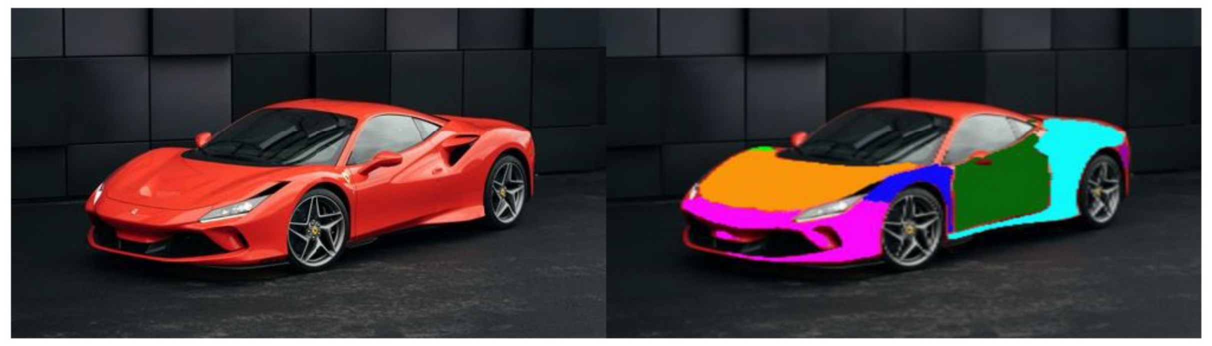
Image Segmentation of Car Parts

Antarlina Mukherjee (s210142), Felipe Olivos (s220050), Oliver Norborg (s174030)
DTU Compute, Technical University of Denmark
Deloitte

Introduction and Dataset



Semantic segmentation of car parts when provided with a car image.

Image Size
- 256x256.

Count of Images:
- 168 'real' images
- 3323 images with augmentation
- 30 images for the test data

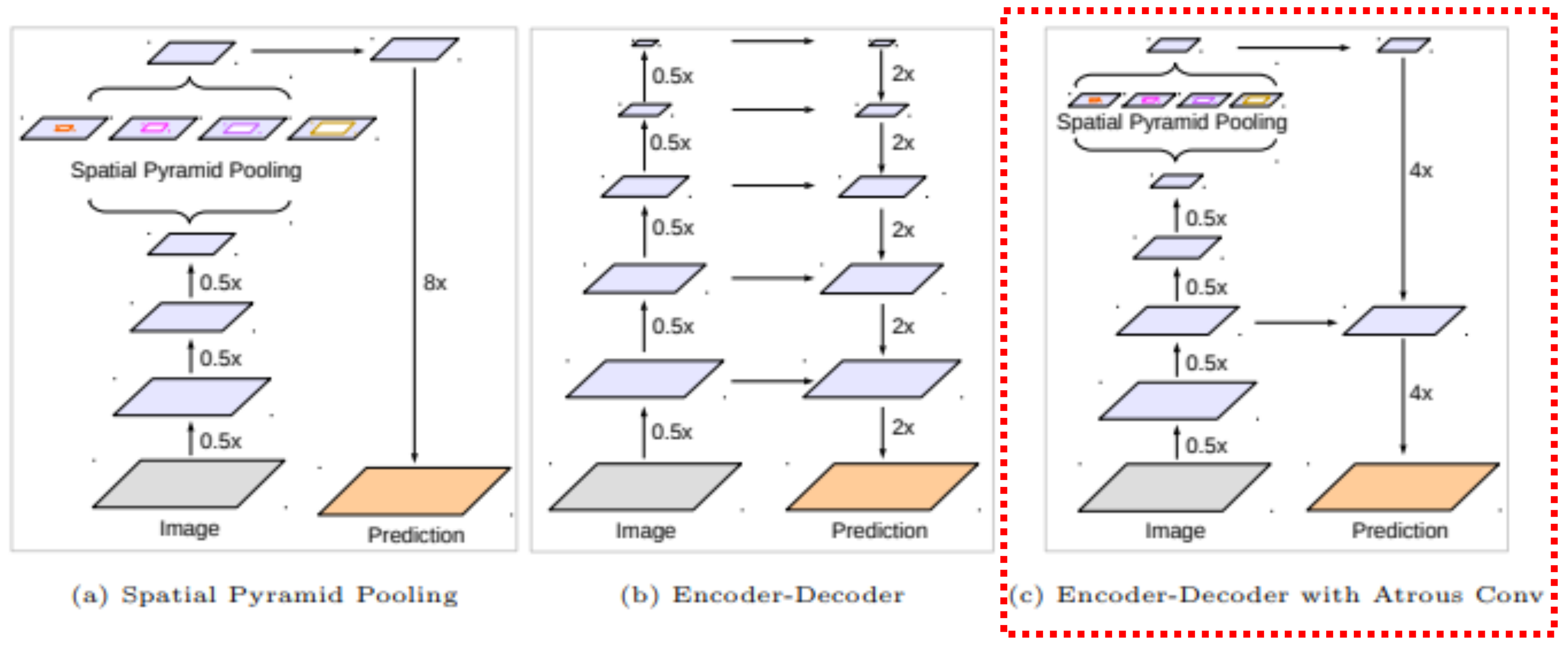
Salient features of the problem
- Each pixel needs to be classified
- The objects are multi scale
- Identify the intricate characteristics of the object (body) and where is it located (position and boundary)

