



**CITY UNIVERSITY  
LONDON**

# **Deep Learning for Image Analysis**

## ***Written Report***

Martin Fixman and Grigorios Vaitsas

2023/2024 Term

# 1 Introduction

In this study we have deployed the freely available Cityscapes dataset [1] to perform semantic segmentation analysis on urban scene imagery using a variety of Deep Learning architectures. Semantic segmentation is a computer vision technique used to identify and label specific parts of an image at pixel level. The main goal is to partition an image into areas (segments) that correspond to different objects or classes of interest (for example person, car, tree, road, building etc.). This technique is widely used in various fields like for instance, in autonomous driving, where it helps cars understand the environment around them, as well as in medical imaging, remote sensing, and robotics. Semantic segmentation is typically performed using deep learning methods, particularly convolutional neural networks (CNNs) and their variants as well as more recent attention-based models.

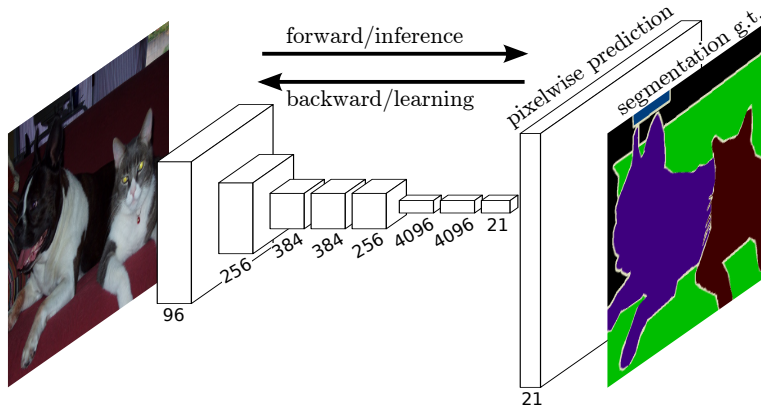
## 2 Dataset

The Cityscapes dataset is a large-scale dataset widely used for training and evaluating algorithms in the fields of computer vision, particularly for semantic understanding of urban street scenes. It features a collection of diverse urban street images from 50 different cities, collected across Germany and some neighbouring countries. The dataset includes over 5,000 fine annotations and 20,000 coarse annotations (see Figure 1) of high-resolution images ( $2048 \times 1024$  pixels). The fine annotations are detailed pixel-wise labels for 30 classes such as road, car, pedestrian, building, and traffic light. The detailed annotations are particularly valuable for our task where the goal is to assign a label to each pixel of the image.



**Figure 1:** Examples of coarse (a) and fine (b) mask annotations superimposed on the corresponding original images

The scenes represent a variety of seasons, daylight conditions, and weather scenarios, providing robust, real-world environments for training models that need to perform under varied conditions. This dataset has been widely used in research for developing and testing algorithms on tasks such as object detection, semantic segmentation, and instance segmentation in urban settings as well as advancing the state-of-the-art in visual perception for autonomous driving systems. The Cityscapes dataset can be accessed in the following address:



