CMSC 636 Data Visualization [Group #2: *SeeBeL*]
Revised Project Proposal & Annotated Bibliography

Sourajit Saha, Shubhashis Roy Dipta, Shalima Binta Manir,
Budhini Amaraneni, Vishnu Gokanakonda, Varun Kunde

1.1. Project Proposal:

In this project we propose to study how Data Visualization can aid machine learning systems users (researchers, engineers, practitioners) analyze the effects of class imbalance on semantic segmentation performance. The domain of the dataset that we work on for designing our proposed visualization is image. Semantic segmentation is a high level computer vision task where each pixel is assigned a class label delineating the kind of object present in that pixel. Image datasets that are usually used [1-8] to train machine learning models to perform semantic segmentation are imbalanced in terms of class distribution. Moreover, class imbalance can negatively impact [9] the performance of semantic segmentation systems and therefore investigating how class imbalance and performance of semantic segmentation models are correlated is a significant research topic that requires investigation and with our proposed visualization, we aim to assist machine learning systems users to visually inspect the aforementioned correlation. Our proposed visualization consists of three key components:

1. User Compares Correlation: The user will be able to compare correlation between class distribution and performance for semantic segmentation. We aim to design this visualization so that users can visually inspect both prediction scores (by the trained model) and the frequency (class distribution) and compare between these two attributes for all of the object classes present in the image dataset of choice. Our goal is to help the user compare this correlation between the aforementioned attributes for all classes so that they have an informed understanding of how the model performs on each class and if at all there exists a correlation between class performance and class distributions.

2. User Discovers Trends: With this visualization, the user will be able to discover trends of how the segmentation performance for each class changes with every training iteration. Traditionally in machine learning literature, the overall performance is empirically visualized for every training iteration, however the overall (averaged across all classes) performance does not depict deviations in performance across classes. Therefore, to visually discern how each class is performing, a visualization that depicts the performance for each class at every iteration of training has significant research value and that is the driving purpose of this particular visualization. Moreover, adaptively changing weights in loss function is a well-known technique [10] to mitigate class imbalance in machine learning literature. With this visualization, the user can discover if the model performs better for particular classes (known as easy class) and not as much for some other classes (known as hard class). In case of discrepancy of performance among classes, it is also possible for the users to adaptively update the weights in the loss function to bolster learning the hard classes with increased weights.

3. User Discovers Distributions: The user will be able to discover distribution for both the model's average predicted probability for a region being classified to the ground truth (also known as target) class and the percentage of pixel area belonging to that region in comparison to the entire image that the user selected. And by pixel area, we refer to a continuous area in the image plane where all of the pixels belong to one particular object class. The goal of this visualization is to help the user discover the distribution of pixels and the amount of confidence that the model assigns to the corresponding target class for a region within an image and we aim to design this visualization over an entire image for all the object regions within the image.

1.2. The problem potentially solved with our proposed visualization:

Our proposed visualization can potentially become a helping tool for researchers who train semantic segmentation models and individuals who are responsible for policy making to deploy these systems in real world application. Semantic segmentation datasets are, more often than not, imbalanced and different techniques such as weighted loss function [10], resampling data [11] are usually used to mitigate these challenges. However, all of these techniques require an understanding of the dataset's statistics. These statistics include distribution of classes across number of images (number of images containing some object may be different than the number of images containing some other object), distribution of classes across amount of pixel area (amount of pixel area containing some object may be different than the amount of pixel area containing some other object). With the knowledge of class distribution, the user then can design loss function by assigning higher weights to the rare classes (low frequency) and lower weights to the abundant classes (low frequency) before initiating the training of the machine learning models. Sometimes, the aforementioned loss weights require adaptive update and to understand in which direction an update is required (if at all), it is essential to know the performance on all the classes as the training iteration advances. And this knowledge can guide the user to adaptively change loss weights, learning rates or to even re-initiate training with a different approach if the performance for rare classes are still not improved which can reduce both time, cost and carbon footprint since training of large machine learning models is expensive, time consuming and power hungry. As described in the previous section, our visualization thus, can help the user depict these statistics, both before initiating the training and while training.

