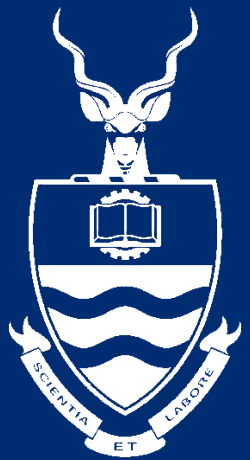


The Application of Attribution Methods to Explain an End-To-End Model For Self-Driving Cars

By: Jason Chalom (711985)

Supervisor: Dr Richard Klein

School of Computer Science and Applied Mathematics

Background

- End to End Learning for Self-Driving Cars by Bojarski et al.
- Implemented an unsupervised CNN model for controlling the steering angle of a vehicle
- "The CNN is able to learn meaningful road features from a very sparse training signal (steering alone)."

Background

- Safety of self-driving cars requires us to understand how the AI's learn and make decisions
- There are two taxonomies:
  - Hierarchical Self-driving
  - end-to-end systems

Models Evaluated

- 6 Models
- Bojorski et al. defined 3 architectures including the End to End model
- Microsoft's Autonomous Driving Cookbook
- Two simple architectures

Model Name	NetSVF	NetHVF		
Layer Type	Layer Output Size	Layer Output Size	Filter	Stride
convolution	32 x 123 x 638	32 x 123 x 349	3 x 3	1 x 1
convolution	32 x 61 x 318	32 x 61 x 173	3 x 3	2 x 2
convolution	48 x 59 x 316	48 x 59 x 171	3 x 3	1 x 1
convolution	48 x 29 x 157	48 x 29 x 85	3 x 3	2 x 2
convolution	64 x 27 x 155	64 x 27 x 83	3 x 3	1 x 1
convolution	64 x 13 x 77	64 x 13 x 41	3 x 3	2 x 2
convolution	96 x 11 x 75	96 x 11 x 39	3 x 3	1 x 1
convolution	96 x 5 x 37	96 x 5 x 19	3 x 3	2 x 2
convolution	128 x 3 x 35	128 x 3 x 17	3 x 3	1 x 1
convolution	128 x 1 x 17	128 x 1 x 8	3 x 3	2 x 2
fully-connected	1024	1024	-	-
fully-connected	512	512	-	-
fully-connected	1	1	-	-

	End to End Learning Model		
Layer Type	Layer Output Size	Filter	Stride
convolution	24 x 31 x 98	5 x 5	2 x 2
convolution	36 x 14 x 47	5 x 5	2 x 2
convolution	48 x 5 x 22	5 x 5	2 x 2
convolution	64 x 3 x 20	3 x 3	5 x 5
convolution	64 x 1 x 18	3 x 3	No Stride
fully-connected	1164	-	-
fully-connected	100	-	-
fully-connected	50	-	-
fully-connected	10	-	-
fully-connected	1	-	-
-	-	-	-
-	-	-	-
-	-	-	-

	Autonomous Driving Cookbook Model		
Layer Type	Layer Output Size	Filter	Stride
convolution	60 x 256 x 16	3 x 3	2 x 2
convolution	29 x 127 x 32	3 x 3	2 x 2
convolution	14 x 63 x 32	3 x 3	2 x 2
fully-connected	6948	-	-
fully-connected	64	-	-
-	10	-	-
-	1	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-

	Test Model 2		
Layer Type	Layer Output Size	Filter	Stride
convolution	60 x 256 x 32	3 x 3	1 x 1
convolution	60 x 256 x 32	3 x 3	1 x 1
fully-connected	122880	-	-
fully-connected	10	-	-
fully-connected	1	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-

	Test Model 1		
Layer Type	Layer Output Size	Filter	Stride
convolution	60 x 256 x 32	3 x 3	1 x 1
fully-connected	122880	-	-
fully-connected	10	-	-
fully-connected	1	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-

Table C.1: Comparison of the four different CNN models [Bojarski *et al.* 2016ab; Spryn and Sharma 2018]

* Includes ReLU and Drop-out Layers

Datasets Used

- Udacity Self-Driving Car Dataset (Training)
- Microsoft’s AirSim Tutorial Dataset (Test)
- The Cityscapes Dataset (Ground truth)

The Udacity Dataset

Data Group	Total Count	Straight Line Image Count	Swerve Image Count	Swerve Ratio
Dataset Before Drop	26720	19102	7618	0.3988
Dataset After Drop	15236	7618	7618	1.0
Train	10780	5390	5390	1.0
Validation	2938	1469	1469	1.0
Test	1518	759	759	1.0

Microsoft's AirSim Tutorial Dataset

Data Group	Total Count	Straight Line Image Count	Swerve Image Count	Swerve Ratio
Dataset Before Drop	46738	34813	11925	0.3425

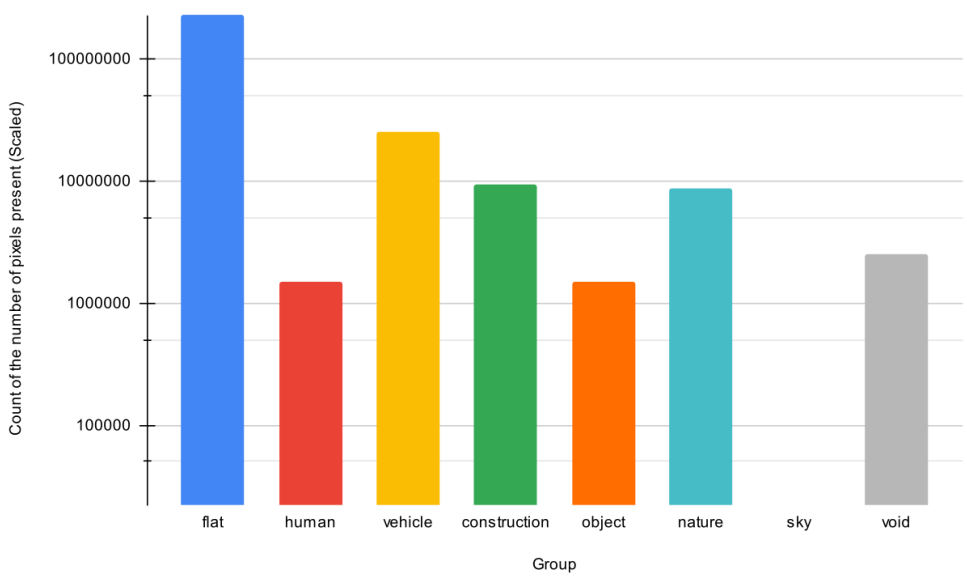
The Cityscapes Dataset

Data Group	Total Count	Straight Line Image Count	Swerve Image Count	Swerve Ratio
Data Set Before Drop	24997	20053	4944	0.2465
Train + Train Extra	22972	18308	4664	0.2547
Validation	500	408	92	0.2254
Test	1525	1337	188	0.1406

The Cityscapes Dataset

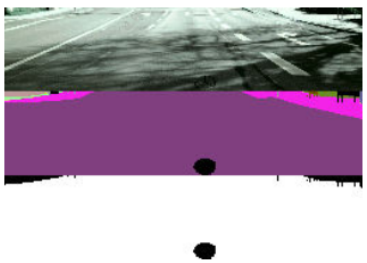
Group Name	Classes
flat	road, sidewalk, parking, rail track
human	person, rider
vehicle	car, truck, bus, on rails, motorcycle, bicycle, caravan, trailer
construction	building, wall, fence, guard rail, bridge, tunnel
object	pole, pole group, traffic sign, traffic light
nature	vegetation, terrain
sky	sky
void	ground, dynamic, static

Table 5.1: Table of Cityscapes labels as divided per group [Cordts *et al.* 2016]