Motivation
- Leveraging pre- trained models for better weight initialization, faster training, and better results
- Observing the importance of image augmentation and its impact on training accuracy and speed
- Experimenting with different architectures and observing the efficiency and effectiveness

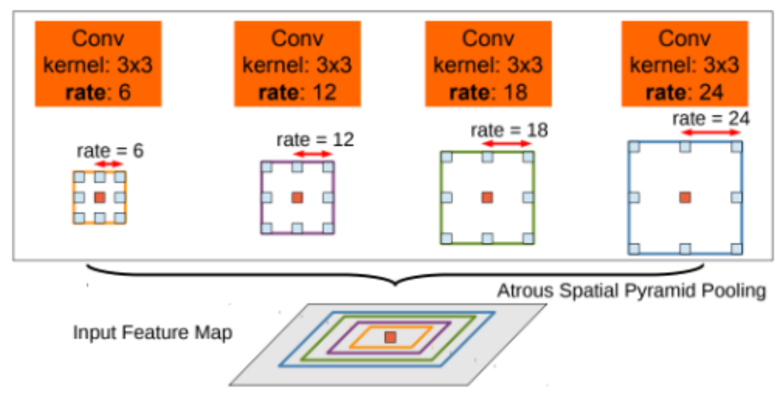
References

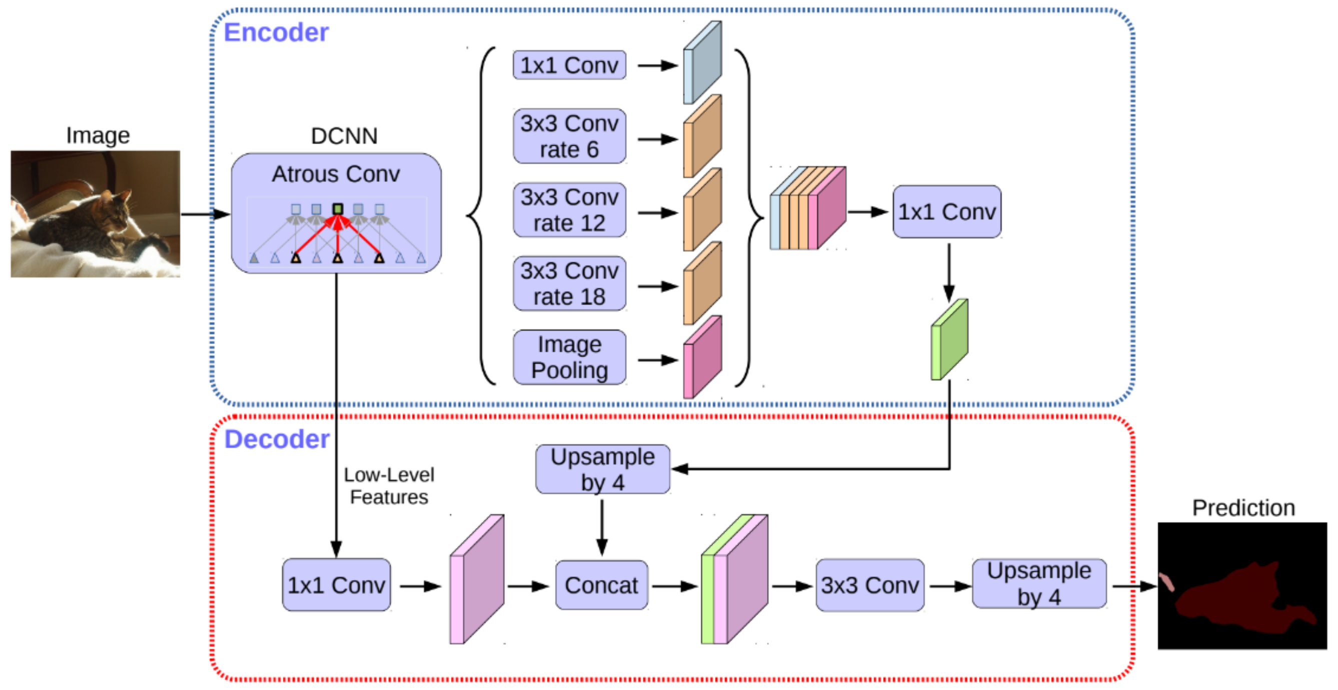
- Liang-Chieh Chen et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: ECCV. 2018.
- Liang-Chieh Chen et al. "Rethinking atrous convolution for semantic image segmentation". In: arXiv:1706.05587 (2017)
- Ronneberger Olaf et al. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: arXiv:1505.04597v1 (2015)