1.3. Typical Users:

(A) Machine Learning (ML) and/or Computer Vision (CV) practitioners/researchers: One might want to compare how the ML/CV system is attending to each class at every iteration and/or what is the correlation between the observed class frequency and/or object size in the dataset and mean IOU per class which is a standard performance metric to evaluate semantic segmentation systems.

(B) Self-driving Car Manufacturers: Discovering how the trained system is performing on each class (good performance of identifying other vehicles and bad performance on identifying pedestrians) is important to decide whether or not to deploy such a system.

(C) Medical Imaging Community (Diagnostics and Research): Discovering the effect of Domain (source of training data i.e., population/ethnicity etc.) on Computer Aided Diagnosis is a significant deciding factor to use such systems.

1.4. Tasks:

(A) Study how abundance of imagery from a particular class and lack of examples from another can lead to biased Computer Vision models (bias towards the more frequently represented class in the dataset). The user will be able to compare correlation between class distribution and performance for semantic segmentation.

(B) The user will be able to discover trends of how performance for each class is changing for semantic segmentation with every training iteration.

(C) The user will be able to discover distribution for both the model's average predicted probability for a region being classified to the ground truth class and the percentage of pixel area belonging to that region in comparison to the entire image that the user selected. To clarify, the user, in the first place, needs to select an image from the validation set and then we will show the aforementioned visualization on it.

2. Datasets:

The dataset that we will use is called the Cityscapes [2] dataset. The following steps are to be followed in order to download the dataset.

- Goto: https://www.cityscapes-dataset.com/
- Navigate to: https://www.cityscapes-dataset.com/downloads/
- Create an account (with institutional email) and log-in in order to download the data.
- Download the following files:
  >> gtFine_trainvaltest.zip (241MB) [md5]
  >> leftImg8bit_trainvaltest.zip (11GB) [md5]

The cityscapes dataset has 5000 images with an identical resolution of 1024 × 2048 pixels. The dataset further contains high quality ground truth labels and all of the images are taken while driving in different urban streets. Cityscapes contains 2975 training images, 500 validation images and 1525 test images contributing to a total of 5000 images. Moreover, the 1525 test images do not contain annotated labels as those labels are not made public and are only used internally to score models submitted to Cityscapes leaderboard. And therefore, we only use the training images for training and validation images for testing to serve the purpose of this project. The annotated labels provided in the dataset account for 30 classes. Although. only 19 classes are used for evaluation and therefore all of our experiments including the preprocessing of the data and the corresponding Visualizations will contain 19 classes.

Such datasets are usually collected via setting up a high resolution camera on the exterior of a vehicle and the images are recorded frame-by-frame. And then each of these images are manually labeled at every pixel adhering to a common set of rules for the entirety of the dataset. The semantics of this dataset are street scene images captured from vehicles and for every image in the dataset there exists a corresponding ground truth image that describes the objects present in the image and the position (pixel location) and name of the object.

Another way to think of this data is that all the pixels in an image contain an object of some kind (people, car, road etc.) and the dataset uses color (different color code to represent different objects) for every pixel to describe what kind of object is present in that pixel, as shown in Figure 1.