The Cityscapes Dataset

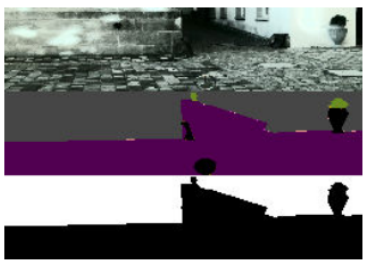
Examples of Each Cityscapes Group



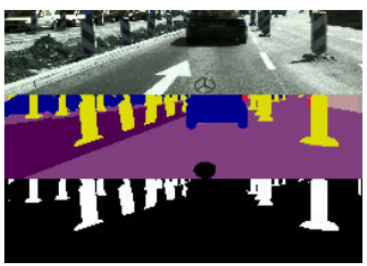
(a) flat

(b) human

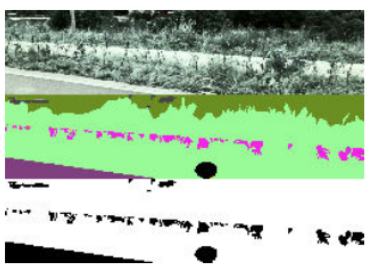
(c) vehicle



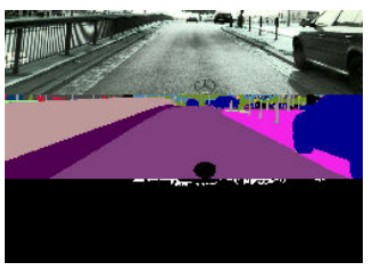
(d) construction



(e) object



(f) nature



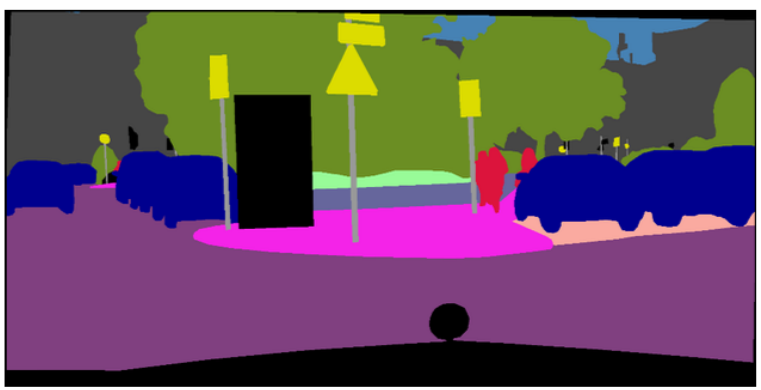
(g) sky

(h) void

Dataset Cropping ROI

(a) Uncropped example frame

(b) Cropped frame



(a) Original Segmentation

(b) Segmentation After Processing

Some Types of Attribution

- Data Analysis
- Generalised model behaviour
- Relationship between inputs and outputs
  - Pertinent parts of the dataset
  - Dataset relationship to the output
  - **Relationship between model layers and the output**
- Feature Analysis

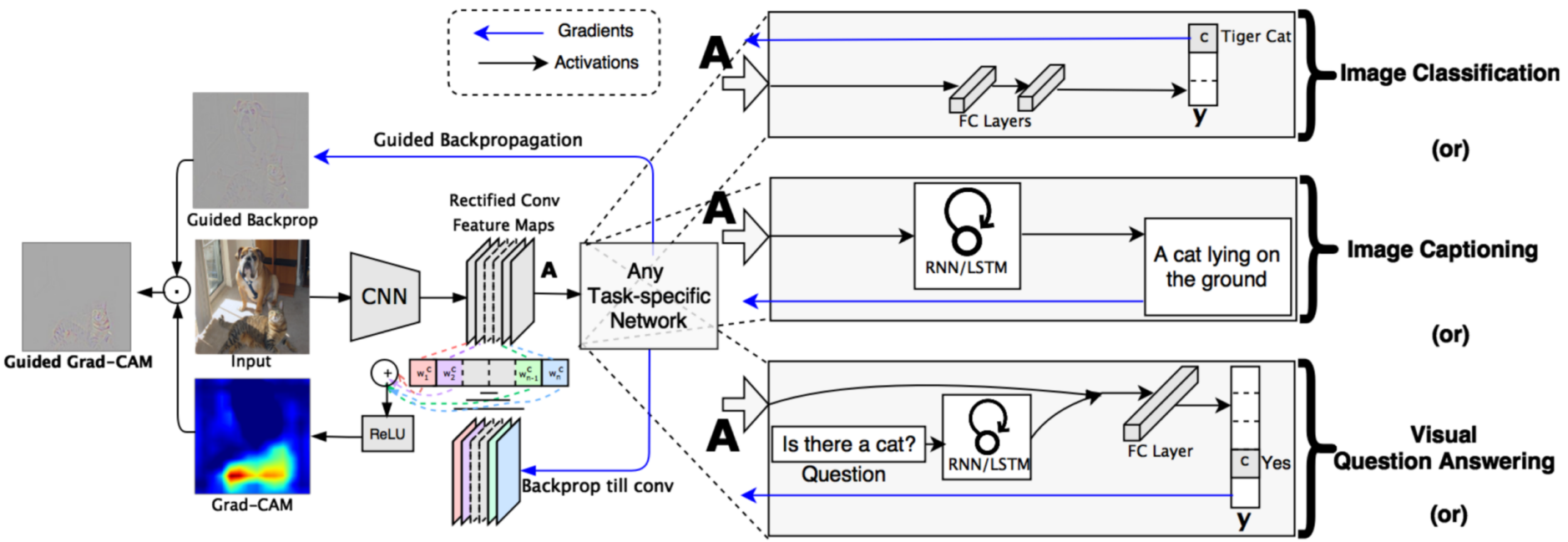
Guidotti et al. (2018)

Saliency Maps

$$\frac{\partial \text{output}}{\partial input}$$

Ancona et al. 2017; Simonyan et al. 2013

Gradient-weight Class Activation Mapping (Grad-CAM)



Selvaraju et al. (2016)

Issues with some attribution methods

Independent of
- Input Data
- Model Parameters

Sanity Checks for Saliency Maps, Adebayo et al. (2018)

Issues with attribution

- Brittle to noise
- Manual visualisation analysis should not be used alone

Adebayo et al. (2018)

Metric of Autonomy (%)

$$autonomy = (1 - \frac{(I*6[seconds])}{E}) \cdot 100$$

where *I* is the number of interventions, and *E* is the elapsed time in seconds.

Bojarski et al. (2016b)

Metric of Autonomy (%)

(Updated)

$$autonomy = (1 - \frac{(I * I_c)}{E}) \cdot 100$$

where I is the number of interventions, Ic is assumed 1, and *E* is the elapsed time in seconds.

Metric of Autonomy (%)

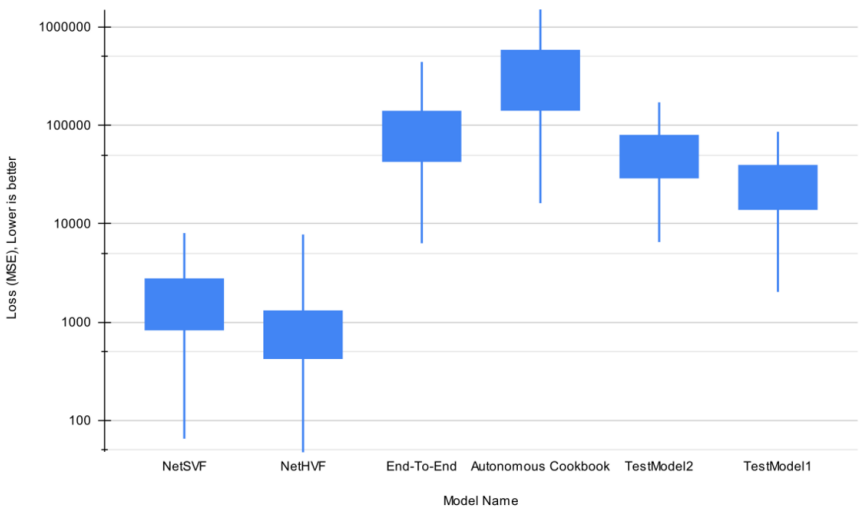
(Number of Interventions)

$$\mathbb{1}(x, \varepsilon) = \begin{cases} 0 & x \leq \varepsilon \\ 1 & otherwise \end{cases}$$

