GradSUM a Method to Quantitatively Characterise and Explain Deep Learning Model Behaviour in Several Domains (Unpublished)

By: Jason Chalom

Supervisor: Professor Richard Klein

RAIL Lab

PRIME LAB

School of Computer Science and Applied Mathematics

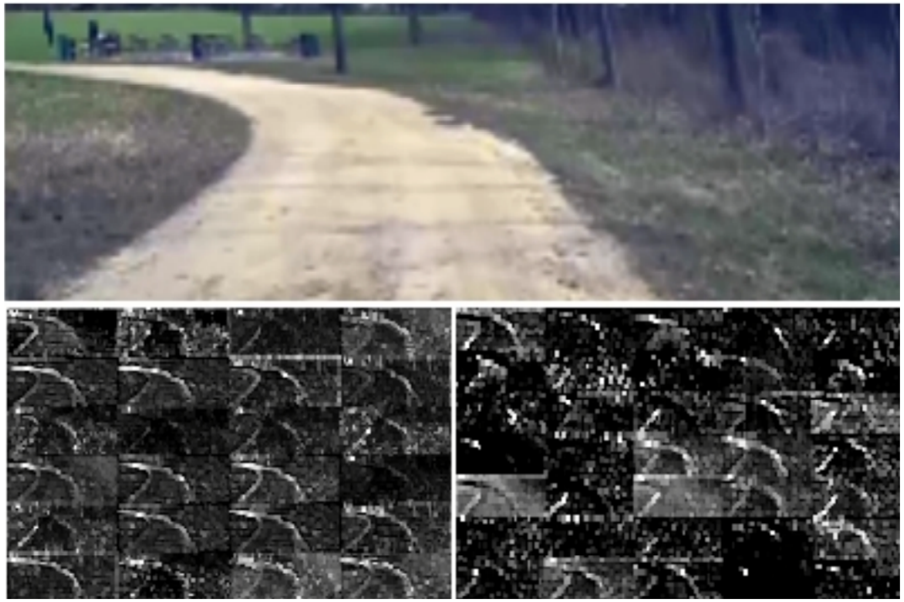
Background

- 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)."

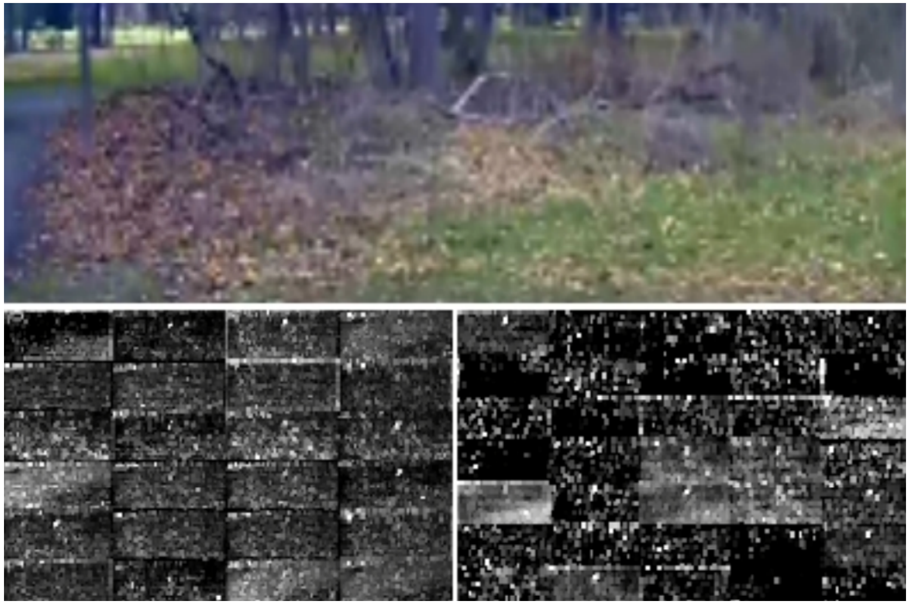
https://github.com/Microsoft/AutonomousDrivingCookbook

Safety Concerns

- What are **meaningful road features**?
- Which aspects **influence** steering angle?
- Understanding the **relationship** between the given data and predicted output



