Ensemble-based Semi-supervised Learning to Improve Noisy Soiling Annotations in Autonomous Driving

Michal Uřičář\(^1\), Ganesh Sistu\(^2\), Lucie Yahiaoui\(^2\) and Senthil Yogamani\(^2\)

\(^1\)Independent Researcher, Czech Republic  \(^2\)Valeo Vision Systems, Ireland

Abstract—Manual annotation of soiling on surround view cameras is a very challenging and expensive task. The unclear boundary for various soiling categories like water drops or mud particles usually results in a large variance in the annotation quality. As a result, the models trained on such poorly annotated data are far from being optimal. In this paper, we focus on handling such noisy annotations via pseudo-label driven ensemble model which allow us to quickly spot problematic annotations and in most cases also sufficiently fixing them. We train a soiling segmentation model on both noisy and refined labels and demonstrate significant improvements using the refined annotations. It also illustrates that it is possible to effectively refine lower cost coarse annotations.

I. INTRODUCTION

Wide-angle cameras are important sensors for certain tasks in autonomous driving, such as automated parking. Figure 1 shows how these cameras are organized in order to get a 360° view around the vehicle. Since these cameras are usually placed externally and exposed to the environmental conditions, one needs to account for the reliability of these sensor input in order to achieve a full autonomy. It is not uncommon that during harsh weather (such as heavy rain, snow, etc.) or when driving in off-road conditions, the surround view cameras get dirty (Figure 1 bottom). In general, there is limited work on fisheye camera perception for tasks such as object detection [1], depth estimation [2], SLAM [3] and multi-task learning [4]. Recently, the problem of dealing with camera soiling in autonomous driving has gained increased attention [5], [6], [7], [8].

Automated driving systems perform well in normal weather conditions and one of its main challenges is to work effectively in adverse weather conditions. Particularly, visual perception algorithms degrade severely in adverse weather conditions. Adverse conditions can cause soiling of camera lens which causes even more severe degradation in performance. Relatively, there is little work addressing to handle this scenario. Porav et al. [9] implemented a de-raining algorithm using PatchGAN-based architecture and a clever data collection mechanism to collect rainy and non-rainy images of the same scene. Sakaridis et al. [10] uses synthetically generated foggy scenes to evaluate the impact of degradation of object detection and propose to defog in order to improve performance. Ki et al [11] propose to use conditional GAN to perform dehazing.

While some work pointed out the problems with manual annotation process and offered artificial generation of training data with pixel-precise annotation [12], [13], there is not much work dedicated to solving the issue of fixing existing imprecise manual annotations or some method to quickly spot the images with annotation errors in large scale datasets. The problem with imprecise annotations is two-fold. Firstly, the classification models learned with the full supervision inherently reflect the quality of the annotations provided in the training. Secondly, the evaluation of the trained models is done on imprecise annotations, so the actual performance analysis is clouded and it can easily happen that a best performing model on real data is discarded due to poorer performance on the imprecisely annotated test data.

The soiling classification and localization task is problematic due to the unclear boundary of individual soiling categories and vague definition of soiling classes, such as when the soiling is already considered to be in the transparent or opaque category. Moreover, due to the annotation acquisition costs, coarse polygonal contours are commonly used instead of precise segmentation masks. Figure 2 illustrates the difficulties of doing manual annotation and shows how ambiguous and subjective it can be.

Ideally, the solution would be to design a model which accounts for the known problems of annotation precision and operates in a semi-supervised manner. However, despite the
enormous progress in semi-supervised learning [14], this is still considered as the machine learning holy grail and far from being applied reliably in the real world scenarios, such as the autonomous driving. The other way of approaching this problem is to design a semi-automatic way of improving the annotation quality and reduce its inconsistencies.

We selected the latter approach and propose to deal with the problem via the concept of pseudo-label [15] ensemble [16]. Pseudo-labeling is an interesting concept of handling imprecise annotations (e.g. prediction of some classifier) for semi-supervised learning. The pseudo-label for a data sample is usually chosen based on the maximum predicted probability and then considered to be the real label in the standard training loop. This is equivalent to entropy regularization [15]. Because of the nature of manual annotations which we use, we consider all labels to be pseudo-labels and train the ensemble classifier in two consecutive stages. First stage selects the pseudo-label randomly, whereas the second stage picks it using the nearest neighbor principle (to the output of the first stage classification). The output from the second stage classifier provides refined annotations, which can be readily used for the standard supervised learning.