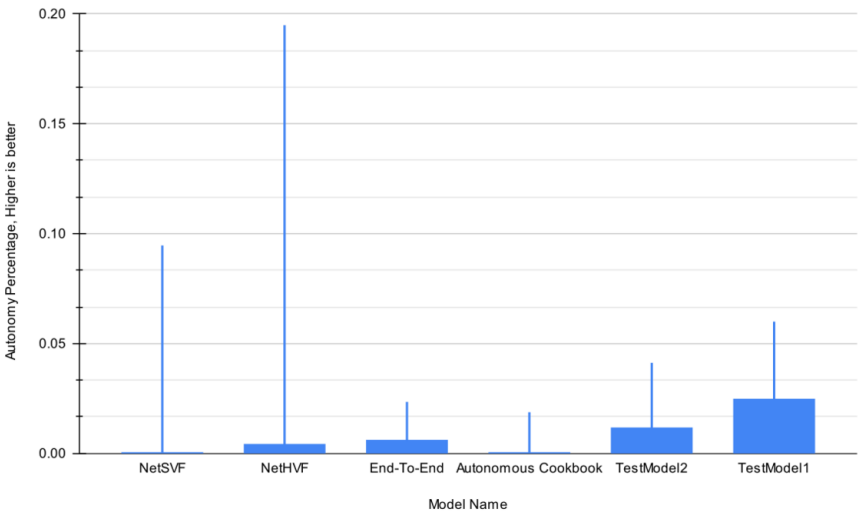
$$\text{number of interventions} = \sum_{n=1}^{N} \mathbb{1}(|\theta_n - \phi_n|, \varepsilon)$$

