TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Hierarchical Multi-Domain Semantic Segmentation with Heterogeneous Labels

MasterThesis

by

Pawnesh Gautam

December 23, 2022

Technische Universität Kaiserslautern,
Department of Computer Science,
67663 Kaiserslautern,
Germany

Examiner: Prof. Dr. Didlier Stricker
Dr.-Ing. René Schuster

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die von mir vorgelegte Arbeit mit dem Thema "Hierarchical Multi-Domain Semantic Segmentation with Heterogeneous Labels" selbstständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Kaiserslautern, den 23.12.2022

_____

Pawnesh Gautam

Abstract

Most semantic segmentation methods usually lack the generality across datasets and do not integrate symbolic artificial intelligence into deep learning methods. The first limitation is usually due to the fact that network is trained on a single limited sample dataset and unavailiability of one large dataset. In this work, multiple heterogeneous datasets are used to train the networks and their performance on the datasets are investigated. The second drawback is overcome by inheriting the relationships of object categories into the network structure through multi-level segmentation heads known as hierarchical network. In this work, the hierarchical semantic segmentation network predict categories of different levels in the hierarchy and aggregate the information into the fine-grained output mask. The rich associations of various categories are encoded in network through a label hierarchy based on human cognition and pixel embedding. Extensive experiments over Cityscapes, Mapillary Vistas, VIPER, ADE20K, Wilddash and Scannet dataset, show that the performance of the cross-domain network on ADE20K, Vistas v1.0, Cityscapes, VIPER, Scannet and Wilddash improves by 11.22%, 8.2%, 5.99%, 5.83%, 8.88%, 9.55% over the baseline network, respectively. In addition, for well-structured datasets, hierarchical networks based on hand-picked label hierarchy further improve the results by 3.08% for Vistas v1.0 and 0.89% for Cityscapes over cross-domain network; for unstructured datasets, hierarchical networks based on pixel-embedded label hierarchy underperform cross-domain networks by 5% for ADE20K, 1.62% for Scannet, and 0.18% for Wilddash.

Contents

List of Figures

List of Tables

1. Introduction

1.1. Semantic Segmentation

The goal of semantic segmentation is to classify each pixel of an input image based on semantic information and to predict the semantic class of each pixel in a given set of labels, as shown in Figure 1.1.

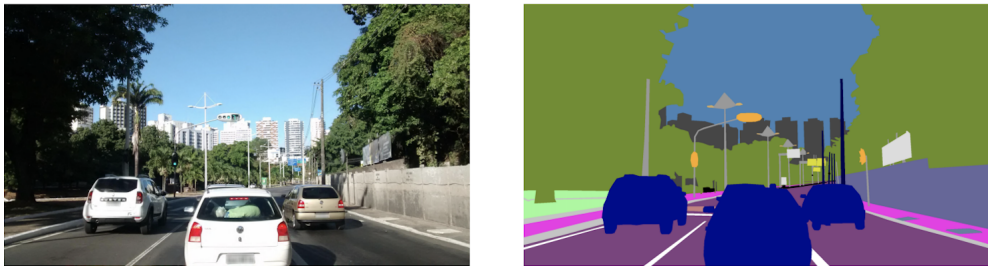


Figure 1.1.: *Semantic segmentation of an image [Neu+17].*

Semantic segmentation is receiving increasing attention from computer vision and machine learning researchers, as it is one of the advanced tasks that pave the way for complete scene understanding. It is also an essential data processing step for robots and other unmanned systems to understand the surrounding scenes. Many applications, such as augmented reality, autonomous driving, and video surveillance, medical image analysis are emerging that require accurate and efficient segmentation mechanisms. Semantic segmentation is also being utilised in scene understanding, human-machine interaction, computational photography, image search engines, and other areas.

In the past there were many computer vision and machine learning techniques that used symbolic models e.g. random forests and conditional random fields to build classifiers for semantic learning. It soon became clear that these methods required a lot of manual work and lacked real learning. Since the deep learning method CNN showed its superiority in the image network competition. The deep learning revolution turned the tide and gave rise to a new generation of segmentation models that have significantly improved their performance and have become the dominant solution for semantic segmentation. Deep learning-based approaches have also become the architecture behind the vast majority of recent successful artificial intelligence systems.

1.2. The Challenges

Although recent transformers [Vas+17] based methods have shown reasonable results, they still can not able to beat human-level performance in real-world

environment. Humans segment the objects in very less time, even if the object is changed geometrically or non-geometrically. Semantic segmentation is still an unsolved problem in computer vision. On the other hand, the performance of machine learning and computer vision algorithms becomes more and more challenging for semantic segmentation as image content becomes more and more complex.

In order to achieve the state-of-art accuracy for deep learning based semantic segmentation approaches, pixel-level annotated images are required, which are limited for many applications or even not available. Models trained on a limited set of images may achieve satisfactory performance, but do not generalize well to other images in real environments. Therefore, deep learning-based approaches require a large and balanced dataset of pixel-level annotations. Next, the main challenges of the algorithm are inference rate and accuracy. The algorithm needs low inference rate and high accuracy so that it can be used in real-time applications. For these reasons, semantic segmentation remains a hot topic in research.

1.3. Objectives and Contributions

One of the goals of this work is to design a single model for semantic segmentation of image-based road and indoor datasets. For this purpose, several datasets are selected to obtain a large dataset, which is further used for training and evaluating the model.

The next goal is to design a hierarchical semantic segmentation model based on the current state-of-the-art approach. The idea behind this approach is to combine symbolic and deep learning based artificial intelligence, such as exploiting the correlation between categories in a deep learning approach.

Finally, an automatic label hierarchy are proposed for the hierarchical network and compared with the hand-picked label hierarchy. The motivation for this approach is to avoid the manual work of constructing the label hierarchy and to view the label hierarchy from a network perspective.

1.4. Thesis Structure

The structure of this thesis is as follows: Chapter 2 aims at providing an overview of semantic segmentation research for supervised deep learning methods. Chapter 3 contains an overview of the dataset and detailed descriptions of the baseline and proposed methods. Chapter 4 describes the technology stack for experimentation and captures the performance of various experiments. It also includes the qualitative and quantitative analysis of the results. To conclude, Chapter 5 reviews the results obtained from this work and highlights some areas of improvement that can be explored in the future.

2. Related Work

For the semantic segmentation task, researchers have been working on this problem since the advent of digital images. The development started from machine learning and heuristic approaches to segment the each pixel in an image. Before the deep neural networks, a considerable number of algorithms have been designed to solve this non-trivial task, such as Watershed algorithm [VS91], Image thresholding [Ots79], K-means clustering [Llo82], Conditional Random fields [SM+12], etc. However, object occlusion usually leads to poor performance for methods based on hand-crafted features. With the advent of deep learning methods, deep learning outperforms hand-crafted techniques across the board for most image-related problems.

For the semantic segmentation, fully convolutional networks (FCNs) [LSD15] show the significant improvement. Especially, the Encoder-decoder based neural networks proven to work well for the semantic segmentation. In recent years, there have been significant advances in semantic segmentation, so it is not possible to capture every aspect and subfield of semantic segmentation. Instead, this section briefly describes the main breakthroughs in recent years in deep learning-based methods for semantic segmentation of images. In addition to standard semantic segmentation methods, deep learning methods based on the label hierarchy are discussed.

2.1. Standard Semantic Segmentation

In recent years, CNNs have became the most successful methods for visual recognition tasks, such as image classification, object detection and semantic segmentation. For semantic segmentation, the earliest deep learning based method is region proposal based approach [Gir+14; Har+14]. However, significant accuracy in semantic segmentation mask prediction has been obtained by using FCNs [LSD15]. FCNs improved the feature representation by providing high level features, which resulted in performance boost. However, the multilayer pooling operation leads to a reduction in the feature map size, which poses a challenge in upsampling the segmentation output to original resolution and recognizing multi-scale objects in the scene.

To overcome aforementioned limitations, a number of research works have been proposed to improve the resolution of feature maps and the receptive field of neurons. Dai et al. proposed the adaptive receptive field using deformable convolution [Dai+17b]. It adds two-dimensional learnable offsets to the regular grid sampling locations in the regular convolution. Another approach is to use dilated convolution [YK15] and deconvolution [NHH15] to maintain the resolution of large feature maps. Chen et al. uses the dilated convolution in

atrous convolution [Che+14] to have an effective fields-of-view without losing the spatial resolution. Further, the contextual information is improved through Atrous Spatial Pyramid Pooling (ASPP) [Che+17; Yan+18], fixed and adaptive spatial pyramid pooling methods [Zha+17; Lin+17; He+19a] by using multi-scale feature fusion. These efforts expand the receptive field by controlling the resolution of the feature maps.

Moreover, in order to retain more detailed information, boundary aware methods [Din+19; Li+20; Yua+20] and nonlinear upsampling techniques using Encoder-Decoder architectures [Che+18a; BKC17; RFB15] are proposed. To further improve the contextual information, self-attention mechanisms [Fu+19; HDK17; Hua+19b; Li+18; Zha+18; Wan+18; YCW20; Hua+19a; Li+19; Zhu+19] are introduced to capture information from all spatial locations, which assist to capture different scales objects along with their long range relations.

Recently, many advanced methods are based on transformers [Vas+17]. Transformer based approaches [Zhe+21; Xie+21; Liu+21; CSK21; Xu+22] has shown impressive results by addressing the long range pixel relationship problems.

2.2. Hierarchical semantic segmentation

Although standard semantic segmentation approaches have a powerful semantic understanding capacity, these methods do not prefer to build network by considering the hierarchical structure of dataset categories. Various experiments [Ber+20; Fro+13; Wu+16] show that considering the relationships among the semantic categories in the network design process leads to significant performance improvements. For these encoding methods, some of the principles are learned from the data, while others are performed by human knowledge. The idea behind these approaches is to combine symbolic artificial intelligence into the deep learning based approaches. The integration of the symbolic and statistical cognitive paradigms shows advantages of reasoning and interpretability of symbolic representations and the promise of robust learning of neural networks. It improves the interpretability of the network and empowers it with more possibilities, while allowing more interaction between labels. Therefore, in order to improve the understanding of human thinking, it seems reasonable to look for ways to integrate symbolic and deep learning methods, rather than focusing on dichotomies. Hierarchical networks replicate the hierarchical thought model of human visual perception and represent the structured nature of our visual environment. In the field of computer vision, existing efforts in hierarchical taxonomy-aware image segmentation can be roughly divided into three groups: (i.) Hierarchical loss (ii.) Label embedding (iii.) Hierarchical architecture.

2.2.1. Hierarchical Loss

Semantic segmentation methods based on hierarchical loss are designed to incorporate the label hierarchy into the loss function in order to produce consistency between the label hierarchy and the predictions. In these methods,

the network is designed without the hierarchical relation of the label space, but the loss function exploits the hierarchical nature of the label space. Deng et al. [Den+14] proposed the hierarchy and exclusion graph, which encodes the label hierarchy into directed acyclic graph and computes the losses defined on it. Moreover, Bertinetto et al. [Ber+20] incorporate the label hierarchy into the cross-entropy loss, which they defined as the weighted sum of the cross-entropies of the conditional probabilities along the label hierarchy from the root to the leaf of the label tree. Muller and Smith [MS20] presented hierarchical loss for semantic segmentation, which is sum of losses at different levels of class abstraction. Li et al. [Li+22] introduced well-structured feature embeddings along with hierarchical loss into the network to improve segmentation. These methods require fewer parameters and computational power than neural networks with hierarchical architecture, as they utilize flat network to accomplish the task. However, this approach only introduces the label hierarchy in the network through the loss function, while the approach proposed in this work utilizes hierarchical loss and the architecture to include the label hierarchy.

2.2.2. Label Embedding

Label embedding based approaches aims to encode hierarchy into embeddings whose relative locations or possible interactions represent the semantic relationships and optimise a loss on these embedded vectors. The DeViSE - a deep visual-semantic embedding [Fro+13] uses labeled data and semantic information gathered from unannotated Wikipedia text [Mik+13] to recognize objects. It maps features to embedded labels, and then uses rank loss to optimize the algorithm. Another approach based on label embedding is represented by Bertinetto et al. [Ber+20], who softened the one-hot encoding based on the label tree-based hierarchical factorization and computed a loss on it. However, the label embedding-based approach also learns the label hierarchy through a loss function rather than introducing additional changes to the network based on the label hierarchy.

2.2.3. Hierarchical Architecture

Hierarchical architecture based approaches incorporate the hierarchy into the structure of a neural network, so that the network is designed to branch, with each branch responsible for identifying a specific level of conceptual abstraction in the label hierarchy. Wu et al. [Wu+16] designed a network containing a label hierarchy that shares a backbone with multiple fully-connected layers, each of which is responsible for label prediction at its level. Bilal et al. [Bil+17] uses CNNs to accommodate fine-grained labels and adds branches in the middle layers to accommodate the coarser-grained labels. Furthermore, Chen et al. [Che+18b] extends this work by proposing a hierarchical semantic embedding framework that uses single feature extractor and multiple branching network to predict categories of each level. It also introduces an attention mechanism that incorporates coarse-grained results to guide the learning of finer-grained features. However, Chang et al. [Cha+21] proposes that only fine-grained

features can improve the learning of coarser-grained branch networks. The approach proposed in this work combines the concepts of hierarchical structure and hierarchical loss with state-of-the-art semantic segmentation models. It forces the network to learn label stratification through hierarchical network design and hierarchical loss.

3. Approach

Now that all fundamentals and related work in the field of image semantic segmentation are covered. The focus of this chapter is to describe the datasets used in this work, present the methodology, and the rationale for the techniques used in this work. Afterwards, in the next chapter, I describe the experiments, implementation details and their comparative study with the baseline architecture.

The structure of this chapter is as follow: Section 3.1 provides an overview of the datasets used for training and evaluating the models, Section 3.2 defines the baseline architecture used for the comparative study, and Section 3.3 discusses the proposed architecture and the advancement in the label hierarchy.

3.1. Datasets

This section includes an overview of the image-based semantic segmentation datasets used in this work. Since most of the research on image semantic segmentation has focused on 2D images, there are many 2D image segmentation datasets available. As one of the goals is to have a model that can be generalized, it is desirable to use different datasets from indoor and outdoor domains. In order to achieve this, the datasets are selected in such a way that they represent variations in various environmental conditions, geographic locations, domains and objects representation. The selection criteria also considers the number of categories in the dataset (i.e. coarse category datasets and finely annotated datasets) and the size of the dataset. In addition, these datasets are divided into two categories based on the variability of the dataset samples: (i) Cross-domain datasets (ii) No cross-domain datasets. Cross-domain datasets include samples recorded under different lighting, weather, and seasonal conditions. On the other hand, the no cross-domain datasets are usually recorded under daylight conditions.

In this work ADE20K [Zho+17], Scannet [Dai+17a], Indian driving dataset [Var+19], Cityscapes [Cor+16], Mapillary Vistas [Neu+17], BDD10K [Yu+20], Wilddash 2.0 [Zen+18] and Playing for benchmarks [RHK17] datasets are selected. These datasets provide appropriate pixel-level labels and represent the diverse nature of the objects in real environments. The segmentation masks for these dataset do not use JPG format because JPG is lossy and the pixel values may change. They use a segmentation mask in 8-bit PNG format with the size of the input image. The subsections includes information about the domain, sample size, sample diversity, number of categories, and sample resolution of each dataset. The overview of all datasets are described in Section 3.1 and more detailed information about each dataset can be found in Appendix A.

Table 3.1.: *Datasets overview: A total of 8 datasets are used in this work. Class count refers to the number of classes in the original label space.*

Dataset	Scenes	#Images (train/val)	#Class
ADE20K [Zho+17]	Natural	20210/2000	150
Cityscapes [Cor+16]	Driving	2975/500	34
Vistas v1.0 [Neu+17]	Driving	18000/2000	66
BDD [Yu+20]	Driving	7000/1000	19
IDD [Var+19]	Driving	6993/981	39
WildDash 2 [Zen+18]	Driving	3413/857	34
ScanNet [Dai+17a]	Indoor	19466/5436	41
VIPER [RHK17]	Synthetic	13367/4959	32

3.1.1. No cross-domain datasets

ADE20K

The ADE20K [Zho+17] dataset provides complete scene annotation, which includes annotation of big objects, their components, and parts of their components. The exhaustively annotated objects are extracted from indoor and outdoor scenes. These objects are annotated by a skilled annotator. Images are carefully segmented by hand and cover a wide range of scene, object, and object portion categories. The consistency of the data and the quality of the annotation are greatly improved compared to automatic/external annotation methods. This dataset includes 150 categories of large and small objects across 20K training, 2K validation, and 3K test images. On average, each image contains 9.9 object classes, and the maximum number of object instances per image is 273, and 419 instances, if parts are also counted.

