CS 415: Computer Vision

Final project

Author:

Gaetano Coppoletta

Email:

gcoppo2@uic.edu

December, 2022

1 Introduction

Our project is called "Image segmentation: from K-means to deep learning". We explored several techniques for image segmentation, let's start defining what image segmentation is.

Image segmentation is the process of partitioning an image into multiple image segments, also known as image regions or image objects, which are sets of pixels. It is typically used to locate objects and boundaries in images. To be more precise, image segmentation is the process of assigning a label to each pixel in an image in a way that pixels with the same label have the same characteristic. The result of image segmentation is a set of segments that collectively cover the entire image. All the pixels in a region are similar with respect to some characteristic. Some images and their results are listed below as examples.

There are two main approaches for image segmentation: unsupervised and supervised learning. With unsupervised techniques we perform clustering for grouping together pixels with similar features. With supervised, instead, it is possible to inject label information and obtain better results. An example is semantic segmentation, which is an approach detecting, for each pixel the correspondent class of the object.

What was our objective? We started from techniques for image segmentation that we have seen during classes, like k-means and mean-shift, and we tried to move forward in order to include semantic meaning into them. We wanted to understand how different approaches works for image segmentation. We experimented with several deep-learning architectures for the specific task of semantic segmentation.

2 Unsupervised learning

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes. In this section we analyze two unsupervised learning techniques that we have used to try to do image segmentation. This has been done to demonstrate how those techniques are not good for our task. We will discuss the limits of those techniques below.

2.1 K-Means

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. The goal of K-means is to group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number, k, of clusters in a dataset. A cluster is a collection of data points aggregated together because of certain similarities. K-means was the first unsupervised approach that we explored. We have different drawbacks for this approach, first of all there is no semantic understanding, second we need to know the number of cluster in advance. [3] The algorithm has been written in python. We tried the algorithm with different values of K, in particular we tried k=2, k=3, k=5 and k=8. The results obtained are showed below.

As we can see from the images, the results are not good and the choice of K is crucial to obtain a decent output, so this technique is not good for image segmentation.

(a) K=2

(b) K=3

(c) K=5

(d) K=8

Figure 1: Results obtained with K-means algorithm

2.2 Mean-shift

The second unsupervised technique that we explored is the mean shift algorithm. It is a non parametric feature space mathematical analysis technique used to locate the maxima of a density function. Its applications include clustering in computer vision and image processing. Mean shift works by shifting data points towards centroids to be in the mean of other points in the region. An important difference respect to K-means is that mean shift assigns clusters to the data without automatically defining the number of clusters based on defined bandwidth.[2]

We used the sklearn python library to perform mean shift. The result obtained is listed below together with the input image.

As we can see the result is not good. Looking at the sky, for example, we can notice that not every pixel is assigned to the same cluster but we have more than one cluster just for the sky, this is clearly not correct. Mean-shift segmentation does not carry any semantic information

Figure 2: Input image

Figure 3: Result

and rather acts like a smoothing filter, for this reason we need to move forward and try different techniques.

3 Dataset

In our project we have used a reduced version of Cityscapes as dataset. The Cityscapes Dataset focuses on semantic understanding of urban street scenes. The original dataset includes 30 classes, but we have simplified it to obtain only 19 classes, in order to reduce complexity. We have used the images of the dataset with fine-annotated labels, this means that for each pixel in each image we have the correspondent class. The label will be used for the supervised learning algorithms. All the images have been taken in different conditions, this includes diversity to our dataset. In particular the images have been taken in 50 different cities, during several month, all during daytime and with good or medium weather conditions. Some examples of images and the corresponding labels are reported below.

(a) Image 1

(b) Label 1

(c) Image 2

(d) Label 2

4 Supervised Learning

Supervised learning is used for problems where the available data consists of labelled examples, in the sense that each data point contains features and an associated label. The goal of supervised learning algorithms is learning a function that maps feature vectors to labels based on example input-output pairs[6]. The dataset described previously contains both the images and the respective labels.

4.1 Training

To train the deep learning models for semantic segmentation we used the Google Colab platform. We used PyTorch and trained from scratch our models and we performed data augmentation like random cropping, random scaling and horizontal flipping during the training. Each model was trained for 30 epochs and the quality of the models was evaluated in terms of precision and mean intersection over union that is metric-specific for semantic segmentation.

4.2 Loss function

We have used two different loss functions: Mean squared error and cross entropy, MSE was used to train the U-Net but it performed badly as expected so to train the other networks we used only CE. Those two loss function are described below.

- **Mean squared error:** measures the amount of error in statistical models. It assesses the average squared difference between the observed and predicted values. When a model has no error, the MSE equals zero. As model error increases, its value increases. The formula for MSE is reported below.

$$MSE = \frac{\sum (y_i - \widehat{y}_i)^2}{n}$$

Where y_i is the i^{th} observed value, is the corresponding predicted value and n is the number of observation.

- **Cross entropy** measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. The cross entropy can be calculated with the formula below.

$$CE = -\sum_{c=1}^{M} y_{o,c} log(p_{o,c})$$

Where M is the number of classes, log is the natural log, y is the binary indicator, 0 or 1, if class label c is the correct classification for observation o, p is the predicted probability observation o is of class c.

