Sensor Training Data Reduction for Autonomous Vehicles

Matthew Tomei
University of Illinois at Urbana-Champaign
tomei2@illinois.edu

Alexander Schwing
University of Illinois at Urbana-Champaign
aschwing@illinois.edu

Satish Narayanasamy
University of Michigan
nsatish@umich.edu

Rakesh Kumar
University of Illinois at Urbana-Champaign
rakeshk@illinois.edu

ABSTRACT

Autonomous vehicles requires good learning models which, in turn, require a large amount of real-world sensor training data. Unfortunately, the staggering volume of data produced by in-vehicle sensors, especially the cameras, make both local storage and transmission of this data to the cloud for training prohibitively expensive. In this work, we explore techniques for reducing video frames in a way that the quality of training for autonomous vehicles is minimally affected. We particularly focus on utility aware data reduction schemes where the potential contribution of a video frame to enhancing the quality of learning (or utility) is explicitly considered during data reduction. Since actual utility of a video frame cannot be computed online, we use surrogate utility metrics to decide what video frames to keep for training and which ones to discard. Our results show that utility-aware data reduction schemes can reduce the amount of camera data required for training by as much as 16× compared to random sampling for the same quality of learning (in terms of IoU).

ACM Reference Format:
Matthew Tomei, Alexander Schwing, Satish Narayanasamy, and Rakesh Kumar. 2019. Sensor Training Data Reduction for Autonomous Vehicles. In *2019 Workshop on Hot Topics in Video Analytics and Intelligent Edges (HotEdgeVideo'19), October 21, 2019, Los Cabos, Mexico.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3349614.3356028

1 INTRODUCTION

Safe and reliable autonomous driving will require good learning models, which, in turn, will require a large amount of real-world training data. In general, the more training data available, the better the inference [7]. In order to generate real-world training data, in-vehicle sensors can be used. Autonomous vehicles are equipped with a large number of sensors. These sensors (e.g., cameras, LIDARs, sonars, radars, GPSes, IMUs, etc.) gather data about the environment in order to recognize and track objects, as well as localize the vehicle. Tracking and localization information is then subsequently used along with human driver input to plan the next set of actions for the physical vehicle. The same sensor data can be used to train and improve learning models, by sending data (in real-time or later) to the cloud, which then help make better driving decisions. This sensor data for training can be produced even by conventional vehicles which have been fitted with sensors specifically to generate such data [6].

Unfortunately, management of this sensor data for training and improving learning models for autonomous vehicles is extremely challenging since the volume of sensor data, especially camera data, produced by an autonomous vehicle can be staggering. Industry analysts expect that an average car can produce about 4 TB of sensor data for just one hour of driving [11], most of it produced by cameras. That is about 100,000× more than an average person's media consumption per day over the Internet today [12]! Both local storage and transmission of the sensor data to the cloud for subsequent use in training learning models can be prohibitively expensive. Even if data is transmitted, sensor training data may overwhelm cloud storage and total cost of ownership (TCO), especially if data needs to be retained for training of future models. Processing large amounts of sensor data for training also has energy and latency costs.

This paper focuses on the following technical challenge – *how do we store or transmit only a fraction of video frames produced by an autonomous vehicle (limited by storage, bandwidth, or energy resources), while achieving the same quality of learning (e.g., training) as can be achieved when all data from the vehicles is transferred to the cloud?*. Stated differently, can we learn equally well with a much smaller amount of data if data subsetting is performed intelligently?

Conventional schemes for data reduction (e.g., based on periodic or random sampling or on compression) do not consider the utility of data for training. Such schemes may also end up discarding data that may be crucial to learning new scenarios.

HotEdgeVideo'19, October 21, 2019, Los Cabos, Mexico

ACM ISBN 978-1-4503-6928-2/19/10…$15.00
https://doi.org/10.1145/3349614.3356028

We show that such *utility-agnostic schemes*, therefore, lead to poor training when data is reduced aggressively. Instead, we argue that data reduction schemes in autonomous vehicles should be *utility-aware*. Such schemes explicitly consider the utility of data for learning when making decisions about whether to log or discard the data.

