The Practical Impact of Data-Centrism on the Example of Autonomous Driving

Raphael Emberger
Project thesis in MSE in Computer Science
Zurich University of Applied Sciences (ZHAW)
Winterthur, Switzerland
emberrap@students.zhaw.ch

***Abstract*—Data centrism is a relatively young methodology in deep learning and is thus sometimes misunderstood. Both research and industry seem to use the term liberally, furthering misconceptions. This contributes to a skewed view and lost potential in the actual practice of data centric AI. In this project thesis, I will present the core of data centrism, as well as demonstrate its practical impact on the example of a computer vision problem in autonomous driving. The Formula Student ZHAW (FSZHAW) team competes among other categories in the "driverless" category in a car-racing competition with self-designed cars. They rely on a sophisticated network of modules with specific functions such as mapping, or constructing the racing lines. One of those modules is the visual perception module which has been trained with a custom dataset. This custom dataset, however, is of low quality and proved itself not very effective for the task. In this project thesis, I construct two new training datasets for the perception module. One is a simple dataset with the typical methods like data cleaning and standard image augmentations for object detection. The second one has more effort put into creating it: I use the segmentation masks to artificially insert new objects into the scene with realistic perspective scaling. I show and explain multiple solutions to improve the perception module's performance on the test dataset and discuss their effectiveness.**

***Index Terms*—data-centric processing, object detection, object recognition**

I. Introduction

For this project, I chose to work on the computer vision task of the Formula Student ZHAW (FSZHAW)'s "driverless"-team, because their dedication to the Formula Student competition is inspiring to me and I saw an opportunity for me to contribute to their cause. The recent emergence of data-centric AI (DCAI) has gained popularity as well, and in a project where the model cannot be too complicated because of inference time or memory restrictions, looking at the problem from a data-centric viewpoint seemed logical to me. With my findings I hope to help the team towards placing high on the leaderboard of the international Formula Student competition and also to contribute to the research about DCAI. Therefore I will give an overview of the current state of DCAI and demonstrate the importance and impact of a data-centric approach on the example of the perception module of the FSZHAW team[1]. In section II-A, the current state-of-the-art of DCAI will be presented. Following that, section II-B will explain the context of the practical part of this project, such as the context, the current state, the model, the datasets etc. Further, I will formulate the main points and sections of DCAI and later (see section III) connect the concepts together in a concrete demonstration of a DCAI-oriented approach to assess and improve the perception system.

II. Background

A. Data-Centric AI

In the recent years, the rise of DCAI signified a shift from a model-centric approach to AI to a data-driven one. Although the data aspect has not exactly been neglected in the field, it could be argued that it has received less attention than rightfully deserved. The key point being that data was only seen as a means to the end.

The term "data-centric AI" was originally coined by Andrew Ng [18] and the concept behind it has received greater attention ever since. DCAI shifted the focus from extracting value *out* of data to seeing the value *in* data itself – similar to how a farmer sees the value they can *get out* of a horse, while a private horse owner sees value *in* the horse itself, as [23] argues. It is a subtle but crucial shift in the mindset of how to approach data science as a whole. This fine distinction can lead to misunderstandings and myths [14], which can in turn contribute to unmet expectations when reading through the literature on DCAI. DCAI is a newer discipline and can still profit from traditional data and ML engineering, accroding to [20].

Andrew Ng defines DCAI as a three legged stool[2], consisting of (i) labeling and crowd-sourcing, (ii) data augmentation, and (iii) data in deployment. The idea behind (i) is that a clean, high-entropy dataset is more valuable to ML/DL than a noisy, incomplete, low-entropy dataset. This seems straight forward, but as [3] says, labeling data objectively and consistently is hard. The source of the data and labels is also important and could for example also by synthetic, which can open up an avenue to dealing with lots of data issues at the cost of having to deal with the transfer form synthetic data to real-world data. Specifically for topics of autonomous driving, simulations and tools for dataset creation

[1]Formula Student ZHAW (FSZHAW): https://fszhaw.ch/en/homepage/

[2]https://datacentricai.org/

like CARLA [4] can be of great value. Synthetic data can also solve issues like small data volume, but there are other solutions like [17, 21, 25]. Also part of (i) are data cleaning and analysis as explored with a data-centric view by [19] and benchmarking datasets instead of only models as introduced by [6, 15]. Data augmentations (ii) have been proven useful for decades, with more recent examples like [1, 21, 25] which take a more data-centric view on the topic. But one thing that often gets neglected is the lifecycle of ML/DL systems in deployment (iii). [22] sets the focus on technical debt in AI with a data-centric lense, because although it is known that models have to be adapted over time, a similar challenge shows itself with data and its lifecycle. Similarly, [13] advocates for reusability of data, which can help future work by leveraging the value in existing curated datasets. Both model and data in deployment are a big topic specifically in continual learning[3].