**Figure 2:** Example of a fully convolutional network used for semantic segmentation [2]

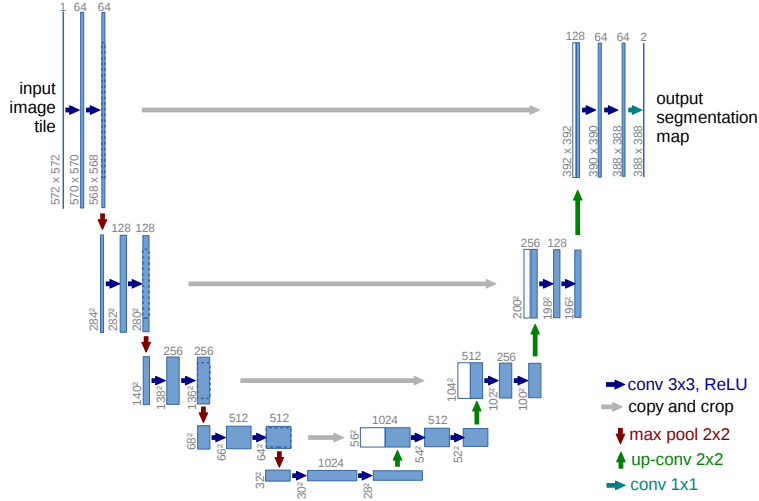
<https://www.cityscapes-dataset.com>

Our particular approach in this study is to create and train at least two models; one with a basic architecture that will form our baseline and a second one where we will be exploring a more complex architecture. We are aiming to demonstrate first of all that both our models are able to perform semantic segmentation on the chosen dataset. Our second goal is to investigate how the choice of various hyper-parameters and tweaks in architecture can affect the accuracy and performance of the models.

### 3 Architectures

Modern, state-of-the-art techniques for image analysis involve convolutional neural networks (CNNs) in their implementation. CNNs can be used to extract vital information from spatial features, which allow us to classify and segment objects in images. A type of neural network architecture designed specifically for semantic segmentation is the Fully Convolutional Network (FCN). This architecture, developed by researchers from UC Berkeley in 2014 [2], has been influential in advancing the field of computer vision, particularly in tasks requiring dense prediction, like semantic segmentation.

Unlike standard convolutional neural networks used for image classification, which typically end with fully connected layers, FCNs are composed entirely of convolutional layers. This design allows them to take input of any size and output segmentation maps that correspond spatially to the input image, providing a per-pixel classification. FCNs transform the fully connected layers found in traditional CNNs (like those in AlexNet or VGGNet) into convolutional layers (Figure 2). This is done by treating the fully connected layers as convolutions with kernels that cover the entire input region. For example a fully connected layer that accepts an input of size  $16 \times 16$  can be reimagined as a convolutional layer with a  $16 \times 16$  filter size. To generate output segmentation maps that match the size of the input image, FCNs use transposed convolution layers (also known as deconvolutional layers) for upsampling. This process helps in recovering the

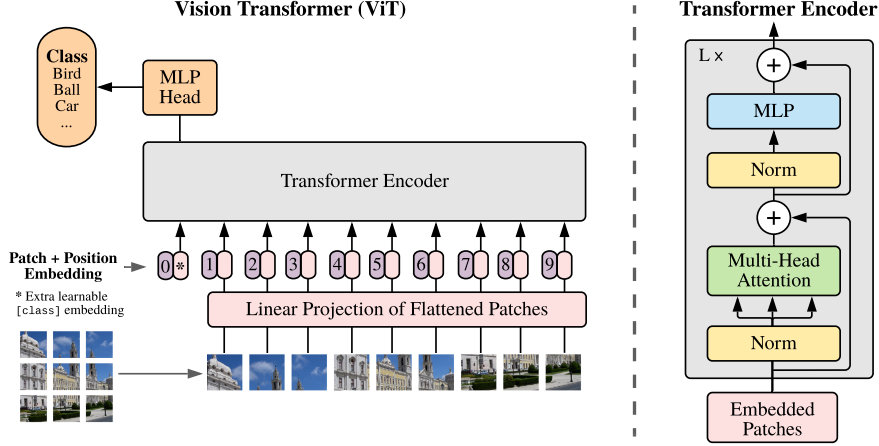


**Figure 3:** Example of the original U-net architecture as presented in [3]

spatial dimensions that are reduced during the pooling or convolutional operations in earlier layers. FCNs often utilize skip connections to combine deep, semantic information from lower layers with the shallow, appearance information in the upper layers of the network. This helps in improving the accuracy and detail of the output segmentation maps, as it allows the network to use both high-level and low-level features.