II. PROPOSED METHOD

The soiling task was first formally defined in [5] based on WoodScape dataset [17]. WoodScape provides segmentation annotations $T = \{x_n, y_n\}_{n=1}^N$. Where $x \in \mathcal{I}^{H \times W}$ denotes the image of a fixed resolution and $y \in \mathcal{Y}$ denotes the per-pixel annotation of a set of four classes, namely $\mathcal{C} \in \{opaque, semi-transparent, transparent, clean\}$. The class opaque means that in that particular region, it is impossible to see through, semi-transparent and transparent are blurry regions where background colors are visible with diminished texture. The difference between the semi-transparent and transparent classes is the ability to recognize the object through the blur. The last class is clean which indicates no soiling present. We follow the same labelling categories and definitions in our work.

Inspired by [16], we propose a two stage approach for learning an ensemble classifier using the pseudo-label (PL) concept introduced by [15]. The ensemble classifier is a neural network, consisting of two separate encoders and one decoder. First encoder processes the input image, while the second encoder processes all channel-wise concatenated pseudo-labels to propagate the information from them.

Algorithm 1 Two-stage ensemble learning

Require: $x_i, PL_i = \{f_1(x_i), f_2(x_i), \ldots, f_9(x_i)\}$
$S(PL_i), i = 1, \ldots, n, m$

{1st stage training}
1: while stopping criterion not met do
2: {Random fit}
3: for $i = 0$ to $m$ do
4: Sample $p \sim \mathcal{U}(1, n)$ \{to select sample\}
5: Sample $q \sim \mathcal{U}(1, 9)$ \{to select PL\}
6: Add sample $\{x_p, S(PL_p), f_q(x_p)\}$ to mini-batch
7: end for
8: Update 1st stage network $\mathcal{H}_1$ weights by back-propagating loss:\n\[
\mathcal{L}(x, c) = -\log \left( \frac{\exp(x[c])}{\sum_j \exp(x[j])} \right)
\]
9: Empty mini-batch
10: end while

{2nd stage training}
11: while stopping criterion not met do
12: {Nearest neighbor fit}
13: for $i = 0$ to $m$ do
14: Sample $p \sim \mathcal{U}(1, n)$ \{to select sample\}
15: $\hat{y} = \mathcal{H}_1(x_p, S(PL_p))$
16: $\hat{q} = \arg\min_{q_1, \ldots, q_9} \mathcal{L}(\hat{y}, f_q(x_p))$
17: Add sample $\{x_p, S(PL_p), f_q(x_p)\}$ to mini-batch
18: end for
19: Update 2nd stage network $\mathcal{H}_2$ weights by back-propagating loss $\mathcal{L}(x, c)$
20: end while

We use 9 PLs in total. First one is the original manual annotation, which is quite rough (polygonal annotation for soiling parts, cf. Figure 2) and error-prone (cf. Figure 5). Next 3 PLs are formed by classification via DeepLabV3+ [18] network with ResNet50 [19] backbone with 3 different approaches for calculating the inference, namely: multi-scale prediction, sliding window prediction and holistic prediction. Another 3 PLs are built analogically by classification output of DeepLabV3+ but this time using the inception [20] backbone. The last 2 PLs consist of the output of an in-house simple soiling segmentation network called MaskSegNet, where the first one is using a model trained on the quarter of the original image resolution ($\mathcal{I}^{H/4 \times W/4}$) and the second is using a model trained on the half of the original image resolution ($\mathcal{I}^{H/2 \times W/2}$). All networks were trained using the non-public part of the soiling part of the WoodScape dataset [17]. In Figure 5, we show several examples of the 4 out of the 9 PLs, so the reader can compare their quality.