The authors of [8] formulate six guiding principles for DCAI: (1) Improving data fit in a systematic way, meaning how well the data represents the complexities of the real world context. (2) Systematic improvement of data consistency, because as mentioned by [3], having consistent labeling is important, although hard. (3) Mutual improvement of both model and data through iteration. There are situations where problems with the training data are not identifiable until models train with it, so having an iterative process where both data and model can get improved is recommended by the authors. (4) Human-centerednesss of "data work", meaning that the human cannot be removed from the process and has to be the focal point in the decisions to make like the shape of the data, how to improve it, etc. (5) AI as a sociotechnical system, meaning to uphold ethics while handling data. (6) Continuous and substantive interactions between AI and domain experts, ensuring that the data still adequately represents the real world domain. Furthermore, the authors of [8] – just like [18] – stress topics like (i) data preparation and augmentation, (ii) crowdsourcing, and (iii) data in deployment.

B. The Perception module of FSZHAW

The autonomous driving system of the race car of FSZHAW consists of an array of interconnected modules. One of which is the perception module, which is the focus of the practical aspect of this work. This is the module that solves the object-recognition task that is crucial to the overall task of autonomous driving. As it is the heart-piece, its performance directly impacts the overall performance of the race car in the "driverless" challenge[4]. The model has to detect four different classes of traffic cones in an image. There are a few restrictions on the complexity, such as no changes in elevation, because the tracks on which the competitions are being held are all flat. This aspect is crucial for section III-B, where we assume a flat surface to calculate an artificial horizon.

The model was chosen to be a YOLOv5 network [12, 9] for its simplicity and good performance without the need of tinkering with its source code. Before this project, the team used a checkpoint trained on the Synthetic Cones dataset (section II-C1). This dataset however breaks the rule of a flat surface as well as the cones being

[3]https://www.continualai.org/

[4]Formula Student, Driverless Cup: https://www.formulastudent.de/about/disciplines/

Fig. 1. The four classes of traffic cones: yellow, blue, orange, and big orange, taken from [7].

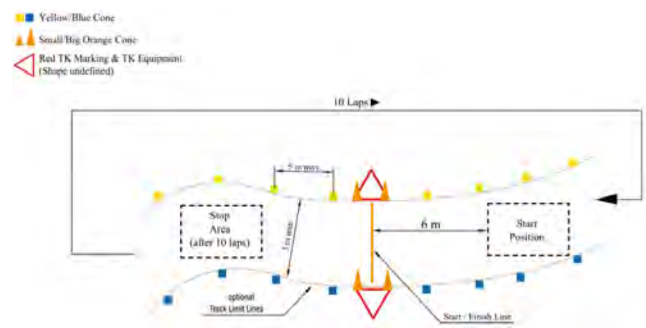

Fig. 2. An example of how the cones are arranged on a racing track, taken from [7].

C. Datasets

The datasets relevant to the formula student competition are object detection datasets with different types of traffic cones as objects to be detected. For the competition, four types of traffic cones (see figure 1) are designated to represent different roles: (i) The small blue cones mark the left border of the track. (ii) The small yellow cones respectively mark the right border of the track. (iii) The small orange cones are used to mark exit and entry lanes. (iv)The big orange cones are placed before and after start, finish, and timekeeping lines.

The small cones have dimensions of $228 \times 228 \times 325$ and the big orange cones have dimensions of $285 \times 285 \times 505$. During competitions, the white/black stripes have "FSG" written on them (with black/white colors respectively). The tracks look similarly as seen in figure 2. The competition handbook [7] goes into further detail.

1) Synthetic Cones Dataset: The Synthetic Cones dataset (see figure 3) consists of images of cones on racetracks and was constructed for FSZHAW by [16]. The cones come in various poses and were randomly set onto the background image of racetracks, which were sourced from the internet. The pictures of the racetracks have various angles, most of which are not consistent with the real-world environment as perceived by FSZHAW's race car's camera. Therefore, this dataset was

Fig. 3. Two images from the Synthetic Cones dataset [12, 16], taken from [12].

