results

A. Quantitative

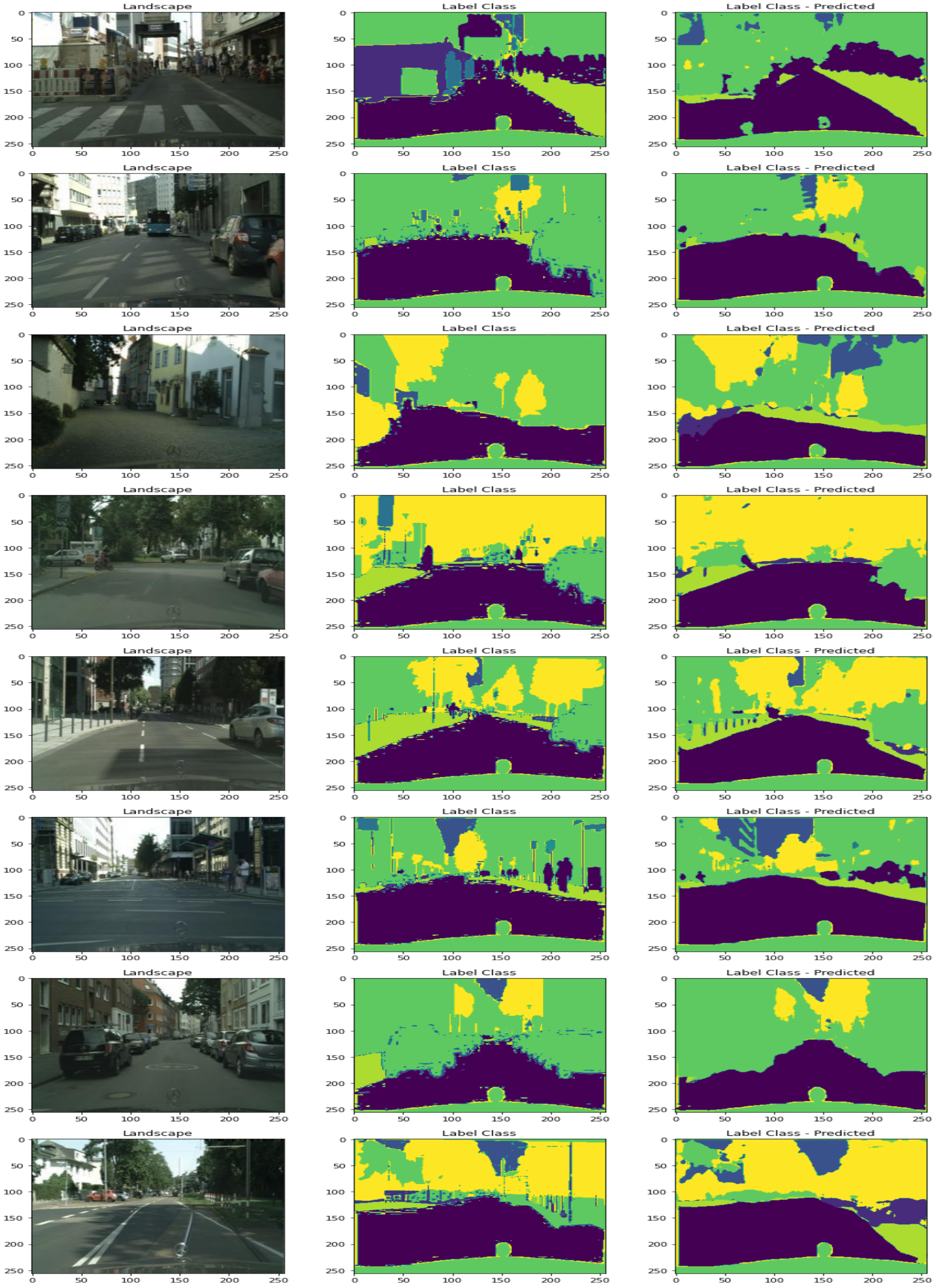
The mean IoU score after training for 10 epochs is 0.9665521





B. Qualitative

The figure below shows the actual image in the first column, the labelled class in the second column and the segmented image predicted by our U-Net model in the last column.