2 UTILITY AWARE REDUCTION

2.1 Definition of Utility

In general, utility is a measure of value attributed to a specific context. Hence, in our case, for a datapoint x, its utility is a measure of the value attributed to it supporting 'autonomous driving.' More specifically, since classifiers are a common tool to achieve autonomous driving, we use utility as a measure to support and improve classifier training.

Let $\Omega = \{x_1,\ldots,x_{|\Omega|}\}$ denote the set of all the recorded data that was available at some point in time. From this dataset we want to select a subset S of samples ($|S| \ll |\Omega|$), such that its utility $U(S)$ is maximal, i.e.,

$$S^* = \operatorname*{argmax}_{S \in \mathfrak{S}} U(S). \tag{1}$$

Here, $U(S)$ is a set utility function, and $\mathfrak{S}$ is the set of all possible reduced sets that satisfies a set of constraints that we care about, (e.g., the maximally available disk space).

To define the utility of data for a classifier, note that we generally care most about accuracy of a classifier on some metric M computed by using a held-out test set $D_{\text{test}} = \{(x,y^*)\}$ which contains pairs of datapoints $x \in X$ unavailable during training and corresponding groundtruth labels $y^* \in Y$. The metric M, e.g., accuracy, Jaccard index (also known as Intersection over Union) etc., compares the classifier output $f_{w(S)}(x)$ to the groundtruth y^*. Hence, we obtain the utility

$$U(S) = \sum_{(x,y^*) \in D_{\text{test}}} M(f_{w(S)}(x), y^*),$$

where we assume that every sample can be processed independently.

2.2 Surrogate Utility Metrics

Unfortunately, there is lack of groundtruth for estimating utility. In addition, even with the groundtruths, evaluating the utility of every set S would be prohibitively expensive. We explore several suitable per-video-frame surrogates for data utility, which remove the dependence of the utility on the groundtruth labels $y(x)$ and reduce the search space significantly.

2.2.1 Entropy. Entropy measures the uncertainty of a distribution $p_w(y|x)$ over labels $y \in Y$ computed by the classifier $f_w(x)$. Intuitively, the more uncertain the classifier, the more potential for a datapoint x to make a big impact during training.

Formally, the entropy based utility is given by

$$U(S) = \sum_{x \in S,\, y \in Y} -p_w(y|x) \ln p_w(y|x). \tag{2}$$

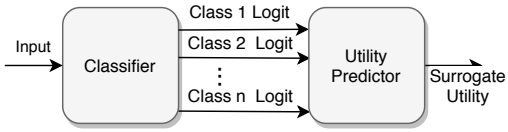

Figure 1: Logits can also be used to predict a utility metric that would otherwise require groundtruth data.

The distribution employed by the classifier is obtained from

$$f_w(x) = \operatorname*{argmax}_{y \in Y} p_w(y|x) := \frac{\exp \hat{f}_w(x,y)}{\sum_{y \in Y} \exp \hat{f}_w(x,y)}, \tag{3}$$

where $\hat{f}_w(x,y)$ are the logits which are transformed into a probability distribution via the soft-max operation given on the right-hand side. The soft-max calculation exponentiates a real number to ensure non-negativity while the denominator guarantees that a sum over all possible $y \in Y$ equates to one, making $p_w(y|x)$ a valid probability distribution.

2.2.2 Approximate Entropy. In some cases, computation of the entropy is not tractable because the output space Y is too large. For instance, for semantic segmentation of images, the size of the output space is $|Y|^M$, where M is the number of pixels in the image. This translates to, in the denominator of Equation 3, a sum over a number of terms that scales exponentially with the number of pixels. This is not computationally tractable, so the probabilities $p_w(y|x)$ (resp. the logits $f_w(x,y)$) are not computed. Instead, only M marginal distributions (resp. their logits) are computed. In the case of semantic segmentation, these marginal distributions are per-pixel. So, to compute approximate entropy, we compute Equation 3 for each pixel and average the results over the whole image.