Fig. 4. Two images from the ZUR Testing Day Cones Dataset. The left image was taken with the mounted GoPro camera and the right one with a reflex camera [12, 16], taken from [12].

determined not to provide adequate training data for this project thesis. For this reason it was no longer considered.

2) ZUR Testing Day Cones Dataset: In an effort to create more significant real-world data, the FSZHAW team has collected a small dataset consisting of images taken from a GoPro camera, mounted on the top of the chair's backrest (see figure 4). This took place in 2021 in Dübendorf, when the team was taking the race car for a test drive.

Unfortunately, this dataset was not labeled to completion and the small volume of curated, labeled data is too small to be of much value. Furthermore, the GoPro has a native fish-eye effect, which distorts the imagery in a way not consistent to the new camera, which makes the little data available even less optimal to use. Because of this, it was decided not to work with this dataset.

Similarly to the initial effort, a new test set was to be constructed with the help of the team during the course of this project. Unfortunately it did not come to pass, as the members responsible were not available for setting up the car and environment in time. Multiple attempts to get a realistic test set have been made, but none of them were fruitful.

3) MIT Cones Dataset: The MIT cones dataset is a small dataset containing real-world imagery from the Formula Student racing team at MIT. It is taken from a camera mounted on their racing car, driving on a track similar to the setup in which the competitions are being held.

Unfortunately, this dataset was not considered a viable option because it did not have a segmentation map, which is important for section III-B. Therefore this dataset was also not considered for this project.

4) TraCon Dataset: The authors of the TraCon dataset [11] created their own dataset of traffic cones by annotating images taken from the H2020 HERON project [10].

Fig. 5. Two images from the MIT Cones Dataset [24, 12, 16], taken from [12].

Fig. 6. Two images from the TraCon Dataset [10].

This dataset does depict traffic cones different from the ones used in the Formula Student competition and is therefore not applicable. However, this work is mentioned explicitly because the authors do point out the lack in datasets for traffic cones.

5) FSOCO Dataset: The FSOCO dataset [26] is the result of a crowd-labeling effort, where teams had to label images with both bounding boxes and segmentation maps (see figure 7) in order to get access to the labeled dataset. In summer 2022 it was released publicly after they had no longer a need to label more data.

Because this dataset includes segmentation maps and has a high number of labeled real-world data, it was ultimately chosen to be used in this project.

D. Model

The model used by the team is YOLOv5 [9, 12] and has shown weakness in detecting cones that are farther away and therefore smaller. It does not suffer greatly from the class imbalance in the various datasets (like in section II-C5), but the imbalance does still show in evaluation metrics and visual outputs (refer to section IV). To combat this, multiple data-driven solutions have been conceptualized and tested in section III.

III. Methods

As explained in section II-D, the baseline model suffers from bad performance for cones which are small in the picture. From a data perspective there are a couple of possible solutions to improve this. By manipulating the representation of the data, the model can be enabled to focus more on its weaknesses. During the course of this project, multiple avenues have been explored, which are as follows.

A. Temporally-aware Frames

The images coming from the camera are temporally correlated, which means that the input and output of successive frames have things in common. Specifically the bounding boxes are similar but have been transformed in accordance

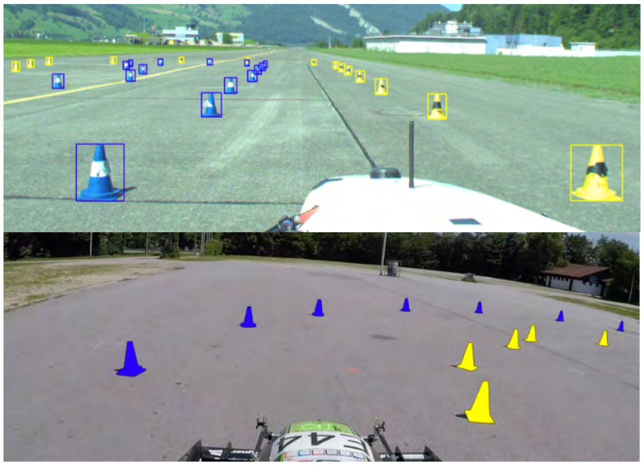

Fig. 7. Two images from the FSOCO Dataset [26]. The upper one shows the ground truth bounding boxes and the lower one the segmentations of those cones.

	bounding boxes		segmentations	
	#	%	#	%
blue cones	86,116	40.3	8,844	44.0
yellow cones	93,257	43.7	8,300	41.3
orange cones	25,991	12.2	2,955	14.7
big orange cones	8,217	3.8	0	0.0
total	213,581	100.0	20,099	100.0

TABLE I

STATISTICS ABOUT THE FSOCO DATASET.

to the car's movement. In order to encourage continuity in predictions and harness the common traits between frames of a video, it was hypothesised that combining frames would help the model to make better predictions. Since the FSOCO dataset (see section II-C5) does not contain videos but still images, a different dataset was used: In a project where object detection is used to detect patients in hospital beds with members of staff walking around, it was deemed feasible to test this hypothesis.

