Contrastive Point Cloud-Image Pre-Training

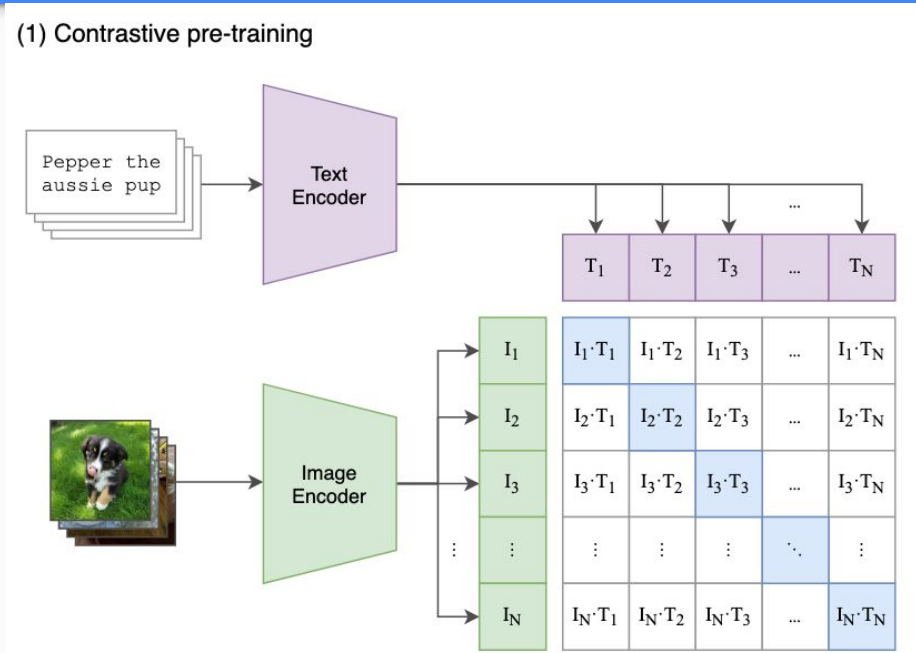
Parker Erickson
CSCI 8980

Introduction

- Sensor Validation is Very Important
  - Can we use one sensor to validate another?
- Validate point-cloud data using an image and vice-versa
- Similar to “Arguing Machines” idea introduced in [1]
  - Two neural networks
  - When they disagree, put humans in the loop

OpenAI’s CLIP

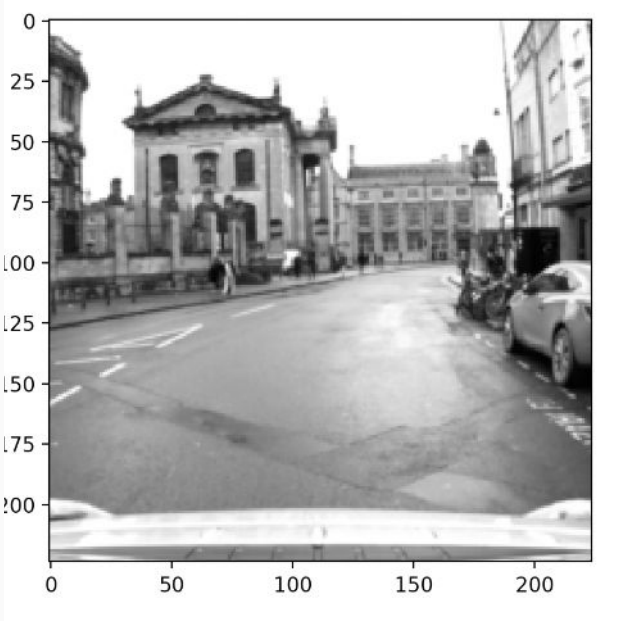
- Contrastive Language-Image Pre-Training [2]
- Encode both image and caption to an embedding
  - Similarity between correct image-caption pairs should be high



Source: [2]

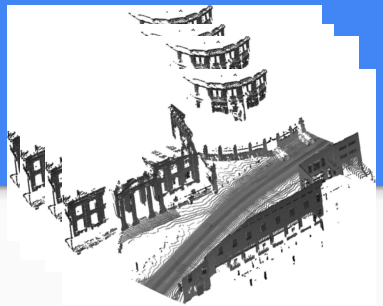
Data

- Uses the Oxford Robot Car Dataset [3]
  - Training Set (80%)
    - 18,580 Image/Point-Cloud Pairs
  - Validation Set (20%)
    - 4,645 Image/Point-Cloud Pairs
- Many LIDAR readings between frames of video
  - Combined readings into large point cloud for each frame/image
- All data is in good weather/daytime