4.3 U-Net

The typical use of convolutional networks is on classification tasks, where the output to an image is a single class label. However, in many visual tasks, the desired output should include localization, i.e., a class label is supposed to be assigned to each pixel. The architecture is reported in Figure 1. [4] The U-Net reported in Figure 1 is a U-shaped encoder-decoder architecture. There are no fully connected layers at the end of the net. So the input image is downsampled by the encoder applying convolution, the activation function used is ReLu, after the encoder we have a bottleneck and then then the decoder upsamples the image by concatenating the results of transposed convolutions to the one of skip connections. We trained the neural network with both mean squared error and cross entropy error. This network is usually trained with CE, but for an experimental purpose we also tried the training with MSE.

As expected, the MSE task is more difficult and the training procedure appeared to be unstable, so it did not produce good results. With MSE we produce only one output, which is unbounded so we need to bring it into the range [0,18].

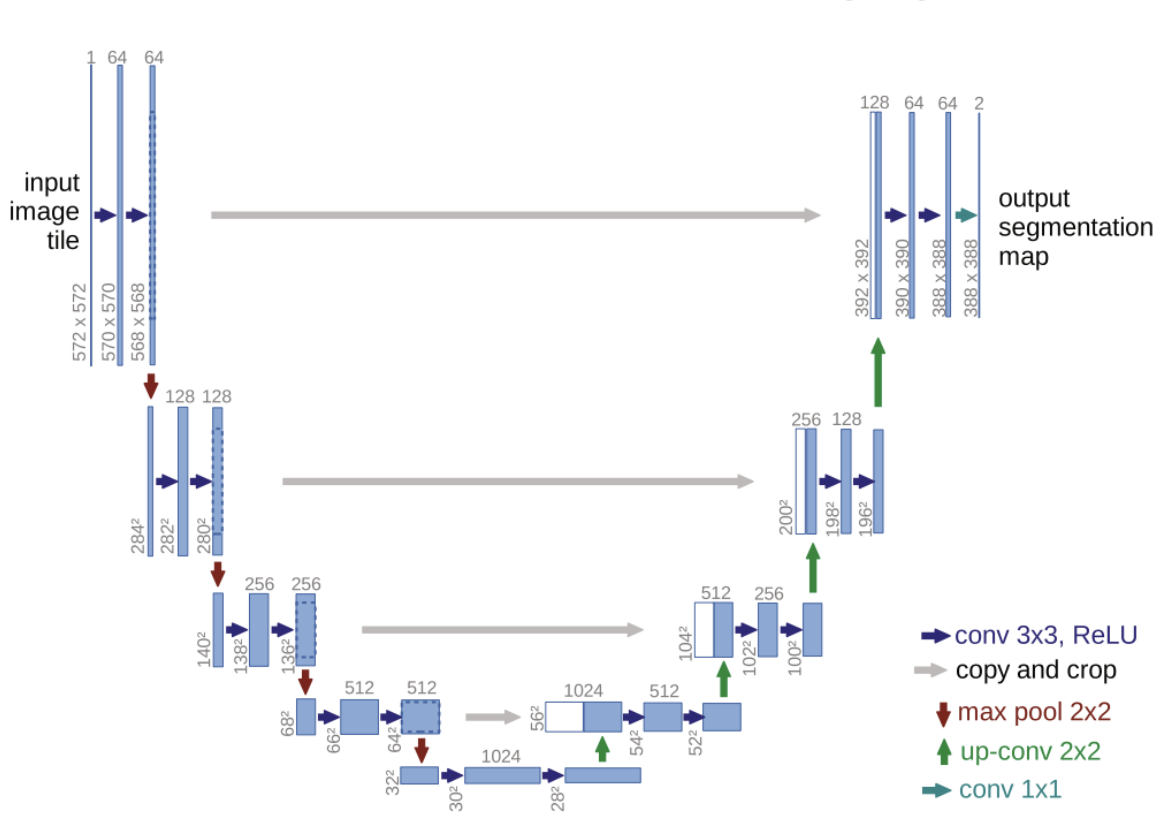


Figure 5: U-Net architecture

With CE the output of our model has 19 channels, one for each label, output i represent the probability that the pixel belongs to the i^{th} class, So we apply the softmax and then we take the maximum with argmax and we assign the class. With cross entropy loss instead our model produced reasonable results. The results are listed below.

4.4 Dilated U-Net

The Dilated U-Net is a modified version of the U-Net in which we appliad dilated (or atrous) convolution instead of performing the normal convolution. In formula we can write:

$$(F * k)(p) = \sum_{s+t=p} F(s)k(t)$$

$$(F *_l k)(p) = \sum_{s+lt=p} F(s)k(t)$$

The first formula is the standard convolution, while the second one is the dilated convolution. We can see that at the summation, it is $s+lt = p$ that we will skip some points during convolution. If l=1 it is a standard convolution, if l>1 it is a dilated convolution.[5] The main impact that

