Final Competition of Deep Learning (Traffic environment Contest, spring 2020)

- Sree Gowri Addepalli (sga297)
- Amartya Prasad(ap5891)
- Sree Lakshmi Addepalli(sla410)

Baseline with Supervised techniques

1. **Road Map prediction** - Binary Classification with **Resnet18 + FC Layer** (Sent **6 images** as **18 channels**). The accuracy on validation set was **0.74** which was our baseline.
2. **Object Detection and Classification** - **Faster RCNN** with **Image stitching**. We achieved an IoU of **0.014** through this technique.

One of the reasons for such performance is **loss of depth info** in **monocular** images.

We also expect performance on the road tasks to have been hampered due to a relatively **small annotated dataset.**

SimCLR (Simple framework for contrastive learning of visual representations)

- SimCLR is a framework with SOTA in Self-Supervised and Semi-Supervised Image Training.
- It provides a model that learns representations by **maximizing agreement** between differently augmented views of the same data example via a contrastive loss in the latent space.

It consists of the following modules:

1. **Data Augmentation:** Generates two correlated views of same image using random cropping, random color distortions and random gaussian blur.
2. **Base Encoder:** Output of last average pooling layer used for extracting representations using resnet18.
3. **Projection Head:** For learned embeddings of sizes 512 and 64.
4. **Contrastive Loss Function:** Takes as input, set of examples (xi and xj) , aims to identify xj in set for given xi. (called **NT-Xent - the normalized temperature-scaled cross entropy loss**).
5. **LARS optimizer** - We tested with Layer wise Adaptive Rate scaling and Adam for optimization.

SSL Contrastive Learning and Depth Estimation

We worked on unlabelled dataset through self supervision techniques.

1. **Monodepth2** - Self supervised Monocular Depth estimation (Learns through **temporal dynamics**)- **6 models (6 cameras)**. It helps us calculate the **depth** of each pixel in the image.
2. **SIMCLR** - **24 channels (6*(3+1))** - feature embeddings of dimensions 512 with depth images being passed as the fourth channel generated through **Monodepth2**.

Results: Heavy model - Could train only for 5 epochs. Achieved a low average training loss of **0.02** on **pretext** task.

Downstream for Object detection

1. **Transfer learned** embeddings from SimCLR with MonoDepth2 to the Regional Proposal Network (RPN) head of the Faster RCNN , a 2D object Detection model for bounding box prediction.

2. **Custom loss :** Combined with IoU based loss (-log(IoU) or Generalized IoU).

$$IoU = \frac{\mathcal{I}}{\mathcal{U}}, \quad \text{where} \quad \mathcal{U} = A^p + A^g - \mathcal{I}$$

$$GIoU = IoU - \frac{A^c - \mathcal{U}}{A^c}$$

3. **RetinaNet-** Also, Implemented transfer learning with RetinaNet, attaching the upstream prediction heads to this model.

Results - Implementation issues on downstream - issues merging the SimCLR and monodepth model with Faster RCNN/RetinaNet due to time constraints.

Downstream for Road Map prediction

Transfer learned embeddings of SimCLR with Monodepth models for RoadMap with semantic segmentation task.

1. Conjoined learned embeddings with **ASPP**(**Atrous Spatial Pyramid Pooling**) Head of DeepLabv3 - (**SOTA semantic segmentation model for road prediction**). ASPP probes an incoming convolutional feature layer with filters at **multiple sampling rates** and effective fields-of-views, thus capturing objects as well as **image context** at **multiple scales**.

Results: 0.66 (vs 0.74 supervised validation accuracy). Possible overfitting on unlabelled data.

2. Also, conjoined learned embeddings with **Linear classifier**.

Results: 0.64 (vs 0.74 supervised validation accuracy).

Miscellaneous - Experiments and ideas.

We experimented with the following architectures but hit roadblocks.

1. **Mono Layout** (Top Down View prediction using DL) - Bad Quality training labels.
2. **Pseudo Lidar** (3D object detection - Point Clouds from depth images) - No access to velodyne calibration data.
3. **RetinaNet** - With only depth images as input (Supervised) - computing resource issues with GPU availability.

Other ideas:

1. **LSTM Layers** - could help learn, temporal dynamics better - possibly better performance.
2. **Video based Contrastive Learning** (eg: VINCE) for better feature learning.
3. **Ensembling** techniques.