To construct the new frame, the three channels (RGB) were repurposed as to not have to change the structure of the model.

1) The red channel was storing the grayscale image of the original frame.
2) The green channel was repurposed to show the magnitude of pixel change from the last to the latest frame on the grayscaled frames. If it is the first frame in the sequence of a video, then the magnitude would be zero. In order to not bias the model too much towards requiring this channel, only half of all the images were assuming a last frame and were therefore non-zero.
3) The blue channel was reserved for storing the shapes of the bounding boxes. The intensity of the pixels did not change with the class of the bounding box. For the same reason as for the "green" channel, this channel would be non-zero half of the times – coinciding with the "green" channel.

Because the FSOCO dataset does not contain videos and those images that were taken in succession were too spaced out, this hypothesis was only tested on the aforementioned dataset of hospital beds. We trained a baseline model (also YOLOv5 [9]) with early stopping to compare the performances.

B. Cone Duplication using Perspective Transformation

As explained in section II-B, the data at hand (specifically FSOCO) consists of perspective images taken on flat ground. This means there is a visible correlation between the size of the bounding boxes and the y-position in the picture. [5] When looking at the bounding boxes, they get smaller the closer they get to the horizon and detections above the horizon are impossible. Taking this into account, by applying linear regression on the size of the bounding boxes to their y position in the image, one can approximate the position of the artificial horizon in the form of a linear equation (see figure 8).

$$s_{\text{bbox}} = y \cdot m + c$$

$$y_{\text{horiz}} = -\frac{c}{m}$$

In order to normalize the data, the large orange cones were scaled down to the same size as the small cones. To make the estimation more stable, the size of a bounding box was calculated with $s_{\text{bbox}} = w_{\text{bbox}} \cdot h_{\text{bbox}}$. Furthermore, to increase accuracy, we considered both the bounding boxes as well as the segmentations per image for this calculation.

With the artificial horizon y_{horiz} and the parameter c, one can calculate how big the bounding boxes are supposed to be at the position $(x, y) \quad y \in [y_{\text{horiz}}, y_{\max}]$, where $y_{\max}$ is the height of the image [5].

For the linear regression, we ignore those annotations which are touching the borders of the image, as they are likely cropped cones and therefore would skew the calculations unfavourably. We also dropped those pictures which had less than six annotations (both bounding boxes and segmentations together, as they are probably overlapping). Furthermore, we calculate the mean squared error of the annotations based on the fitted linear equation and selected 25 to be a good threshold, above which we do not process the image further, as the variance is deemed too high to achieve confident and precise results.

For duplicating cones, we make a list of five times the amount of existing annotations or up to 100, while oversampling the small orange cones, to combat the class imbalance[6]. Before we select a new position for a cone, we create a mask, which is zero if there are no bounding boxes and the y position is in $[y_{\text{horiz}}, y_{\max}]$. Then we randomly select a new position for the next segmentation in the list anywhere from $x \in [0, w_{\text{img}}], y \in [y_{\text{horiz}}, y_{\text{seg}_{\max}}]$, where we set $y_{\text{seg}_{\max}} = \min(h_{\text{img}}, \max(y \cdot 1.5, y + 20))$ to not scale up the segmentation too much, which would impact the quality of the cone. Then, we randomly select a new position, transformation the bounding box to the new size, check against the mask if there is overlap and accept the new position if

[5]Note that the big orange cone is bigger than the other three classes of traffic cones in our dataset.

[6]Note that it is shown in table I that there are no segmentations for the big orange cones.