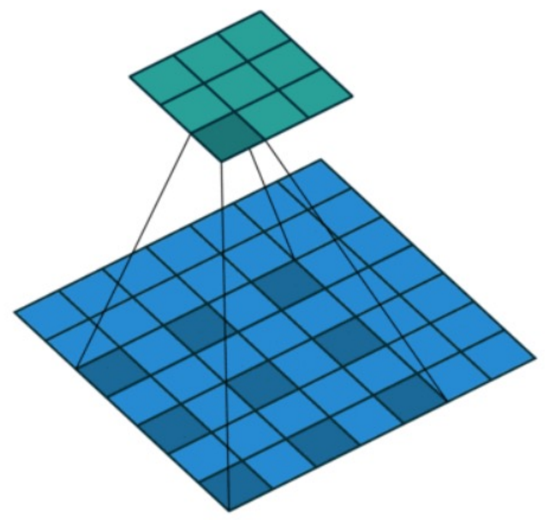
Architecture



Spatial Pyramid Pooling: Down sampling → Pyramid Pooling
- Resampling a feature layer at various rates (that determines the holes in the filter) and capturing information from **different fields of view** thanks to the **atrous convolution**.
- Capturing useful image context at various scales.







Encoder
Encode multi-scale contextual information by performing filters in multiple **fields-of-view**.
The last feature map (high-level semantic information) is used in the encoder-decoder approach (DeepLabv3+)

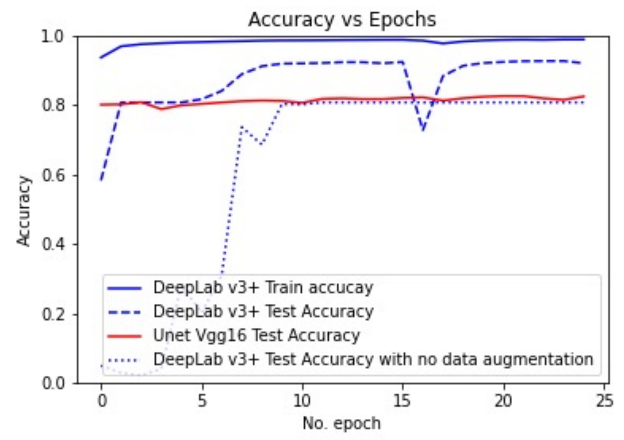
Decoder
Encoder features are **concatenated** with the corresponding low-level features from the network backbone that have the same spatial resolution.
Works by capturing more distinct boundaries of images.

Results

Training
- Using resnet50 instead of resnet101 for faster training and less memory use
- Adding an extra convolutional block to the image pooling
- Evaluation was based on the f1-score and trying to increase that score. F1 is a measure of performance of classification
- F1-Score = $2 * \frac{Recall * Precision}{Recall + Precision}$
- Loss function is a SparseCategoricalCrossentrop - Computes the crossentropy loss between the labels and predictions
- Using an Adams optimizer with a learning rate of 0.001

Architecture	F1-Score	Computational Time for 25 Epoch*	Trainable parameters
UNET	0.283	10 min 25 sec	18,548,105
Deeplabv3+	0.615	5 min 50 sec	11,821,673

*Computational Resourses: Google Colab - Intel Xeon CPU@2.20GHz, 13GB RAM



Conclusion and Future Work

Conclusion
- Deeplabv3+ is a better (higher accuracy and f1 score, less trainable parameters) and faster choice than the UNET architecture in this scenario
- A "heavier" architecture doesn't ensure better performance, but an efficient model does

Future Works
- Investigate other pretrained weights such as resnet101 or resnet152
- Test different values for learning rate, as well as testing different optimizers and loss functions
- Get more 'real' data