Approach

- Perform same contrastive loss learning process as CLIP model
- Instead of a text encoder, use a point cloud encoder
- A sensor validation task is created to determine if an image/point cloud pair is the correct matching or not
  - Cosine similarity between the image and point cloud embeddings



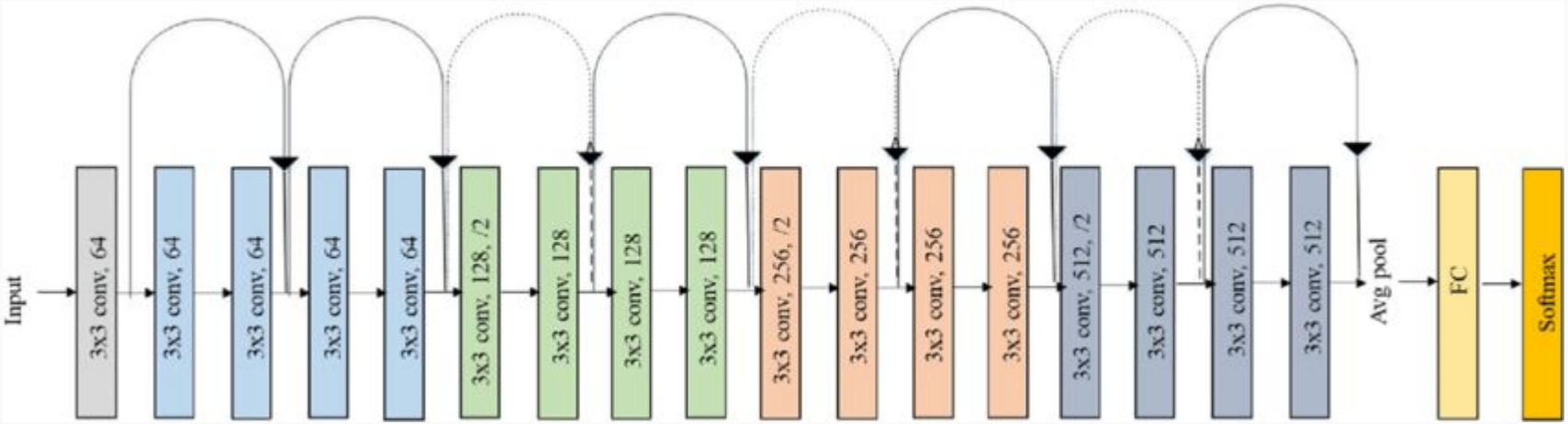
Point Cloud Encoder

Image Encoder

I_1P_1	I_1P_2	...	I_1P_n
I_2P_1	I_2P_2	...	I_2P_n
...	...	...	...
I_nP_1	I_nP_3	...	I_nP_n

Vision Model

- Utilize ResNet18 architecture
  - Evaluate both pretrained and non-pretrained versions



Source: https://www.researchgate.net/figure/Original-ResNet-18-Architecture_fig1_336642248

Point Cloud Models

- PointNet [4]
- GAT [5]
  - 4 Attention Heads
  - Most Memory Hungry
- GCN [6]
- Both the GCN and GAT architectures use a graph constructed of k = 6 Nearest Neighbors
- All algorithms implemented in PyTorch Geometric
- Mean and Max Pooling are Evaluated