We describe the full learning scheme in Algorithm 1. It consists of two consecutive stages. In the first stage, one of the 9 pseudo-labels is randomly selected as the correct label and the ensemble network is trained to minimize the cross-entropy loss. The second stage uses the same architecture as the first stage, but this time the nearest neighbor pseudo-
Fig. 3: Proposed architecture. RGB image and channel-wise concatenated pseudo-labels (PL) are used as inputs. They are passed through two independent encoders. The output of both encoders is concatenated and passed to the decoder which produces refined soiling segmentation annotation.

Fig. 4: Three sets of samples are presented. Left: Original image, Middle: Manual annotation and Right: Ensemble refined annotation. Class color coding: Green — transparent, blue — semi-transparent, red — opaque, and original image color — clean.

Since the variance of the images and the soiling patterns is not very high, we use the common trick applicable to semantic segmentation networks by selecting a random crop of the image and corresponding crop of its annotation. This also allows us to use standard data augmentation, such as random rotation or horizontal/vertical flipping.

III. EXPERIMENTAL EVALUATION

In this section, we summarize the conducted experimental evaluation and obtained results.

A. Experimental Setup

We made use of an internal dataset consisting of 20k samples. They were split into train/val/test datasets in the ratio of 6 : 2 : 2 based on stratified sampling techniques discussed in [21]. To evaluate the refined (i.e. ensemble generated) annotations, two PSPnet [22] models with ResNet-50 [19] backbone were trained. One model was trained...
Fig. 5: Three sets of qualitative results. Top row: Original image, manual annotation and ensemble refined annotation. Next row: Individual classifier predictions (ensemble input) from MaskSegNet, DeepLabV3+ with the ResNet and DeepLabV3+ with the Xception backbone. Only 4 out of 9 pseudo-label outputs are shown here due to limited space. Class color coding: Green — transparent, blue — semi-transparent, red — opaque, and original image color — clean.
with the manual annotations and the second one with the refined annotations. Both models are trained under the same configuration settings like optimizer, number of epochs, etc.

The networks are trained using the train/val sets and evaluated using the test set. Key Performance Indicators (KPIs) are computed for both the networks using three different annotation variants for the same test set. The first annotation variant composes of the original manual annotations, the second one composes of the refined, ensemble generated, annotations, and the third annotation set was generated by computing the intersection between the manual and the ensemble labels where only pixels annotated with the same class in both annotations were considered (Figure 6). This intersection area pixels indicate the model performance on regions where both the manual annotations and ensemble annotations are aligned. Two metrics are used for the comparison: i) IoU (Intersection over Union) between the ground truth (GT) and the prediction, ii) accuracy of the prediction.

B. Quantitative results

Table I summarizes the obtained results in terms of IoU and prediction accuracy. The discussion of the results follows.

a) Model trained on manual annotation: Mean IoU score for this model on manual annotations is 65.26%. While the model performs well on clean and opaque classes it has relatively low score for transparent and semi-transparent classes. This is understandable as the transparent and semi-transparent classes are hard to annotate due to fuzziness in the definition, resulting in noisy annotations.

The mean IoU obtained with the ensemble annotation test set (74.89%) is considerably higher than the one obtained when testing on the manual annotations (65.26%). For both annotations the model performance remains quite similar and improvements are observed in noisy categories of classes. This shows that there is a label noise in transparent and semi-transparent refined annotations compared to manual ones. This conclusion can hold further better,

- If both the models perform better on ensemble annotations, showing that ensemble generated annotations are less noisy
- If the model trained on ensemble annotations outperforms the model trained on manual annotations while evaluated on manual annotations and ensemble annotations, indicating relatively better generalization.