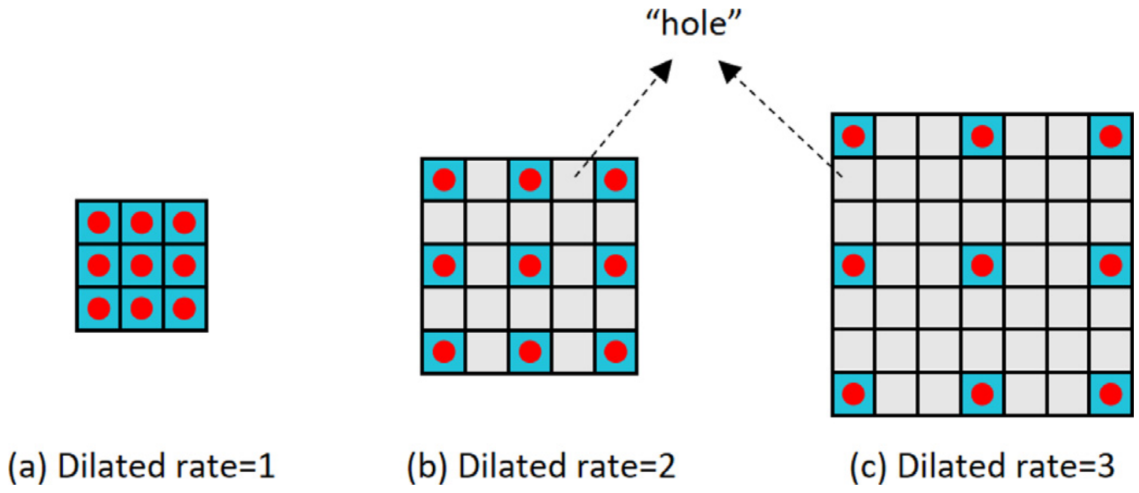


Figure 6: Dilated convolution

the dilated convolution has is that it enlarges the field of view, maintaining the number of parameters constant, so we obtain better performance but the same computational cost. Our dilated U-Net performs dilated convolution in both the encoding and the decoding stage. During the encoding l is progressively decreased from 16 to 1, while during the decoding phase l is increased progressively from 1 to 16. This is illustrated more in details in 1

Layer	1	2	3	4	5	6	7	8
Convolution	3×3	3×3	3×3	3×3	3×3	3×3	3×3	3×3
Dilated rate	1	1	2	4	8	16	1	1
Receptive field	3×3	5×5	9×9	17×17	33×33	65×65	67×67	69×69

Table 1: Dilated convolution schema used during the decoding phase. In the encoding phase, the schema is specular.

We trained the dilated U-Net only with the cross entropy loss given that it performed best with the simple U-Net.

4.5 BiSeNet

Another architecture that we tried was the Bilateral Segmentation Network [1]. This compact model has two branches: a spatial path and a context path, we will now explain the details of those two paths.

- **Spatial path:** Spatial information and the receptive field are crucial to achieving high accuracy. However, it is hard to meet these two demands simultaneously. The spatial path is used to preserve the spatial size of the original input image and encode affluent spatial information. The Spatial Path contains three layers. Each layer includes a convolution with stride = 2, followed by batch normalization and ReLU. Therefore, this path extracts the output feature maps that is 1/8 of the original image. It encodes rich spatial information due to the large spatial size of feature maps.

- **Context path:** the context path is designed to provide sufficient receptive field. The context path utilizes lightweight model and global average pooling to provide large receptive field. The context path utilizes lightweight model and global average pooling to provide large receptive field. In this work, the lightweight model can downsample the feature map fast to obtain large receptive field, which encodes high level semantic context information. Then there is a global average pooling on the tail of the lightweight model, which can provide the maximum receptive field with global context information. Finally, it is combined with the up-sampled output feature of global pooling and the features of the lightweight model. In the lightweight model, it is deployed U-shape structure to fuse the features of the last two stages, which is an incomplete U-shape style.

The Feature Fusion Module (FFM) is used to merge the two feature maps coming from the spatial path and the context path. The architecture is shown in 8. Different from the architectures described previously, BiSeNet is basen on a ResNet that is pretrained on ImageNet, this makes its prediction much better even if it was trained for the same number of epochs as the other models.