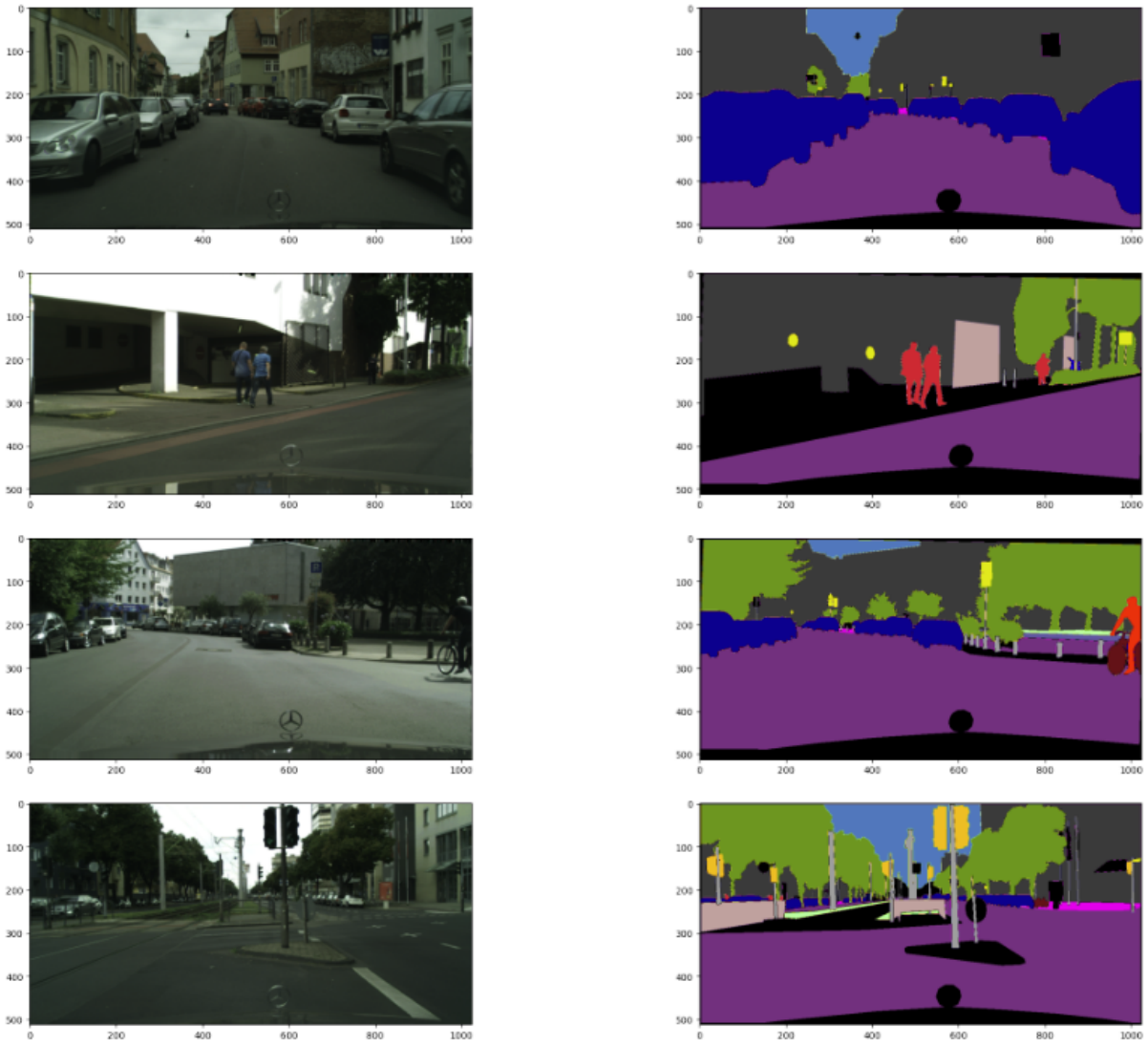

Figure 1: A Snippet of Cityscapes Dataset. *Left column:* Street scene images, *Right column:* Ground truth labels (red is person, blue is car etc.) describing what object each pixel represents.

The dataset type for Cityscapes is geometry since images convey information of items with explicit spatial positions on a 2D surface and therefore the data type for cityscapes are items and spatial positions. One of the limitations on Cityscapes dataset is abundance of background class meaning only a few objects of interest are labeled, leaving a lot of pixel space unlabeled (background class). Moreover, since we are visualizing the performance on the cityscapes dataset, one of the challenges in getting good performance on this dataset is the limited number of labeled training images [3,4].