Landscape
Label Class
Label Class - Predicted
0
50
100
150
200
250

4. Conclusion

We have successfully implemented the concept of image segmentation using the U-Net model. The project was implemented using the PyTorch framework, and the cityscapes dataset was used. U-Net was initially developed for biomedical image segmentation. By dividing an image into different segments, we can easily process only the needed segments instead of processing the whole image. Image segmentation is a key building block of computer vision technologies and algorithms. It is used for many practical applications, including medical image analysis, computer vision for autonomous vehicles, face recognition and detection, video surveillance, and satellite image analysis.

5. Related Works

Deep Learning for Automatic Image Segmentation in Stomatology and Its Clinical Application

In the dentistry field, the fundamental source of data for biomedical operations is radiographic images. The X-rays of the teeth, jaws, gums, and bones are used to check the condition of the oral health of the patient. In the field of stomalogical research, deep learning and segmentation of images for biomedical results have become active research topics. The images are categorised for the mdical professionals to weigh the advantages and disadvantages to obtain conclusions. Common stomatological images are classified into five types: panoramic radiography, dental X-rays, intraoral scanning (IOS), MSCT, and CBCT. Each type is specific for clinical purposes, according to its imaging principles. It is necessary to separate the teeth, jaws, and backdrop in the semantic segmentation task without differentiating between the individuals in each group ("Tooth" or "Jaw"). In contrast, the individuals in each category (such as "Tooth" or "Jaw") must be distinguished in the instance segmentation task, where both the category label and the instance label (within the class) are necessary.

The most common methods of image segmentation in stomatology are built on the concept of CNN. For the automatic segmentation of teeth, the UNet network is used in panoramic radiography. The image segmentation method also used the method to segment teeth in complex cases such as tooth loss, defect, filling, and fixed bridge restoration, achieving a Dice similarity coefficient (DSC) of 0.744 on the CBCT dataset.

6. References

[1] Abdulateef, Salwa & Salman, Mohanad. (2021). A Comprehensive Review of Image Segmentation Techniques. Iraqi Journal for Electrical and Electronic Engineering. 17. 166-175. 10.37917/ijeee.17.2.18.

[2] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 7, pp. 3523-3542, 1 July 2022, doi: 10.1109/TPAMI.2021.3059968.

[3] O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

[4] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481-2495, 1 Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[5] “Fully Convolutional Networks for Semantic Segmentation” ***Jonathan Long, Evan Shelhamer, Trevor Darrell***; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431-3440

[6] Yutong Cai, Yong Wang, "MA-Unet: an improved version of Unet based on multi-scale and attention mechanism for medical image segmentation," Proc. SPIE 12167, Third International Conference on Electronics and Communication; Network and Computer Technology (ECNCT 2021), 121670X (7 March 2022);

[7] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui and J. Long, "A Survey of Clustering With Deep Learning: From the Perspective of Network Architecture," in IEEE Access, vol. 6, pp. 39501-39514, 2018, doi: 10.1109/ACCESS.2018.2855437.

[8] A. Alem and S. Kumar, "Deep Learning Models Performance Evaluations for Remote Sensed Image Classification," in IEEE Access, vol. 10, pp. 111784-111793, 2022, doi: 10.1109/ACCESS.2022.3215264.

[9] N. A. Nezla, T. P. Mithun Haridas and M. H. Supriya, "Semantic Segmentation of Underwater Images using UNet architecture based Deep Convolutional Encoder Decoder Model," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, pp. 28-33, doi: 10.1109/ICACCS51430.2021.9441804.

[10] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *ArXiv*, *abs/1505.04597*.