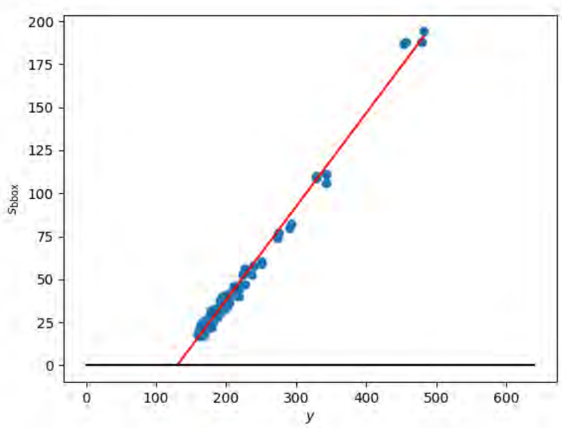

Fig. 8. A sample plot of s_{bbox} plotted against the y position of the bounding boxes, with the red line showing the fitted line. The intersection of the red line with the black line shows where the artificial horizon was determined to be.

Fig. 9. A sample image where the cone duplication algorithm with perspective transformation was applied to.

there is no overlap and it is not crossing outside the image. Otherwise we try up to 19 more times. We then copy the RGB values of the cone selected by the segmentation as a mask over to the new position and save the new bounding box. The duplicated cone is flipped horizontally at random to counteract overfitting.

The figure 9 demonstrates the effect of the algorithm: There are new cones, many of which are far away, increasing the hard to detect cones. This was hypothesized to increase the performance of the model by putting more stress on small cones and increasing the overall amount of annotations.

C. *Custom Domain Loss*

Inspired by the perspective transformation from section III-B and [21], it was hypothesized that adding a domain-specific loss would improve continuity and prevent the detection in places which do not agree with the perspective relationship between the bounding boxes and the artificial horizon. The idea of this is to use the linear regression to fit a line through the proposals of the network and then calculate the mean squared error to pose as an additional loss for the model to optimize. For this, we changed scaled the proposals for the big orange cones by a factor so that they have potentially the same size as the small cones. Then, we perform linear regression on the cones and return the mean squared error against the fitted line as the loss.

IV. Results

We trained a baseline model on the FSOCO (section II-C5) dataset for 300 epochs and achieved a mAP@0.5 score of 74%. When looking at the confusion matrix (figure 10), it is evident that the strong class imbalance (refer to table I) does not impact the performance greatly.

The model still detects big orange cones the worst, but there is only little overlap between the big orange cone and the small orange cone. Nonetheless, the small amount of large orange cones does make it the least detected class. Also important to note is that it does still struggle to differentiate between cones and background.

A. *Temporally-aware Frames*

The baseline model takes almost twelve times as long to achieve an almost identical performance as the temporally-aware-frames (refer to section III-A) model (see table III). Unfortunately the gain in performance is not as big as hypothesized. However, the model does learn faster than the baseline model.

This shows that although the performance does not increase significantly, the training can be made faster using this method. However, this does involve reducing an RGB image to grayscale, which is not a good idea for the traffic cones datasets (sections II-C1 to II-C3 and II-C5, as they rely on the colors of the traffic cones for classification. Furthermore, those datasets do not contain video material like it was used on the hospital bed dataset. This means that actual video data is needed for this transformation to work.

So although promising, a better transformation should be found to preserve the colors and use a different dataset.

B. *Cone Duplication using Perspective Transformation*

We trained a model on the data according to section III-B and tested it along with the baseline model on the untouched test set (see table II). The result seems to suggest that the juice was not worth the squeeze; the improvement in performance is negligible.

There is only minimal differences between the confusion matrix from this new model (figure 11) and the one from the baseline (figure 10).

It seems that the proposed method in section III-B does not improve the model's performance significantly. It can be ruled out that the quality of the data dropped, because the transformations have been applied very carefully with great deliberation. The resulting dataset does look realistic and does indeed mirror what the idea behind it is. But the model evidently does not profit from this augmentation. This seems to suggest that the model itself needs adjustments to see an improvement in performance.

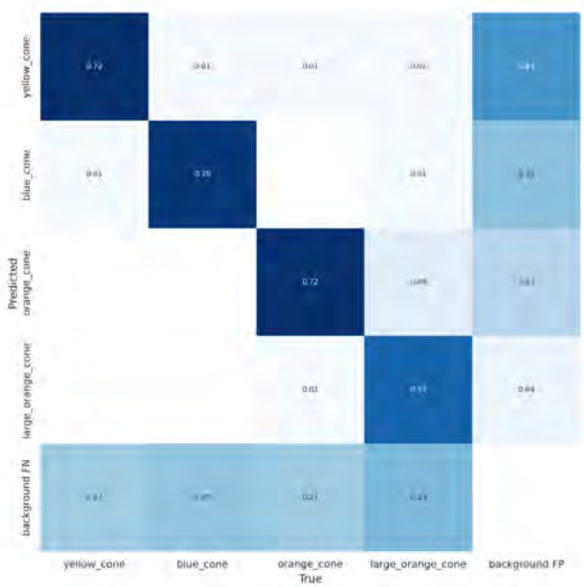

Fig. 10. The confusion matrix of the baseline, tested on the normal test set.