Scannet

Scannet [Dai+17a] is an RGB-D video dataset containing 2.5M views in 1513 scenes acquired in 707 unique indoor spaces annotated with 3D camera poses, surface reconstructions, and semantic segmentations. It provides annotation with estimated calibration parameters, 3D surface reconstructions, textured meshes, camera poses, pixel wise object level semantic segmentations, and CAD model placements for a subset of the scans. This large dataset includes 19466 training and 5436 validation images with 41 categories. Scannet contains a variety of spaces, ranging from small (e.g. bathroom, closet, utility room) to large (e.g. classroom, apartment, office). It also provides a 2D projected frames for image semantic segmentation and an open source annotation framework for dense RGB-D reconstruction. These 2D projected frames are used for the experiments in this report.

3.1.2. Cross-domain datasets

Cityscapes

Cityscapes [Cor+16] is a substantial database focused on understanding the semantics of urban streetscapes. The database consists of a set of 20K weakly annotated frames and 3475 frames with high quality pixel-level annotations, as well as a series of stereo video sequences taken at street locations in 50 cities. These frames are recorded in 2048×1024 resolution, mainly from moving vehicles in Germany and its neighboring countries, over a period of several months, including spring, summer and autumn. This dataset does not include severe weather conditions, such as heavy rain, and snow. In addition, the dataset provide label hierarchy, where samples are grouped into top level eight categories i.e. flat surfaces, people, cars, buildings, objects, nature, sky, and others. Further these categories are divided into 34 classes of semantic and dense pixel annotations. The pixel-level 3475 frames are divided into 500 validation and 2975 training images. In this report, high quality pixel-level annotations are used in order to maintain the consistency with other datasets.

Indian Driving Dataset (IDD)

Most of the mentioned road datasets have explicit infrastructure such as lanes, traffic lights, road marking, and low variability of foreground and background objects. The IDD [Var+19] contains unstructured and structured scenes from real environments. Samples are recorded in 182 driving sequences on rural and urban roads with 10,004 finely annotated images with 39 categories. The dataset provide 4 level label hierarchy and layered polygon annotation masks similar to cityscapes [Cor+16] dataset. In addition to unstructured scenes, it provides a high number of motorcycles, animals, vehicles objects, and even some new categories like auto rickshaws and drivable roads. These scenes are recorded at 720×720 and 1080×1080 resolution from various weather conditions such as day, afternoon, dawn, dusk, fog, dust and shadows.

Mapillary Vistas

Mapillary Vistas dataset [Neu+17] is a large scale road dataset consisting of 18k training and 2k validation images with 66 categories in version 1.0 and 150 categories in version 2.0. These samples are captured from various devices with varying viewing angles and recorded a large number of outdoor scenes in Europe, North & South America, Asia, Africa and Oceania. These scenes are recorded under a variety of weather, and lighting conditions. The minimum image resolution is 1920×1080 and around 90% of the samples are from roadside views. As of today, it is the largest and most diverse open dataset.

BDD

BDD100K [Yu+20] is a large scale dataset. It is designed for multi-task learning and includes 10 different tasks to evaluate autonomous driving. In this report,

the BDD10K dataset consists of 10K 2D images for semantic segmentation task is used. These samples are recorded in 720×720 resolution from a dashboard camera under different geographical, environmental and weather conditions.

Wilddash

Wilddash [Zen+18] dataset is constructed with an aim to test computer vision algorithms against the robustness and their usability in real-world automotive applications. It includes diverse set of samples from around the world with different lighting, weather and environmental conditions to reduce bias in dataset. It consists of 3413 training and 857 validation images with 34 categories, similar to cityscapes dataset with a resolution of 896×896.

Playing for benchmarks (VIPER)

Playing for benchmarks [**Viper**] is a artificial dataset. The samples are recorded while driving, riding, and walking in the virtual world under five different environmental conditions, such as daytime, sunset, rain, snow and night. It consists of 25,064 frames at 1920×1080 resolution that are annotated with pixel wise semantic segmentation, semantic instance segmentation, instance-level semantic boundaries, object detection and tracking, 3D scene layout, optical flow and ego-motion. These samples are divided into 134K, 50K and 70K frames for training, validation and test sets with an objective that these sets covers a balanced distribution of the obtained data.

Mixed Dataset

Mixed dataset is a custom dataset that is generated by mapping individual datasets to a common dataset with a common label space for this work. The dataset is constructed to address the understanding of generic, indoor and semantic street scenes by reducing the drawbacks of one specific dataset. The individual datasets suffer from a bias toward recording modalities, location-specific objects, overall number of object categories, urban outdoor indoor scenes, and appearance of the objects. This dataset attempts to overcome these shortcomings with a simple and naive mixing technique of these datasets. The mapping techniques follows following principles: (i.) The semantically similar labels across the datasets are merged e.g. Cityscapes-car $\mapsto$ *car* and Vistas-car $\mapsto$ *car*. (ii.) If a class is superset of multiple classes, the superset class is ignored e.g. {pole, bridge, tunnel, street light, utility pole} $\mapsto$ VIPER-infrastructure, therefore VIPER-infrastructure class is ignored. The final label space contains 191 categories, high variability in class objects and noisy labels for some categories. This dataset is consist of 100K samples from ADE20K, Scannet, Cityscapes, Mapillary Vistas v1.0, VIPER, Wilddash and 1K from BDD10K, IDD dataset respectively with 191 categories for common label space.

3.2. Baseline Architecture

3.2.1. DeepLabv3+

The image-based semantic segmentation network are mainly based on the so-called Encoder-Decoder (ED) architectures, which is comprised of two parts: Encoder gradually reduces the spatial dimension and decoder restores the object details and spatial dimensions.

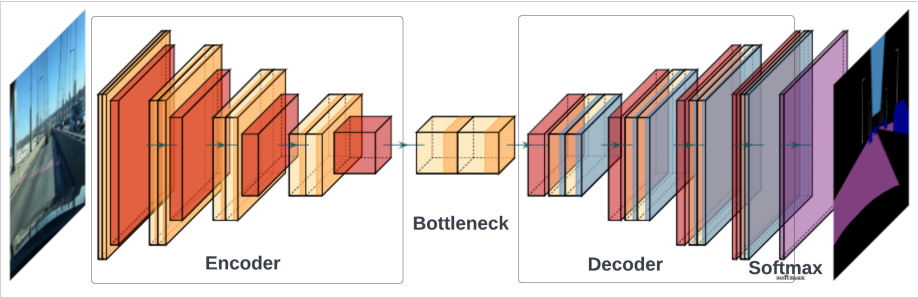


Figure 3.1.: *Standard encode-decoder architecture based semantic segmentation network. Illustration based on TikZ script [Iqb18].*

The encoder stage extracts the features from the input image and the decoder reconstructs the output of appropriate dimensions based on the information provided by encoder. The semantic segmentation network based on ED architecture is illustrated in Figure 3.1. The convolutional neural network-based encoder downsamples the resolution of the input image multiple times, which results in a low resolution feature map for the decoder, as shown in Figure 3.2. The final low-resolution feature map compromises the prediction accuracy and boundary information.

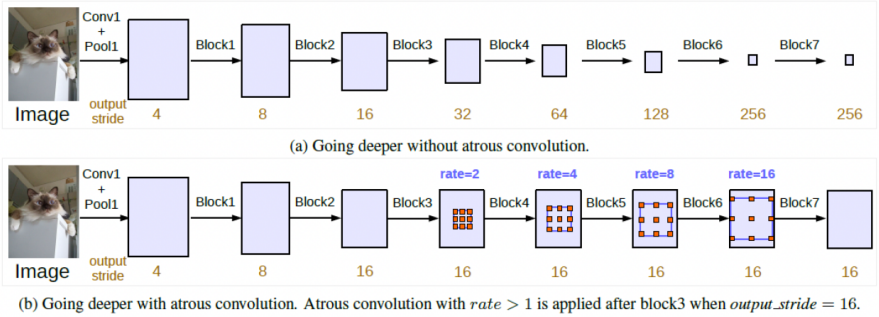


Figure 3.2.: *Encoder phase and their output stride (a) without and (b) with atrous convolution.[Che+18a]*

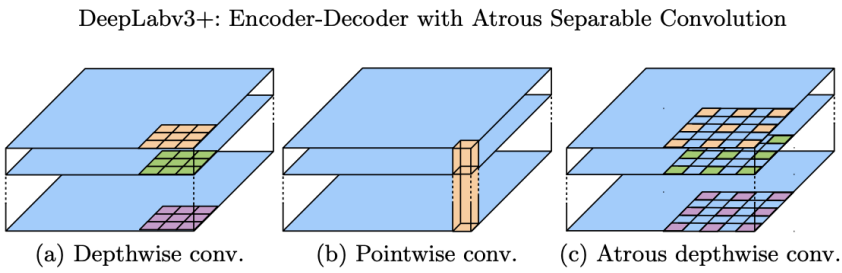


Figure 3.3.: *(a) Depthwise convolution, (b) pointwise convolution (c) atrous separable convolution with rate = 2.[Che+18a]*

Segmentation Head

The state-of-art deep learning network, DeepLabv3+ [Che+18a] for semantic segmentation based on 2D images overcomes these shortcomings by determining the atrous convolution rate based on output stride, and aggregating context around a feature using multi-scale atrous convolution. It is a simple and effective semantic segmentation network for the 2D images in non-transformer based category, it is chosen as segmentation head for the baseline and proposed network architecture for all the experiment in this report. The architecture of DeepLabv3+ is shown in the Figure 3.4. The segmentor is consist of (i.) Encoder: The backbone is used to extract features from high resolution input image and transform them to a lower resolution feature vector. The atrous spatial pyramid pooling (ASPP) consist of four parallel atrous separable convolution and global average pooling is used to capture the multi-scale contextual information from the feature extractor. Atrous separable convolution is atrous convolution with depthwise convolution as shown in Figure 3.3. Atrous convolution adjusts the field of view of the filter to capture multi-scale information from the backbone and the depthwise separable convolution is used to reduce the computation cost and number of parameters. Depthwise convolution applies single filter on each input channel and subsequently combines the outputs using pointwise convolution. (ii.) Decoder: This module fetches the high-level upsampled features from the ASPP and concatenates them to the low-level features from the backbone. The decoder module along with encoder is illustrated in Figure 3.4.

Encoder

DeepLabv3+ supports Auto-DeepLab, MobileNetv2, Xception, ResNet, PNASNet encoders. Since there are various deep and shallow backbone networks, choosing a shallow network can lead to overfitting problems, while choosing a deep convolutional neural network can lead to vanishing gradient problem. Therefore, the network should have enough capacity to be able to solve the vanishing gradient problem. Although there are various methods to solve the vanishing gradient problem by using auxiliary losses in the middle or last layer, none of these solutions seems to really solve the problem. On the other side, Residual Network (ResNet) address this problem by using shortcut connection

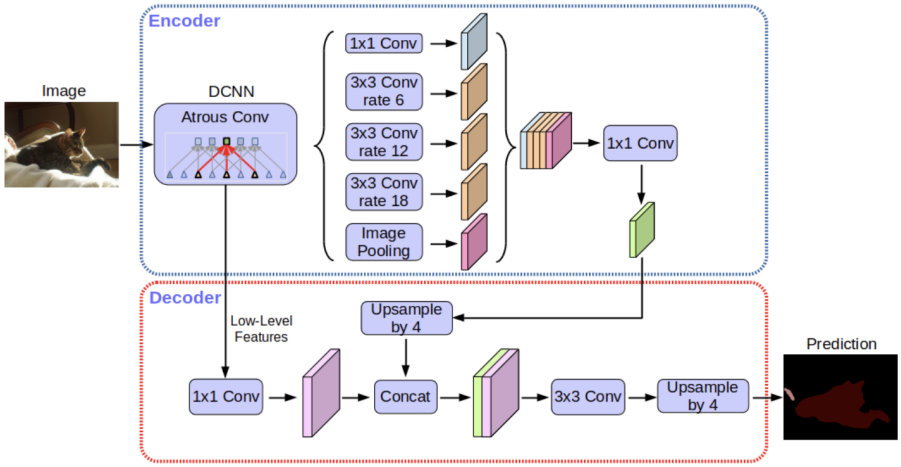


Figure 3.4.: *DeepLabv3+ architecture: Extracts the multi-scale contextual information using ASPP and concatenate these information with low-level information to up-sample the segmentation mask [Che+18a]*

and identity mapping between the stacked layers. The shortcut connections connects the output of one earlier convolutional layer to the input of another future convolutional layer. It is also known as residual connection. Residual connection between the layers allow the gradients to flow to earlier layers, which helps to avoid vanishing gradient problem. Various refinement and tricks of residual networks has been proposed in [He+19b] paper. The three ResNet tweaks have been shown in Figure 3.5. The computation cost of ResNet-C is low, quadratic to the width or height of the kernel, and the network is selected for this work. Further, ResNet-C backbone with 101 layers is selected to train the DeepLabv3+ network in this report.

Training and Inference

During the training of baseline network, batch of input images are passed through encoder, which outputs the features. Then, the decoder recovers the details and spatial dimensions of the objects, resulting in the segmentation mask of the input images. In training phase, only sample from the training set are presented to the model. Afterwards, the performance of the model is measured with samples from the validation or test set. The random 100 samples from each validation set are used to decide the hyperparameters of the model. Once all the optimizations are done and no further modifications to the model are planned, the whole validation/test data is used to evaluate the model performance. Although there are many ways to construct batches of training samples for Mixed dataset e.g. for a batch size of 8: taking each image from each dataset, in this work, the batch samples are selected randomly from the datasets. During inference, the test images are passed to the trained model and performance is measured using intersection-over-union metrics. Additionally, Multi-scale and flip data augmentations are not considered for the test data

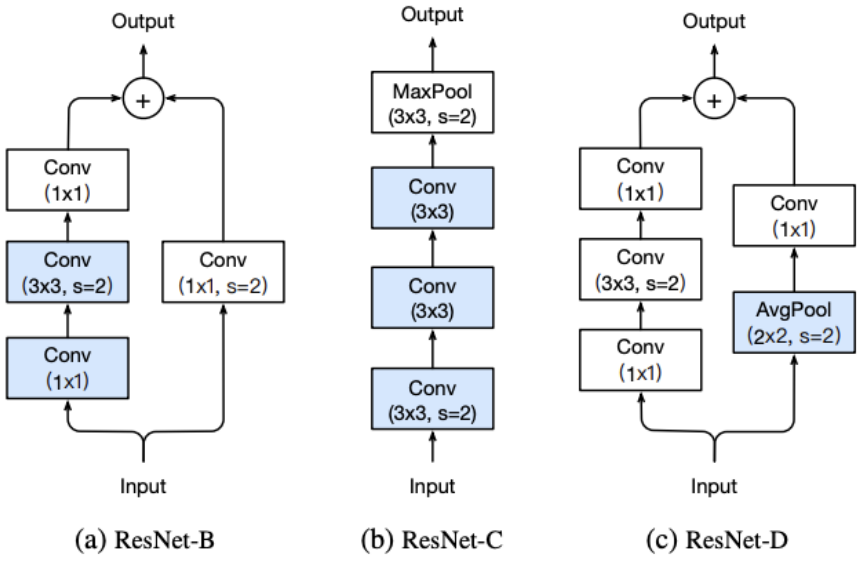


Figure 3.5.: *ResNet variants. ResNet-B adjusts the downsampling block. ResNet-C alters the input stem and ResNet-D further changes the downsampling block.[He+19b]*

evaluation. The detailed information is available in Section 4.1.

Addressing the Class Imbalance

Every dataset mentioned in Section 3.1 suffers from category imbalance, which can lead to undesirable biases in the model during the training stage. There are various techniques to address this problem, which are accompanied by unbalanced dataset, such as class specific weights for the loss function, oversampling of underrepresented classes or downsampling of overrespresented classes. This poses the challenge of selecting the appropriate class weights. In many cases, class weights have not proved to be well suited. Therefore, in this work, the online hard example mining (OHEM) algorithm [SGG16] is used for training the model. It is a bootstrapping method that alters the stochastic gradient descent method by sampling from examples in a non-uniform manner based on the current loss of each example considered.

