Identifying Label Errors in Object Detection Datasets by Loss Inspection

Marius Schubert
IZMD, University of Wuppertal
Germany
schubert@math.uni-wuppertal.de

Tobias Riedlinger
IZMD, University of Wuppertal
Germany
riedlinger@math.uni-wuppertal.de

Karsten Kahl
IZMD, University of Wuppertal
Germany
kkahl@math.uni-wuppertal.de

Daniel Kröll
ControlExpert GmbH
Germany
d.kroell@controlexpert.com

Sebastian Schoenen
ControlExpert GmbH
Germany
s.schoenen@controlexpert.com

Siniša Šegvić
University of Zagreb
Croatia
sinisa.segvic@fer.hr

Matthias Rottmann
IZMD, University of Wuppertal
Germany
rottmann@math.uni-wuppertal.de

Abstract

Labeling datasets for supervised object detection is a dull and time-consuming task. Errors can be easily introduced during annotation and overlooked during review, yielding inaccurate benchmarks and performance degradation of deep neural networks trained on noisy labels. In this work, we introduce a benchmark for label error detection methods on object detection datasets as well as a theoretically underpinned label error detection method and a number of baselines. We simulate four different types of randomly introduced label errors on train and test sets of well-labeled object detection datasets. For our label error detection method we assume a two-stage object detector to be given and consider the sum of both stages' classification and regression losses. The losses are computed with respect to the predictions and the noisy labels including simulated label errors, aiming at detecting the latter. We compare our method to four baselines: a naive one without deep learning, the object detector's score, the entropy of the classification softmax distribution and a probability margin based method from related work. We outperform all baselines and demonstrate that among the considered methods, ours is the only one that detects label errors of all four types efficiently, which we also derive theoretically. Furthermore, we detect real label errors a) on commonly used test datasets in object detection and b) on a proprietary dataset. In both cases we achieve low false positives rates, i.e., we detect label errors with a precision for a) of up to 71.5% and for b) with 97%.

Figure 1. Example image from the Pascal VOC 2007 test dataset with two labeled boats marked by the blue boxes and multiple unlabeled boats.

1. Introduction

Nowadays, the predominant paradigm in computer vision is to learn models from data. The performance of the model largely depends on the amount of data and its quality, *i.e.* the diversity of input images and label accuracy [10, 17, 18, 21, 23]. Deep neural networks (DNNs) are particularly data hungry [34]. In this work, we focus on the case of object detection where multiple objects per scene belonging to a fixed set of classes are annotated via bounding boxes [9, 30].

In many industrial and scientific applications, the labeling process consists of an iterative cycle of data acquisition, labeling, quality assessment, and model training. Labeling data is costly, time consuming and error prone, e.g. due to inconsistencies caused by multiple human labelers or a change in label policy over time. Therefore, at least a partial automation of the label process is desirable. One research direction that aims at this goal is automated label error de-

tection [8, 27, 32]. The extent to which noisy labels affect the model performance is studied by [1, 38]. Wu *et al*. [38] observe that the model is able to tolerate a certain amount of missing annotations in training data without losing too much performance on Pascal VOC and Open Images V3 test sets. In contrast, Buttner *et al*. [1] show that inaccurate labels in terms of annotations size in training data yields to significant decrease of test performance for calculus detection on bitewing radiographs. Other methods model label uncertainty [26, 31] or improve robustness w.r.t. noisy labels [2, 11, 24, 43].

Up to now, automated detection of label errors has received less attention. There exist some works on image classification datasets [27, 28, 35], one work on semantic segmentation datasets [32] and some works for object detection [16, 22]. Label errors may affect generalization performance, which makes their detection desirable [28]. Furthermore, there is business interest in improving and accelerating the review process by partial automation.

Here, we study the task of label error detection in object detection datasets by a) introducing a benchmark and b) developing a detection method and compare it against four baselines. We introduce a benchmark by simulating label errors on the BDD100k [41] and EMNIST-Det [30] dataset. The latter is a semi-synthetic dataset consisting of EMNIST letters [5] pasted into COCO images [25] of which we expect to possess highly accurate labels. The types of label errors that we consider are missing labels (*drops*), correct localization but wrong classification (*flips*), correct classification but inaccurate localization (*shifts*), and labels that actually represent background (*spawns*). We address the detection of these errors by a novel method based on monitoring instance-wise object detection loss. We study the effectiveness of our method in comparison to four baselines. Then, we demonstrate for commonly used object detection test datasets, such as BDD100k [41], MS-COCO [25], Pascal VOC [9] and Kitti [12], and also for a proprietary dataset on car part detection that our method detects label errors by reviewing moderate sample sizes of 200 images per dataset. Our contributions can be summarized as follows: a) we introduce a novel method based on the instance-wise loss for detecting label errors in object detection, b) we introduce a benchmark for identifying four types of label errors on BDD100k and EMNIST-Det, and c) we apply our method to detect label errors in commonly used and proprietary object detection datasets and manually evaluate the error detection performance for moderate sample sizes.