Further improving upon the architecture of the fully convolutional network, the U-net, first introduced in a paper by Olaf Ronneberger, Philipp Fischer and Thomas Brox in 2015 [3], has become very popular due to its efficiency and effectiveness, especially where data is limited. U-Net’s architecture is distinctive because of its symmetric shape, which resembles the letter “U” (see Figure 3). It consists of two main parts: the left side of the U-net is the contracting path (encoder) which is designed to capture the context of the input image. This path is essentially a typical convolutional network that consists of repeated application of convolutions, followed by rectified linear unit (ReLU) activations, and max pooling for downsampling. Each of these steps helps in capturing the features at different levels of abstraction. On the right side of the network is the expansive path (Decoder), which performs the task of precise localization using transposed convolutions for upsampling. This part of the network gradually recovers the spatial dimensions and detail of the input image. The upsampling in the expansive path is combined with the high-resolution features from the contracting path via skip connections. These connections are crucial as they help in propagating context information to higher resolution layers, which enhances the accuracy of the output segmentation maps. U-Net’s design is particularly effective in providing good segmentation results even with small training datasets, which has led to its widespread adoption and adaptation in different segmentation tasks beyond its initial medical imaging focus.

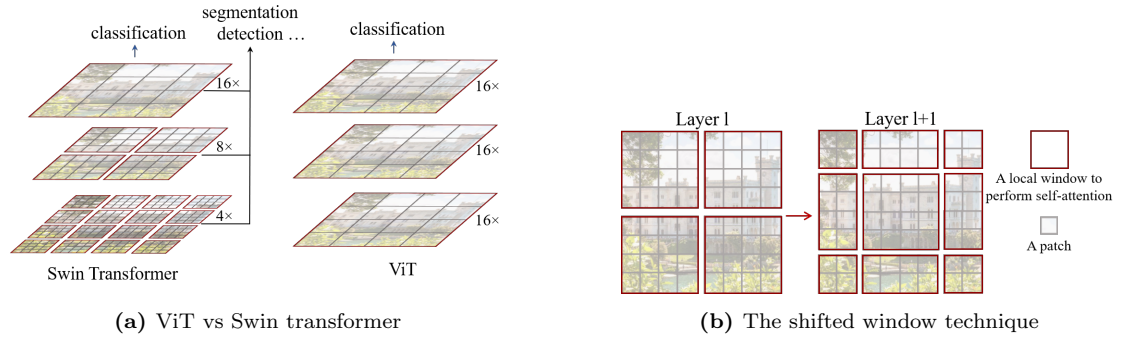
The last architecture we explored in our study was a Vision Transformer (ViT), which is



**Figure 4:** Model overview of a Vision Transformer [3]

a novel approach to applying transformer models, primarily used in Natural Language Processing (NLP) tasks, to image recognition. In the paper “An Image is worth  $16 \times 16$  Words: Transformers for Image Recognition at scale” [4], the authors proposed a method that treats an image as a sequence of fixed-size patches (similar to words in NLP). Each patch is embedded similarly to how tokens are embedded in NLP, and then processed by a standard transformer encoder. An overview of this model is depicted in Figure 4.

Vision transformers can achieve excellent results, outperforming many advanced CNN architectures, particularly when pre-trained on very large datasets. However, the computational complexity especially when applied on high-resolution images increases rapidly, especially for dense prediction tasks like semantic segmentation. For this reason, an improved approach has been suggested called the Swin Transformer model [5]. In contrast to the Vision Transformer that processes the entire image at once by dividing it into fixed-size patches and applying global self-attention to all patches, the Swin transformer divides the image into smaller, non overlapping windows and applies local window-based self-attention (Figure 5(a)). This reduces the computational complexity because the number of interactions is limited to within a window rather than across the entire image. Additionally, the Swin transformer introduces a novel mechanism of shifting these windows in alternate layers, thus enabling cross-window connections. This helps in integrating local and global contextual information more effectively (Figure 5(b)). Finally, the fact that the Swin transformer starts with smaller windows and gradually merges them, reducing resolution while increasing the depth of features, enables a hierarchical architecture that processes features at multiple scales. This is beneficial for various vision tasks like object detection and segmentation, where multi-scale features are crucial. These advantages make the Swin transformer suitable as a general purpose backbone for various vision tasks and as will be explained later, we found this model to perform very well in our particular study.