3.3. Proposed Architecture

3.3.1. Hierarchical DeepLabv3+

Motivation

In general, the baseline architecture assumes that each category is inherently disjoint and faces difficulties in distinguishing between these categories. These categories are confused with other categories, such as "bird" with "vegetation",

Table 3.2.: *Confusion matrix of baseline network trained on Mapillary Vistas v2.0 dataset. The gray section refers to low performing categories and white section to high scoring categories. The last column denotes the percentage of pixel count with respect to whole dataset for the category.*

Confusion Matrix (Accuracy)					Pixels % w.r.t. corpus
	Bird	Vegetation	Sky	Building	
Bird	0.0	0.44	0.10	0.07	9.91e-6
	Bike Lane	Road	SideWalk	Crosswalk	
Bike Lane	0.27	0.57	0.06	0.02	3.0e-3
	Curb cut	Curb	Sidewalk	Drive way	
Curb Cut	0.20	0.32	0.23	0.02	8.0e-3
	Parking	Road	Sidewalk	Ground	
Parking	0.159	0.55	0.13	0.04	2.0e-3
	Tunnel	Wall	Road	Vegetation	
Tunnel	0.253	0.392	0.13	0.05	6.0e-4
	Boat	Car	Dynamic	Building	
Boat	0.03	0.44	0.17	0.11	8.9e-5
	Caravan	Truck	Bus	Car	
Caravan	0.0	0.568	0.19	0.18	7.53e-5
	Trailer	Car	Truck	Other Vehicles	
Trailer	0.0	0.45	0.38	0.088	1.0e-4
	Sky	Road	Vegetation	Pole	
Sky	0.96	0.01	0.006	0.002	2.9e-1
	Snow	Road	Ground	Vegetation	
Snow	0.88	0.04	0.001	0.007	3.0e-3
	Vegetation	Building	Sky	Terrain	
Vegetation	0.94	0.01	0.008	0.007	1.4e-1
	Pole	Building	Vegetation	Utility Pole	
Pole	0.62	0.10	0.07	0.04	8.0e-3
	Bicycle	Motorcycle	Building	Bicyclist	
Bicycle	0.67	0.05	0.04	0.037	5.0e-4
	Person	Building	Vegetation	Car	
Person	0.83	0.04	0.01	0.016	3.0e-3

"boat" with "car", etc. To analyze it in more detail, the accuracy based confusion matrix was recorded on the baseline model of Mapillary Vistas v2.0 [Neu+17] for some low and high scoring categories and their pixel percentage in Table 3.2. A categories like "bird" shows that the object is associated with "vegetation", "sky" and "building", which suggests that the object is either sitting on vegetation/building or flying in the sky. And it is true in natural environment. Similarly, for "curb cut", it is mainly related to "curb", "sidewalk", and "driveway". Thus, it is clear that these categories are not disjoint in nature. Furthermore, the table shows that the network straggle to understand structural differences between the categories e.g. "caravan" and ["Truck", "Bus", "Car"], and the hierarchical relationships among the categories, regardless of the number of pixels. This raises the question that whether a

baseline network is adequate for distinguishing all the categories.

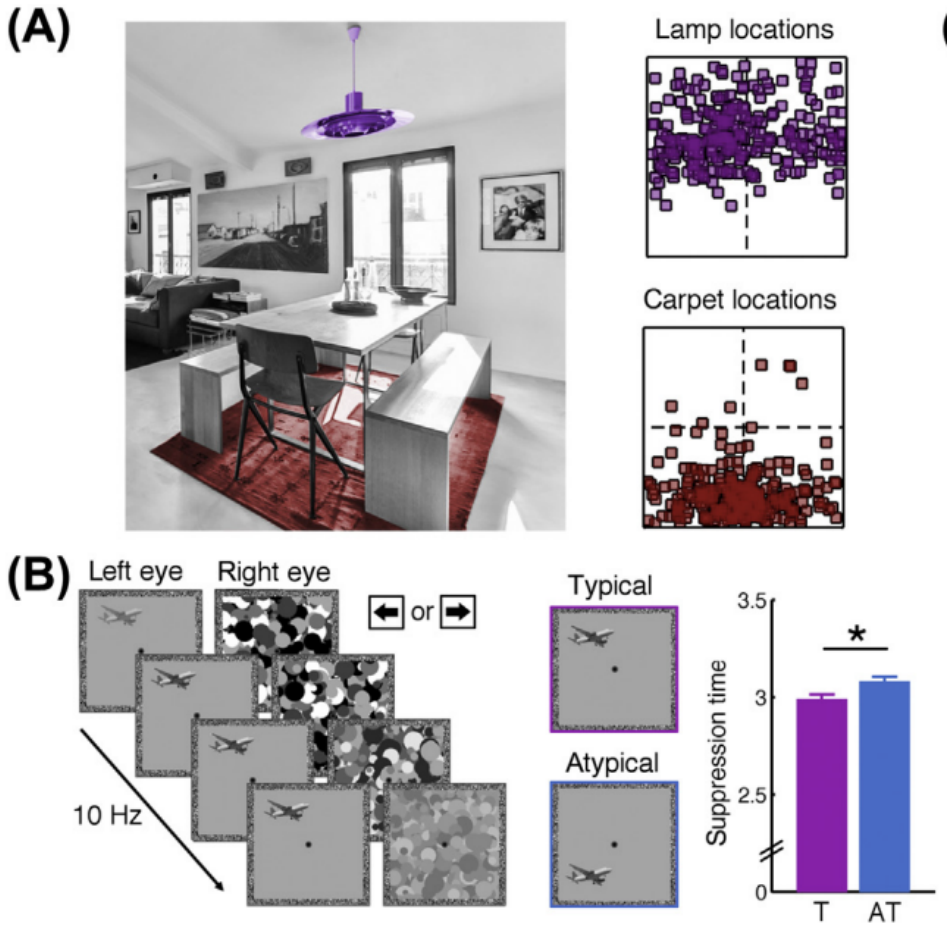


Figure 3.6.: *(a) Location of carpet and lamp object over 250 images. Carpet and Lamp commonly appears on lower and upper portion of the visual field respectively. (b) Human's suppression time in normal and abnormal situation to identify the airplane [Kai+19].*

In a natural environment, if a human is asked to recognize a "car" object in a video sequence, the human would first focuses on the foreground of the scene, then on the vehicles, then he/she identifies the "car" in the scene faster, depending on the absolute position of the object in the field of view and its relative position to other objects [Kai+19]. The authors of "Object Vision in a Structured World" [Kai+19] also found that the suppression time to recognize a object in typical situation is shorter compared to atypical situations. It is shown in the Figure 3.6 that if carpet and lamp is positioned in the normal situation means at lower and upper part of the visual field respectively, they are faster to recognize. Furthermore, this infers that if objects are systematically distributed in complex scene, it reduces the degree of ambiguity to segment the

object.

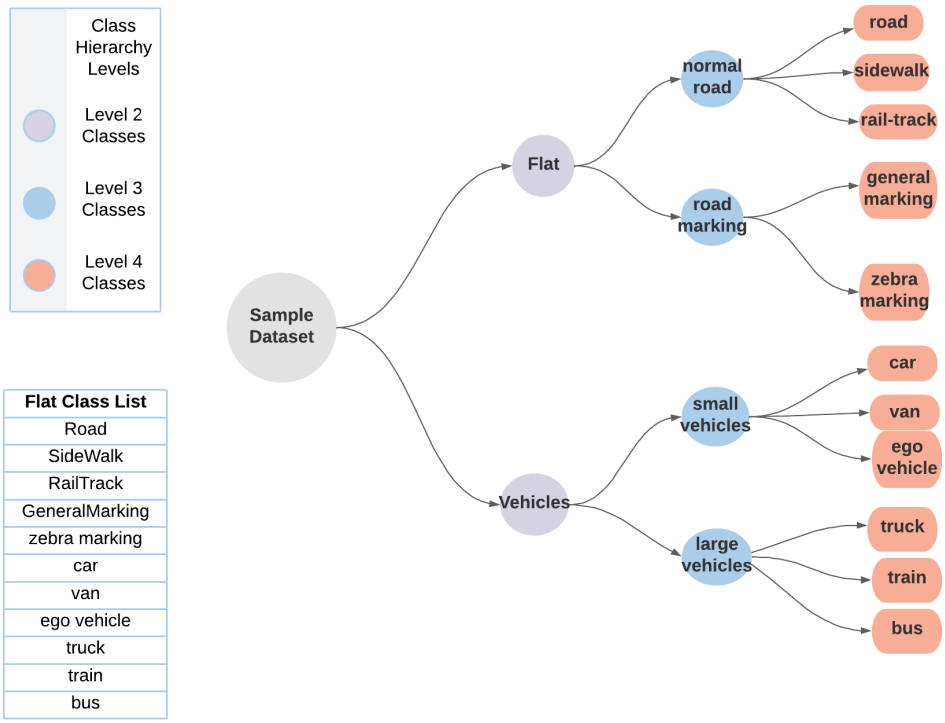
Therefore, to overcome these aforementioned limitations, I approach the problem here from the perspective of network design and label hierarchy, where the network is designed in such a way that it inherits the hierarchy of categories, with an associated segmentation head for each level of conceptual abstraction. This "hierarchical network" is based on DeepLabv3+ [Che+18a] segmentation head and the ResNet-101 backbone. The hierarchical architecture adapts to these basic concepts of perception processing and hierarchical nature of objects in structured world by using: (i.) Label hierarchy: It groups labels into multi level label spaces based on semantic similarity among the classes. It also attempts to group classes based on their position in visual field and relative locations to other classes such as road, sidewalk, parking and rail track are subclass of flat category. (ii.) Additional segmentation heads: The segmentation heads are used to maintain the label hierarchy in the network and to provide a specific header for each group of categories, which also assist in maintaining the balance of categories between intra and inter datasets samples by providing equal attention to categories irrespective of their sample size.

Architecture

With an aim to include basic perception processing and hierarchical nature of object categories into the semantic segmentation network. The baseline network is transformed into hierarchical network by first converting the flat label space into tree-based label hierarchy, then the segmentation head in baseline network is replaced with multiple segmentation head. The number of segmentation head in hierarchical network depends on the label hierarchy, i.e. it is equal to the number of nodes in the tree-based label hierarchy.

Hierarchical Label Space: The hierarchical label space is constructed by formalizing the common label space into a tree-structured label hierarchy. The tree-based label hierarchy $\mathcal{H}$ consists of nodes $\mathcal{N}$ and edges $\mathcal{E}$. Each level l of the tree represent the level of label hierarchy. Levels are denoted in ascending order from top to bottom of the tree and depth of the tree is indicated by d. Each node $n \in \mathcal{N}$ represent the category in the label space and edge $e \in \mathcal{E}$ denotes the relationship between the categories. The root of $\mathcal{H} = (\mathcal{N}, \mathcal{E})$ i.e. n_{l1_1} represent the first node at level 1, which groups datasets categories into one $[n_{l2_{1..k}}, n_{l3_{1..k}}, ..., n_{ld_{1..k}}] \in n_{l1_1}$. The intermediate node e.g. n_{l2_1} of the tree represent the most general class and leaf node at level d denotes the most-fine grained class.

In order to simplify the concept, here I consider an sample dataset with 11 categories i.e. "Road", "Sidewalk", "Rail Track", "General Marking", "Zebra Marking", "Car", "Van", "Ego vehicle", "Truck", "Train", and "Bus". These flat categories are transformed into label hierarchy. In this example, the label hierarchy is hand-picked based on human cognition to distinguish the labels. Here, the generalized categories (super categories) at each level of the tree is selected based on the human's understanding of the visual world. The intermediate nodes at second level $n_{l2_{1,2}} \mapsto [\text{Flat}, \text{Vehicles}]$ and at third level $n_{l3_{1..4}} \mapsto [\text{NormalRoad}, \text{RoadMarking}, \text{SmallVehicles}, \text{LargeVehicles}]$ are



Flat Class List
Road
SideWalk
RailTrack
GeneralMarking
zebra marking
car
van
ego vehicle
truck
train
bus

Figure 3.7.: *Label hierarchy for sample dataset.*

the generalized labels for the leaf node (fine grained) classes. Tree-based hierarchical label space for the sample dataset is shown in Figure 3.7 and loss $\mathcal{L}$ for this hierarchy is computed by aggregating the loss at each node except leaf nodes of the tree $\mathcal{H}$ as shown in Equation (3.1).

$$\mathcal{L}(\mathcal{H}) = \mathcal{L}_{n_{l1_1}} + \mathcal{L}_{n_{l2_{1,2}}} + \mathcal{L}_{n_{l3_{1..4}}} \tag{3.1}$$

The tree based label hierarchy e.g. Figure 3.7 is integrated into the semantic segmentation network by replacing the baseline semantic segmentation neural network 3.1 with hierarchical network 3.8. Hierarchical network is consist of single shared encoder and multiple segmentation heads based on the dataset label hierarchy. The number of segmentation head S in the hierarchical network is calculated by number of nodes in the tree-based label hierarchy. The encoder ϕ_{ENC} extracts the dense features $\mathcal{X} = \phi_{ENC}(\mathcal{I}) \in \mathbb{R}^{\mathcal{H}\times\mathcal{W}\times\mathcal{C}}$ from the input image $\mathcal{I}$. Then the features $\mathcal{X}$ are passed through each segmentation head ϕ_{SEG_s} to get the unnormalized score vectors (logits). These logits are further normalized using softmax function such as $\hat{\mathcal{Y}}_s = \text{softmax}(\phi_{SEG_s}(\mathcal{X})) \in [0, 1]$. The normalized score vectors $\hat{\mathcal{Y}}_s$ and ground truth of each segmentation head $\mathcal{Y}_s$ are further used to optimize the loss $\mathcal{L}$. The total categorical cross entropy loss of the network is computed by summing the loss of each segmentation head as shown in Equation (3.2).

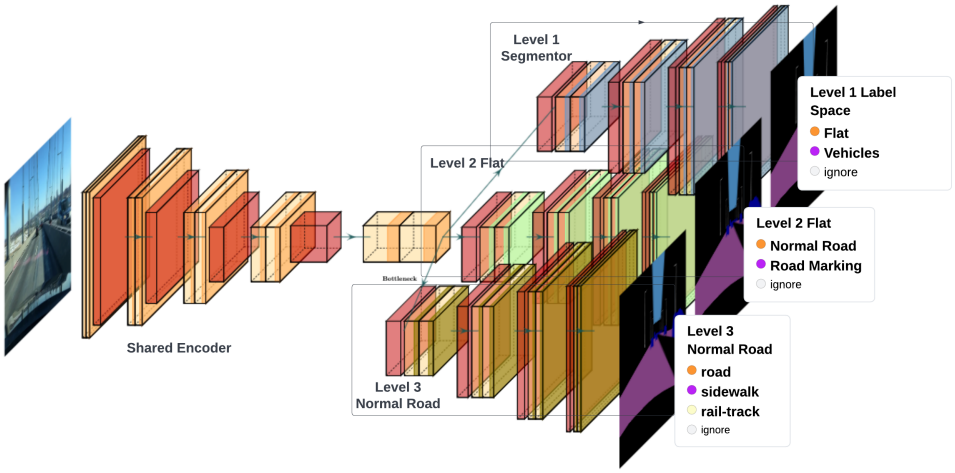


Figure 3.8.: *Hierarchical semantic segmentation network based on ED architecture. Partial hierarchical network based on label hierarchy in Figure 3.7. Network is illustrated using TikZ [Iqb18].*

$$\mathcal{L}(\hat{\mathcal{Y}}, \mathcal{Y}) = -\sum_{s}^{S} \mathcal{Y}_s log(\hat{\mathcal{Y}}_s) \tag{3.2}$$

Hierarchical Network: In order to comprehend the hierarchical semantic segmentation network, the label hierarchy shown in Figure 3.7 is taken to demonstrate the sample hierarchical network architecture as shown in Figure 3.9. The network includes single shared encoder and multiple decoder heads. The multiple decoder heads are placed at each node in the tree-based label hierarchy. Each head is responsible to identify subset of categories. The features from encoder $\mathcal{X}$ are concurrently passed through all the heads e.g. "Level 1 Seg Head", "Level 2 Normal Road", ..., "Level 3 large vehicle head". Each segmentation head generate the score map for their label spaces e.g. Level 1 head include Flat and Vehicles categories and level 3 large vehicles head consists of three categories truck, train and bus. Given the score vector of each segmentation head w.r.t. the label space of the head, final loss is calculated and optimised using backpropagation algorithms.