Unpaved Road



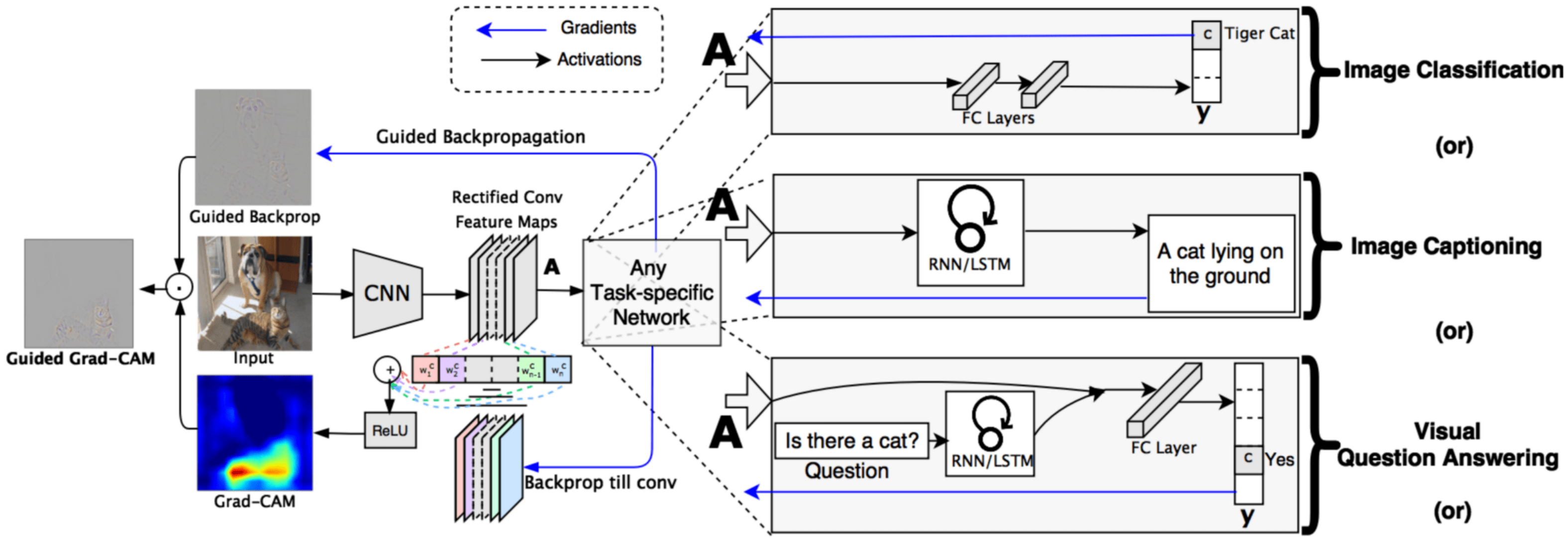
Forrest Scene

Explainable AI

GradCAM and related gradient methods

https://github.com/jacobgil/pytorch-grad-cam

GradCAM



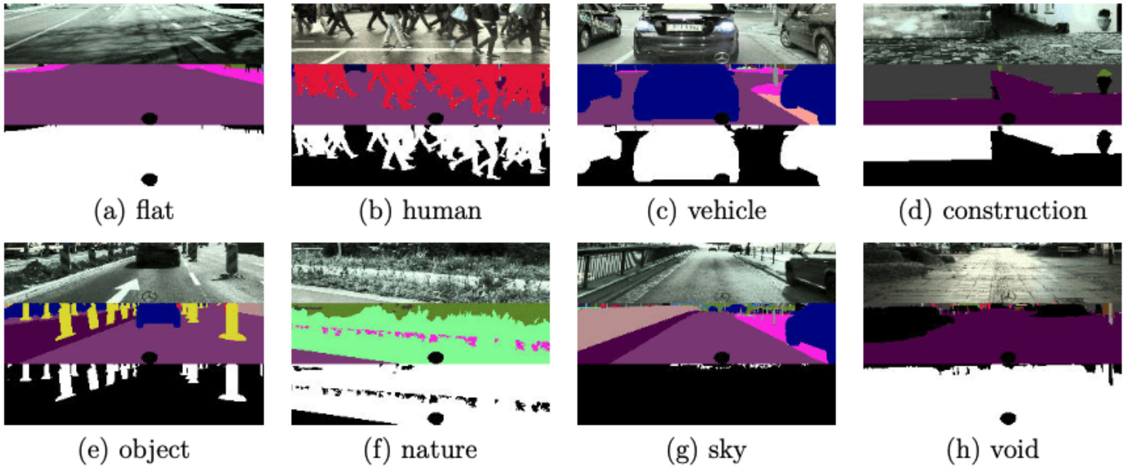
Grad-cam: Visual explanations from deep networks via gradient-based localization, Selvaraju et al.

Sanity Checks for Saliency Maps (Adebayo et al.)