where ε is the error margin, θ is the predicted angle, φ is the expected angle

Trained Model Random Test Results

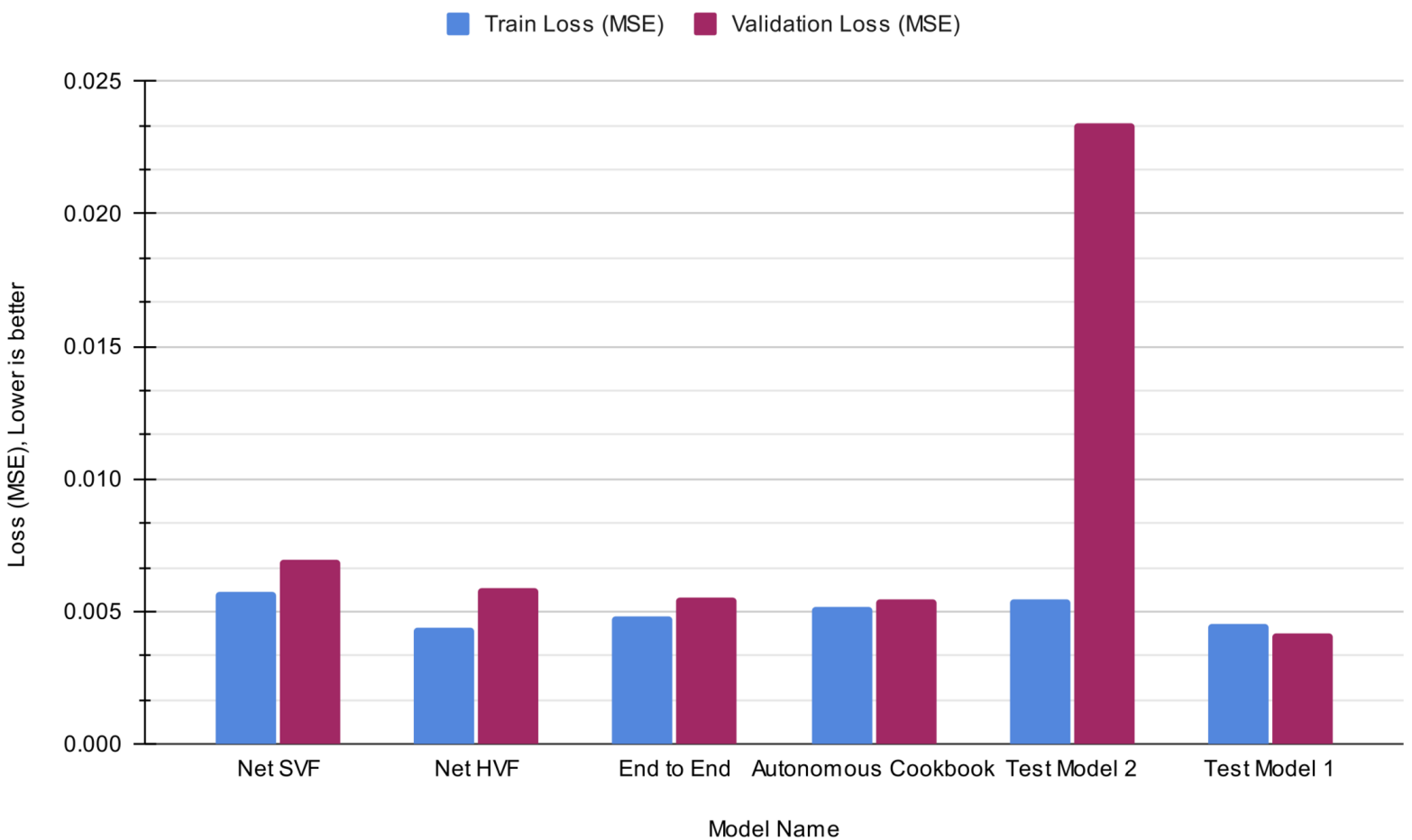


(a) Test Loss Results

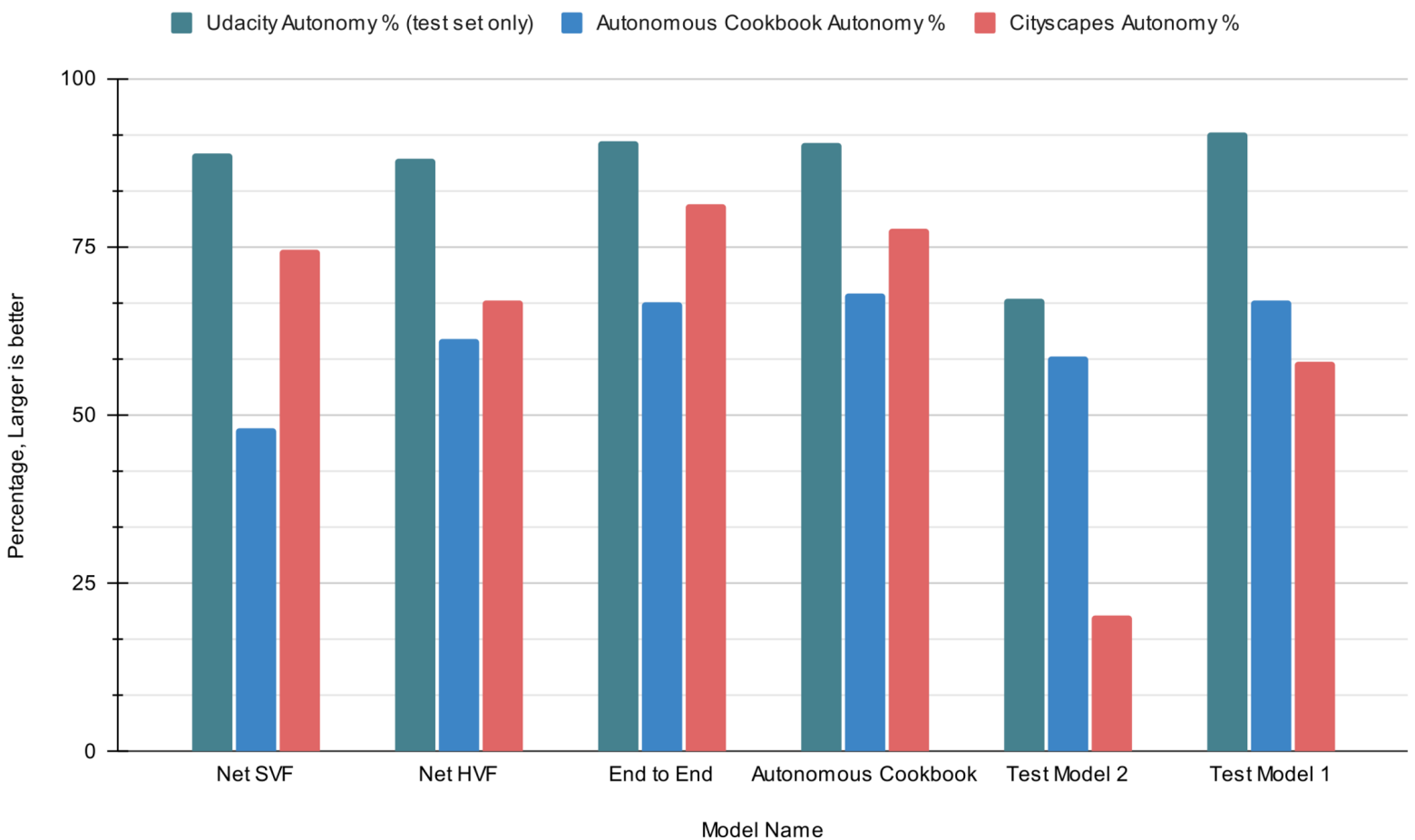


(b) Autonomy Results

Trained Model Results



Trained Model Results



GradSUM

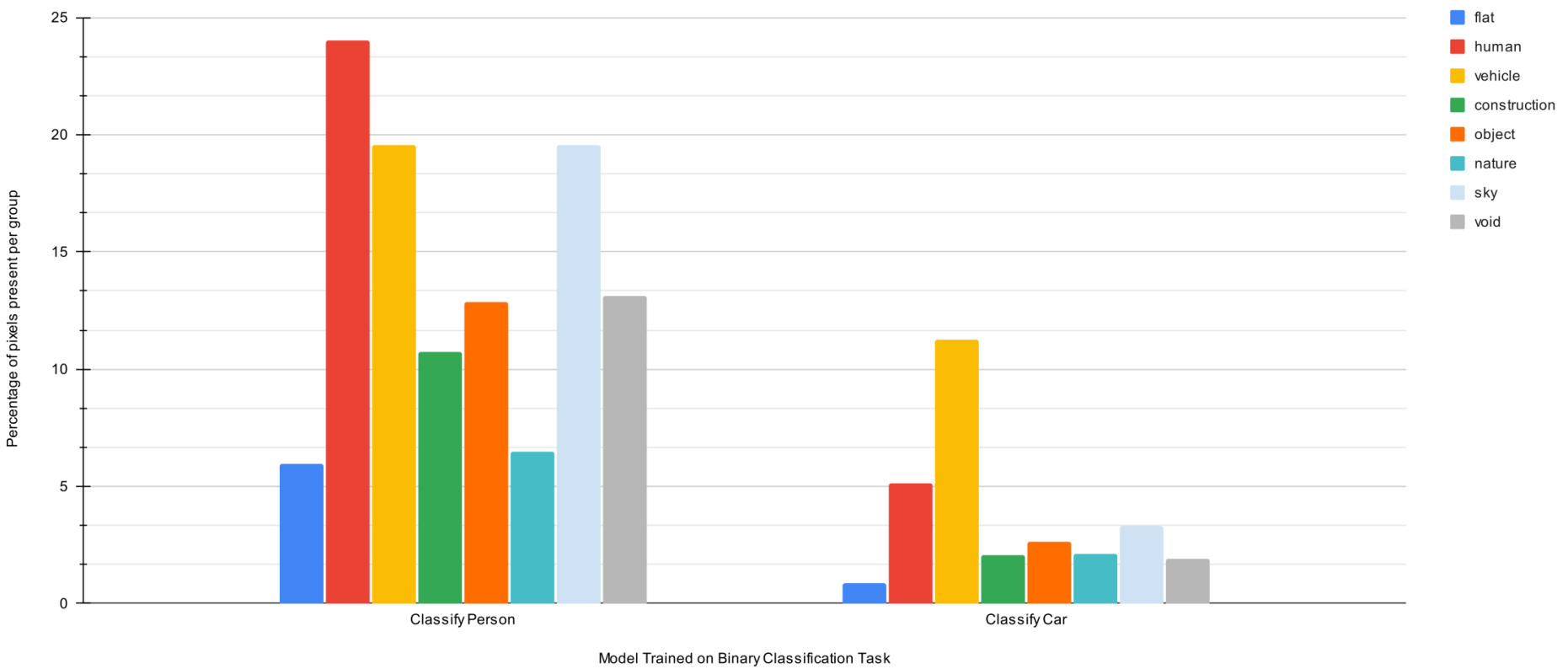
Using a ground truth dataset (Cityscapes), generate a sum (per segmentation class), of each pixel of that class weighted by the activation map and divided by the total number of pixels of that class

GradSUM

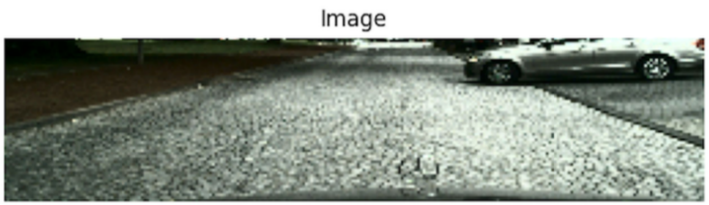
$$G_k = \frac{\sum_{i=0}^{I} q_{ki}}{\sum_{i=0}^{I} P_{ki}} \times 100$$

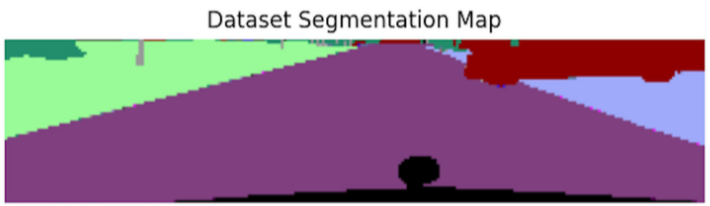
Where **k** is as specific class, *Pk* is the map of pixels for a class category, ***Gk*** is the activation percentage for that class, ***i*** is the index in the segmentation and activation maps