Training and Inference

The network is trained by passing a random set of images into the network as a batch and parsing the ground truth mask based on the label space of the segmentation head. For the total loss of the network, the losses of all segmentation heads are collected and weighted with different hyperparameters to obtain the total objective to be minimized as shown in Equation (3.3).

$$\mathcal{L}^{total} = \sum_{j} \lambda^j . \mathcal{L}^j \tag{3.3}$$

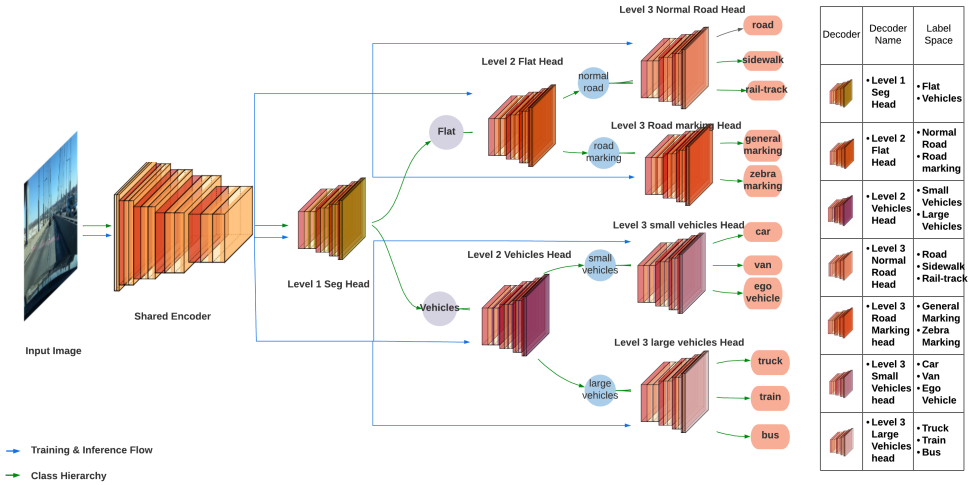


Figure 3.9.: *Hierarchical semantic segmentation network for the sample dataset based on label hierarchy in Figure 3.7.*

The hyperparameters λ^j of Equation (3.3) are chosen to be 1.0, 0.1 and 0.1 for the three levels of label hierarchy.

During the inference process, the test image passes through a shared encoder and then through multiple segmentation heads. Every segmentation head in hierarchical network generates the output mask for their label space e.g. level 1 segmentation head segments the input image into level 1 label space and subsequently level 2 and level 3 segmentation heads. Further, the output of all levels are combined to form the final fine-grained per-pixel segmentation.

3.3.2. Learning a label hierarchy

So far, I have discussed the baseline architecture and proposed network for semantic segmentation with hand-picked label hierarchy. This section explores more about the design of the tree based label hierarchy. First, it discusses naive approach for constructing the label hierarchy. Secondly, it discusses feature embedding based label hierarchy.

Hand-picked label hierarchy

For a dataset, the construction of naive label hierarchy is based on human cognition abilities. In hand-picked label hierarchy semantically similar objects are grouped together. For a three level of label hierarchy, three level of concept abstraction are introduced e.g. first all vehicles are grouped into "small" and "large" vehicles category followed by more generalized vehicles category. For a sample dataset, I constructed such label hierarchy as shown in Figure 3.7. The visually similar objects are grouped into one category like car, van, ego vehicle are grouped to small vehicles and truck, train, bus are grouped into large vehicles categories. Further small vehicles and large vehicles categories are grouped into vehicle category. This is the simplest grouping of the categories

based on the human's ability to understand the relations and hierarchies among the labels.

Pixel-embedding based label hierarchy

The motivation behind this approach is that networks may have a different understanding of relationships and hierarchies than humans. Therefore, in this approach, the label hierarchy is constructed based on the features from the backbone. In order to extract the pixel-embedding from the model, the baseline model is first trained and once all the optimizations are done, the pixel-embedding are extracted for the validation set through the projection head for each categories. The average of these pixel-embedding for a category represent that category. Furthermore, this representation is used to form the multi-level clusters to mimic the human cognition based label hierarchy. K-Means and Hierarchical Clustering algorithm are analyzed to form the multi level clusters. However, these algorithm tends to form a unbalanced clusters. Further, regularized version of K-Means [YGS19] is used to overcome unbalanced clusters and hyper-parameter for the number of clusters is choosen based on hand-picked label hierarchy. As an exemplary, if hand-picked hierarchy has 2 nodes at top level of the hierarchy, embedding based hierarchy choose 2 as a number of clusters for the top level.

4. Experiments and Results

4.1. Experiments

This section summarizes the experimental details and performance for the baseline and proposed methods for semantic segmentation. The standard DeepLabv3+ is chosen as the baseline method and the network is further extended as a hierarchical network. Section 4.1.1 describes in-depth details of the baseline experiments, where separate networks are trained for datasets: ADE20K, Cityscapes, VIPER, Vistas v1.0, Vistas v2.0, Scannet, Wilddash and their performance is documented. Section 4.1.2 designed a single network trained on mixed dataset with a common label space and separately captures its performance on ADE20K, Cityscapes, VIPER, Vistas v1.0, Scannet, and Wilddash, respectively. In later parts of this report, this model is referred as a cross-domain network. In addition, the cross-domain network is transformed into a hierarchical network, where the label hierarchy is designed based on human cognition and backbone pixel-embedding to group similar categories into a multi-level of label hierarchy. The human cognition based label hierarchy experiments are referred as hand-picked hierarchical networks and more details about it are presented in Section 4.1.3. The label hierarchy based on pixel-embedding is described in Section 4.1.4. Later, hierarchical network with hand-picked hierarchy and pixel-embedding based hierarchy are experimented on a single dataset i.e. Mapillary Vistas v2.0 in Section 4.1.5 and Section 4.1.6. The experimental setup and common configurations for these experiments are summarized below.

Experiment pipeline

To efficiently perform these experiments on multiple GPUs, a experimental pipeline is constructed based on OpenMMLab [Con20]. The pipeline includes following steps: (i.) Pre-processing pipeline step includes annotation loading, mapping the annotation to respective class ids, resizing the inputs, random cropping, random flipping, photometric distortion and normalization of inputs. (ii.) Training step, the experiment configurations such as optimizer type, initial learning rate, momentum, weight decay, learning rate policy, samples per GPUs and checkpoints configs are set and network training is executed. (iii.) Evaluation is performed for the model based on best checkpoint available on the validation set. (iv.) Inference step, the test input images are passed through the trained model to get the output mask for bench-marking the model.

Experiment Hyper-parameters

During the training phase, some parameters must be considered to change in order to obtain the best performance of the proposed network on the problem to be solved and to modify the behavior of each layer. For these experiments, standard hyper-parameters are considered so that they can be easily compared with other available methods. Therefore, these experiments are trained with stochastic gradient (SGD) as an optimizer, with momentum and weight decay set to 0.9, and 0.0005, respectively, polynomial learning policy with power of 0.9 and standard categorical cross entropy loss function. The detailed experimental configuration is mentioned in Table 4.1. For comparison purposes, the training is kept consistent across experiments. The experiments in Sections 4.1.1 to 4.1.4 are trained for 600K iterations and Sections 4.1.5 to 4.1.6 are trained for 240K iterations.

Table 4.1.: *Common experimental configurations.*

Setup Parameters	Description
Network Architecture	Backbone: ResNet 101 v1c Segmentation Head: DeepLabv3+
Training Config	Dataset: Training set for learning and random 100 images from validation set for tunning. Batch size: 32 Resize: multi-scaled before crop Crop: random 512 × 512 Flip: random with 0.5 Pretrained: ImageNet Initial learning rate: 0.007 Optimizer: SGD Momentum: 0.9 Mixed precision: FP16 Learning rate: Polynomial decay with power set to 0.9 and end learning rate set to 1.8e-4.
Test/Validation Config	Dataset: Whole validation/test set Batch size: 1 Crop: Whole image Multi scale: No Flip: No

Evaluation Criteria

The performance of the semantic segmentation models are recorded using intersection-over-union (IoU) metrics, also known as Jaccard Index. The IoU metrics takes into account true positives (TP), false positives (FP) and false negatives (FN) classifications. Using binary bitmaps, the evaluation metric can be reformulated as in Equation (4.1). The overall performance on a dataset is

computed by mean of the IoU of each class. It is represented as mIoU. This is most common and widely used evaluation metric for the semantic segmentation tasks.

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{4.1}$$

Datasets

In addition to the dataset configuration mentioned in Table 4.1, in each experiment, the training set of the dataset is used to train the model and random 100 samples are selected from the validation set to tune the model. Then, during evaluation, whole validation set is used for measuring the performance. However, except for Wilddash, most of the datasets have a separate validation set for evaluation. Thus, for the Wilddash dataset, the 700 images of training set are used for validation.

4.1.1. [Baseline] Semantic Segmentation on Individual Datasets

The purpose of this experiment is to establish a baseline network and record its performance on validation set. Afterwards, the performance of this network will be compared with other experiments. As a baseline network, ResNet with 101 layers is used for features extraction and DeepLabv3+ [Che+18a] for segmentation head and standard hyper-parameters for training and inference. The baseline network follows the common experimental configuration mentioned in Table 4.1.

Baseline experiment trains DeepLabv3+ network for each dataset (ADE20K, Cityscapes, VIPER, Vistas v1.0, Scannet and Wilddash) separately and evaluate it on its validation set. Each model has different number of classes depending on the label space of its dataset. The results are evaluated on the validation set of the dataset using IoU metrics. At the end, this experiment presents 6 models and their performance on validation set in Table 4.2.

4.1.2. [Cross-domain] One Semantic Segmentation Network for All Datasets.

The goal of this experiment is to design a single semantic segmentation network that performs well on ADE20K, Scannet, Cityscapes, Mapillary vistas v1.0, Wilddash and VIPER dataset. This experiment is inspired by the Robust Vision Challenge (RVC-2022), which also aims to design a single network that is applicable to all datasets without providing any dataset specific information into the network. For this task, I considered the DeepLabv3+ segmentation head with backbone ResNet-101 and hand-picked common label space. This network is trained with deep supervision technique and OHEM to balance the data samples. In this experiment, single network is trained on mixed dataset mentioned in Section 3.1.2 and then separately evaluated on ADE20K, Scannet, Cityscapes, Mapillary vistas v1.0, Wilddash and VIPER dataset. As part of this experiment, two network were trained on mixed dataset, first model is evaluated on validation set and its performance is recorded in Table 4.2.

Table 4.2.: *Results of the experiments: Section 4.1.1 and Section 4.1.2. The top half of the table shows the performance of baseline network and bottom half represent the performance of the cross-domain network.*

Experiment: Baseline		
Training Dataset	Validation Dataset	Validation mIoU (%)
ADE20K	ADE20K	22.67
Cityscapes	Cityscapes	68.7
VIPER	VIPER	52.89
Vistas v1.0	Vistas v1.0	31.14
Scannet	Scannet	43.4
Wilddash	Wilddash	48.7
Experiment: Cross-domain		
Training Dataset	Validataion Dataset	Validation mIoU (%)
Mixed Dataset	ADE20K	33.89
	Cityscapes	74.69
	VIPER	58.72
	Vistas v1.0	39.34
	Scannet	52.28
	Wilddash	58.25

Table 4.3.: *RVC-2022 cross-domain model performance on test set.*

RVC-2022		
Training Dataset	Test Dataset	Test mIoU (%)
Mixed Dataset	ADE20K	30.7
	Cityscapes	79.4
	VIPER	65.2
	Vistas v1.0	47.39
	Scannet	54.5
	Wilddash	45.5

This model is further compared with other models in this report. Another identical network is trained for much longer iterations and evaluated on test set of each dataset. This model is submitted to the RVC-2022 challenge and its performance is shown in Table 4.3. It achieved 3rd rank in RVC-2022. In addition to the common configuration, the cross-domain network consists of 191 classes. In comparison to baseline experiment, where the network is trained on single dataset and evaluated on same dataset, cross-domain network is trained on mixed dataset and evaluated separately on each dataset. The cross-domain network performance is documented in Table 4.2.

Table 4.4.: *Results of hand-picked and pixel-embedding based hierarchical model. These experiments are described in Section 4.1.3 and Section 4.1.4, respectively.*

Experiment: Handpicked Hierarchy		
Training Dataset	Validation Dataset	Validation % mIoU
Mixed Dataset	ADE20K	27.72
	Cityscapes	75.58
	VIPER	54.31
	Vistas v1.0	42.42
	Scannet	48.76
	Wilddash	57.60
Experiment: Pixel Embedding Hierarchy		
Training dataset	Validataion set	Validation % smIoU
Mixed Dataset	ADE20K	28.89
	Cityscapes	74.71
	VIPER	60.71
	Vistas v1.0	41.70
	Scannet	51.02
	Wilddash	58.07

4.1.3. [Hand-picked Hierarchy] Hierarchical Semantic Segmentation Network with Hand-picked Label Hierarchy on Mixed Dataset.

The goal is to construct a network that inherits the hierarchical nature of the label space and include more segmentation heads to ease the confusion among the classes. The hierarchical network is designed to accommodate the natural hierarchy in the label space. This network is trained on mixed dataset with 191 categories arranged in multiple level of hierarchy similar to Figure 3.7. The hand-picked label hierarchy for the mixed dataset 3.1.2 is shown in Figure B.1. The label hierarchy is consist of 3 levels, where bottom level represents the fine-grained categories of the dataset. The top most level in the hierarchy consists of 7 super categories: "Vehicles", "Flat", "Nature Objects", "Traffic Objects", "Construction", "VRU", and "Indoor Objects". These super categories include sub categories and sub-sub categories. The categories in the hierarchy are grouped in such a way that the each category share some structural and location in visual field similarity with in a super category. In order to achieve a well organized and less ambiguous hierarchical structure, some of the categories are ignored during the construction of hand-picked hierarchy. One of example is "infrastructure" category of VIPER dataset. It includes "traffic objects", "bridges", "poles" etc. Therefore, this label is ignored during the training to maintain the integrity of label hierarchy. The hand-picked label hierarchy for mixed dataset includes 31 segmentation heads, which means that there are 31 nodes in the tree-based label hierarchy. The label hierarchy and number of parameters varies with the dataset. Since the

mixed dataset consist of 191 categories, this results in multiple segmentation heads. After mapping the classes according to the hand-picked hierarchy shown in Figure B.1, the network is trained and optimized, and further evaluated on validation set of each dataset separately. The training and hyper-parameters setting are exactly similar to the baseline experiments. The performance of the model for each dataset is described in Table 4.4.

4.1.4. [Pixel-embedding Hierarchy] Hierarchical Semantic Segmentation Network with Pixel-embedding based Label Hierarchy on Mixed Dataset.

Since hand-picked hierarchy varies from person to person, the purpose of this experiment is to propose a method for constructing the label hierarchy based on network features and evaluate its performance against a hand-picked hierarchy. To achieve this, pixel-embedding based label hierarchy using K-means algorithm is experimented on mixed dataset. The hierarchy in the previous model Section 4.1.3 is replaced by pixel-embedding based label hierarchy shown in Figure B.2. This approach consists of two phases of training. In the first stage, a standard segmentation network is trained for the dataset and then, using projection head, backbone features for each class are extracted. In the second stage, a clustering algorithm is used on the basis of these features to derive a tree-based label hierarchy , which is then used to construct the hierarchical network. Since standard clustering algorithms tend to form unbalanced multi-level clusters, the regularized K-means algorithm [YGS19] is used and in order to determine the hyper-parameter for the number of clusters in the multi-level hierarchy, the number of multi-level clusters are taken from hand-picked hierarchy. Finally, from a human perspective, the pixel-embedding hierarchy yields a set of multi-level hierarchy with random subcategories in each super class. Now, to evaluate its perform against the hand-picked hierarchy, the network is trained with mixed dataset and evaluated for each dataset separately. The performance of this network is shown in Table 4.4.