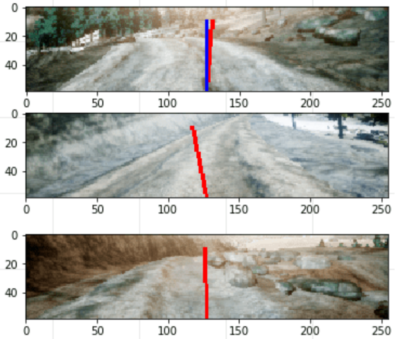
- Many methods partially reconstruct the input data
- Brittle to noise and interference (misleading results)
- Many of the advanced guided methods dont have an adequate relationship between the input data and output nodes of a network
- Some methods (like some saliency maps) may not work with features that have a negative effect on the output

Available Datasets

Cityscapes

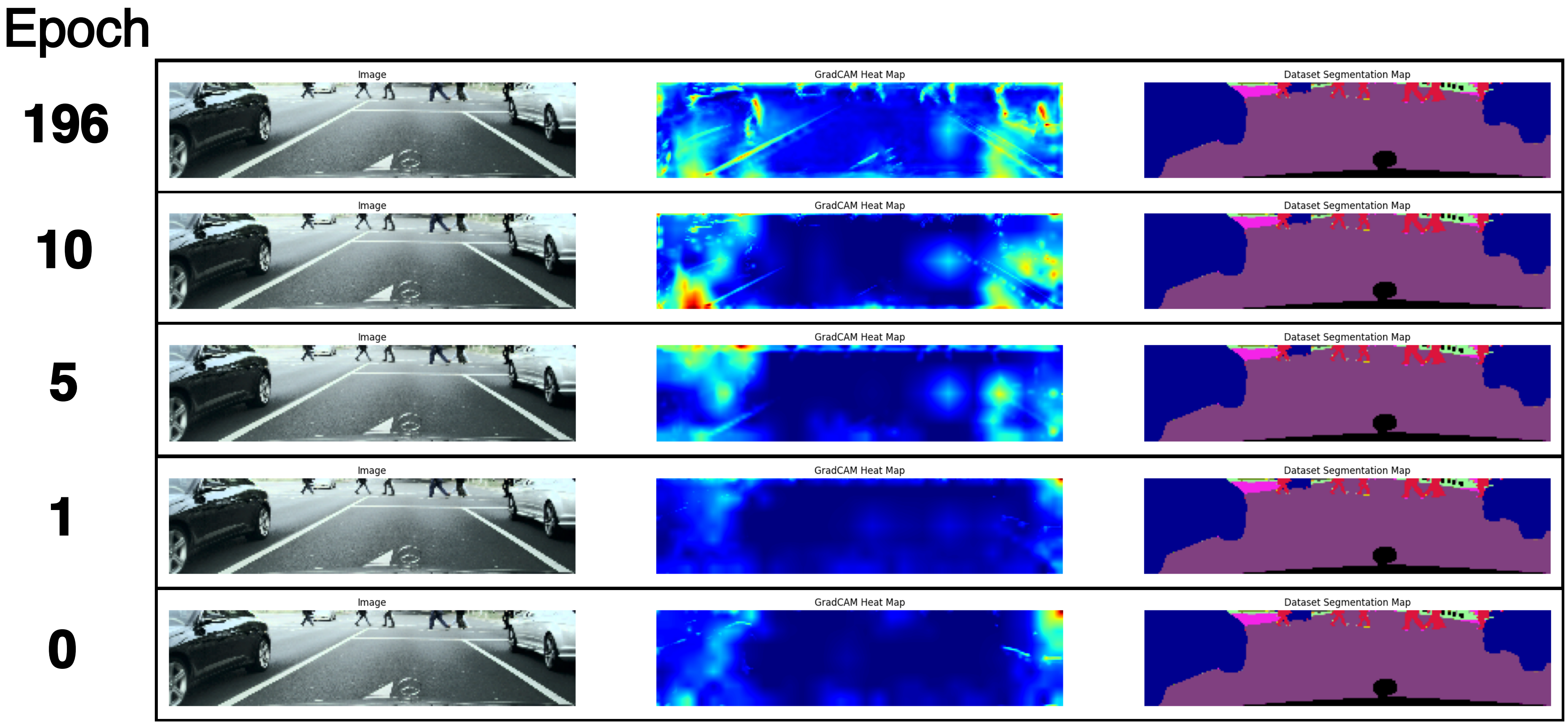


Microsoft Self-Driving Cookbook



Udacity

and many others ...