GradSUM Classification Example

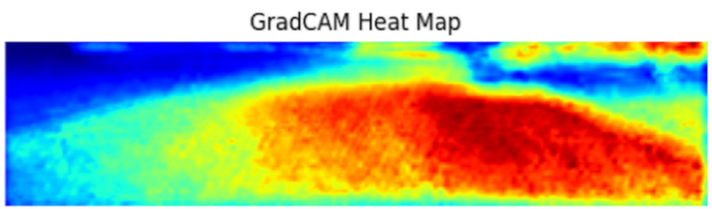


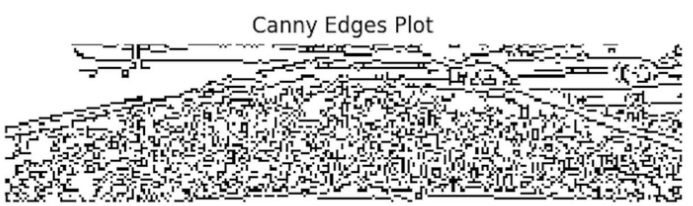
Canny Edges of Cityscapes





(a) Original RGB image and corresponding *fine* segmentation map





Canny Edges of Cityscapes

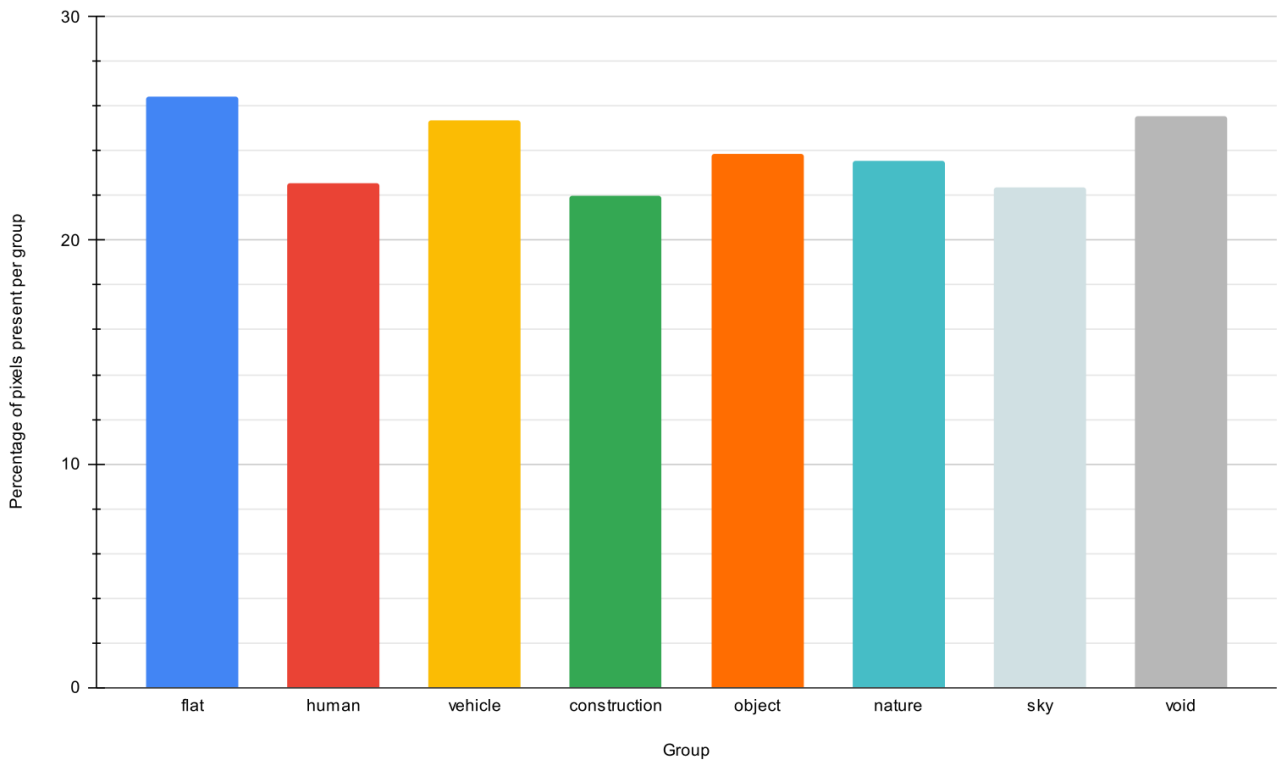
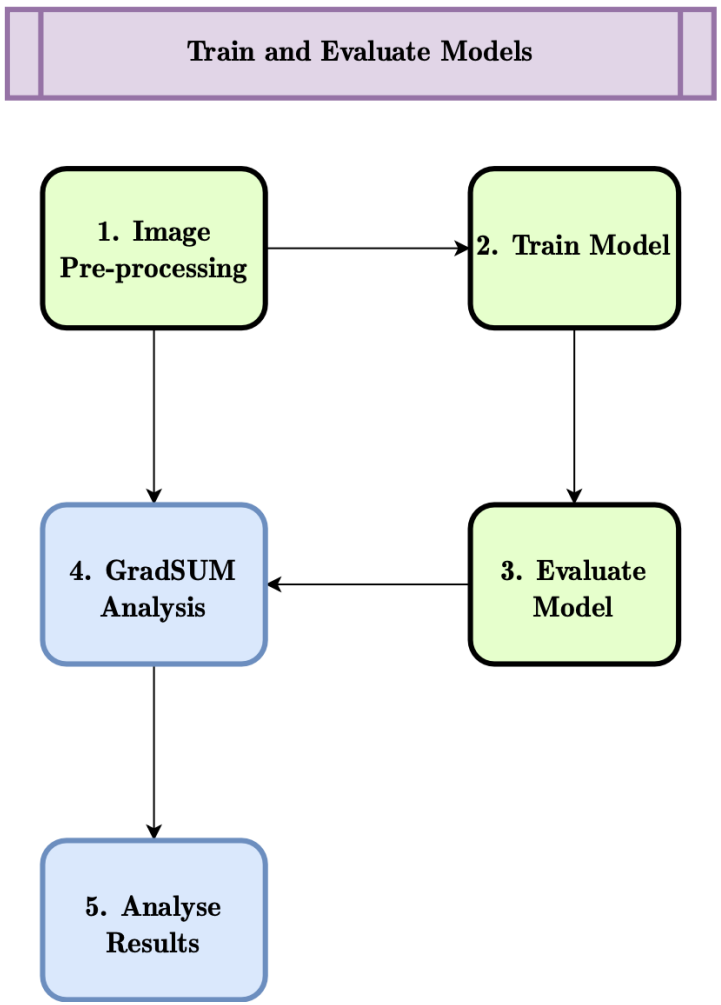


Figure 5.15: Resultant canny edge group profile

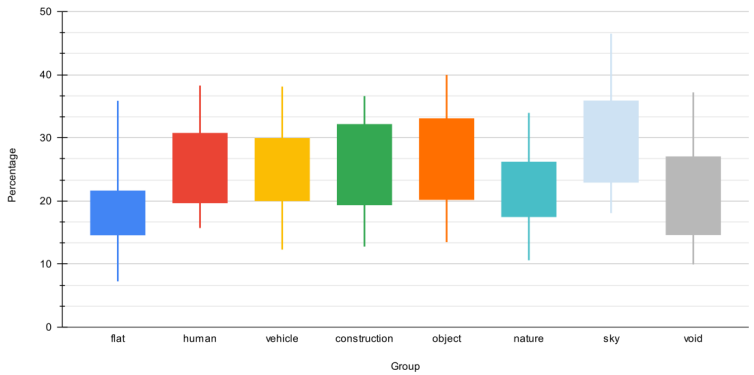
Experimental Setup



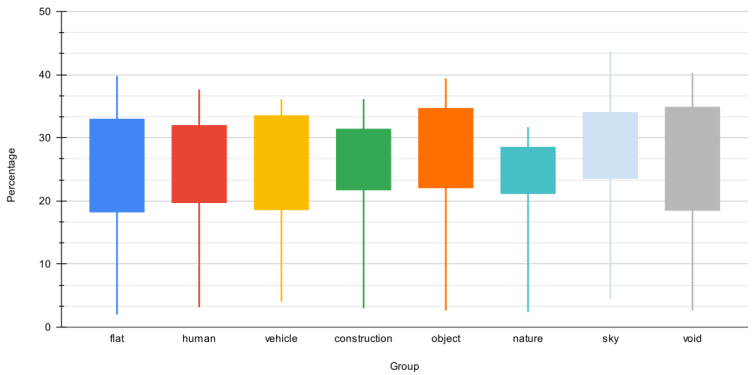
Experimental Setup

- Training included generating **10** randomly initialised models (small weights, **Xavier Uniform**) and training for **10** epochs
- The **best** model (by validation loss) was then fully trained
- Pre-processing of images (including cropping and scaling was also done)

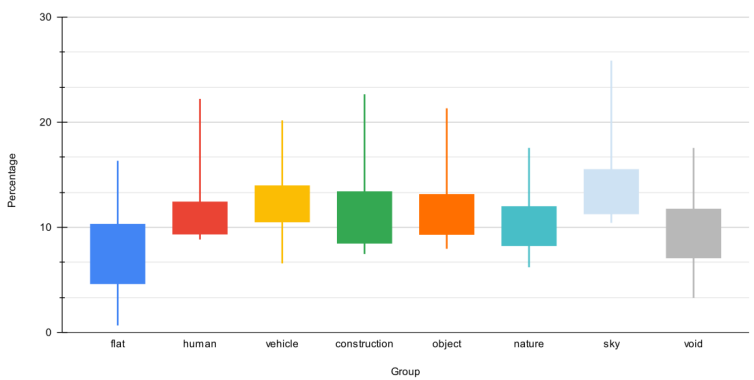
Random GradSum Results



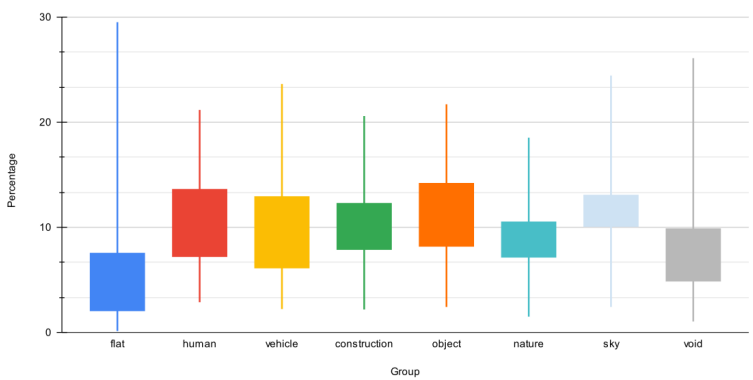
(a) NetSVF (x100)



(b) NetHVF (x100)

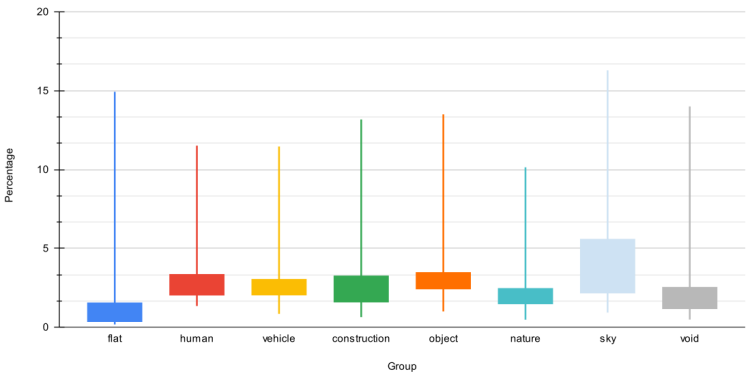


(a) End to End (x100)