Table 4.5.: *Hand-picked and pixel-embedding based hierarchical network result on Mapillary Vistas v2.0 dataset. These experiments are described in Section 4.1.5 and Section 4.1.6.*

Training Dataset	Validation Dataset	Experiment Name	Validation mIoU (%)
Mapillary Vistas v2.0	Mapillary Vistas v2.0	Baseline	28.64
		Hand-picked hierarchy	30.72
		Pixel-embedding hierarchy	28.15

4.1.5. [Hand-picked hierarchy]: Hierarchical Semantic Segmentation Network with Hand-picked Label Hierarchy on Mapillary Vistas 2.0 dataset.

So far, hierarchical network based experiments in Section 4.1.3 and Section 4.1.4 are conducted on mixed dataset and the performance of the hierarchical network does not show the consistent improvement or deterioration in the performance across the datasets. Therefore, the performance of hierarchical network is further analyzed using a single dataset. As part of this experiment, the hand-picked hierarchical network is trained with a single dataset and evaluated on its validation set. The hand-picked label hierarchy for the dataset is shown in Figure B.3. Once the classes are mapped, the network is trained with single dataset i.e. Mapillay Vista v2.0 for 240K iterations with SGD optimizer and polynomial learning rate. Mapillay Vistas v2.0 is selected due to its fine-grained label space and large data samples, which shows a very well hierarchical structure in the label space. The network performance on Mapillary Vistas v2.0 dataset is described in Table 4.5.

4.1.6. [Pixel-embedding Hierarchy] Hierarchical Semantic Segmentation Network with Pixel-embedding based Label Hierarchy on Mapillary Vistas 2.0 dataset.

This experiment is conducted on hierarchical semantic segmentation network, and the purpose of this experiment is to evaluated the pixel-embedding based hierarchy on a well-structured dataset. Therefore, this experiment uses the hierarchical network proposed in Section 3.3.1 with pixel-embedding based hierarchy for Mapillary Vistas v2.0 dataset and further evaluated on validation set. In order to construct the hierarchy based on network features, it requires two training phases as described in Section 4.1.4. The performance of the pixel-embedding based hierarchical network on Mapillary Vistas v2.0 is described in Table 4.5.

4.2. Qualitative & Quantitative Results Analysis

So far, Section 4.1 has covered the experimental setup, configuration and experimental results for baseline, cross-domain and hierarchical networks. In this section, a qualitative and quantitative analysis of the experimental results is presented. The analysis is divided into three subsections. First, the effectiveness of the cross-domain network on the baseline network is analyzed. Then, the performance of the hand-picked label hierarchy over the pixel-embedding based label hierarchy in the hierarchical network is analyzed. At the end, the hand-picked hierarchical network and cross-domain network are analyzed based on the overall performance of the dataset.

Table 4.6.: *Cross-domain and hierarchical model performance comparison with baseline model.*

	Baseline	Experiment: Cross-domain	Experiment: Hand-picked	Experiment: Pixel-embedding
Validation Dataset	% mIoU	+/- % mIoU	+/- % mIoU	+/- % mIoU
ADE20K	22.67	+ 11.22	+ 5.05	+ 6.22
Cityscapes	68.7	+ 5.99	+ 6.88	+ 6.01
VIPER	52.89	+ 5.83	+ 1.42	+ 7.82
Vistas v1.0	31.14	+ 8.2	+ 11.28	+ 10.56
Scannet	43.4	+ 8.88	+ 5.36	+ 7.62
Wilddash	48.7	+ 9.55	+ 8.9	+ 9.37
	Baseline		Experiment: 4	Experiment: 5
Vistas v2.0	28.64		+ 2.08	- 0.64

4.2.1. Baseline Vs Cross-Domain

Although both models described in Section 4.1.1 and Section 4.1.2 are trained identically and have the same hyperparameters, the cross-domain model trained on the mixed dataset shows a significant improvement in the average IoU of each dataset compared to the baseline model, as shown in Table 4.6. Compared with the baseline model, the cross-domain model showed the highest improvement for the ADE20K and Wilddash datasets, with 11.22% and 9.55%, respectively. In addition, the cross-domain model trained with the mixed dataset has 191 categories, which is higher than any model trained in baseline experiment. The higher number of categories in the -domain model should introduce more confusion between categories, but it does not seem to be the case here. There can be one possible reason for this: The increase in size of the dataset with the common label space. The mixed dataset includes ADE20K, Cityscapes, VIPER, Vistas v1.0, Scannet, Wilddash, BDD10K, and IDD, where most of the datasets are related to the road dataset, except for Scannet and ADE20K. Every dataset has different number of categories, and in order to construct the mixed dataset, these label spaces are converted to one common label space with 191 categories.

To investigate the impact of mixed dataset, baseline model and cross-domain model trained on ADE20K and mixed dataset, are considered for analysis. The ADE20K baseline model with 151 categories is trained on the ADE20K training set, which includes around 20K samples and evaluated on the ADE20K validation set. In contrast, the cross-domain model trained on the mixed dataset with 191 categories includes approx 190K samples and is evaluated on a ADE20K validation set. If the performance improvement is due to the size of the dataset, it may be due to one of these factors: (i.) The increase in mIoU due to the increase in the number of pixels in under-sampled (minority) categories, (ii.) An increase in the number of samples in the entire dataset. Table 4.7 shows that increasing the number of pixels in a category does not infer any reliable conclusion that it increases the IoU of that category (e.g. "mirror", "carpet"). But it also does not show the negative impact of the

Table 4.7.: *Impact of number of pixels count on the class IoU for baseline and cross-domain model.*

ADE20K	Experiment: Baseline		Experiment: Cross-Domain	
	# of pixels	% IoU	+/- in # of pixels	% IoU +/-
Road	1.8e+8	71.7	+ 1.08e+11	+ 9.32
Van	2.88e+6	2.7	+ 2.0e+8	+ 9.9
Bus	4.07e+6	43.9	+ 1.66e+9	+ 50.9
Pole	2.97e+6	13.4	+ 1.45e+9	+ 28.7
Flower	6.78e+6	5.69	+ 0.0	+ 34.01
Door	5.54e+7	19.72	+ 4.7e+9	+ 17.08
Table	5.16e+7	35.77	+ 5.55e+9	+ 15.13
Shelf	2.8e+7	27.60	+ 1.40e+9	+ 9.67
Mirror	2.4e+7	17.79	+ 0.0	+ 24.21
Carpet	2.11e+7	27.3	+ 0.0	+ 20.4
Armchair	2.09e+7	14.23	+ 0.0	+ 16.31

category having an additional number of pixels. On the other hand, it definitely says that increasing the number of samples in the dataset makes the network more effective for all categories, even if the number of pixels in that category is low. Now, it can be said that the increase in IoU is due to the increase in number of samples in mixed dataset and to an extent the effect of increase in number of pixels for some categories. Training with large dataset changes the network's backbone and decoder behaviour as well. When analysing the feature maps, it is found that the baseline model trained with ADE20K dataset shows less confidence on object segmentation, while cross-domain model trained with mixed dataset shows higher confidence on segmentation. As an example, for "street light" category, feature maps of cross-domain model are confident than the baseline model as shown in Figure 4.1. Both the models have approx 60.157M parameters but model with mixed dataset takes larger time to train the model due to large number of samples compared to model trained with single dataset.

4.2.2. Hand-picked Vs Pixel-embedding based Hierarchical Network

Hand-picked and pixel-embedding based label hierarchy methods for hierarchical network on mixed dataset is described in Section 4.1.3 & Section 4.1.4 respectively. In addition, another hierarchical network trained with Mapillary Vistas v2.0 dataset is described in Section 4.1.5 & Section 4.1.6 for hand-picked and pixed-embedding label hierarchy. In this section, both of these label hierarchy approaches are analyzed from two perspectives: (i.) Model trained on mixed dataset (ii.) Model trained on single dataset.

The limitation of this hierarchical semantic segmentation network is that if a category is not recognized by the top level segmentation heads, then

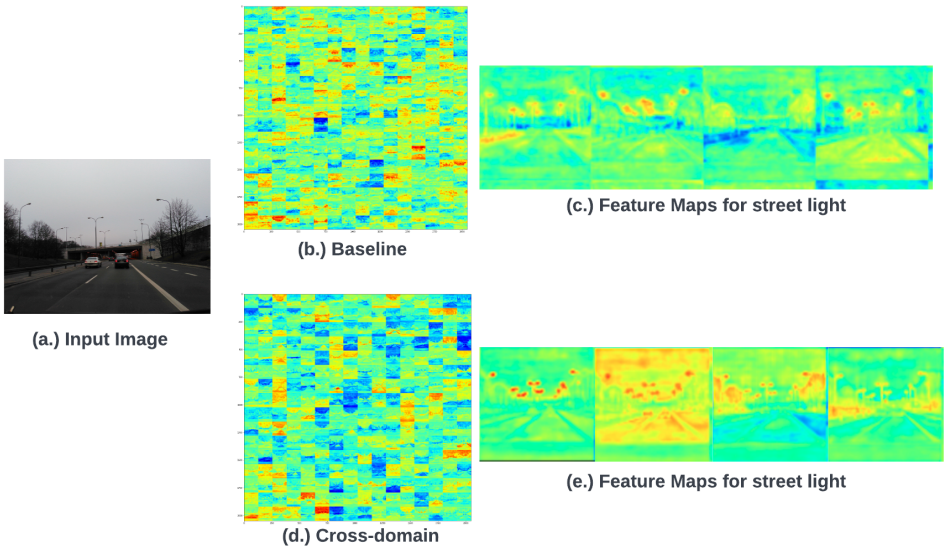


Figure 4.1.: *Comparison of feature maps: (a.) represent the input image that is passed through the baseline and cross-domain model. All 512 channels of the last convolutional layer of segmentation head of baseline and cross-domain network are shown in (b.) & (d.), respectively. Additionally, the feature maps focused on the streetlights are extracted for both models and shown in (c.) and (e.).*

even if the bottom-level segmentation head recognizes the object, it cannot be taken into account in the final output mask. More details about the flow of inference is mentioned in Section 3.3.1. Therefore, the performance of hand-picked and pixel-embedding hierarchical model are analyzed on the basis of top level segmentation head performance and that infers that the hand-picked label hierarchy is better for the datasets, where the subcategories are semantically related to each other within a top level categories, such as for Cityscapes, Vistas datasets. The Cityscapes and Vistas datasets have "vehicle", "flat", "construction", "traffic objects", "nature objects" and "VRU" as super categories in the label hierarchy, and sub-categories within these are semantically related and have less variability within the subcategories compared to other datasets. For the Vistas v1.0 and Cityscapes datasets, the hand-selection model improves over the cross-domain model by 3.08% and 0.89%, respectively, and over the pixel-embedding model by 0.72% and 0.87%. Mapillary Vistas v2.0 trained on single dataset supports the above statement by showing an improvement of 2.72% over pixel-embedding model. The VIPER dataset could also have shown better performance incomparison to pixel-embedding based hierarchy but due to ignorance of "infrastructure" category in the hand-picked label hierarchy makes it incomparable.

On other hand, pixel-embedding based hierarchy show a better improvement over the hand-picked hierarchy for datasets such as Scannet, ADE20K and Wilddash. By analysing the performance of every segmentation head and

samples of top level categories for each dataset, the generalized categories or top level categories in the label hierarchy of Scannet, ADE20K and Wilddash dataset contains subcategories that have less semantic relationships with each other and have higher object variability within the subcategories. As an example, top level category in the label hierarchy "furniture" in ADE20K dataset consists of subcategories "bed", "sofa", "counter", "coffee table" are shown in Figure 4.2. These subcategories within the "furniture" category varies in shape, appearance and their location in visual field compared to subcategories of generalized categories in the label hierarchy of Cityscapes, VIPER, and Vistas datasets. This infers that the pixel-based hierarchy perform better for the datasets, that have distinct categories and that it is difficult to construct hand-picked label hierarchy based on similarity.

Figure 4.2.: *Top level category "Furniture" in label hierarchy of mixed dataset is consist of subcategories "box", "chair", "bench", "window" and so on. Every sub-category in this category varies in size, appearance and their location in visual field.*

4.2.3. Hand-picked hierarchical Vs Cross-domain Network

According to the previous analysis, the cross-domain model performs better than the baseline model on each dataset, while the hand-picked label hierarchy performs better than the pixel-embedding based label hierarchy on Cityscapes, and Vistas or well hierarchically structured datasets. Furthermore, in this section, the hand-picked hierarchical semantic segmentation is compared with cross-domain semantic segmentation network on various datasets. The cross-domain and hand-picked experiments described in Section 4.1.2 and Section 4.1.3, respectively, are trained with mixed dataset and evaluated separately on each

datasets as shown in Table 4.6 and experiment preseneted in Sections 4.1.5 to 4.1.6 are trained with Mapillary Vistas v2.0 and tested on its validation set.

Table 4.6 shows that the cross-domain experiment shows better performance for ADE20K, Scannet, and Wilddash datasets, with an improvement of 6.17%, 3.52% and 0.65% IoU over hand-picked hierarchical model, respectively. The VIPER dataset also shows improvement, but the main reason for this is the presence of "infrastructure" category in cross-domain experiment. On other side, Cityscapes and Vistas v1.0 show improvement of 0.89% and 3.08% in overall IoU over cross-domain model.

After analyzing the performance of various levels of segmentation heads and samples analysis, its been found that the hand-picked hierarchy work better than cross-domain models for those datasets where the top level categories in the label hierarchy have semantic similarity in its subcategories. A further improvement in the performance of hand-picked label hierarchy is found for the datasets that has a well hierarchical structure and fine-grained annotations. Additionally, the multi-level clusters of label hierarchy in Cityscapes and Vistas datasets have been found to be more closely related than the indoor and Wilddash dataset. The hand-picked hierarchy shows a significant improvement of 2.08% over baseline model for Mapillary Vistas v2.0, and 3.08% improvement for Vistas v1.0 over cross-domain model. This indicate that the performance of hand-picked label hierarchy can be further improved with large hierarchical structured and fine-annotated dataset for hierarchical network. Another advantage of using a hierarchical model is the improvement in identifying difficult classes or the worst performing classes of the cross-domain and baseline model. The hierarchical model performs better for the minority categories, such as the identification of the "flag" and "ship" categories, by 26% and 29% over the baseline and cross-domain models, respectively. The comparative study of the performance of the minority and majority categories are described in Appendix C. Sample images from road dataset and indoor datasets, as well as the output masks from each experiments, are shown in Figure 4.3 and Figure 4.4, respectively.