How the model's heatmaps change per training epoch

(Cityscapes Dataset Sample, NetHVF Model)

The parts

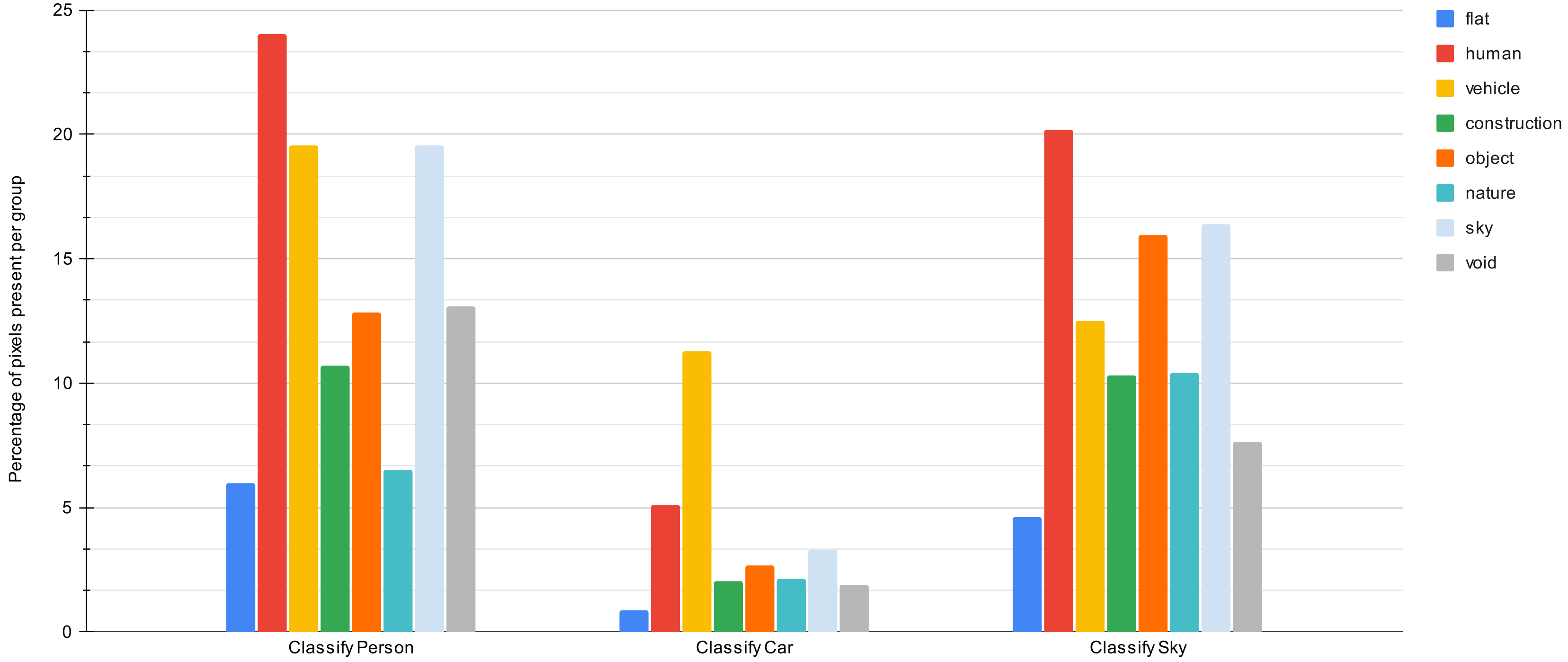
- Can **train** CNN models to predict steering angle
- Have many datasets that include **semantic** information
- Have algorithms to highlight **regions** of **importance** from the input of CNN models
- Manual **analysis** is subjective

GradSUM

GradCAM (or any heat-map)

+

Segmentation Maps



GradSUM

1. Split out each **segmentation category** into segmentation maps and corresponding input images.
2. Generate **Grad-CAM** for each **input** for each **category**
3. Compute the **element-wise product** of the Grad-CAM map and input image
4. Then compute the **pixel percentage** activated for that segmentation category