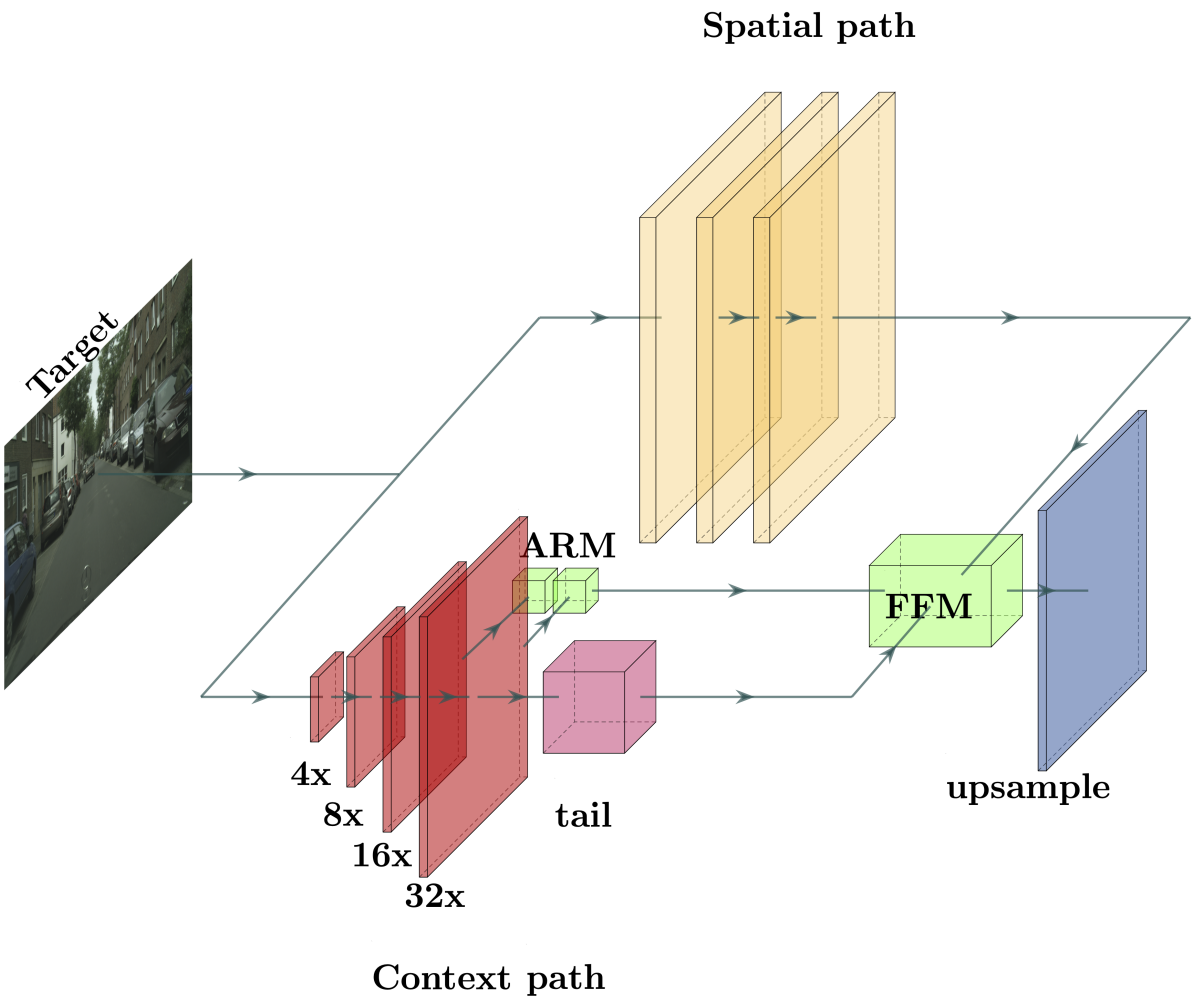


Figure 7: BiSeNet architecture

4.6 Results

In this section we describe the results of the architectures described before. First we show the evolution of the loss, mIoU and precision over 30 epochs. The U-Net with MSE model is not reported because its values are out of scale.