From the Table I the overall IoU of the manual annotations trained model is better on the ensemble test set, it is important to highlight the increase in performance per classes: transparent (+16.21%), semi-transparent (+15.26%) and opaque (+6%). The same conclusion can be drawn regarding the mean accuracy computed (+9.1%).

b) Model trained on ensemble annotation: The model trained on ensemble annotations improved by +5.43% in mean IoU score, compared to the manual annotations trained model when testing on ensemble test set. This model performs slightly better than the model trained on manual annotations overall (mean IoU score). This shows that the model trained on ensemble annotations can generalize well on manual annotations as well. More importantly, this model outperforms the manual annotation model by +2.15% and +1.72% for transparent and semi-transparent classes respectively.

c) Comparison of results computed on Intersection set: So far we have shown results on inter and intra annotation domains for both models. But it is also important to compare the performance of the models on a common domain. To compare both the networks on a common domain, we evaluated both models on an “intersection” test set (see Figure 6). Intersection test set labels are generated as the pixels that shares the same class in both manual and ensemble generated annotations, respectively. The obtained results show that the network trained on the ensemble annotation performs better than the other one. While the IoU for the class “clean” stays the same, the IoU for the three other classes increased significantly for the model trained on the ensemble generated annotations (transparent: +4.55%, semi-transparent: +5.39% and opaque: 2.09%). Along with the IoU KPI, the overall accuracy increased as well (+3.73%). Hence, our conclusion that the ensemble generated annotations contain less label noise compared to the manual annotations is proven via the evaluation of the model performance on segmentation tasks.

C. Manual visual inspection

Since the goal of the proposed ensemble classifier is fixing the annotations, we cannot rely solely on the numbers presented in the previous paragraph. Therefore, we conducted an experiment with a manual inspection of the generated annotations. Manual inspection was done as an AB test by a team of 3 image annotation experts with the knowledge of the soiling detection task. Tests are carried using a Python based software tool, where the original image along with its manual annotation and its generated ensemble annotation were showed (see Figure 4). Each reviewer visually evaluated 280 images and had to select which annotation is better in his opinion. Table II shows the score obtained after this manual inspection. The percentage of the better manual annotation is summarized in the first column, the better ensemble annotation is shown in the second column, and of similar annotations in the third column. It can be observed that in 42.70% of the cases, ensemble annotations where considered better than the manual annotation (14.59%). It is important to notice, that in the 42.7% of the annotation that where found similar, clean images have been reviewed too.
TABLE I: Qualitative results of the two models trained with manual or ensemble annotations. Each model has been tested on 3 different test sets (Manual, Ensemble or Intersection test sets). Accuracies obtained on Ensemble test set are higher than when testing on manual test set for both networks. The Intersection test sets allows us to compare the performance of the 2 networks on a common domain and shows that the network trained on Ensemble annotations performs better.

| Model                      | Trained on Manual Annotations | Trained on Ensemble Annotations |
|----------------------------|-----------------------------|-------------------------------|
|                            | Manual | Ensemble | Intersection | Manual | Ensemble | Intersection |
| Class                      | IoU   | Acc     | IoU   | Acc     | IoU | Acc     | IoU   | Acc     |
| clean                      | 94.36 | 96.72   | 95.41 | 96.37   | 86.15 | 86.44   | 94.11 | 96.82   |
| transparent                | 35.77 | 47.56   | 51.98 | 68.57   | 38.21 | 73.89   | 37.92 | 51.17   |
| semi_transparent           | 49.52 | 68.01   | 64.78 | 82.23   | 50.7 | 87.41   | 51.26 | 60.2    |
| opaque                     | 81.39 | 95.74   | 87.39 | 95.22   | 81.97 | 98.44   | 81.15 | 91.87   |
| Mean                       | 66.20 | 76.5    | 74.89 | 85.6    | 64.26 | 86.55   | 66.11 | 77.26   |

The number of images manually inspected covers 20% of the hold out test set.

IV. CONCLUSIONS

Camera lens soiling annotations are difficult to annotate precisely due to less defined boundaries and semantic ambiguities. In this work, we propose to refine the manual coarse polygonal annotations to fine precise annotations using an ensemble pseudo-label model. We demonstrate significantly improved annotations using manual visual verification and quantitative results based on a soiling segmentation model. It also illustrates that coarse manual annotations are sufficient to significantly lowering the annotation costs. Precise annotations are crucial for robust models and we demonstrate a semi-supervised approach to improve the annotations. In future work, we plan to jointly train the ensemble annotation refiner and the learning model in an active learning loop.