2.2.3 Dropout Entropy. Previous work has shown that soft-max based entropy does not always sufficiently represent uncertainty of the model [3, 5, 8]. To overcome this weakness, it has been proposed to sample multiple probability distributions $p_w(y|x)$ via inference with different random applications of dropout [5]. More specifically, dropout randomly sets deep net activations in the computation to zero before passing them on to the next layer. Hence different probability distributions are obtained and their variance gives us a potentially more robust surrogate utility metric that is more costly to compute because multiple deep net forward passes are required.

2.2.4 Accuracy. It is possible that a surrogate utility requiring the groundtruth outperforms any other. For example, we may want to use accuracy as a surrogate utility, but knowing whether a prediction was correct requires the groundtruth. Consider however that we would expect higher confidence predictions, as determined by logits, to be more likely to be accurate. To approximate accuracy, we could train some accuracy inference engine offline that takes logits (available at runtime) as input and use the predictor as shown in Figure 1.

3 METHODOLOGY

3.1 Classification Task

We evaluated our data reduction strategies on an inference engine performing semantic segmentation. Semantic segmentation is the classification of each pixel according to what that pixel represents in the image. We chose DeepLab v2 [1] as our semantic segmentation baseline network since it provides a convenient tradeoff between accuracy and training time. On the challenging Pascal VOC benchmark [4], DeepLab v2 achieves a compelling 80% mean average precision (current state-of-the-art is 89% achieved by DeepLab v3). On the Cityscapes dataset, DeepLab v2 achieves 70.4% mean IoU while DeepLab v3 achieves 82.1%. On our machines (nVidia Titan Xp), training of DeepLab v2 takes 14 hours, while training of DeepLab v3 is reported to take 3.65 days using a single GPU [2]. In addition, sources for both training and testing are publicly available, simplifying reproducibility of our setup. We evaluate our data reduction schemes on the Cityscapes and Berkeley Deep Drive (BDD) datasets. To compare the result of training the semantic segmentation engine on different reduced data sets, we used the classical Jaccard Index [10]. The Jaccard Index, also known as the Pascal VOC Intersection-over-union (IoU) metric is a standard on semantic segmentation datasets. For a given constraint on data, we report the maximum IoU over all training runs that used amounts of data smaller than that constraint.

One limitation of the Cityscapes and BDD datasets is the relatively small number of frames in these datasets (5000 and 8000 respectively). Also, the creators of the datasets already performed some subsampling of frames when they chose which ones to label (i.e., the frames in these datasets are not contiguous). This reduces the perceived impact of our techniques since one would expect significant redundancy across contiguous frames generated in a vehicle. To try to account for this, we developed a synthetic dataset using a set of over 100,000 unlabeled images that are contiguous in time and available as a part of the Cityscapes dataset. These images comprise the full set of images collected during drives in Frankfurt. We labeled the images using a pretrained Deeplab V3 model, which we tested to have a mean IoU of 78% on the validation set, and used these labels as the ground truth. For images that were included in the original Cityscapes dataset, we used the given human generated labels. We used the original Cityscapes validation set to test the models trained on this synthetic dataset.

3.2 Utility-Agnostic Sampling-based Camera Training Data Reduction

Our naïve reduction schemes perform utility-agnostic sampling from an image stream, as shown in Figure 2. We have blown up the image stream to show the individual images sorted by time and then location and vice versa. This allows

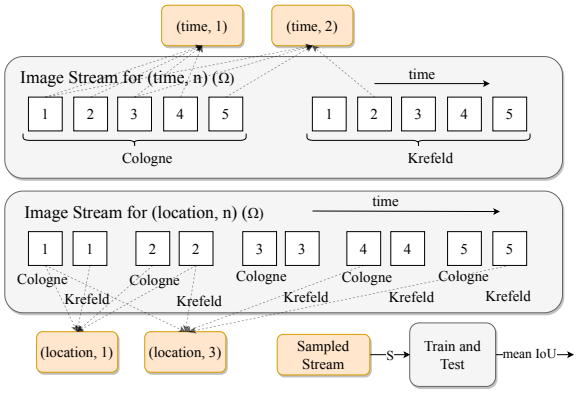

Figure 2: We evaluated image streams sampled in time with both minimum and maximum variation in time.