2. Related Work

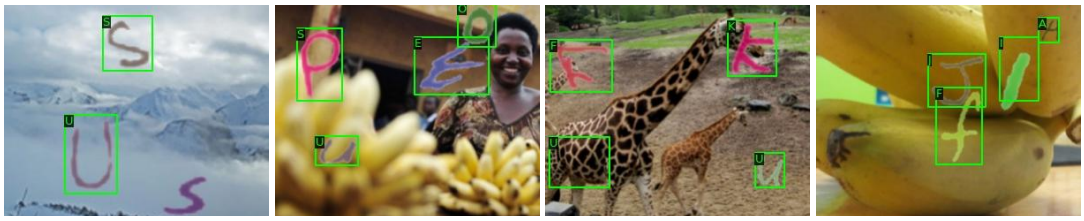
The influence of noisy labels in the training as well as in the test data is an active and current research topic. The labels for commonly used image classification datasets are noisy [28] and this also applies to object detection. Figure 1 shows an image from the Pascal VOC 2007 test dataset containing just two labeled boats, but clearly more can be seen.

For the task of image classification, some learning methods exist that are more robust to label noise [13–15, 19, 27, 29, 37, 40, 42]. Also the task of label error detection has been tackled in [4, 28] and theoretically underpinned in [27]. Chen *et al*. [4] filter whole samples with noisy labels but individual label errors are not detected. Northcutt *et al*. present label errors in image classification datasets and study to which extent they affect benchmark results [28] followed by the introduction of the task of label error detection [27]. The latter introduces a confident learning approach, assuming that the label errors are image-independent. Then, the joint distribution between the noisy and the true labels with class-agnostic label uncertainties is estimated and utilized to find label errors. This method allows to find label errors on commonly used image classification (*i.e*. MNIST or ImageNet) and sentiment classification datasets (Amazon Reviews), resulting in improved model performance by re-training on cleaned training data. This line of works has been recently extended to the task of multi-label classification in [35], where a single object is shown per image but may carry multiple labels.

For object detection, Wu *et al*. [38] as well as Xu *et al*. [39] study how noisy training labels affect the model performance, observing that the model is reasonably robust when dropping labels. To counter label errors in object detection, methods that model label uncertainty [26, 31] or more robust object detectors have been developed [1, 11, 20, 24, 39, 43]. Buttner *et al*. [1] simulate label errors and introduce a co-teaching approach for more robust training with noisy training data. For the task of label error detection, Koksal *et al*. [22] simulate different types of label errors in video sequences. Predictions and labels of consecutive frames are compared and then manually reviewed to eliminate erroneous annotations. Hu *et al*. [16] introduce a probability differential method (PD) to identify and exclude annotations with wrong class labels during training.

For semantic segmentation, a benchmark is introduced by Rottmann and Reese [32] to detect missing labels. For this purpose, uncertainty estimates are used to predict for each false positive connected component whether a label error is present or not. Detection is performed by considering the discrepancy of the given (noisy) label and the corresponding uncertainty estimate.

Our work introduces the first benchmark with four types of label errors for label error detection methods on object detection datasets as well as a label error detection method (that detects all four types of label errors) and a number of baselines. The label error detection methods simulate a) different types of label errors and detect these with the help of a tracking algorithm [22] for images derived from video sequences or b) class-flips and identify these via a probability differential (PD) [16], where, however, the focus



(a) Label Drop (b) Class Flip (c) Label Spawn (d) Loc. Inaccuracy

Figure 2. Examples of the different types of simulated label errors. The images are from the EMNIST-Det test dataset [30].

is on training. For our benchmark, we randomly simulate four types of label errors and detect them simultaneously with a new and four baseline methods, including PD. In our method, the discrepancy between the prediction or expectation of the network and the actual labels is used to find label errors. This discrepancy is determined by the classification and regression loss from the first and second stage of the detector. This allows to find not only simulated but also real label errors on commonly used object detection test datasets.

3. Label Error Detection

In this section we describe our label error benchmark as well as the setup and evaluation for real label errors on commonly used object detection test datasets and a proprietary dataset. We describe which datasets are used, which types of label errors are considered and the way we simulate label errors inspired by observations that we made in commonly used datasets and by related work. We then introduce our detection method as well as four additional baseline methods. This is complemented with evaluation metrics used to compare the methods with each other on our label error benchmark and the evaluation procedure for commonly used test datasets where we manually review the findings of our method for moderate sample sizes.