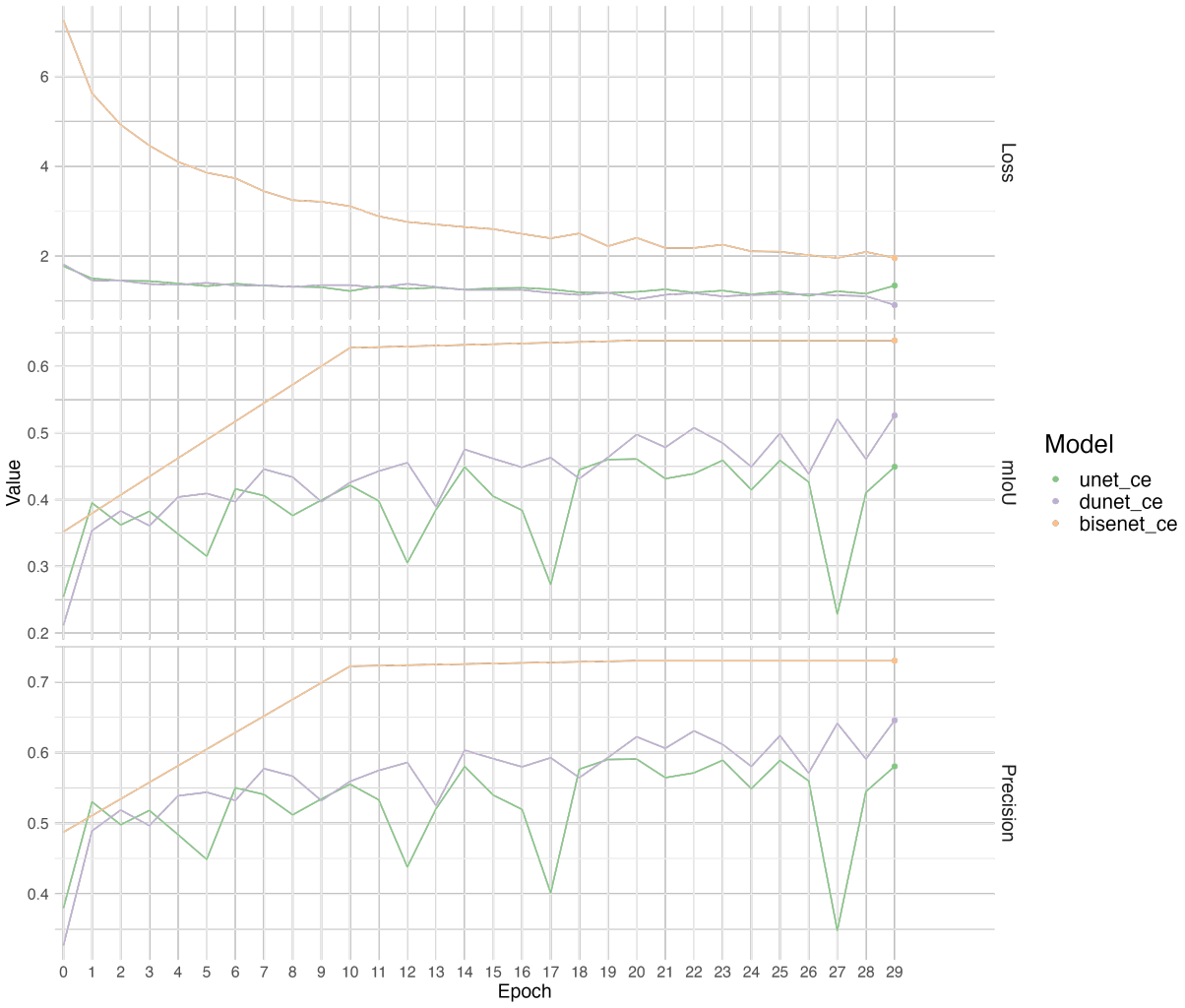


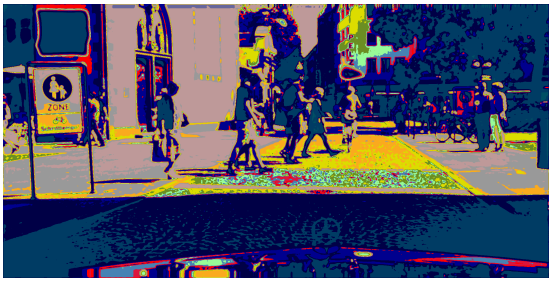
Figure 8: Loss, mIoU and precision over epochs

4.6.1 U-Net

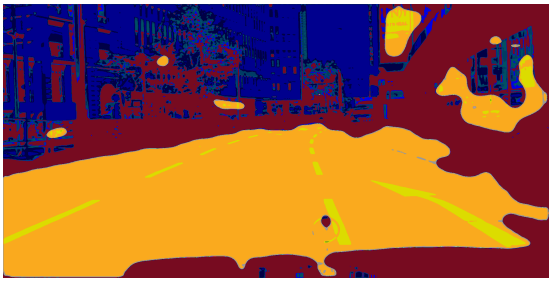
As expected, the MSE task is more difficult and the training procedure appeared to be unstable, so it did not produce good results, precision and mIoU are reported in 2. Instead, our CE model produced reasonable results. Some results are listed below. For more results see the appendix.

Model	Precision	mIoU
U-NET (MSE)	0.326	0.211
U-NET (CE)	0.591	0.461

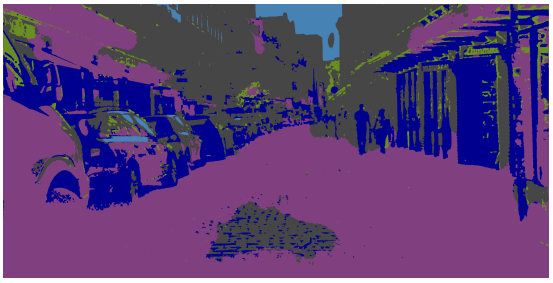
Table 2



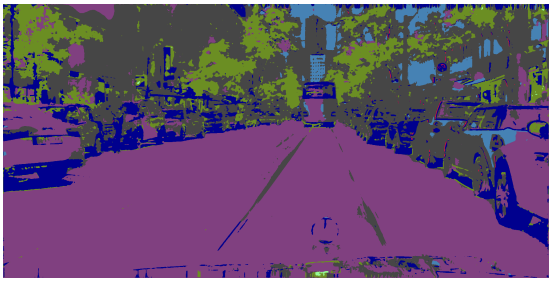
(a) MSE result 1



(b) MSE result 2



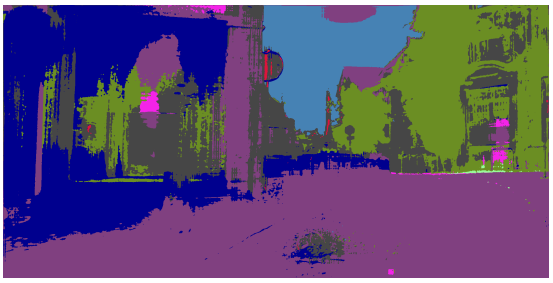
(c) CE result 1



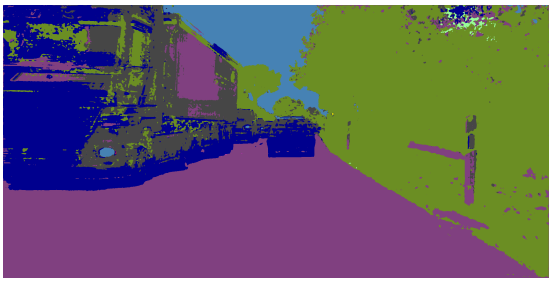
(d) CE result 2

4.6.2 Dilated U-Net

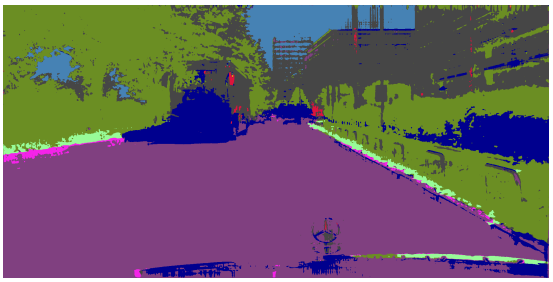
Looking at the result we notice that the dilated U-Net performed betted than the simple U-Net both in term of precision and mIoU. In particular we obtained a precision of 0.591 and a mIoU of 0.461. Some results are reported below. For more results see the appendix.