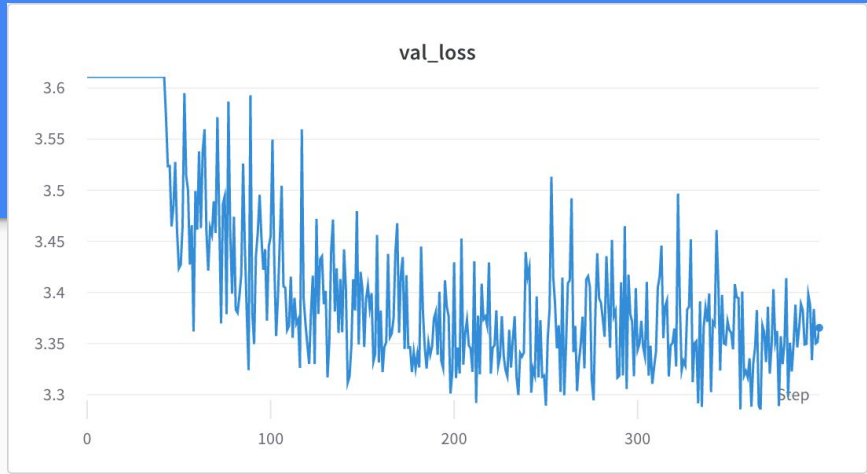
Putting it Together

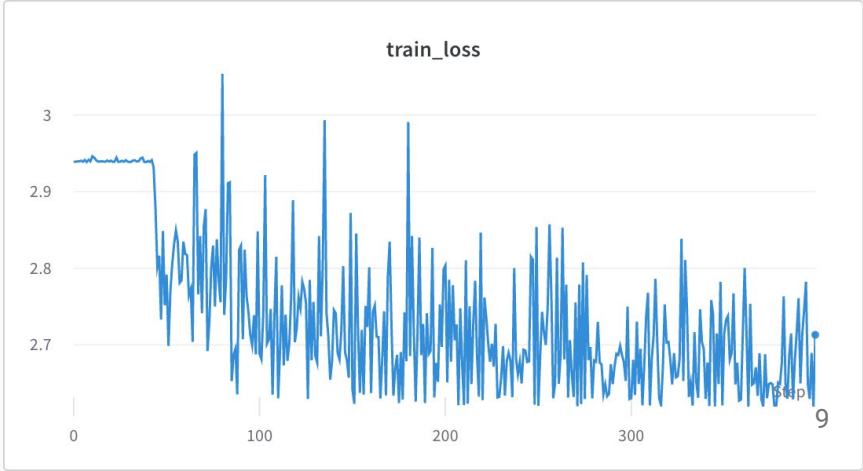
- Linear Projection Layer is added to both models' output to map into correct joint-embedding dimension
- Similarity matrix is computed within the minibatch
  - Image-only and point cloud-only similarity matrices are also calculated and averaged together
- Loss function:
  - I is the image model output, and G is the point cloud model output:
  - $L_{\text{Point Cloud}} = \text{CrossEntropy}(IG^T, (II^T + GG^T)/2)$
  - $L_{\text{Image}} = \text{CrossEntropy}(IG^T, (II^T + GG^T)/2)$
  - $L_{\text{Complete}} = {}^1\!/\!n\ \Sigma(L_{\text{Image}} + L_{\text{Point Cloud}})/2$

Preliminary Results

Pretrained ResNet, PointNet Encoders

- Increasing model's capacity (more parameters) decreases validation loss
  - Suggests that currently underfitting
  - Hopefully will be able to do multi-GPU training, currently memory limited
- Batch sizes need to be larger to reduce spiky training curves
  - Currently with small batch sizes (64), there is high variance between batches
  - Batches should be large enough to represent complete dataset
- Joint embedding dimension of 256 seems to work well
- Tracking experiments with Weights & Biases





Preliminary Results (Cont.)

	PointNet	GAT	GCN
ResNet18 - Pretrained	0.1714	OOM	0.1548
ResNet18 - Not Pretrained	0.1733	OOM	0.1609

Validation F1 Score - Similarity Threshold 0.90 - Mean Pooling

	PointNet	GAT	GCN
ResNet18 - Pretrained	0.1771	OOM	0.1587
ResNet18 - Not Pretrained	0.1728	OOM	0.1787