3.1. Label Error Benchmark

In the following, we distinguish between the label error benchmark and the detection of real label errors, where in the former, label errors are simulated (and therefore known) and the performance of the five different methods are evaluated automatically. In the latter case, label errors are not simulated but real and therefore automated evaluation is impossible as the real label errors are unknown. To enable a reliable evaluation, only datasets containing almost no real label errors are used for the benchmark. We observe that commonly used datasets in object detection, such as MS-COCO, Pascal VOC or Kitti, contain significant amounts of label errors, thus they are not suitable for the benchmark. Nevertheless, to demonstrate the performance of our instance-wise loss method on these datasets, a moderate sample size of 200 label error proposals are manually reviewed and counted for each dataset.

3.1.1 Datasets

For our benchmark we use the semi-synthetic EMNIST-Det dataset and BDD100k, referred to as BDD. EMNIST-Det consists of $20{,}000$ training and $2{,}000$ test images. To have the best possible labels for BDD, we filter the training and validation split, such that we only use daytime images with clear weather conditions. This results in $12{,}454$ training images and the validation split is split into equally-sized test and validation sets, each consisting of 882 images.

3.1.2 Simulated Label Errors

We consider four different types of label errors: missing labels (*drops*), correct localization but wrong classification (*flips*), correct classification but inaccurate localization (*shifts*), and labels that actually represent background (*spawns*). Any dataset is equipped with a set of G labels, *i.e.* $\mathcal{Y} = \{b^{(i)} : i = 1, ..., G\}$ where each label is a tuple $b^{(i)} = (x^{(i)}, y^{(i)}, w^{(i)}, h^{(i)}, c^{(i)})$ containing the box center $(x^{(i)}, y^{(i)})$, the box extent $(w^{(i)}, h^{(i)})$ and a class index $c^{(i)}$ from the set of classes $\{1, ..., C\}$. Let $\mathcal{I} = \{1, \dots, G\}$ be the set of indices of all boxes $b^{(i)} \in \mathcal{Y}$, $i = 1, \dots, G$. We now describe all types of label errors applied to $\mathcal{Y}$ and we make the assumption that a single label is only perturbed by one type of label error instead of multiple types. We choose a parameter $\gamma \in [0, 1]$ representing the relative frequency of label errors.

Drops For dropping labels, we randomly choose a subset $\mathcal{I}_d$ of $\mathcal{I}$ with cardinality $|\mathcal{I}_d| = \lfloor \frac{\gamma}{4} \cdot G \rfloor$. We drop all labels $\mathcal{Y}_d = \{b^{(i)} : i \in \mathcal{I}_d\}$ and denote $\mathcal{I}_{\setminus d} = \mathcal{I} \setminus \mathcal{I}_d$. Analogously, $\mathcal{Y}_{\setminus d} = \mathcal{Y} \setminus \mathcal{Y}_d$.

Flips For flipping class labels, we randomly choose a subset $\mathcal{I}_f$ of $\mathcal{I}_{\setminus d}$ with cardinality $|\mathcal{I}_f| = \lfloor \frac{\gamma}{4} \cdot G \rfloor$ and copy $\tilde{\mathcal{Y}}_f = \mathcal{Y}_f = \{b^{(i)} : i \in \mathcal{I}_f\}$. Then, we randomly flip the class of every label in $\tilde{\mathcal{Y}}_f$ to a different label. We denote $\mathcal{I}_{\setminus f} = \mathcal{I}_{\setminus d} \setminus \mathcal{I}_f$ and $\mathcal{Y}_{\setminus f} = (\mathcal{Y}_{\setminus d} \setminus \mathcal{Y}_f) \cup \tilde{\mathcal{Y}}_f$.

Shifts To insert *shifts*, we change the localization of labels. We randomly choose a subset $\mathcal{I}_{sh}$ of $\mathcal{I}_{\setminus f}$ with cardinality

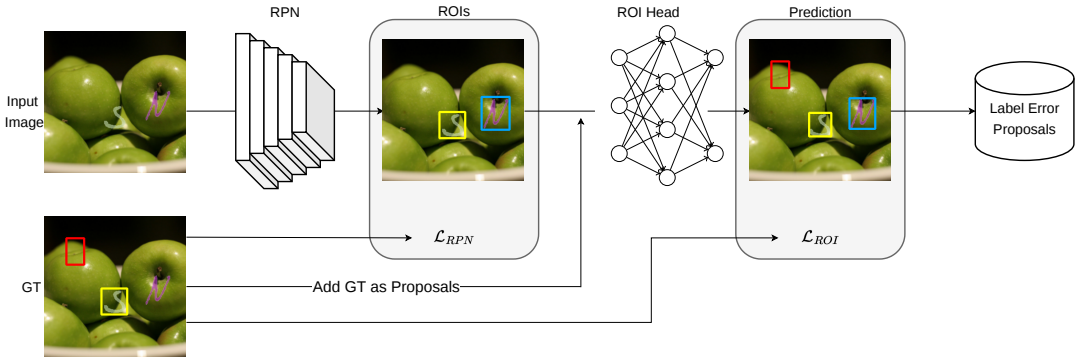


Figure 3. Visualization of our instance-wise loss method for detecting label errors. The red label indicates a *spawn*, the blue one a *drop* and the yellow one a correct label.