**Figure 5:** Characteristics of the Swin transformer[5]

## 4 Training Methodology

### 4.1 Transforming and Enhancing Data

In our final runs, the images in the input data are resized to  $768 \times 768$ . This size presents a good balance between model performance and training speed: the final results of the models were almost identical to using the entire size of  $2048 \times 1024$ . Additionally, this new size makes it possible to use pretrained Swin2 in the entire data instead of having to separate the images or stretch it separately.

The loss of the final mask is calculated on the small, resized image. When evaluating, the final mask is stretched to its original larger size.

Additionally, we experimented with creating random transforms of the input data, including random flips and random crops.

In different experiments these transforms either replaced or augmented the original training data. However, this is not present in the final models as their results weren't conclusive: since the original data is complete enough, adding these transforms only made the results on the validation set worse.

### 4.2 Loss function

The choice of loss function is crucial for the performance of our model.

In this assessment, we experimented with three different loss functions.

#### **Categorical Cross-Entropy Loss**

All pixels (except the ones marked as background) are equally weighted.

#### **Intersection over Union Loss**

A differentiable version of IoU score, which should ideally work to maximise it.

#### **Dice Loss**

Categories that appear less often are weighted higher[6].

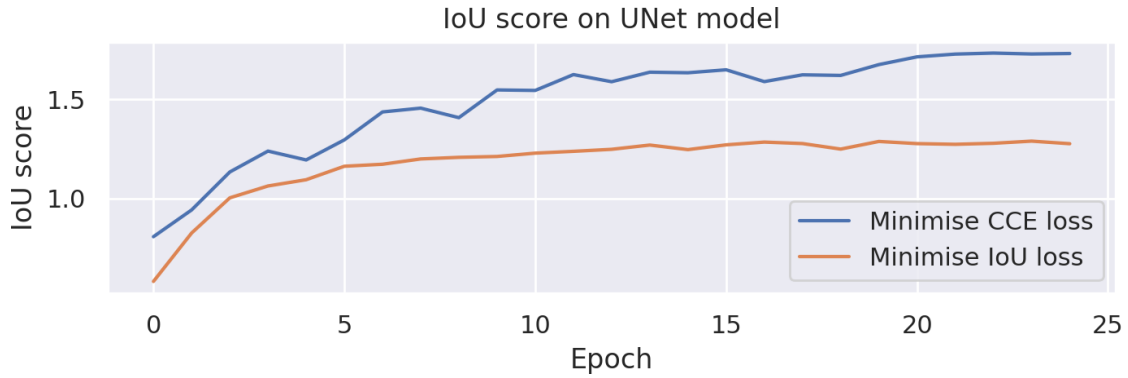
Using **Categorical Cross-Entropy** provided better results in the relevant metrics, including IoU score, when compared to both dice loss and IoU loss.

This surprising result, which can be seen in Figure 6, is likely due to the difference in depth of the loss functions. Since it contradicts some existing literature[6], it warrants more exploration later.

### 4.3 Optimiser and other Hyperparameters

There are many possible hyperparameters to enhance the model.

#### **Loss Function**



**Figure 6:** IoU score for UNet models trained to minimise both categorical cross-entropy and a related IoU loss. Surprisingly, the one that optimises CCE has better results.

## Optimiser

Which optimiser to use to manage the gradient

## 5 Results

## 6 Conclusions

## 7 Reflections

## References

- [1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016.
- [2] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.
- [3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.
- [4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.
- [5] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, 2021.



- [6] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. *Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations*, page 240–248. Springer International Publishing, 2017.