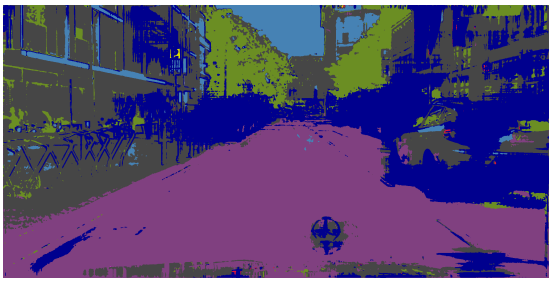
(a) Dilated U-Net result 1



(b) Dilated U-Net result 2



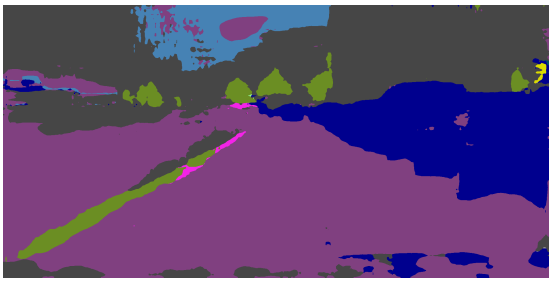
(c) Dilated U-Net result 3



(d) Dilated U-Net 4

4.6.3 BiSeNet

The BiSeNet is based on a ResNet that was pre-trained on ImageNet, for this reason it obtains better results. In particular we have obtained a precision of 0.730 and a mIoU of 0.639. Some results are listed below. For more results see the appendix.



(a) BiSeNet 1

(b) BiSeNet result 2

5 Conclusion

In this section I will discuss some observation about the work done in order to do this project. Image segmentation has improved a lot since the introduction of deep learning techniques. As we

have seen from the section about unsupervised learning, those techniques are not able to obtain results comparable to the ones obtained with deep learning techniques. From this project I have had the opportunity to learn more about neural networks and how they works, in particular was interesting trying to understand why different loss function causes so much difference in the performance. We have learned how some architecture works and we have implemented them from scratch, using the paper as reference. The BiSeNet is the architecture that has the best performance, this is because it has been pretrained on ImageNet. What can be do to improve the results? We can train the network for more epochs and see how better the model could be. As further work I would like to study other architectures that are able to perform image segmentation. Another topic that I would like to study is panoptic segmentation.

6 Appendix

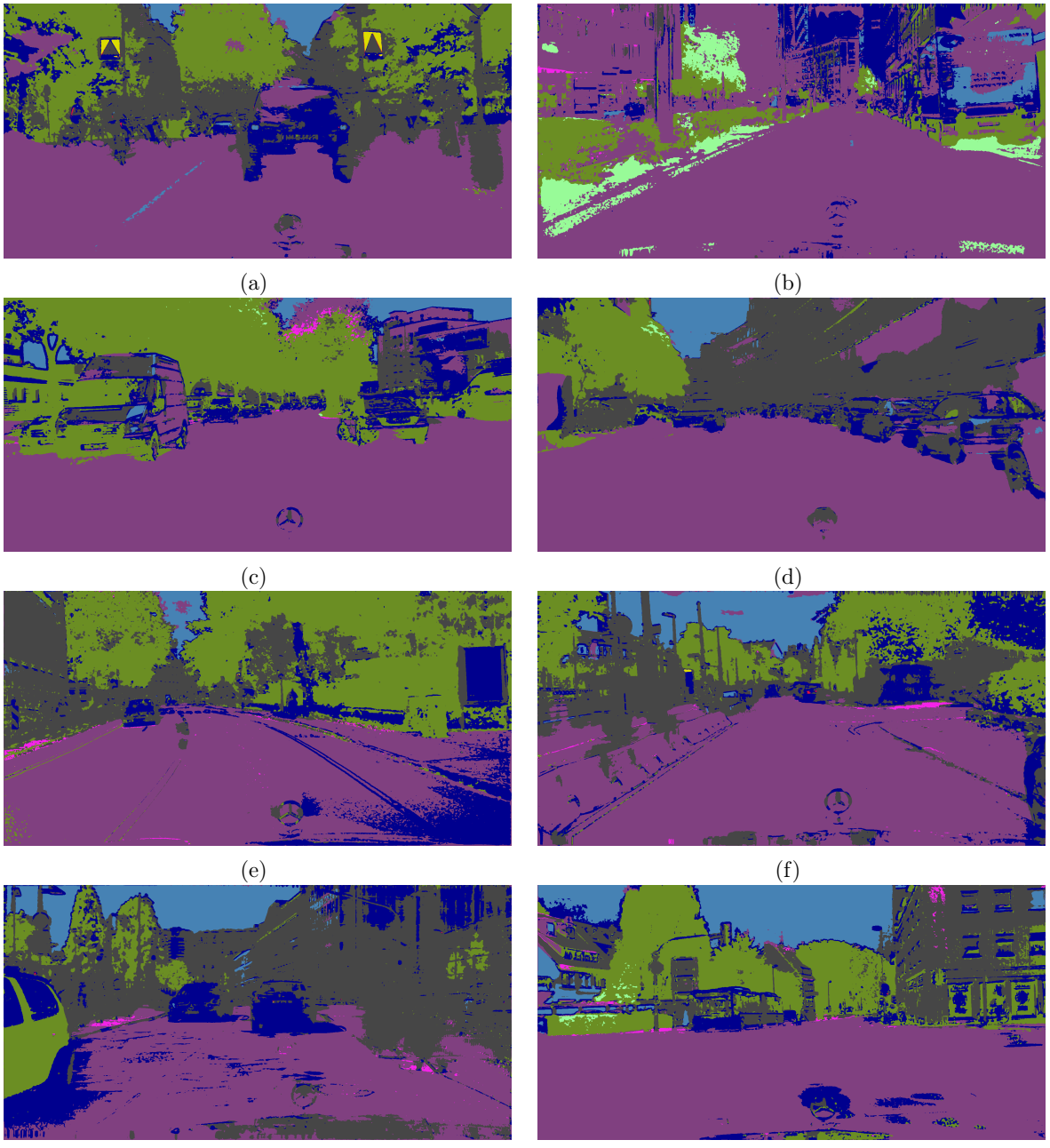
In this section we report some results obtained with the various network.