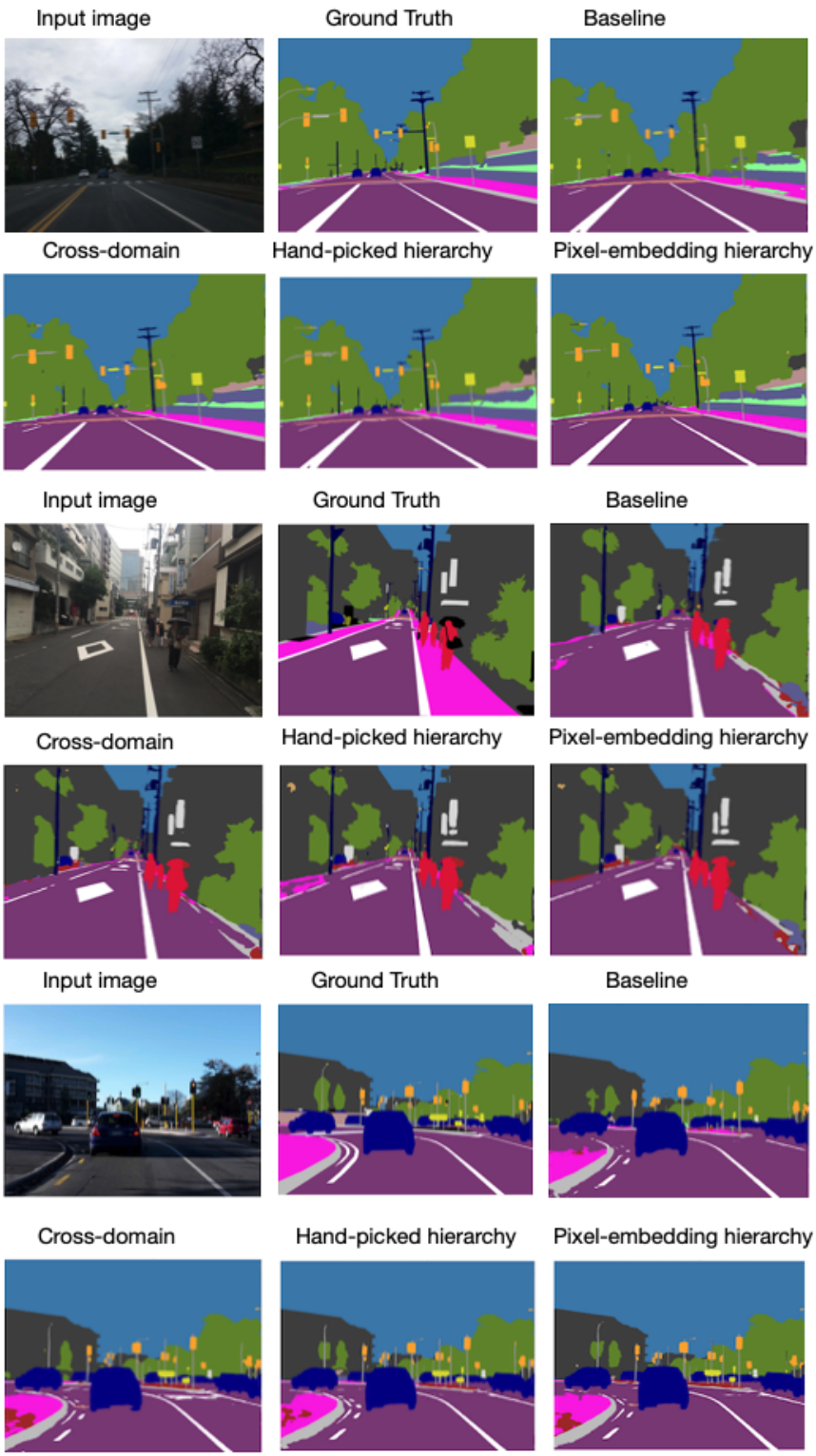


Figure 4.3.: *Vistas v1.0: Input image and models output masks from various experiments. The output masks are quite similar from different experiments, but a closer analysis shows that hierarchical experiments show better segmentation masks.*

Figure 4.4.: *ADE20K: Input image and models output masks from the experiments. For the first sample, the pixel-embedding hierarchy shows better segmentation masks, while for the second sample, the cross-domain and pixel-embedding hierarchy experiment perform better.*

5. Conclusion and Future Works

This chapter summarizes the contributions of the work and possible extensions which can be considered for future research.

5.1. Conclusion

This work investigates the impact of multiple heterogeneous datasets and the proposed hierarchical network based on DeepLabv3+ to overcome the generality and absence of symbolic AI from deep learning methods. First, the baseline network based on DeepLabv3+ was trained and evaluated on single dataset. Second, to address the unavailability of large multi-domain dataset, a large multi-domain dataset was build based on simple mapping technique without manually relabelling the labels and cross-domain network based on DeepLabv3+ was trained on this large dataset. For the large dataset, ADE20K, VIPER, Scannet, Cityscapes, Vistas and Wilddash datasets were concatenated to form one dataset with common label space and later this was used for experiments. Third, hierarchical semantic segmentation network were proposed to integrate categories relations into network architecture based on hand-picked and pixel-embedding based label hierarchy. The behaviour of hierarchical networks were captured on large multi-domain and single Mapillary Vistas v2.0 dataset against the hand-picked and pixel-embedding based label hierarchy. The cross-domain network uses fewer parameter and computational power compared to hierarchical networks. The parameters and computational power of hierarchical networks vary with the underlying label hierarchy of the dataset, the depth of the label hierarchy, and the shared layers. The cross-domain network outperformed the baseline network for all dataset, while hand-picked label hierarchy based hierarchical network outperform every other methods for well-structured datasets i.e. Mapillary Vistas and Cityscapes, but the network required relatively higher parameter and computational power. The hierarchical network have also shown effectiveness in improving the segmentation of minority and difficult classes. On other hand, pixel-embedding label hierarchy based hierarchical network outperformed over handpicked-hierarchy based hierarchical network and underperformed over cross-domain network for the datasets, that has weak correlations in categories. This work have proven that the performance of the network can be improved by symbolic AI in deep learning methods, while further research is needed for the purpose of generalized models.

5.2. Future works

- One problem that plagues all experiments with respect to Mixed dataset is the naive flat universal taxonomy, where dataset-specific categories are mapped to subsets of a universal set of disjoint elementary classes without considering the abnormalities in object annotation. In order to overcome this problem, manually relabelled dataset e.g. MSeg [Lam+20] or heuristic principles can be used.

- Cross-domain approach lacks in the performance of the minority classes and small object categories, this can be improved by increase the receptive field of the neurons or the balancing mechanism of the dataset.

- The bottleneck of proposed hierarchical method is their dependence on the higher-level segmentation head, which can be reduced by using a two-level label hierarchy or by considering the output of the lower-level segmentation head, even if the higher-level segmentation head does not identify objects by using confidence scoring methods.

- In the pixel embedding based label hierarchy approach, multi-level label hierarchies are constructed based on the features extracted from the backbone. This approach does not consider the semantic relationships between categories and the number of their occurrences in the dataset when designing the multilevel label hierarchy, and can be improved by adding knowledge of the semantic relationships from the human perspective and the number of pixels in the categories in the dataset.

- Hierarchical network loss with a tunable hyperparameter λ. All experiments were performed with the default parameters specified in the base paper, and performance may be improved if these parameters are tuned according to the dataset and its labeling hierarchy.

- The cross-domain & hierarchical approach still suffers from some false positives, which will be a source of interference when applied to real autonomous driving scenarios. Reducing this false positive rate could be a focus of research.

- Finally, the possibility of sharing features from parent to child segmentation heads in the label hierarchy can also be verified in the hierarchical network.

Bibliography

[Ber+20] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. "Making better mistakes: Leveraging class hierarchies with deep networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12506–12515.

[Bil+17] Alsallakh Bilal, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. "Do convolutional neural networks learn class hierarchy?" In: *IEEE transactions on visualization and computer graphics* 24.1 (2017), pp. 152–162.

[BKC17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.

[Cha+21] Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. "Your" Flamingo" is My" Bird": Fine-Grained, or Not". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11476–11485.

[Che+14] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. "Semantic image segmentation with deep convolutional nets and fully connected crfs". In: *arXiv preprint arXiv:1412.7062* (2014).

[Che+17] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848.

[Che+18a] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 801–818.

[Che+18b] Tianshui Chen, Wenxi Wu, Yuefang Gao, Le Dong, Xiaonan Luo, and Liang Lin. "Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding". In: *Proceedings of the 26th ACM international conference on Multimedia*. 2018, pp. 2023–2031.

[Con20] MMSegmentation Contributors. *OpenMMLab Semantic Segmentation Toolbox and Benchmark*. Version v1.2.0. Aug. 2020. URL: `https://github.com/open-mmlab/mmsegmentation`.

[Cor+16] 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". In: *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*. 2016, pp. 3213–3223.

[CSK21] Bowen Cheng, Alex Schwing, and Alexander Kirillov. "Per-pixel classification is not all you need for semantic segmentation". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 17864–17875.

[Dai+17a] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes". In: *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*. 2017.

[Dai+17b] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. "Deformable Convolutional Networks". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[Den+14] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. "Large-scale object classification using label relation graphs". In: *European conference on computer vision*. Springer. 2014, pp. 48–64.

[Din+19] Henghui Ding, Xudong Jiang, Ai Qun Liu, Nadia Magnenat Thalmann, and Gang Wang. "Boundary-aware feature propagation for scene segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 6819–6829.

[Fro+13] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. "Devise: A deep visual-semantic embedding model". In: *Advances in neural information processing systems* 26 (2013).

[Fu+19] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. "Dual attention network for scene segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 3146–3154.

[Gir+14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[Har+14] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. "Simultaneous detection and segmentation". In: *European conference on computer vision*. Springer. 2014, pp. 297–312.

[HDK17] Adam W Harley, Konstantinos G Derpanis, and Iasonas Kokkinos. "Segmentation-aware convolutional networks using local attention masks". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 5038–5047.

[He+19a] Junjun He, Zhongying Deng, Lei Zhou, Yali Wang, and Yu Qiao. "Adaptive pyramid context network for semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7519–7528.

[He+19b] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. "Bag of tricks for image classification with convolutional neural networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 558–567.

[Hua+19a] Lang Huang, Yuhui Yuan, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. "Interlaced sparse self-attention for semantic segmentation". In: *arXiv preprint arXiv:1907.12273* (2019).

[Hua+19b] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. "Ccnet: Criss-cross attention for semantic segmentation". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 603–612.

[Iqb18] Haris Iqbal. *HarisIqbal88/PlotNeuralNet v1.0.0*. Version v1.0.0. Dec. 2018. DOI: 10.5281/zenodo.2526396. URL: https://doi.org/10.5281/zenodo.2526396.

[Kai+19] Daniel Kaiser, Genevieve L Quek, Radoslaw M Cichy, and Marius V Peelen. "Object vision in a structured world". In: *Trends in cognitive sciences* 23.8 (2019), pp. 672–685.

[Lam+20] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. "MSeg: A composite dataset for multi-domain semantic segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2879–2888.

[Li+18] Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. "Pyramid attention network for semantic segmentation". In: *arXiv preprint arXiv:1805.10180* (2018).

[Li+19] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. "Expectation-maximization attention networks for semantic segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9167–9176.

[Li+20] Xiangtai Li, Xia Li, Li Zhang, Guangliang Cheng, Jianping Shi, Zhouchen Lin, Shaohua Tan, and Yunhai Tong. "Improving semantic segmentation via decoupled body and edge supervision". In: *European Conference on Computer Vision*. Springer. 2020, pp. 435–452.

[Li+22] Liulei Li, Tianfei Zhou, Wenguan Wang, Jianwu Li, and Yi Yang. "Deep Hierarchical Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1246–1257.

[Lin+17] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1925–1934.

[Liu+21] 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". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10012–10022.

[Llo82] Stuart Lloyd. "Least squares quantization in PCM". In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[Mik+13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[MS20] Bruce Muller and William A. P. Smith. "A Hierarchical Loss for Semantic Segmentation". In: *Proc. of the International Conference on Computer Vision Theory and Applications (VISAPP)*. Vol. 4. 2020, pp. 260–267.

[Neu+17] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. "The mapillary vistas dataset for semantic understanding of street scenes". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 4990–4999.

[NHH15] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. "Learning deconvolution network for semantic segmentation". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1520–1528.

[Ots79] Nobuyuki Otsu. "A threshold selection method from gray-level histograms". In: *IEEE transactions on systems, man, and cybernetics* 9.1 (1979), pp. 62–66.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[RHK17] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. "Playing for benchmarks". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2213–2222.

[SGG16] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. "Training region-based object detectors with online hard example mining". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 761–769.

[SM+12] Charles Sutton, Andrew McCallum, et al. "An introduction to conditional random fields". In: *Foundations and Trends® in Machine Learning* 4.4 (2012), pp. 267–373.

[Var+19] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. "IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019, pp. 1743–1751.

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[VS91] Luc Vincent and Pierre Soille. "Watersheds in digital spaces: an efficient algorithm based on immersion simulations". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 13.06 (1991), pp. 583–598.

[Wan+18] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. "Non-local neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7794–7803.

[Wu+16] Hui Wu, Michele Merler, Rosario Uceda-Sosa, and John R Smith. "Learning to make better mistakes: Semantics-aware visual food recognition". In: *Proceedings of the 24th ACM international conference on Multimedia*. 2016, pp. 172–176.

[Xie+21] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. "SegFormer: Simple and efficient design for semantic segmentation with transformers". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12077–12090.

[Xu+22] Jing Xu, Wentao Shi, Pan Gao, Zhengwei Wang, and Qizhu Li. *UperFormer: A Multi-scale Transformer-based Decoder for Semantic Segmentation*. 2022. DOI: 10.48550/ARXIV.2211.13928. URL: https://arxiv.org/abs/2211.13928.

[Yan+18] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. "DenseASPP for Semantic Segmentation in Street Scenes". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[YCW20] Yuhui Yuan, Xilin Chen, and Jingdong Wang. "Object-contextual representations for semantic segmentation". In: *European conference on computer vision*. Springer. 2020, pp. 173–190.

[YGS19] Ke Yang, Vasilis Gkatzelis, and Julia Stoyanovich. "Balanced ranking with diversity constraints". In: *arXiv preprint arXiv:1906.01747* (2019).

[YK15] Fisher Yu and Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions". In: *arXiv preprint arXiv:1511.07122* (2015).

[Yu+20] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. "Bdd100k: A diverse driving dataset for heterogeneous multitask learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2636–2645.

[Yua+20] Yuhui Yuan, Jingyi Xie, Xilin Chen, and Jingdong Wang. "Segfix: Model-agnostic boundary refinement for segmentation". In: *European Conference on Computer Vision*. Springer. 2020, pp. 489–506.

[Zen+18] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernandez Dominguez. "Wilddash-creating hazard-aware benchmarks". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 402–416.

[Zha+17] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. "Pyramid scene parsing network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2881–2890.

[Zha+18] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. "Psanet: Point-wise spatial attention network for scene parsing". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 267–283.

[Zhe+21] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 6881–6890.

[Zho+17] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. "Scene parsing through ADE20K dataset". In: *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*. 2017.

[Zhu+19] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. "Asymmetric non-local neural networks for semantic segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 593–602.

A. Detailed Dataset Overview

This section provides an overview of the dataset samples, their annotations, and the pixel distribution in the Mixed dataset. Each dataset provides pixel-level annotation and labeling space information.

The VIPER dataset labels vehicle occupants as "people", while "poles", "utility poles", "bridges", and "tunnels" are mapped to the "infrastructure" category. Some of the samples are shown in Figure A.1. The cityscape dataset provides annotations for the data samples, with coarse categories divided into 34 categories. For example, "road markings" and "curb/curb cuts" are part of the sidewalk. Samples from the Cityscapes dataset are shown in Figure A.3. The Mapillary Vistas dataset finely annotates the scenes, which results in an increased number of categories in the label space. Roadside objects are finely annotated as "trash cans", "catch basins", "connection boxes", "cctv-cameta", etc. Even potholes and manholes are annotated in the scenes. The upgraded annotations provide a more refined level of annotation in Mapillary Vistas v2.0 dataset. Figure A.2 shows the samples of Mapillary Vistas v1. On the other hand, the wilddash dataset provides unstructured samples from around the world. It is designed to test the robustness of computer vision algorithms. Although the classes of the dataset are similar to Cityscape, the variability of the samples is high. Wilddash samples are shown in Figure A.4.

The ADE20K dataset includes both outdoor and indoor images. The masks of these images are annotated for each object and its parts. Some of these samples are shown in Figure 4.4. Another dataset: Scannet dataset includes only indoor images with coarse classification masks. Most of the categories are the same as in the ADE20K dataset. The dataset samples and their segmentation masks are shown in Figure A.6.

The Mixed dataset is a hybrid dataset that provides a label space with 191 categories. These categories are derived from the indoor and road datasets. Each dataset contributes to the high variability and annotation style of the objects in this dataset. To analyze the distribution of categories in this dataset, the pixel distribution of these categories compared to the number of pixels is shown in Figure A.7.