Now, there are multiple stages of transformation that we need to perform on the data. Firstly, since we need to get our data ready to be trained, we need to resize the images from the original resolution (1024 x 2048) to 512 x 1024 pixels to match the computational capacity of our training hardware (Nvdia RTX 3090 GPU). Training the dataset will yield prediction and we will store the prediction for all the training iterations. Now, we perform these transformations because predictions for all classes at every iteration are some of the attributes in our proposed visualizations as described in section 1.1. We further perform more transformations to compute class distributions because this is another attribute in our proposed visualization (described in section 1.1). We will traverse through the entire training set images and compute the total amount of pixel area occupied by each object class which we will finally normalize by the total pixel area in all the images in the training set to estimate class distribution. Finally, for the third visualization (depicted in section 1.1) we need to compute the area and mean prediction probability for all continuous object areas in the images to visualize the surface area and mean prediction distribution for the user. For further clarification, mean IOU (mIOU) is a standard performance metric used to measure semantic segmentation performance and IOU stands for intersection over union which is computed between the ground truth mask and predicted mask.

3. Project Contribution:

There are three visualizations that we are proposing for this project, each with a unique purpose as we described in section 1.1. In the following, we sketch how and what information the user will be able to uncover with each of our proposed visualizations and how they are different from the concurrent and/or related visualizations in driving their respective purposes.

The purpose of our first visualization is to help the user compare correlations between class distributions and performance for each class. Being able to compare these two attributes is quintessential to building robust semantic segmentation systems and no current work in computer vision and visualization offers a solution or visualization that allows users to compare the correlations between class distribution and performance for semantic segmentation datasets under one unified design. There are studies that discover the distribution of different classes in the cityscapes dataset in terms of number of pixels occupied by each class [2,14], proportion of labeled pixels for each class [2], number of images for each class [2] etc. These studies however are limited to analyzing (discovering) class distribution alone and do not compare any correlation between class distribution and performance. On the other hand, different studies - to depict the efficacy of their respective machine learning models - have reported [12,13] a detailed overview of performance for each class trained on various semantic segmentation datasets, yet once again lacking to show any connection between performance and class distribution. To the best of our knowledge, our visualization is the first to address a correlation between the aforementioned attributes. We plan to use Bubble Charts as our idiom for this visualization. We will use the X axis to denote class labels for the dataset, Y axis to plot the model accuracy (mIOU: a number between 0 to 100). And finally, with the size (diameter) of the bubble we denote how rare or frequent a class is in terms of pixel area it occupies.

The purpose of our second visualization is to give the user the ability to discover the trends of how performance is changing after every iteration of training. Since the current machine learning literature [9] lacks visualization of per class performance for every iteration, we hope to bridge the aforementioned gap with our proposed visualization (second visualization). We plan to use a facet grid (multi plot-grid). In this case we use multiple bar plots (for each class in the dataset) with iterations in the horizontal spatial position and IOU at every iteration in the vertical spatial position axis. And therefore each barplot alone will allow the user to discover trend as the model trains for a fixed number of iterations and with the multi plot-grid the user can globally discover trend although that will impose some degree of cognitive load and we will, in the future, attempt to design a visualization that circumvents this issue.

Finally, with our third visualization the user will be able to discover distribution for both the model's average predicted probability for a region being classified to the ground truth (also known as target) class and the percentage of pixel area belonging to that region in comparison to the entire image that the user selected. And by pixel area, we refer to a continuous area in the image plane where all of the pixels belong to one particular object class. The goal of this visualization is to help the user discover the distribution of pixels and the amount of confidence that the model assigns to the corresponding target class for a region within an image and we aim to design this visualization over an entire image for all the object regions within the image. We plan to use choropleth maps as an idiom for this visualization. Furthermore, we plan to create an interactive design for this visualization so that when the user selects an area/segment of the image, two distribution bars pop up with the average prediction and pixel area for the selected area/segment.