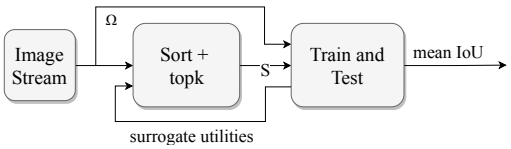

Figure 3: Surrogate utility-based data reduction

us to consider sampling based on both primarily time or primarily location of the images. The format of the labels on the reduced image streams in orange in Figure 2 is (variation type, step size). The variation type tells us whether we order the stream by time and then location (time), or by location and then time (location). The step is made through the stream with the specified order. This sampling specification format is also used in legends in later figures.

3.3 Utility-Aware Camera Training Data Reduction

An example of the flow for utility-aware data reduction is shown in Figure 3. As opposed to Figure 2 where there was no feedback from the train and test block, we now require feedback in the form of surrogate utility data used to rank incoming video frames. The surrogate utility computation is appended to the inference task as described in Section 2.

3.4 Surrogate Utility Metrics

We evaluated all the surrogate utility metrics described in Section 2. We also trained and tested accuracy predictors described in Section 2 using all the machine learning classifiers available in scikit-learn [9]. We then implemented the trained perceptron and also the entropy calculation from Section 2 in Tensorflow so we could evaluate their computational cost relative to the cost of generating logits and taking the argmax (the baseline inference task). We used Tensorflow's built in profiling functionality to measure the latencies of baseline inference, calculating entropy, and predicting accuracy.

To get surrogate utility values, we used an initial randomly reduced training set. We chose a random reduction by 32× for the Cityscapes and BDD datasets, and a random reduction

of 1024× for the much larger synthetic dataset. The random reductions resulted in IoU losses of 13.6%, 12.9%, and 17% for the three datasets respectively. These reductions were chosen to cause a drop in IoU significant enough to differentiate the utility-aware metrics from the utility-agnostic ones without making the surrogate utility measurements useless. The randomly reduced training sets are used to train Deeplab V2, and then the surrogate utility values used for all the utility-aware evaluations are generated using this trained model.

4 RESULTS

4.1 Quantifying Impact of Data Reduction on Training Quality

Figure 4 shows the impact of naïve (utility-agnostic) sampling schemes on mean IoU for the Cityscapes dataset. The baseline mean IoU of the network trained without any sampling was 0.704. Random sampling (averaged over three runs with different seeds) resulted in performance 12% below the optimal IoU for an 8× reduction in data. The schemes that minimize variation in location (*time, n*) have similar performance to random reduction, while the schemes that maximize variation in location (*location, n*) perform better than random, with their lines only intersecting for the lowest step size and for lower reductions. This suggests that variation in location is more important than variation in time for this data set. This is not surprising since the dataset has already been sampled in time, removing most of the potential savings. The best naïve sampling scheme (*location, 4*) resulted in performance 8% below the optimal IoU for an 8× reduction in data size. The large reduction in training quality for even a modest data reduction suggests that naïve sampling may not be a good fit for camera data reduction for autonomous vehicles.

Figure 5 shows the performance of utility-agnostic metrics on the synthetic Cityscapes dataset which only includes images from Frankfurt. For this dataset, the mean IoU without sampling was 0.651. Since all the images are gathered in the same city, we do not report any location based results. Due to higher density in time of the synthetic dataset, a period of 32 for the artificial dataset is equivalent to a period of ∼ 1 image for the original dataset in terms of the amount of time between images collected. We also see that, unlike with the original dataset, large sampling periods now significantly outperform random sampling for moderate amounts of reduction. At very large and very small amounts of reduction, any set of images collected start to look closer to a set of random images, so the IoU is similar. This same pattern is repeated with utility-aware metrics. A sampling period of 256 images results in 64× data reduction before the IoU starts to degrade compared to IoU without sampling. Random sampling starts exhibiting IoU degradation (vs. no sampling) even at a 4× reduction.