Figure A.1.: *VIPER dataset samples and its corresponding segmented mask in second row. Dataset also provide annotation of humans inside the vehicles.[RHK17]*

Figure A.2.: *Mapillary Vistas v1 images and its fine grained annotation of road lanes and traffic objects.[Neu+17]*

Figure A.3.: *Examples of cityscapes images and its corresponding segmentation mask. Cityscapes provides coarse segmentation mask of road side scenes.[Cor+16]*

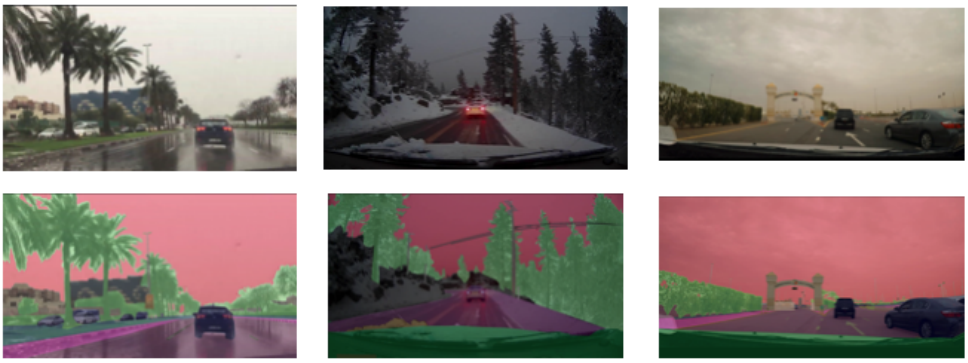
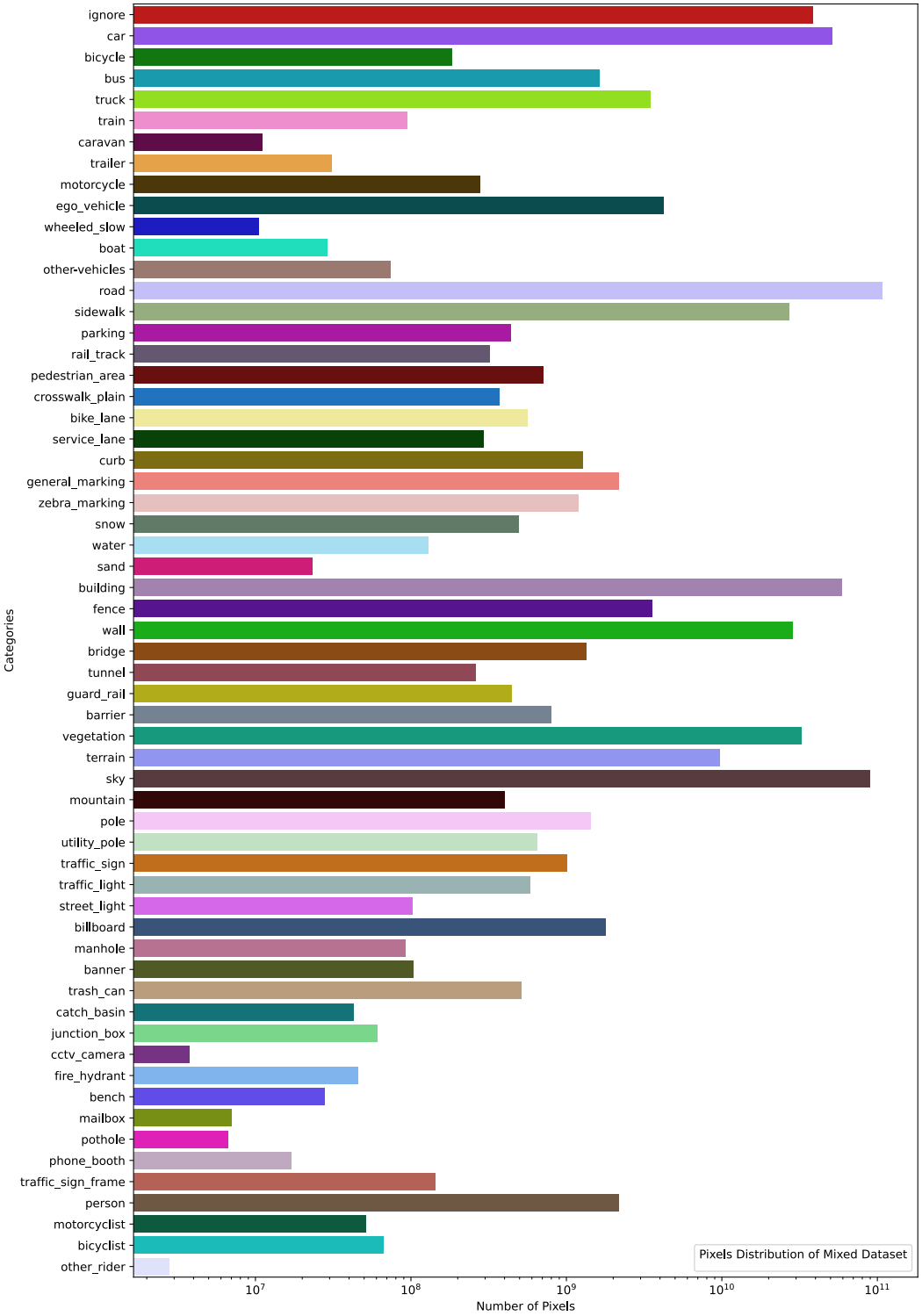
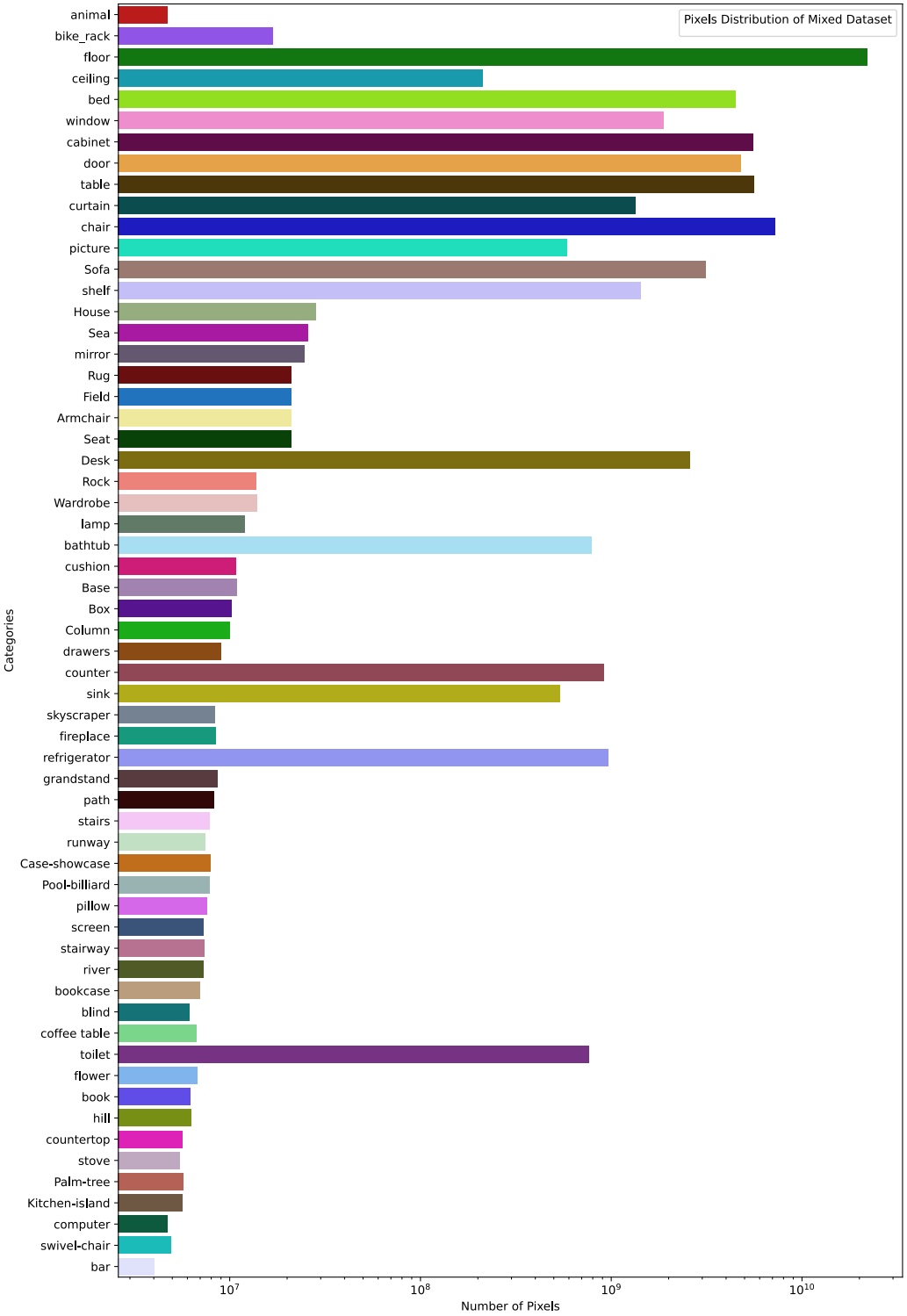


Figure A.4.: *Wilddash samples and its segmentation mask. Dataset provides images from harsh conditions.[Zen+18]*

Figure A.5.: *Examples of ADE20K images and its corresponding semantic label maps from indoor and outdoor scenes.[Zho+17]*

Figure A.6.: *Scannet images and its corresponding semantic label maps. [Dai+17a]*


ignore
car
bicycle
bus
truck
train
caravan
trailer
motorcycle
ego_vehicle
wheeled_slow
boat
other-vehicles
road
sidewalk
parking
rail_track
pedestrian_area
crosswalk_plain
bike_lane
service_lane
curb
general_marking
zebra_marking
snow
water
sand
building
fence
wall
bridge
tunnel
guard_rail
barrier
vegetation
terrain
sky
mountain
pole
utility_pole
traffic_sign
traffic_light
street_light
billboard
manhole
banner
trash_can
catch_basin
junction_box
cctv_camera
fire_hydrant
bench
mailbox
pothole
phone_booth
traffic_sign_frame
person
motorcyclist
bicyclist
other_rider
Categories
Pixels Distribution of Mixed Dataset
10^7
10^8
10^9
10^{10}
10^{11}
Number of Pixels


Pixels Distribution of Mixed Dataset
animal
bike_rack
floor
ceiling
bed
window
cabinet
door
table
curtain
chair
picture
Sofa
shelf
House
Sea
mirror
Rug
Field
Armchair
Seat
Desk
Rock
Wardrobe
lamp
bathtub
cushion
Base
Box
Column
drawers
counter
sink
skyscraper
fireplace
refrigerator
grandstand
path
stairs
runway
Case-showcase
Pool-billiard
pillow
screen
stairway
river
bookcase
blind
coffee table
toilet
flower
book
hill
countertop
stove
Palm-tree
Kitchen-island
computer
swivel-chair
bar
Categories
10^7
10^8
10^9
10^{10}
Number of Pixels

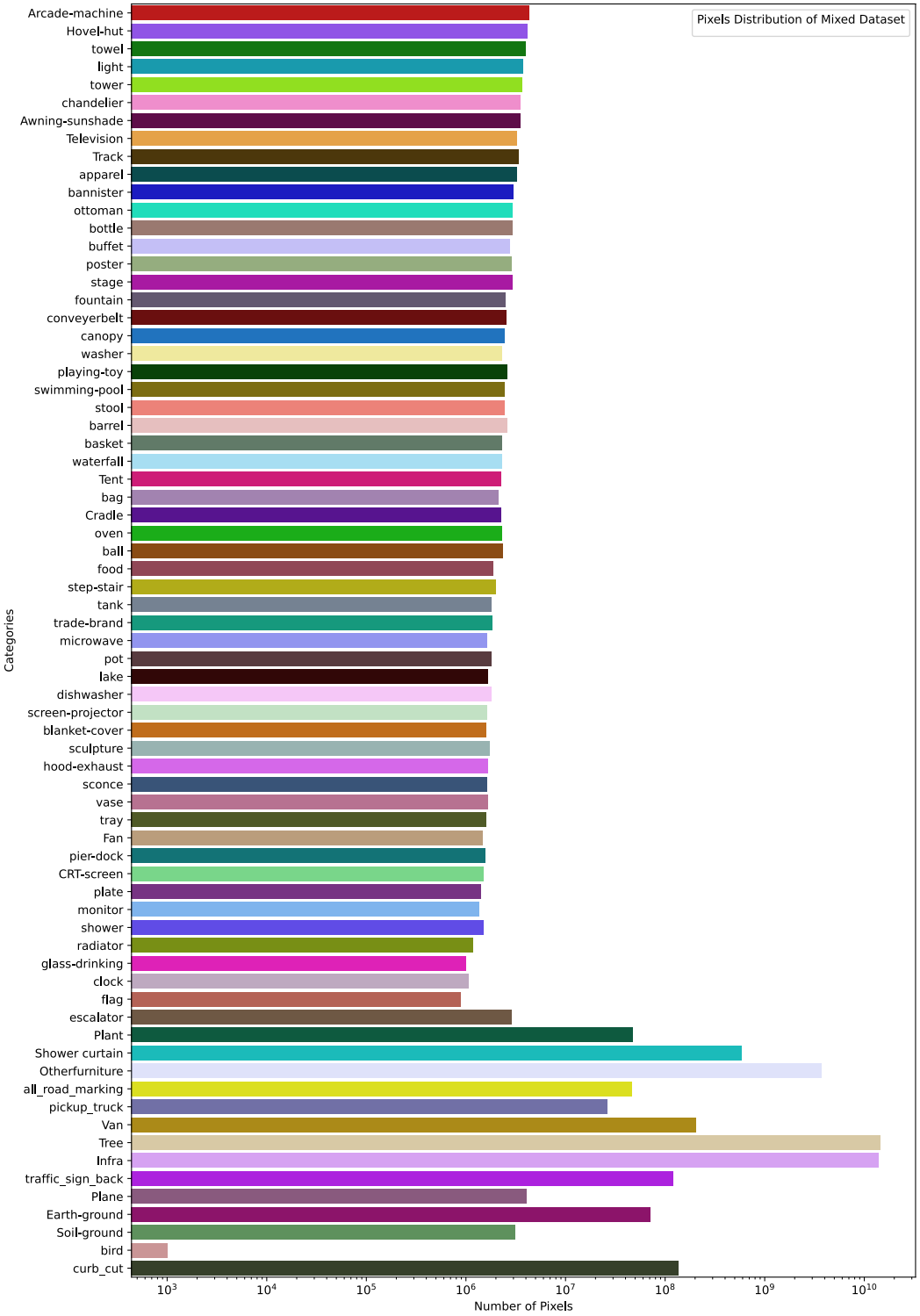


Figure A.7.: *Mixed dataset: Categories distribution against the number of pixels.*

B. Label Hierarchy

The hand-picked label hierarchy for Mixed dataset is constructed by aggregating the semantically similar categories into one group. For this dataset three level label hierarchy was introduced, which resulted into 7 super-categories at first level, 23 categories at second level, and fine-grained categories at the bottom stage. Hand-picked label hierarchy for Mixed dataset is shown in Figure B.1. Similary, hand-picked label hierarchy is constructed for Mapillary Vistas v2.0 dataset as shown in Figure B.3.

The pixel-embedding based label hierarchy for Mixed dataset is based on network understanding of features for the category. These features are further used to design three level clustering through regularized K-means algorithm. The pixel-embedding based label hierarchy is shown in Figure B.2. The categories within the clusters do not show any similarity from the humans perspective.