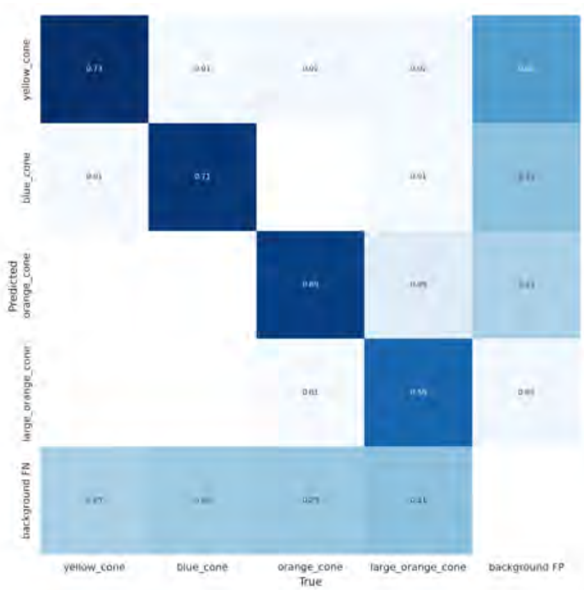

Fig. 11. The confusion matrix of the model trained with the cone duplication dataset (ConeDup, see sections III-B and IV-B), tested on the normal test set.

C. Custom Domain Loss

The custom domain loss (refer to section III-C) did not behave like loss functions usually behave: Looking at figure 13, it is clear that the model has trouble optimizing this loss. As seen in table II, the performance also did neither increase not decrease. And although the desired effect did not come to pass, the model did not completely fail.

It appears that the loss function is hard to optimize because it takes into account all predictions – even those with low confidence that would be discarded by the non-max-suppression (NMS) algorithm.

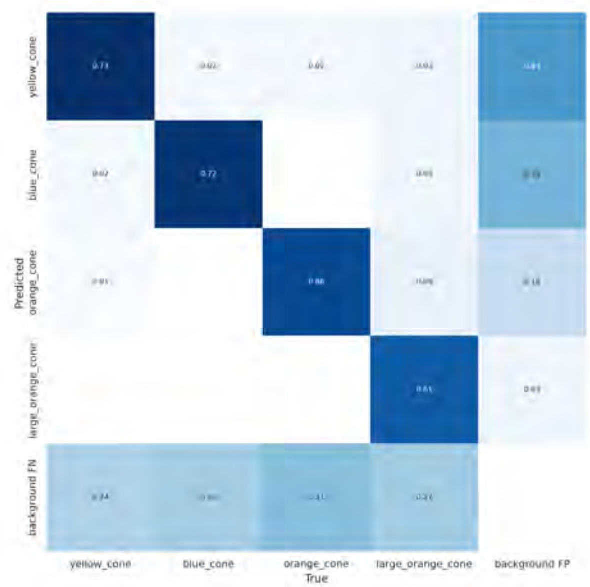

Fig. 12. The confusion matrix of the custom domain loss method (see sections III-C and IV-C), tested on the normal test set.

		FSOCO mAP@0.5	
YOLOv5 model	epochs	normal	ConeDup
baseline	300	73.9%	71.3%
ConeDup	300	**74.1%**	**73.3%**
DomLoss	300	73.9%	73.0%

TABLE II

COMPARISON BETWEEN THE BASELINE MODEL AND THE MODEL TRAINED USING THE PROPOSED METHODS IN SECTIONS III-B AND III-C ON BOTH FSOCO TEST SETS: THE ORIGINAL ONE AND THE ONE GENERATED USING CONEDUP (SEE SECTION III-B).

YOLOv5 model	epochs	mAP@0.5
baseline	119	87.2%
TaF	**10**	**88.9%**

TABLE III

COMPARISON BETWEEN THE BASELINE MODEL AND THE MODEL TRAINED USING THE TEMPORALLY-AWARE FRAMES HOSPITAL BED DATASET (SEE SECTIONS III-A AND IV-A).