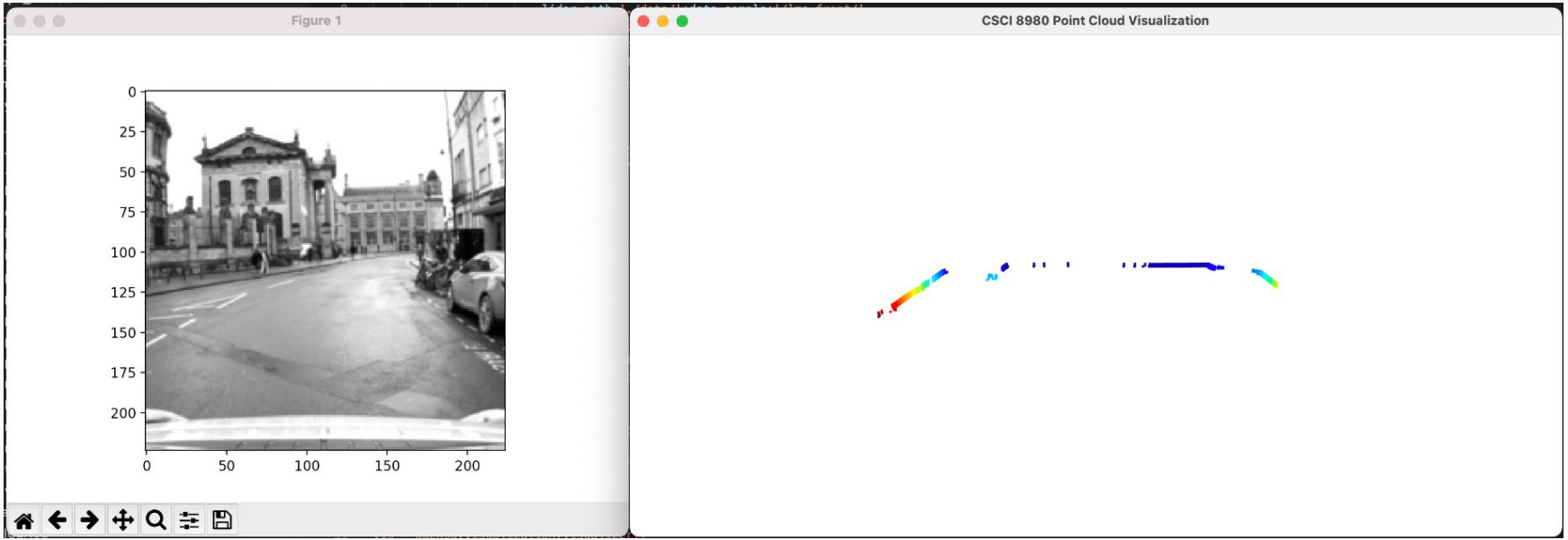
Validation F1 Score - Similarity Threshold 0.5 - Mean Pooling

	PointNet	GAT	GCN
ResNet18 - Pretrained	3.365	OOM	3.407
ResNet18 - Not Pretrained	3.355	OOM	3.32

Final Validation Loss - Batch Size 64 - Mean Pooling

OOM - Out of Memory

Visualization



Challenges

- Very memory intensive model and data
  - Multi-GPU training should help (Data Distributed)
  - Currently working through a driver/package compatibility issue on MSI
- Efficient evaluation of sensor validation task
  - For each threshold, create a boolean mask of logit matrix
  - Sum the diagonal for true positives, batch size - true positives = false negatives
  - Sum upper triangular matrix (offset by 1) for false positives
- Should look into better data cleaning/quality if more parameters and larger batch sizes don’t fix
  - Possibly use multiple frames to be able to construct larger pointcloud
  - Memory cost - already at a premium
  - Time of inference - system loses benefits the longer it takes to build inference data

Future Work

- Visualize important features in image using libraries like Captum
- Evaluate more vision models
  - Deeper ResNets
  - Vision Transformers
- Further hyperparameter searches, ablation studies
- Use the image embedding to generate depth estimations
  - Could you pretrain for deployment in an environment with only a single camera available?
- Evaluate how well does a trained model transfer to other datasets
- Utilize a sequence of images/point clouds?

Conclusion

- Very compute intensive - probably only feasible for self-driving cars
  - Maybe you can fit one (image or point cloud) encoder on embedded device?
- The model architecture is doable in theory
  - Loss function works
  - Probably need more parameters than currently configured with
- So far, results have been mediocre at best
  - Current memory requirements are a bottleneck
  - Larger batch sizes and models might improve performance
  - Might need to embed larger point clouds - limits real-time inference rate

References

[1] L. Fridman, L. Ding, B. Jenik and B. Reimer, "Arguing Machines: Human Supervision of Black Box AI Systems That Make Life-Critical Decisions," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 1335-1343, doi: 10.1109/CVPRW.2019.00173.

[2] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever,
“Learning Transferable Visual Models From Natural Language Supervision,” *CoRR*, March 2021.

[3] W. Maddern, G. Pascoe, C. Linegar and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset", *The International Journal of Robotics Research (IJRR)*, 2016.

[4] R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77-85, doi: 10.1109/CVPR.2017.16.

[5] P. Velickovic, G. Curcurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, “Graph Attention Networks”, *ICLR*, January 2018

[6] T. Kipf, M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” *ICLR,* February 2017