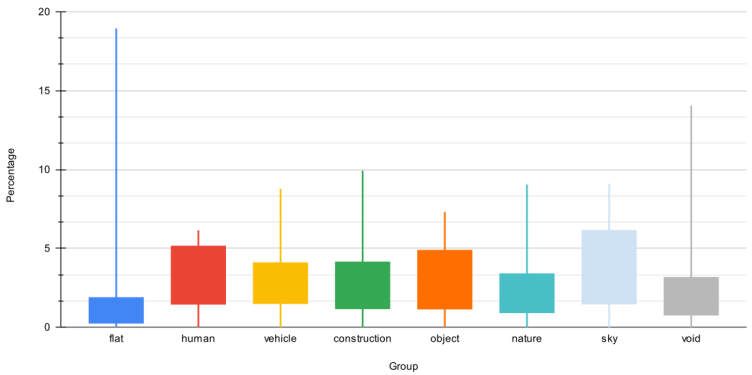


(b) Autonomous Cookbook (x100)

Random GradSum Results

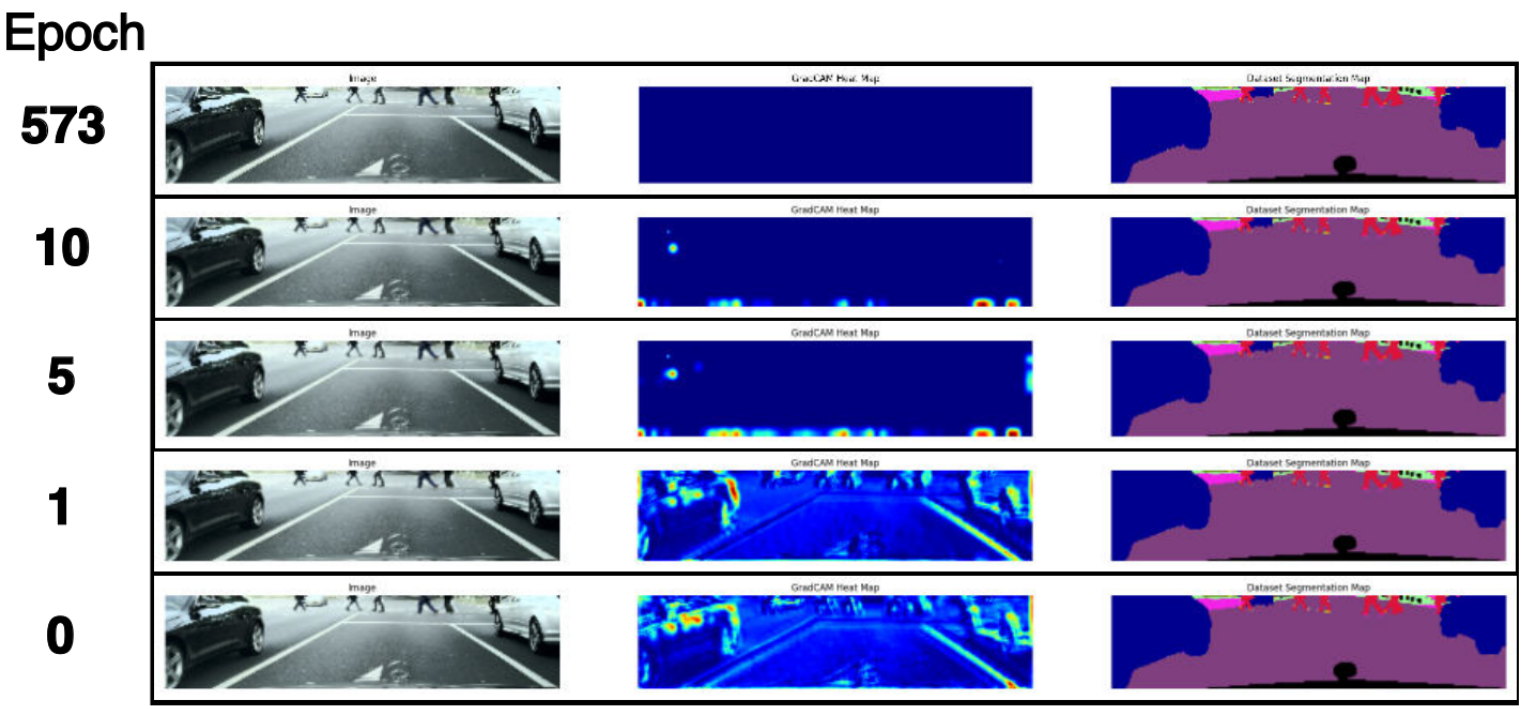


(a) TestModel2 (x100)



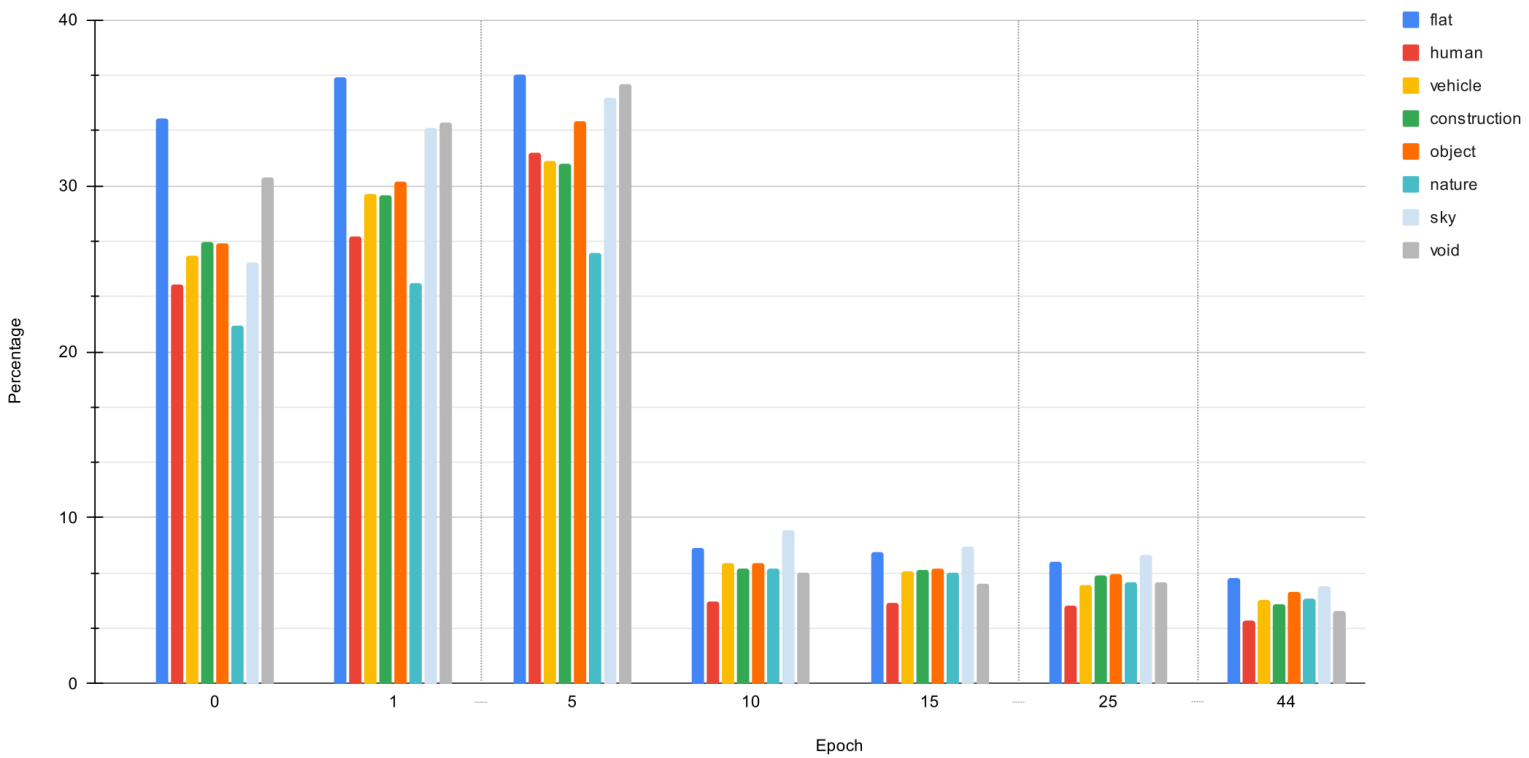
(b) TestModel1 (x100)