GradSUM

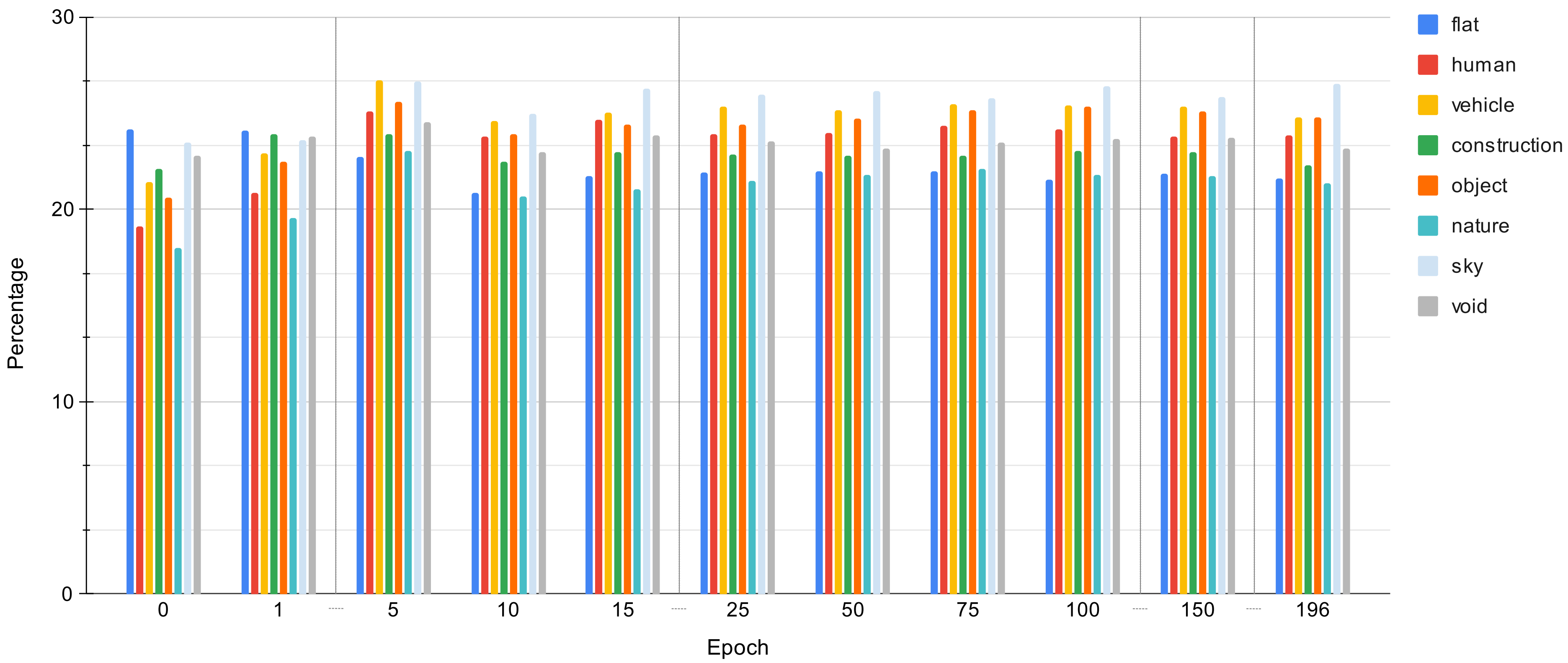
Pseudo-code for the GradSUM analysis scheme

Algorithm 1 An algorithm for the GradSUM scheme

```
K ← {K^flat, K^human, ...}  ▷ The available segmentation groups in ground truth dataset
for k in K do
    for w, h in InputImage do
        if InputImage[w,h] is in group  k then
            P[k][w][h] ← 1
        else
            P[k][w][h] ← 0
        end if
    end for
    N[k] ← Sum(P[k])        ▷ This is the sum of all pixels present for the given group k
    M ← GradCam(InputImage, model)
    G[k] ← (P[k] ⊙ M) / N[k] · 100  ▷ G is the GradSUM result, and ⊙ is the element-wise product
end for
```

$$K \leftarrow \{K^{\text{flat}}, K^{\text{human}}, \ldots\}, \quad N[k] \leftarrow \mathrm{Sum}(P[k]), \quad G[k] \leftarrow \frac{P[k] \odot M}{N[k]} \cdot 100$$

Percentage of activation of pixels per category averaged for each sample (cityscapes)



NetHVF Model (Trained on Udacity Dataset)

What we did

- 6 Models
- 2 Datasets (Udacity, Cityscapes)
- 3 Sanity check experiments
- Other model performance metrics also compared (autonomy of driving, MSE)
- Model comparison

Possible Issues

- Performance cost
- Needing semantic data
- Accuracy and granularity of the semantic data

Next Steps

- CNNs -> Vision Transformers
  - Can **Attention** maps be used?
  - **GradCAM** comparison
- Other domains
  - **NLP**, textual input instead of images
  - Generate similar heat-maps against **textual** input to produce similar **profile** graphs
  - Would need to have a **ground truth** dataset of categorised textual data

https://github.com/TRex22

https://www.linkedin.com/in/jasonchalom/

https://twitter.com/trex2218