V. CONCLUSION AND DISCUSSION

The proposed hypotheses only brought about minimal improvement to the baseline model. It seems that the model might profit more from model-centric adjustments than from data-driven ones due to it not being able to handle small objects in general. There is also a lack of video data available for this traffic cone detection use-case, with which it would be possible to leverage the strong point of fast convergence (as demonstrated and discussed in sections III-A and IV-A).

It stands to reason that investing in labeled video data would bring about benefits to the learning speed of the model. This way other, more costly optimizations can be incorporated without having to train the models for many epochs. We also identify a need to investigate into the effectiveness of the temporally-aware-frames method (see section IV-A) in frames with movement. It should be emphasized that this method should be improved in such a way that the color information

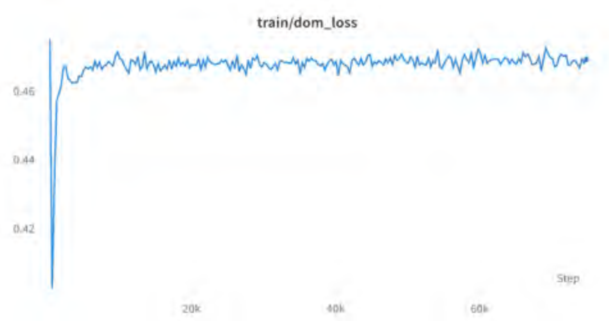

Fig. 13. The custom domain loss during the training.

is retained, which is crucial for detecting the four classes of traffic cones.

The results from the cone duplication with perspective transformation (see section IV-B) show that the model is not well equipped enough to detect small cones. To fix this, there are a couple of possible solutions to try out: (i) Applying a visual transformation to the image that enlarges small objects using a perspective transformation, similar to a fish-eye. This would make it easier for models to detect small shapes. (ii) Adjust the model so it can handle small objects better: (ii.a) The authors of [2] demonstrate a model-centric approach to achieve up to 6.9% improvement in mAP. (ii.b) Change the convolutional depth of YOLOv5.

As already pointed out in section IV-C, the custom domain loss probably failed because it took into account every prediction without applying NMS to them. Although costly, this could improve the usefulness of the domain loss greatly. What the domain loss is also lacking is punishing proposals above a calculated artificial horizon. This however assumes that an appropriate horizon can be found (i.e. $m > 0$). If no horizon can be found, then the loss should still be bigger than if it does find a horizon and penalized all proposals above that y value. Otherwise the loss function would increase once an artificial horizon is found. There are lots of other considerations to be made in the design of the loss function, but ultimately it should prove useful to models.

A different hypothesis which could be explored would be leveraging test subsets for explainability. The FSOCO dataset is very diverse: (i) There are differing levels of illumination and times of day represented in the data. (ii) Different ground types are available. (iii) Images taken from cameras from varying pitch and roll angles exist. (iv) Cameras have different levels of elevation. (v) The traffic cones may have the “FSG” logo written on their stripes (as mentioned in section II-C). (vi) The traffic cones may be positioned in a way that makes them show up at an angle or extra wide. All those attributes could be made use of by enriching the labels with that extra information and then selected for test sets representing a specific attribute that might have an influence on the model’s performance. Using such an array of test sets would allow for a more detailed error analysis and higher explainability of the model.

There are a lot of potential improvements to be made to the model and it is suggested to also consider model-centric approaches, because data-centrism alone also has its limits.

References

[1] Anima Anandkumar. *Data Augmentation*. 2022. URL: https://datacentricai.org/data-augmentation/.

[2] Aduen Benjumea et al. “YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles”. In: *CoRR* (2021). DOI: 10.48550/arXiv.2112.11798.

[3] Michael Bernstein. *Labeling and Crowdsourcing*. 2022. URL: https://datacentricai.org/labeling-and-crowdsourcing/.

[4] Alexey Dosovitskiy et al. “CARLA: An Open Urban Driving Simulator”. In: *Proceedings of the 1st Annual Conference on Robot Learning*. 2017, pp. 1–16.

[5] Raphael Emberger and Daniel Kluvanec. *Verbal Discussion*. Jan. 2023.