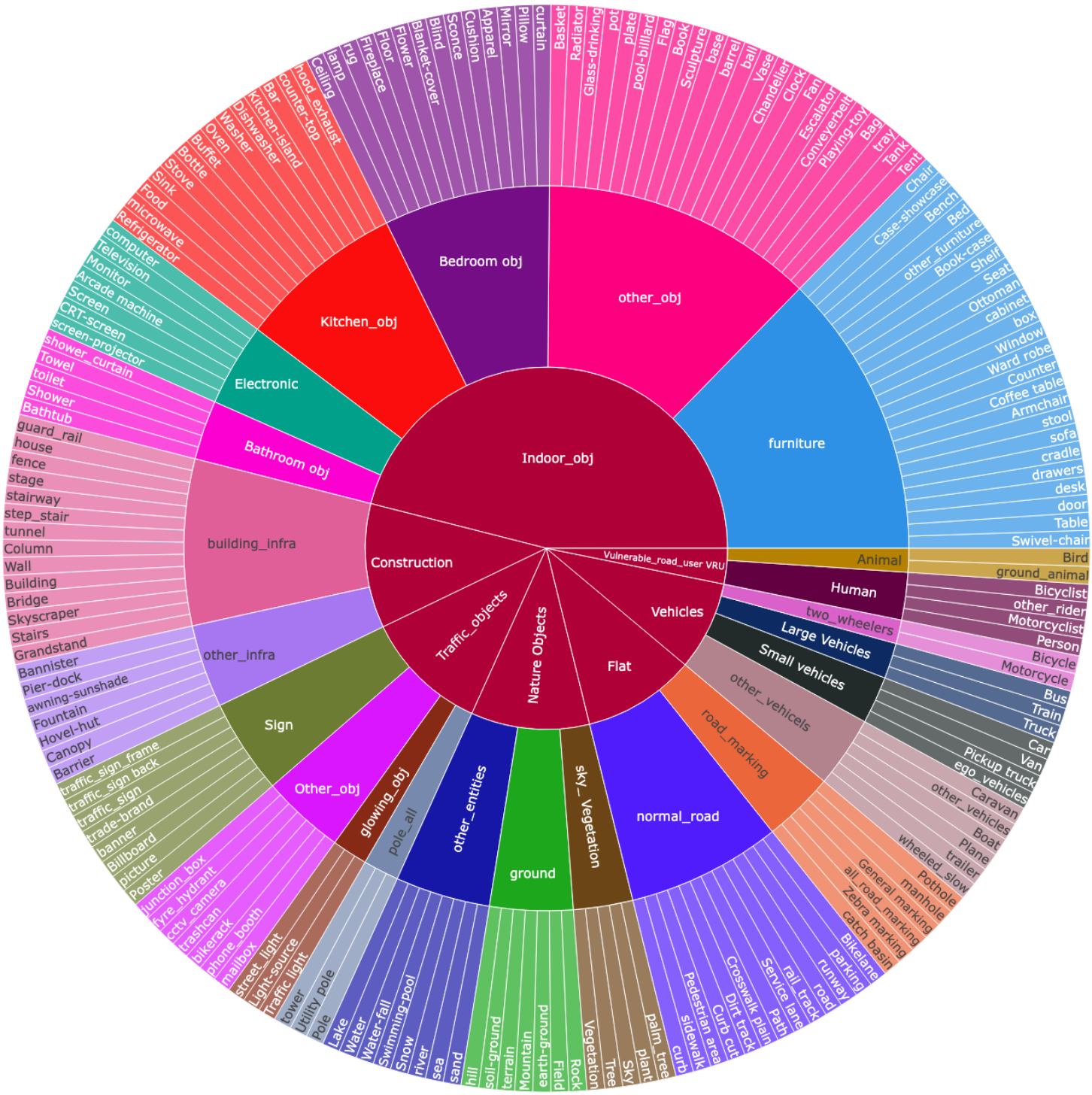


Figure B.1.: *Hand-picked label hierarchy for Mixed dataset. The inner circle consist of 7 super classes represent the top level in the hierarchy and out most circle represent the fine-grained classes in the dataset.*

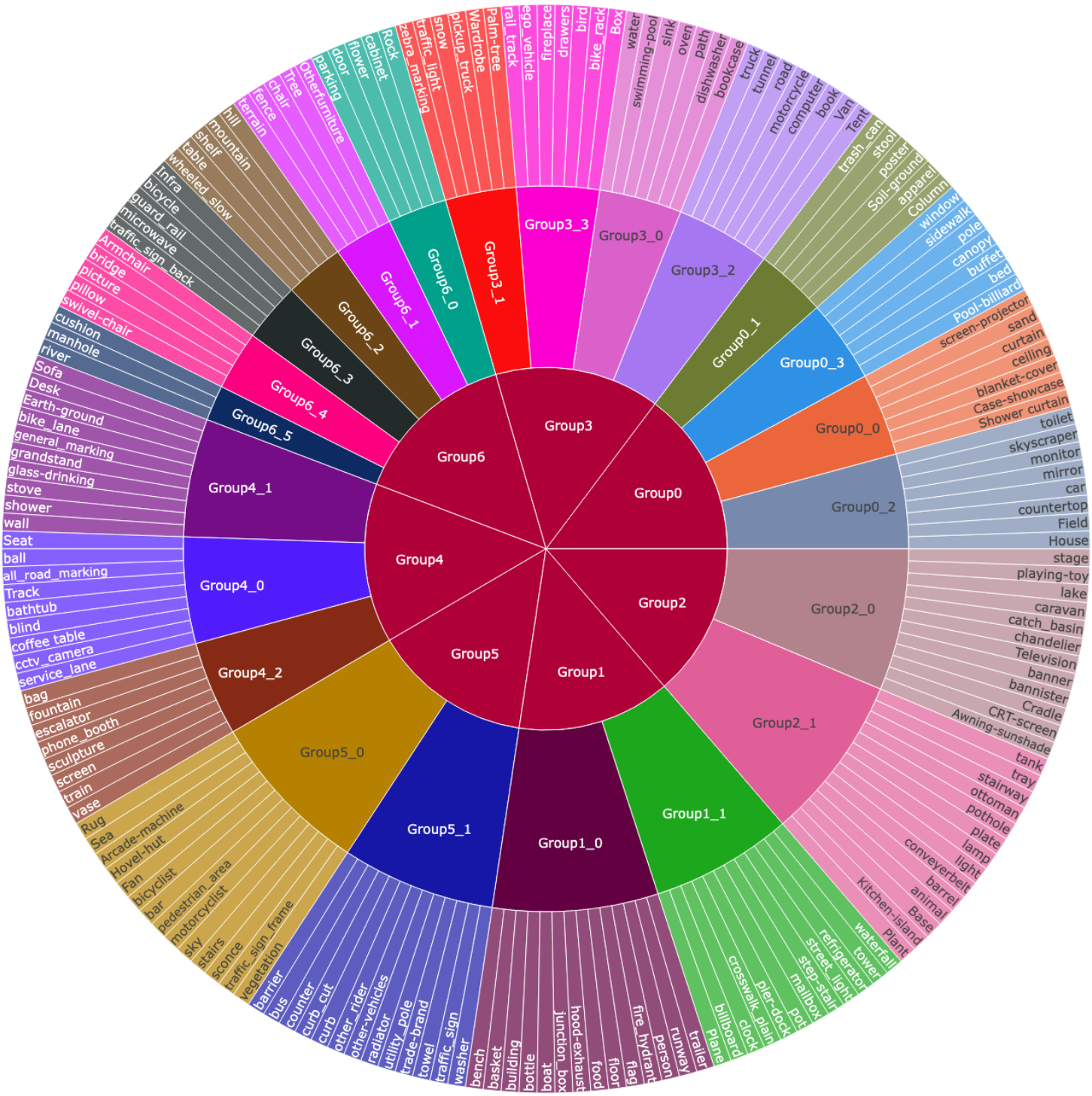


Figure B.2.: *Pixel-embedding based label hierarchy for Mixed dataset.*

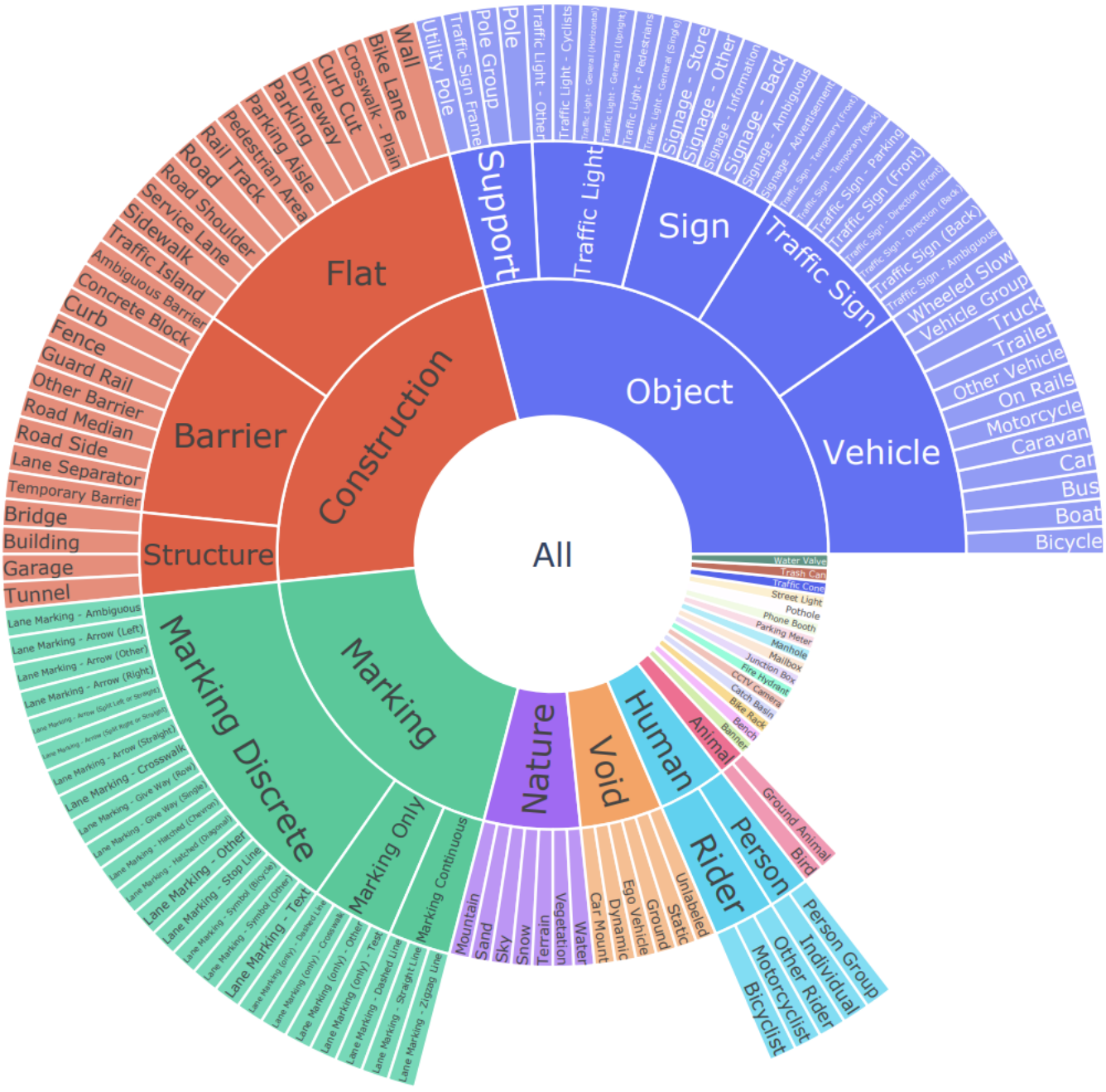


Figure B.3.: *Hand-picked label hierarchy for Mapillary Vistas v2.0 dataset [Li+22].*

C. Detailed Evaluation

In this section, the experiments are compared against the baseline model top performing and worst performing categories.

For ADE20K, Scannet, and Wilddash, cross-domain experiment shows overall better results except for few categories. Performance of these categories are shown in Table C.1, Table C.5, Table C.6. As discussed in Chapter 4, that the hand-picked hierarchy performs better for structured datasets. This conclusion is also supported by the results shown in Table C.2, Table C.3, Table C.4.

Table C.1.: *ADE20K: Best & worst performing categories of baseline model are compared against the performance of other experiments.*

	Categories	Baseline mIoU %	Experiment: Cross-domain	Experiment: Hand-picked hierarchy	Experiment: Pixel-embedding hierarchy
0	flag	0	0	26	0
1	bag	0	0	3	1
2	barrel-	0	9	0	0
3	stool	0	12	0	19
4	canopy	0	1	0	0
5	bannister-	0	2	1	3
6	land-	0	0	0	1
7	dirt-track	0	1	0	0
8	booth-	0	30	20	37
9	clock	0	0	15	20
10	sky	91	94	93	94
11	building-	74	79	78	78
12	billiard-	72	91	82	85
13	road-	72	81	77	80
14	floor	69	77	75	76
15	car-	68	82	83	81
16	tree	65	70	70	69
17	ceiling	64	74	75	74
18	grass	64	64	66	64
19	wall	63	72	70	70

Table C.2.: *Cityscapes: Best & worst performing categories of baseline model are compared against the performance of other experiments.*

	categories	Baseline	Experiment: Cross-domain	Experiment: Hand-picked hierarchy	Experiment: Pixel-embedding hierarchy
0	motorcycle	43	62	67	66
1	wall	44	61	62	62
2	fence	49	64	64	64
3	rider	53	59	64	61
4	truck	54	82	83	83
5	terrain	57	66	63	65
6	traffic_light	57	71	74	71
7	train	58	80	74	70
8	pole	58	67	69	67
9	traffic_sign	70	79	81	79
10	road	97	98	98	98
11	sky	94	95	95	95
12	car	92	95	96	96
13	vegetation	91	93	93	93
14	building	90	93	93	93
15	sidewalk	80	86	85	86
16	person	75	81	83	82
17	bus	72	87	88	87
18	bicycle	71	76	78	77
19	traffic_sign	70	79	81	79

Table C.3.: *VIPER: Best & worst performing categories of baseline model are compared against the performance of other experiments.*

	Categories	Baseline	Experiment: Cross-domain	Experiment: Hand-picked hierarchy	Experiment: Pixel-embedding hierarchy
0	trash	0	0	0	0
1	mobilebarrier	0	19	28	39
2	van	10	15	19	19
3	chair	11	18	30	24
4	fence	19	30	28	30
5	billboard	31	41	39	41
6	motorcycle	42	66	68	69
7	trafficsign	46	61	50	63
8	trafficlight	47	65	53	64
9	infrastructure	53	63	1	65
10	car	96	97	97	97
11	road	95	96	96	96
12	sky	94	96	94	96
13	sidewalk	86	90	89	90
14	truck	79	88	86	88
15	building	77	81	70	82
16	bus	74	80	84	82
17	tree	73	78	74	77
18	vegetation	60	64	60	66
19	terrain	59	63	59	62

Table C.4.: *Vistas v1.0: Best & worst performing categories of baseline model are compared against the performance of other experiments.*

	Categories	Baseline	Experiment: Cross-domain	Experiment: Hand-picked hierarchy	Experiment: Pixel-embedding hierarchy
0	banner	0	7	26	20
1	car_mount	0	0	0	0
2	wheeled_slow	0	0	5	13
3	trailer	0	4	1	7
4	caravan	0	13	0	13
5	boat	0	0	29	2
6	pothole	0	0	0	0
7	phone_booth	0	3	8	2
8	mailbox	0	0	11	4
9	fire_hydrant	0	24	52	39
10	sky	97	98	98	98
11	ego_vehicle	87	91	91	90
12	vegetation	87	89	89	89
13	car	85	90	90	89
14	road	83	86	85	85
15	building	82	86	87	86
16	snow	74	78	81	80
17	bridge	65	74	71	74
18	bus	64	78	75	78
19	lane_marking	62	67	66	65

Table C.5.: *Scannet: Best & worst performing categories of baseline model are compared against the performance of other experiments.*

	Categories	Baseline	Experiment: Cross-domain	Experiment: Hand-picked hierarchy	Experiment: Pixel-embedding hierarchy
0	picture	30	43	45	35
1	counter	35	44	42	48
2	window	36	42	37	42
3	bookshelf	36	40	36	36
4	curtain	38	43	37	47
5	door	41	55	49	50
6	cabinet	45	54	48	50
7	chair	45	62	54	56
8	sofa	52	69	63	69
9	table	52	62	58	58
10	floor	73	79	78	81
11	bed	60	66	61	67
12	wall	55	63	66	64
13	showercurtain	53	62	59	60
14	table	52	62	58	58
15	sofa	52	69	63	69
16	chair	45	62	54	56
17	cabinet	45	54	48	50
18	door	41	55	49	50
19	curtain	38	43	37	47

Table C.6.: *Wilddash: Best & worst performing categories of baseline model are compared against the performance of other experiments.*

	Categories	Baseline	Experiment: Cross-domain	Experiment: Hand-picked hierarchy	Experiment: Pixel-embedding hierarchy
0	pickup-truck	26	56	55	53
1	fence	27	46	45	46
2	van	28	34	34	36
3	traffic_sign	28	56	58	59
4	road_marking	29	34	42	38
5	street_light	30	43	45	45
6	wall	30	44	42	39
7	guard_rail	35	41	43	44
8	car	36	48	52	52
9	billboard	38	47	33	45
10	person	94	95	95	95
11	ego	84	90	89	89
12	road	84	88	88	88
13	terrain	83	85	85	85
14	building	73	80	81	80
15	truck	73	80	77	80
16	rider	59	74	76	74
17	bus	56	74	70	70
18	sky	52	56	57	57
19	caravan	45	65	58	60