A Appendix: Additional results U-NET (MSE Loss)

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 12: Additional U-NET with mean squared error loss predicted segmentation maps.

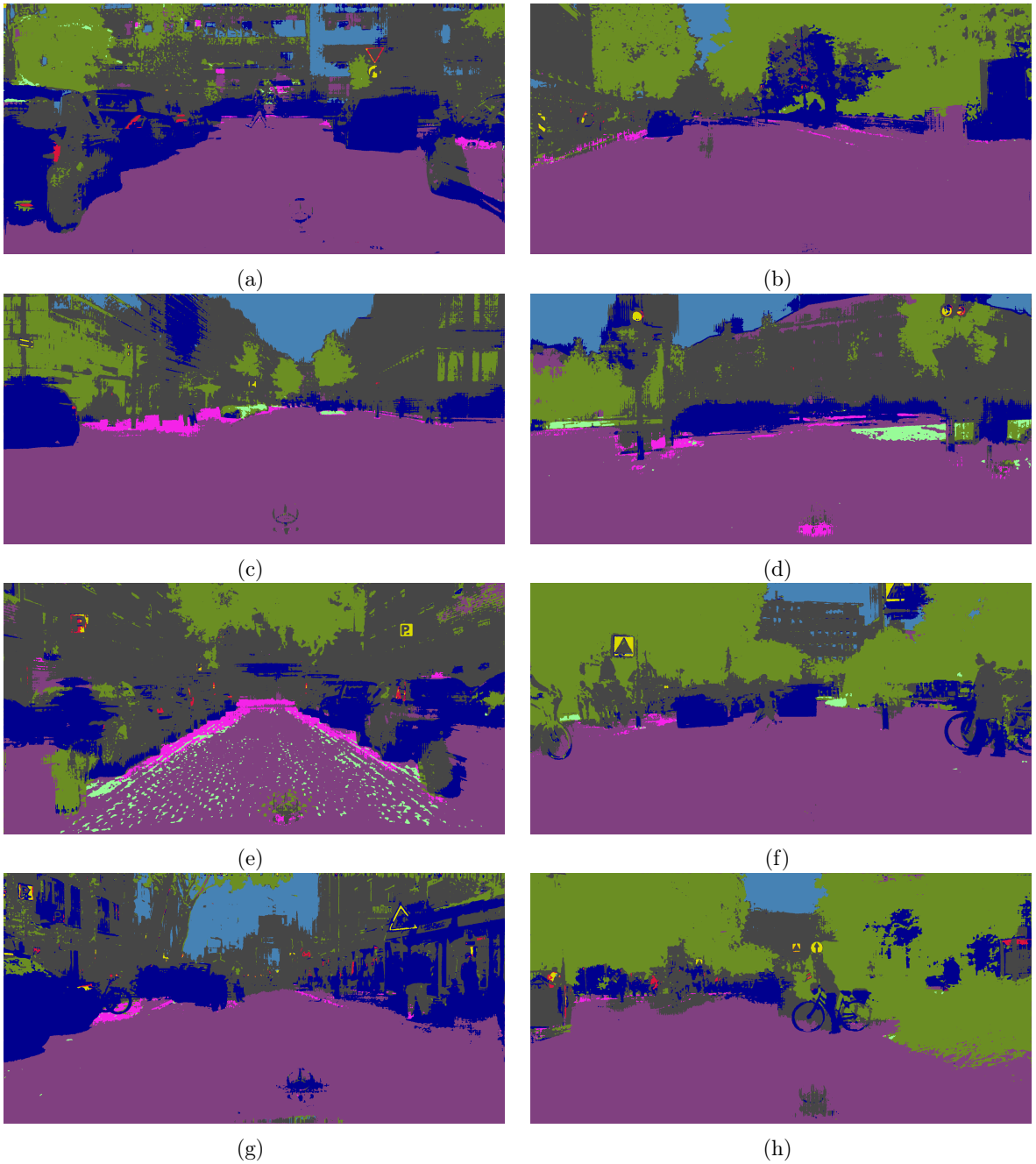
B Appendix: Additional results U-NET (CE Loss)



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 13: Additional U-NET with cross-entropy loss predicted segmentation maps.

