Intel® oneAPI Hackathon 2023

Team Name:

Single Londe

Problem Statement:

Object Detection for Autonomous Vehicles

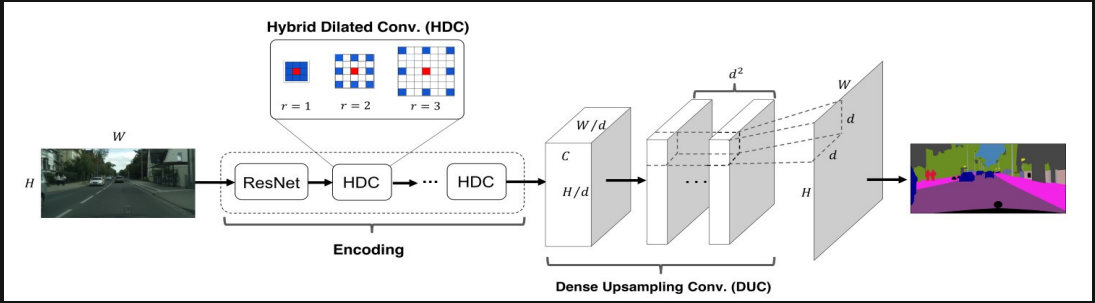
CitySegNet

CitySegNet is a **deep learning prototype** designed for semantic image segmentation on the **Cityscapes dataset.** The Cityscapes dataset is a popular benchmark for urban scene understanding, consisting of high-resolution images captured from various cities. The goal of **semantic image segmentation** is to assign a class label to each pixel in an image, enabling fine-grained understanding of urban scenes.

The **Cityscapes Dataset** is a large dataset that focuses on semantic understanding of urban street scenes. The dataset contains **5000 images with fine annotations** across 50 cities, different seasons, varying scene layout and background. The dataset is annotated with **30 categories**, of which 19 categories are included for training and evaluation (others are ignored)

Architecture

We use **DUC (Dense upsampling convolution**) which is a **CNN based model** for semantic segmentation which uses an image classification network **(ResNet) as a backend** and achieves improved accuracy in terms of mIOU score using two novel techniques. The first technique is called **Dense Upsampling Convolution** (DUC) which generates pixel-level prediction by capturing and decoding more detailed information that is generally missing in bilinear upsampling. Secondly, a framework called **Hybrid Dilated Convolution (HDC**) is proposed in the encoding phase which enlarges the receptive fields of the network to aggregate global information. It also alleviates the checkerboard receptive field problem ("gridding") caused by the standard dilated convolution operation.



Architecture contd.

Network	DS	ASPP	Augmentation	Cell	mIoU
Baseline	8	4	*yes*	*n/a*	72.3
Baseline	4	4	*yes*	*n/a*	70.9
DUC	8	*no*	*no*	1	71.9
DUC	8	4	*no*	1	72.8
DUC	8	4	*yes*	1	74.3
DUC	4	4	*yes*	1	73.7
DUC	8	6	*yes*	1	74.5
DUC	8	6	*yes*	2	74.7

Table 1. Ablation studies for applying ResNet-101 on the Cityscapes dataset. **DS**: Downsampling rate of the network. **Cell**: neighborhood region that one predicted pixel represents.

- We used **the DeepLab-V2 [3] ResNet-101** framework to train our baseline model.
- The network has a downsampling rate of 8, and dilated convolution with rate of 2 and 4 are applied to res4b and res5b blocks, respectively.
- The prediction maps and training labels are **downsampled by a factor of 8 compared** to the size of original images, and bilinear upsampling is used to get the final prediction.
- The image size in the Cityscapes dataset is 1024 × 2048, which is too big to fit in the GPU memory, we partition each image into twelve **800×800 patches** with partial overlapping,

We train the **ResNet-DUC network** the same way as the baseline model for **20 epochs**, and achieve a mean IOU of **74.3%** on the validation set, a 2% increase compared to the baseline model.

Result

mean Intersection Over Union (mIOU) is the metric used for validation. For each class the intersection over union (IOU) of pixel labels between the output and the target segmentation maps is computed and then **averaged over all classes** to give us the mean intersection over union (mIOU).

```
In [7]: print("mean Intersection Over Union (mIOU): {}".format(metric.get()[1]))

mean Intersection Over Union (mIOU): 0.819220680835
```

Tech Stack

Intel® Neural Compressor
An open-source Python library supporting popular model compression techniques on all mainstream deep learning frameworks (TensorFlow, PyTorch, ONNX Runtime, and MXNet).

Intel® Extension for PyTorch*
Intel® Extension for PyTorch* extends PyTorch* with up-to-date features optimizations for an extra performance boost on Intel hardware.

Processing

Pre-Processing

The **DUC (Dense Upsampling Convolution)** layer divides the image **into d^2 subparts**, where d represents the downsampling rate. To ensure accurate reshaping of the image after passing through the DUC layer, a small border is added to the input image. This extrapolation helps maintain the integrity of the reshaped image. Following this step, the image undergoes normalization through mean subtraction.

Post-Processing

The output tensor is reshaped and **resized to give the softmax map** of shape **(H X W X label_num).** The raw label map is computed by doing an argmax on the softmax map. The **script cityscapes_labels.py** contains the **segmentation category labels** and their corresponding color map. Using this the colorized segmented images are generated.

References

- All models are from the paper Understanding Convolution for Semantic Segmentation.
- TuSimple-DUC repo, MXNet
- Intel® Neural Compressor
- DUC github object-detection-segmentation by jcwchen