[6] Sabri Eyuboglu et al. “Dcbench: A Benchmark for Data-Centric AI Systems”. In: *Proceedings of the Sixth Workshop on Data Management for End-To-End Machine Learning*. DEEM ’22. Philadelphia, Pennsylvania: Association for Computing Machinery, 2022. ISBN: 9781450393751. DOI: 10.1145/3533028.3533310.

[7] FormulaStudentGermany. “FSG Competition Handbook 2023”. In: (Jan. 2022). URL: https://www.formulastudent.de/fileadmin/user_upload/all/2023/important_docs/FSG23_Competition_Handbook_v1.0.pdf.

[8] Mohammad Hossein Jarrahi, Ali Memariani, and Shion Guha. *The Principles of Data-Centric AI (DCAI)*. 2022. DOI: 10.48550/ARXIV.2211.14611.

[9] Glenn Jocher et al. *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation*. 2022. DOI: 10.5281/ZENODO.3908559.

[10] Iason Katsamenis et al. “Robotic Maintenance of Road Infrastructures: The HERON Project”. In: (May 2022). DOI: 10.48550/ARXIV.2205.04164.

[11] Iason Katsamenis et al. “TraCon: A novel dataset for real-time traffic cones detection using deep learning”. In: (May 2022). DOI: 10.48550/ARXIV.2205.11830.

[12] Florian Kittelmann. “Integration and Improvement of the Perception System”. In: (June 2022).

[13] Bernard Koch et al. “Reduced, Reused and Recycled: The Life of a Dataset in Machine Learning Research”. In: *CoRR* (2021). DOI: 10.48550/arXiv.2112.01716.

[14] Antonia Marcu and Adam Prügel-Bennett. “On Data-centric Myths”. In: *CoRR* (2021). arXiv: 2111.11514.

[15] Mark Mazumder et al. “DataPerf: Benchmarks for Data-Centric AI Development”. In: *arXiv* (July 2022). DOI: 10.48550/arXiv.2207.10062.

[16] Thomas Signer Mirjam Abegglen Céline Keller. “Perception for Driverless Electrical Racing Car”. In: (July 2021).

[17] Mohammad Motamedi, Nikolay Sakharnykh, and Tim Kaldewey. “A Data-Centric Approach for Training Deep Neural Networks with Less Data”. In: *CoRR* (2021). DOI: 10.48550/arXiv.2110.03613.

[18] Andrew Ng. *Data-Centric AI*. June 2021. URL: https://landing.ai/data-centric-ai/.

[19] Hima Patel et al. “Advances in Exploratory Data Analysis, Visualisation and Quality for Data Centric AI Systems”. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD ’22. Washington DC, USA: Association for Computing Machinery, 2022, pp. 4814–4815. ISBN: 9781450393850. DOI: 10.1145/3534678.3542604.

[20] Neoklis Polyzotis and Matei Zaharia. “What can Data-Centric AI Learn from Data and ML Engineering?” In: *CoRR* (2021). arXiv: 2112.06439.

[21] Pascal Sager et al. “Unsupervised Domain Adaptation for Vertebrae Detection and Identification in 3D CT Volumes Using a Domain Sanity Loss”. In: *Journal of Imaging* 8 (2022). URL: https://www.semanticscholar.org/paper/Unsupervised-Domain-Adaptation-for-Vertebrae-and-in-Sager-Salzmann/87e3f7b375765b42546a3c0f2e56da1ff9d2718e.

[22] David Sculley. *Data in Deployment*. 2022. URL: https://datacentricai.org/data-in-deployment/.

[23] Thilo Stadelmann, Tino Klamt, and Philipp H. Merkt. “Data Centrism and the Core of Data Science as a Scientific Discipline”. In: 2022. URL: https://www.semanticscholar.org/paper/Data-Centrism-and-the-Core-of-Data-Science-as-a-Stadelmann-Klamt/00a54ba31ec58ef2d6f636a023d072b99fe0afca.

[24] Kieran Strobel et al. “Accurate, Low-Latency Visual Perception for Autonomous Racing:Challenges, Mechanisms, and Practical Solutions”. In: (July 2020). DOI: 10.48550/ARXIV.2007.13971.

[25] Lukas Tuggener et al. “Real World Music Object Recognition”. In: 2023.

[26] Niclas Vödisch, David Dodel, and Michael Schötz. “FSOCO: The Formula Student Objects in Context Dataset”. In: *SAE International Journal of Connected and Automated Vehicles 5.12-05-01-0003 (2022)* 5.1 (Dec. 13, 2020). DOI: 10.4271/12-05-01-0003.