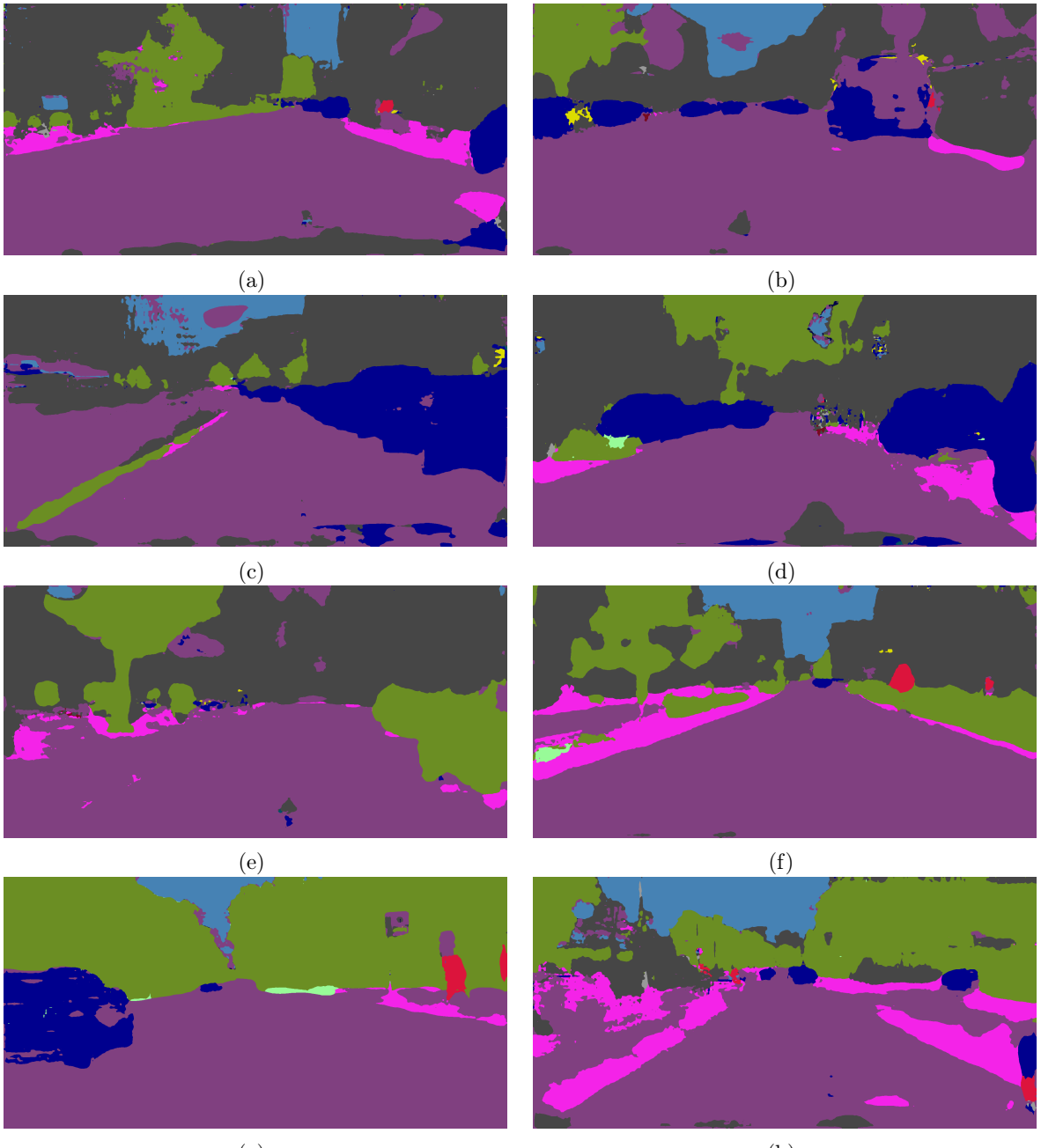
C Appendix: Additional results Dilated-UNET



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 14: Additional Dilated-UNET with cross-entropy loss predicted segmentation maps.

D Appendix: Additional results BiSeNet



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 15: Additional BiSeNet with cross-entropy loss predicted segmentation maps.

References

[1] Chao Peng Changxin Gao Gang Yu Nong Sang Changqian Yu, Jingbo Wang. Bisenet: Bilateral segmentation network for real-time semantic segmentation.

[2] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1197–1203 vol.2, 1999.

[3] Nameirakpam Dhanachandra, Khumanthem Manglem, and Yambem Jina Chanu. Image segmentation using k -means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54:764–771, 2015. Eleventh International Conference on Communication Networks, ICCN 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Data Mining and Warehousing, ICDMW 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Image and Signal Processing, ICISP 2015, August 21-23, 2015, Bangalore, India.

[4] Philipp Fischer Olaf Ronneberger and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. 2015.

[5] Sik-Ho Tsang. Review: Dilatednet — dilated convolution (semantic segmentation). 2018.

[6] Wikipedia. Supervised learning.