We were not able to evaluate naïve metrics (e.g., *(location, n) or (time, n)*) for the BDD dataset since neither time nor location metadata is available for the BDD images labeled for semantic segmentation. Therefore, the best utility agnostic metric is random sampling by default, which is presented next to the utility aware metrics in Figure 7. The IoU without sampling on this dataset was 0.571. The drop in absolute IoU compared to Cityscapes can be explained mostly by the larger number of classes. The IoUs for the car and road classes are 0.89 and 0.943 respectively for BDD compared to 0.915 and 0.970 for Cityscapes. Note that the reduction in IoU (as a function of data reduction) for random is larger for BDD compared to Cityscapes (∼ 6% compared to ∼ 2% at 4× reduction). This pattern holds for other metrics as well and implies that there is more variation in the images gathered for the BDD dataset compared to Cityscapes, making each image relatively more important for training.

Figure 6 compares utility aware schemes against the best performing utility agnostic technique (location, 4) for the Cityscapes data set. Entropy based methods perform the best for higher data reductions, achieving almost 2× the data reduction for a similar drop in mean IoU compared to naïve schemes. While this result is slightly non-intuitive given that other intelligent metrics such as accuracy and IoU use more information (i.e., groundtruth), it bodes well for online utility-aware data reduction since the simplest surrogate utility to compute also performed the best. We also note that adding a constraint on closeness of samples in time (Entropy, 1 versus Entropy, 4) had at worst a negative effect, further supporting a simpler sampler implementation. We attribute the difference in the random result from Figure 4 to the use of a different set of random samples.

Note that at 2× data reduction, all techniques (including random sampling) have a mean IoU within 1% of the baseline (0.704). Ideally, mean IoU would scale up with increasing data. Our results suggest either a limitation in the complexity of DeeplabV2 or in the amount of useful data in the Cityscapes dataset. Accuracy performs slightly better than the entropy based surrogate utility metrics at 2× reduction, achieving a 1% increase in mean IoU compared to entropy's 0.4% increase.

Figure 7 also compares utility aware schemes to random data selection for the BDD dataset. We see that the intelligent surrogate metrics again achieve close to a 2× reduction in data for the same reduction in mean IoU when compared to random sampling. E.g., the IoU for Entropy at 4× reduction is larger than the entropy for Random at 2× reduction.

Figure 8 shows utility aware schemes and the best performing naïve scheme for the synthetic dataset. Similar to the naïve schemes, the performance of utility aware schemes relative to random is better than it was for the smaller datasets. All utility-aware schemes other than the accuracy based scheme have a similar IoU to random subsetting for 16× less data. Even the naïve metric (time, 256) performs comparably to most intelligent metrics. These results suggest that the benefits from intelligent data reduction increase with the amount of data. This makes sense intuitively since with more data, there is greater flexibility to make intelligent decisions.

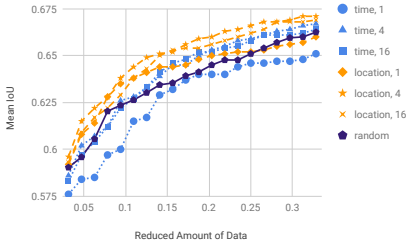

Figure 4: Utility-unaware techniques vs random.

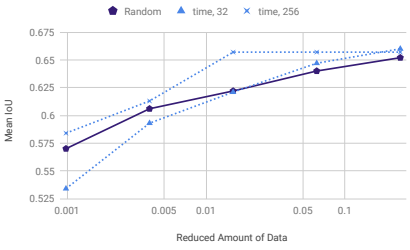

Figure 5: Utility-unaware techniques for the synthetic dataset.

format	quality	cmpr. ratio	mIoU
png	N/A	1×	0.704
png8	N/A	7.8×	0.659
jpg	100	1.4×	0.704
jpg	50	22×	0.687
jpg	1	61×	0.45

Table 1: Effectiveness of compression.

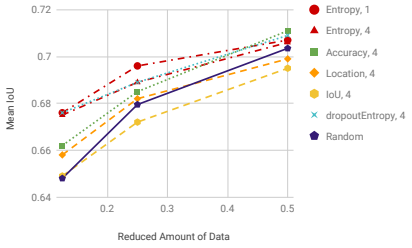

Figure 6: Utility-aware techniques reduce the impact of sampling.