REFERENCES

[1] H. Rashed, E. Mohamed, G. Sistu, V. Ravi Kumar, C. Eising, A. El-Sallab, and S. Yogamani, “Generalized Object Detection on Fish eye Cameras for Autonomous Driving: Dataset, Representations and Baseline,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 2272–2280.

[2] V. R. Kumar, S. Milz, C. Witt, M. Simon, K. Amende, J. Petzold, S. Yogamani, and T. Pech, “Monocular fisheye camera depth estimation using sparse lidar supervision,” in 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2018.

[3] N. Tripathi and S. Yogamani, “Trained trajectory based automated parking system using visual slam,” arXiv preprint arXiv:2001.02161, 2020.

[4] V. R. Kumar, S. Yogamani, H. Rashed, G. Sistu, C. Witt, I. Leang, S. Milz, and P. Mader, “Omnidet: Surround view cameras based multi-task visual perception network for autonomous driving,” IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 2830–2837, 2021.

[5] M. Uričař, P. Krčžek, G. Sistu, and S. Yogamani, “Soilingnet: Soiling detection on automotive surround-view cameras,” in 2019 22nd International Conference on Intelligent Transportation Systems (ITSC), 2019.

[6] M. Uričař, P. Krčžek, D. Hurých, I. Sobh, S. Yogamani, and P. Denny, “Yes, we GAN: applying adversarial techniques for autonomous driving,” in Electronic Imaging, 2019.

[7] M. Uricar, J. Ulicny, G. Sistu, H. Rashed, P. Krizek, D. Hurych, A. Vobecky, and S. Yogamani, “Desoiling dataset: Restoring soiled areas on automotive fisheye cameras,” in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2019.

[8] P. Maddu, W. Doherty, G. Sistu, I. Leang, M. Uricar, S. Chenupati, H. Rashed, J. Horgan, C. Hughes, and S. Yogamani, “Fishyemultinet: Real-time multi-task learning architecture for surround-view automated parking system,” in Proceedings of the Irish Machine Vision and Image Processing Conference, 2019.

[9] H. Porav, T. Bruhs, and P. Newman, “I can see clearly now: Image restoration via de-raining,” in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 7087–7093.

[10] C. Sakaridis, D. Dai, and L. Van Gool, “Semantic foggy scene understanding with synthetic data,” in International Journal of Computer Vision. Springer, 2018.

[11] S. Ki, H. Sim, J.-S. Choi, S. Kim, and M. Kim, “Fully end-to-end learning based conditional boundary equilibrium gain with receptive field sizes enlarged for single ultra-high resolution image dehazing,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018.

[12] M. Uričař, G. Sistu, H. Rashed, A. Vobecký, V. R. Kumar, P. Krčžek, F. Burger, and S. Yogamani, “Let’s Get Dirty: GAN Based Data Augmentation for Camera Lens Soiling Detection in Autonomous Driving,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), January 2021, pp. 766–775.

[13] F. Pizzati, P. Cerri, and R. de Charette, “Model-based occlusion disentanglement for image-to-image translation,” in Computer Vision – ECCV 2020, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 44–66.

[14] J. E. Van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” Machine Learning, vol. 109, no. 2, pp. 373–440, 2020.

[15] D.-H. Lee et al., “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in Workshop on challenges in representation learning, ICMIL, vol. 3, no. 2, 2013.

[16] H. Zheng, Y. Zhang, L.-Y. Yang, P. Liang, Z. Zhao, C. Wang, and D. Z. Chen, “A new ensemble learning framework for 3d biomedical image segmentation,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, 2019, pp. 5909–5916.

[17] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, et al., “Woodscapes: A multi-task, multi-camera fisheye dataset for autonomous driving,” in The IEEE International Conference on Computer Vision (ICCV), 2019.

[18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[19] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in Proc. of CVPR, 2016.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Computer Vision and Pattern Recognition (CVPR), 2015.

[21] M. Uricar, D. Hurych, P. Krizek, and S. Yogamani, “Challenges in designing datasets and evaluation for autonomous driving,” in Proc. of VISAPP. INSTICC, 2019.

[22] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2881–2890.