4. Project Charter:

Team Name: **SeeBeL** *[shorthand for **See**ing Is **BeL**ieving]*

Group roles:

1. **Sourajit Saha - *Code Monkey***
2. **Shubhashis Roy Dipta - *Code Monkey***
3. **Shalima Binta Manir - *Resource Manager***
4. **Vishnu - *Technical Writing Manager***
5. **Varun - *QA/QC Manager***
6. **Budhini - *Project Manager***

Meeting Time & Date: Every Wednesday 8-9 PM

Preferred Writing Tool: Overleaf, Google Docs

Preferred Collaboration Tool: Github

Communication Preference: Google Meet, Discord

Conflict Resolution Strategy: Voting

Topic Area: Machine Learning, Computer Vision, Visualization

Tools: Seaborn, Matplotlib, Tableau

5. Related Work:

Cost Curves [15] is an essential visualization method to uncover machine learning model performance over a range of class distributions. The idiom for cost curve is line plot and the authors plot probability cost of a wide range of sampling distribution over a dataset and their corresponding normalized expected cost. Moreover, to encode data, the authors have used horizontal and vertical spatial positions and point marks for the user to find trend and correlation. However, generalizing the cost curve method to semantic segmentation model is computationally (both memory and time) expensive since (1) cityscapes has 19 trainable classes and the cost curve method uses only two classes to sample a wide range of distribution samples and (2) the complexity of a segmentation model is much higher than that of a classification model. Furthermore, from a visualization perspective the cost curve is not easily interpretable and it takes domain specific knowledge on the user's end and added cognitive load to compare a correlation between class distribution and model performance.. Therefore, we only propose to report model performance and class distributions of the dataset using Bubble Charts to compare correlation between them in our project.

In Natural Language Processing (NLP), attention based models have achieved state-of-the-art results in different domains of NLP tasks. But the interpretability of how they work remains a mystery. In this work [16], the authors have used different idioms to increase the interpretability of attention based models. Authors have provided three interactive idioms, (1) attention-head view, (2) model view and (3) neuron view. In the attention-head view, authors have explored the self-attention weights for one or multiple heads for each input. Subsequently, in the model view the authors have presented a high level overview of the whole model's attention heads. And in the neuron view, authors have drawn an idiom to show the individual neuron weights with respect to query, key, value. All of the views are interactive with the option to choose layers and heads (1, 3) and zoom in on the individual head (2). Even if our proposal is in a different domain (Computer Vision) than this work, this paper motivated us to pursue our goal to make the vision models more interpretable by using different visualizations. Another main difference is that, in our work we have used visualization to explore both the training and prediction stage rather than only focusing on prediction model.

Another study [17] was recently conducted which proposed radial bar chart based visualization of classification performance. They have used the radial position to represent model performance and the color and angular position to represent classes. The authors further show how they draw different numbers of samples from a dataset and report the error rate for all of them after certain iterations. However, using such design for our purpose will be cumbersome as we have more number of classes in cityscapes and discovering trends from different figures (since it requires one visualization per iteration in this design mechanism) will increase cognitive load on the user's part.

This study [18] proposes a visualization that depicts the internal information processed by an autonomous vehicle so that the user, with a reduced cognitive load, can access that information and decide whether or not to trust the vehicle's next set of steps and take control of the vehicle to themselves. It allows the users to perceive the vehicle's detection capabilities and the authors processed the semantic segmentation of road scene images for image classification based on pixels. They use color hue for categorical attributes to show the color distributions of the object visualization. Also, different shapes are used as identity channels to show dynamic and static objects. Furthermore, box plots and violin plots are used to show the distribution of data points. More specifically violin plots are used to show the perceived ability to detect dynamic objects and static objects and the ranking of the system. The focus of this visualization (using cityscapes dataset) is making decisions on the use of color to maximize Situation Awareness rating technique (SART score), it does not depict the class distribution per object. On the other hand, we plan to visualize class distribution (pixel area or number of images) per object class in this project.