Sample GradSum Result



(a) End to End

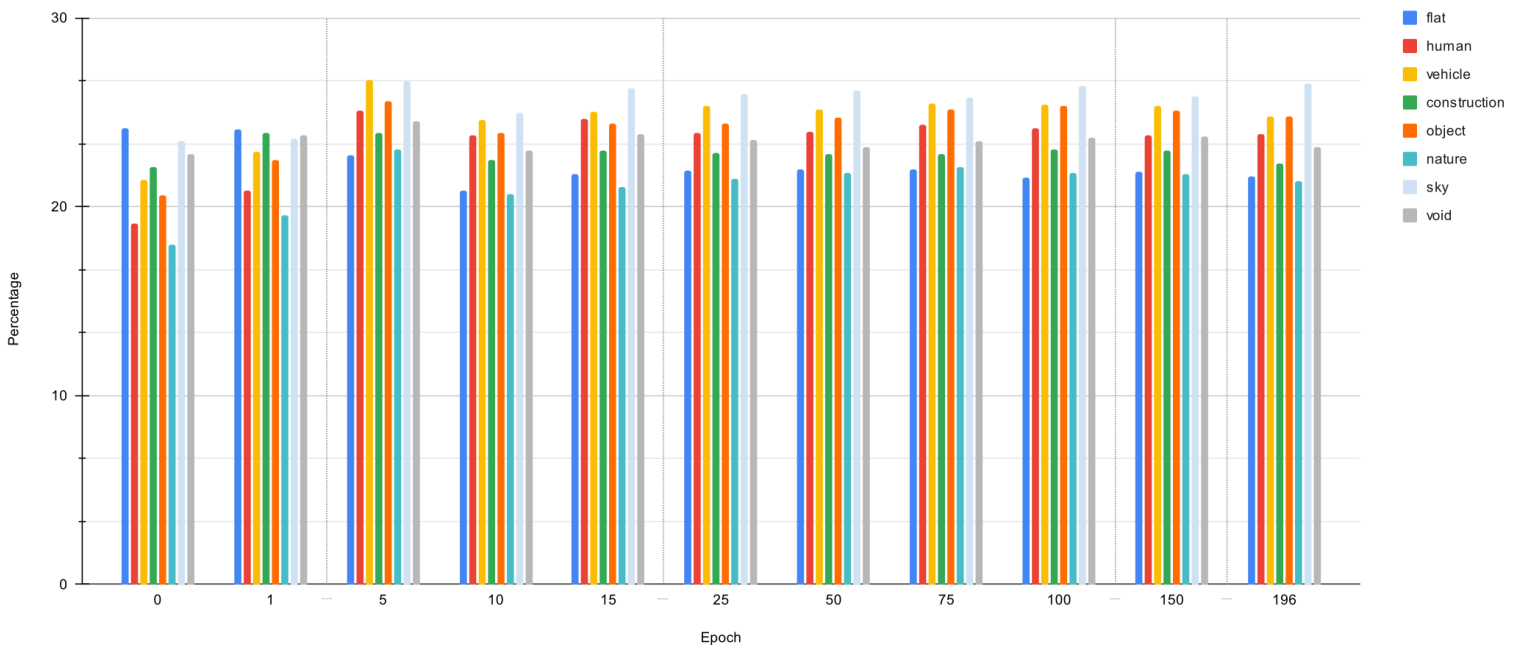
GradSUM Results



(a) NetSVF Profile

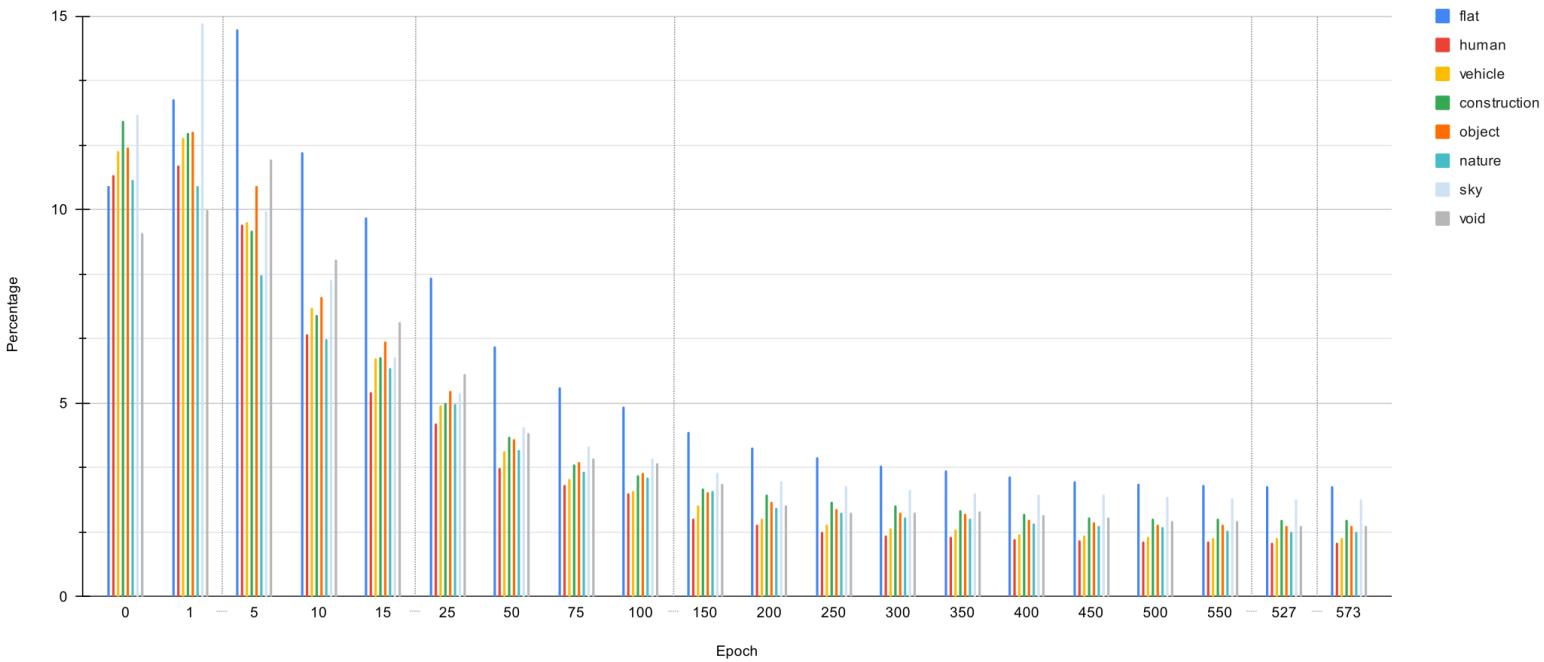
* model with the most parameters

GradSUM Results



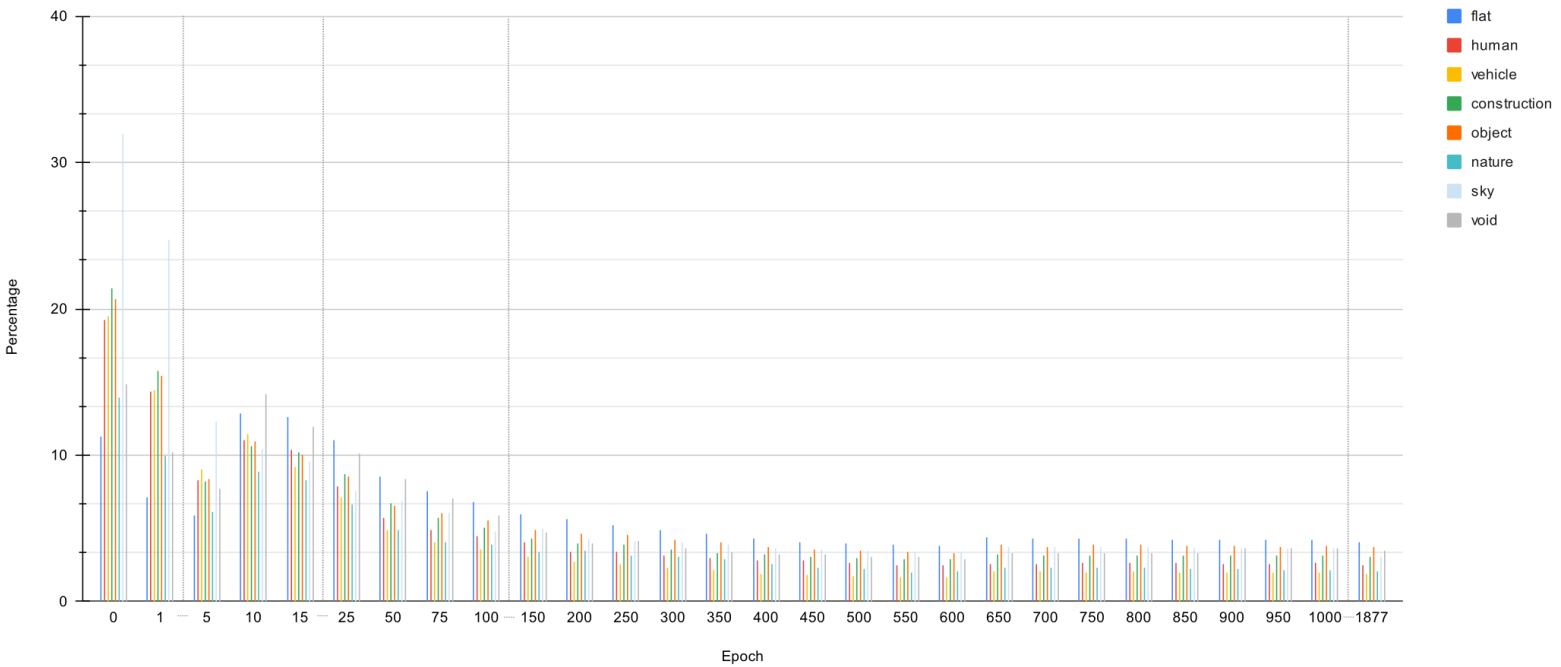
(b) NetHVF Profile

GradSUM Results



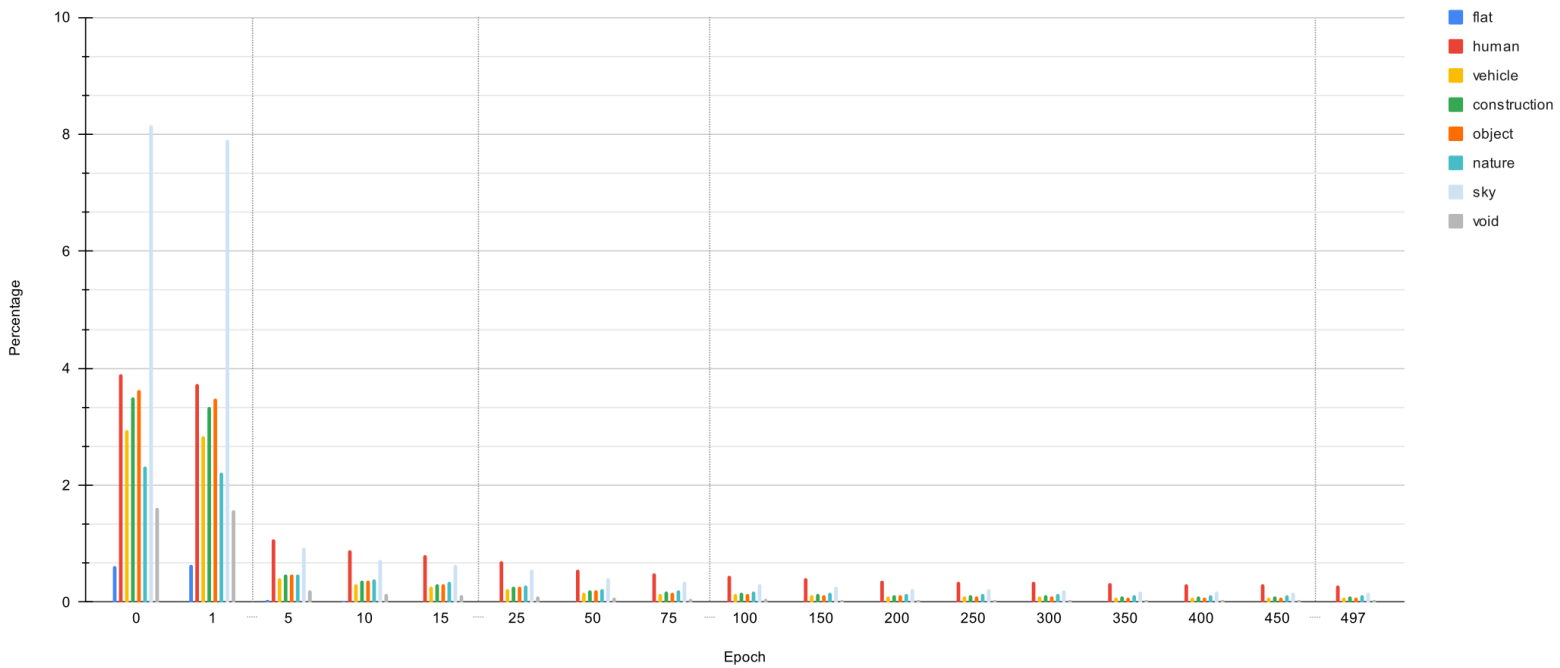
(a) End to End Profile

GradSUM Results



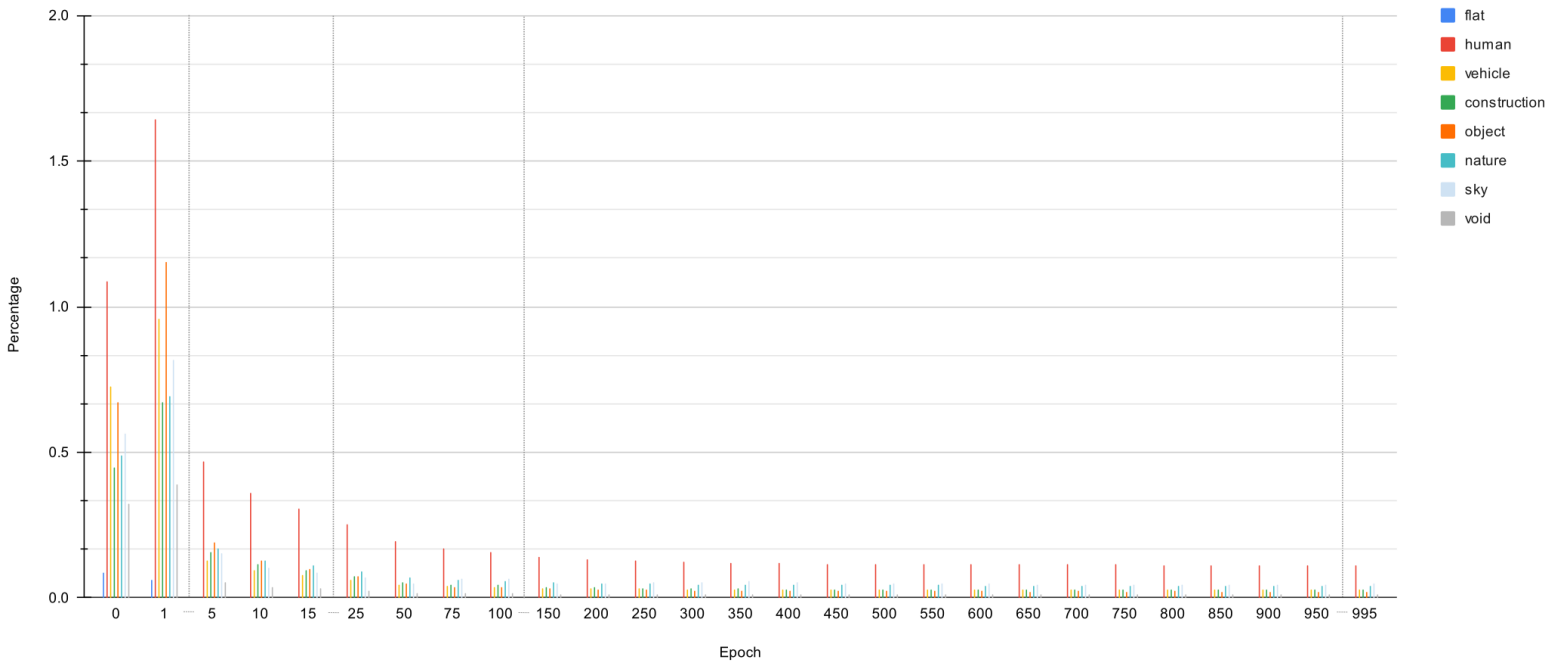
(b) Autonomous Cookbook Profile

GradSUM Results



(a) TestModel2 Profile

GradSUM Results



(b) TestModel1 Profile

Outcome of the Study

- **3** Models **Learned** Meaningful Features
- **3** Did **not** learn meaningful features
- The **End to End** Model **did** learn meaningful features

- Model **architectures** plays an important role

Possible Limitations

- **Performance** of the analysis scheme
- Needing high **quality** semantic data
- Accuracy and granularity of the semantic data (Only **fine** segmentations was utilised)

Future Work

- Use GradSUM with **other** attribution methods
- **Improve** ground truth datasets
- Improve model **architecture** and selection
- Evaluate **different** regression tasks