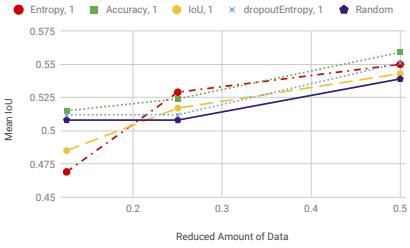

Figure 7: The benefit of surrogate metrics for the BDD dataset.

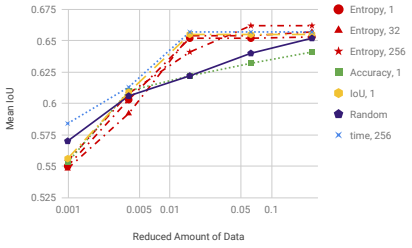

Figure 8: Utility-aware metrics on the synthetic dataset.

Figure 9: The human labeled vs synthetic labeled images.

4.2 Interaction with Compression

Table 1 shows the effect on mean IoU for the Cityscapes dataset when compression is applied by itself. Results show that large data reductions from compression with significant reductions in IoU. On applying compression before before sampling, we found that when using entropy as our surrogate utility and reducing the data by 8 and 4× using sampling, the reduction in mean IoU due to compression is 1.6% and -0.6% respectively, which is less than the 2.4% we saw without sampling. This suggests that compression and utility-aware schemes are at worst orthogonal, and possibly complimentary.

4.3 Validation of Synthetic Dataset

The synthetic dataset is a mix of inference generated and human generated labels. The mean IoU when training with only the human generated labels is only 0.502, much lower than the value of 0.651 when we include the inference generated labels. This tells us that the inference generated labels are useful for training.

We also measured similarity between inference generate and human generated labels by looking for any bias towards either type of image when performing data reduction with the synthetic data set. Figure 9 shows how many human generated labels we collect for different total amounts of data collected. We see that the images with inference generated labels are chosen at just about the same rate as those with human generated labels. This suggests that the inference generated labels are just as useful as the human generated ones for training and not more.

5 CONCLUSION

We focused on the problem of camera training data reduction such that impact on the quality of training is minimized. We showed that naive data reduction schemes that do not consider utility of data for training have limited effectiveness. We explored utility aware data reduction where the potential contribution of a video frame to enhancing the quality of learning (or utility) is considered during data reduction. Since actual utility of a video frame cannot be computed online, we used surrogate utility metrics to decide what video frames to keep for training and which ones to discard. Our results show that utility-aware data reduction schemes can reduce the amount of camera data required for training by as much as 16× compared to random sampling for the same quality of learning (in terms of IoU).

REFERENCES

[1] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *arXiv:1606.00915 [cs]* (June 2016). arXiv: 1606.00915.

[2] Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv:1706.05587 [cs]* (June 2017). arXiv: 1706.05587.

[3] Denker, J. S., and LeCun, Y. Transforming Neural-Net Output Levels to Probability Distributions. In *Advances in Neural Information Processing Systems 3*, R. P. Lippmann, J. E. Moody, and D. S. Touretzky, Eds. Morgan-Kaufmann, 1991, pp. 853–859.

[4] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[5] Gal, Y., and Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *arXiv:1506.02142 [cs, stat]* (June 2015). arXiv: 1506.02142.

[6] Geiger, A., Lenz, P., and Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. CVPR* (2012).

[7] Harris, D. Baidu's chief scientist explains why computers won't take over the world just yet, Sept. 2015.

[8] Kendall, A., Badrinarayanan, V., and Cipolla, R. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. *arXiv:1511.02680 [cs]* (Nov. 2015). arXiv: 1511.02680.

[9] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[10] Real, R., and Vargas, J. M. The Probabilistic Basis of Jaccard's Index of Similarity. *Systematic Biology 45*, 3 (Sept. 1996), 380–385.

[11] Winter, K. For self-driving cars, thereâĂŹs big meaning behind one big number: 4 terabytes. "https://newsroom.intel.com/editorials/self-driving-cars-big-meaning-behind-one-number-4-terabytes/".

[12] Xfinity. What is the median usage of people on your network today?, June 2018.