This visualization [19] aims to learn insights in the training dataset from the corresponding classification results. Furthermore, this study aims to find how classification performance is related to class distribution by visually inspecting how far a predicted class is from the ground truth class, in the class map space. The authors further show how the authors use scatter plot as their choice of idiom and the horizontal spatial position is used to denote class distance and the vertical axis spatial position is used to denote the probability that the model assigns to alternative classes. One of the major issues with this visualization is that it needs to be repeated for each class and therefore using this mechanism to visualize the semantic segmentation performance for all the classes (19 of them) will be challenging and discovering trend from separate graphs will pose added cognitive load on the user's part.

InstanceFlow [20] is a sophisticated visualization tool that allows users to compare learning behavior between iterations on an instance level. They use a sankey diagram with glyphs to depict temporal analysis of the training process. With their proposed visualization the authors offer a unified solution to discover trends between performance at different iterations, at the expense of a complicated visualization and added cognitive load. We aim to allow the user to discover the exact trends, with a much simpler visualization with lesser strains on cognition.

References:

1. A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," International journal of robotic research, vol. 32, no. 11, pp. 1231–1237, 2013.
2. M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3213–3223.
3. T.-Y. Lin et al., "Microsoft coco: Common objects in context," in Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, 2014, pp. 740–755.

4. B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ade20k dataset,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 633–641.
5. C. Couprie, C. Farabet, L. Najman, and Y. LeCun, “Indoor semantic segmentation using depth information,” arXiv preprint arXiv:1301.3572, 2013.
6. R. Mottaghi et al., “The role of context for object detection and semantic segmentation in the wild,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 891–898.
7. G. J. Brostow, J. Fauqueur, and R. Cipolla, “Semantic object classes in video: A high-definition ground truth database,” Pattern Recognition Letters, vol. 30, no. 2, pp. 88–97, 2009.
8. J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, “The 2017 davis challenge on video object segmentation,” arXiv preprint arXiv:1704.00675, 2017.
9. M. S. Hossain, J. M. Betts, and A. P. Paplinski, “Dual Focal Loss to address class imbalance in semantic segmentation,” Neurocomputing, vol. 462, pp. 69–87, 2021.
10. M. J. Jozani, É. Marchand, and A. Parsian, “On estimation with weighted balanced-type loss function,” Statistics & Probability Letters, vol. 76, no. 8, pp. 773–780, 2006.
11. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” Journal of artificial intelligence research, vol. 16, pp. 321–357, 2002.
12. X. Zhang et al., “Dcnas: Densely connected neural architecture search for semantic image segmentation,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 13956–13967.
13. T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, “Gated-scnn: Gated shape cnns for semantic segmentation,” in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 5229–5238.
14. G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. Jawahar, “IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments,” in 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019, pp. 1743–1751.
15. C. Drummond and R. C. Holte, “Cost curves: An improved method for visualizing classifier performance,” Machine learning, vol. 65, pp. 95–130, 2006.
16. J. Vig, “A Multiscale Visualization of Attention in the Transformer Model,” in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2019, pp. 37–42.
17. A. Luque, M. Mazzoleni, A. Carrasco, and A. Ferramosca, “Visualizing classification results: Confusion star and confusion gear,” IEEE Access, vol. 10, pp. 1659–1677, 2021.
18. Colley, Mark, et al. ”Effects of semantic segmentation visualization on trust, situation awareness, and cognitive load in highly automated vehicles.” Proceedings of the 2021 CHI conference on human factors in computing systems. 2021.
19. J. Raymaekers, P. J. Rousseeuw, and M. Hubert, “Class maps for visualizing classification results,” Technometrics, vol. 64, no. 2, pp. 151–165, 2022.

20. M. Pühringer, A. Hinterreiter, and M. Streit, “InstanceFlow: Visualizing the evolution of classifier confusion at the instance level,” in 2020 IEEE visualization conference (VIS), 2020, pp. 291–295.