$|\mathcal{I}_{sh}| = \lfloor \frac{\gamma}{4} \cdot G \rfloor$ and copy $\tilde{\mathcal{Y}}_{sh} = \mathcal{Y}_{sh} = \{b^{(i)} : i \in \mathcal{I}_{sh}\}$. For the shift of a box $\tilde{b}^{(i)} \in \tilde{\mathcal{Y}}_{sh}$, the new values $\tilde{y}$, $\tilde{h}$ are determined analogously to $\tilde{x} = \mathcal{N}(x, 0.15 \cdot w)$ and $\tilde{w} = \mathcal{N}(w, 0.15 \cdot w)$ drawn from a normal distribution with itself as the expected value and $0.15 \cdot w$ as the standard deviation. To avoid the *shift* being too small or too large, the parameters are repeatedly chosen until the intersection over union (IoU) of the original label $b^{(i)} \in \mathcal{Y}_{sh}$ and $\tilde{b}^{(i)} \in \tilde{\mathcal{Y}}_{sh}$ is in the interval of $[0.4, 0.7], \forall i = 1, ..., \lfloor \frac{\gamma}{4} \cdot G \rfloor$. We denote $\mathcal{I}_{\setminus sh} = \mathcal{I}_{\setminus f} \setminus \mathcal{I}_{sh}$ and $\mathcal{Y}_{\setminus sh} = (\mathcal{Y}_{\setminus f} \setminus \mathcal{Y}_{sh}) \cup \tilde{\mathcal{Y}}_{sh}$.

Spawns For spawning labels, we randomly choose a subset $\mathcal{I}_{sp}$ of $\mathcal{I}_{\setminus sh}$ with cardinality $|\mathcal{I}_{sp}| = \lfloor \frac{\gamma}{4} \cdot G \rfloor$ and copy $\tilde{\mathcal{Y}}_{sp} = \mathcal{Y}_{sp} = \{b^{(i)} : i \in \mathcal{I}_{sp}\}$. Then, we assign every label $\tilde{b}^{(i)} \in \tilde{\mathcal{Y}}_{sp}$ randomly to another image. Since in our experiments all images in a dataset have the same resolution, this ensures that objects do not appear in unusual positions or outside of an image. For instance, a car in BDD is more likely to be found on the bottom part of the image rather than in the sky. We denote the set of noisy labels as $\tilde{\mathcal{Y}} = \mathcal{Y}_{\setminus sh} \cup \tilde{\mathcal{Y}}_{sp}$.

One example per label error type is shown in Fig. 2. Note that the number of labels G is unchanged as the number of *drops* and *spawns* is the same.

3.2. Baseline Methods

The four baselines that we compare our instance-wise loss method with are based on a) inspecting the labels without the use of deep learning, b) the box-wise detection score c) the classification entropy of the two-stage object detectors and d) the probability differential from [16].

Naive Baseline We introduce a naive baseline to show the significant improvement of deep learning in label error detection for object detection over manual label review. We assume that all label errors can be smoothly found by taking a single look at all existing noisy labels and the (actually unknown) *drops*, *i.e.* by performing $\lfloor (1 + \frac{\gamma}{4}) \cdot G \rfloor$ operations. This simplified assumption is of course unrealistic, however the corresponding results can serve as a lower bound for the effort of manual label review.

Detection Score Baseline The detection score baseline works as follows: For a given image from the set of all images of the dataset $z \in Z$, a neural network predicts a fixed number N_0 of bounding boxes for the first stage $\hat{\mathcal{B}}_{0,z} = \{(\hat{x}^{(i)}, \hat{y}^{(i)}, \hat{w}^{(i)}, \hat{h}^{(i)}, \hat{s}_0^{(i)}) : i = 1, \ldots, N_0\}$, where $\hat{x}^{(i)}, \hat{y}^{(i)}, \hat{w}^{(i)}, \hat{h}^{(i)}$ represent the localization and $\hat{s}_0^{(i)} \in [0, 1]$ the objectness score. Then, we add the boxes of the labels as proposals for the second stage to ensure that at least one prediction exists for each label, which is particularly important for the detection of *spawns*. For this purpose, each ground truth label from $\mathcal{Y}$ is assigned with a detection score of $\hat{s}_0 = 1$. After adding the labels to $\hat{\mathcal{B}}_{0,z}$, only those N_1 boxes that remain after class-independent non-maximum suppression (NMS) and score thresholding on $\hat{s}_0$ with $s_\epsilon \geq 0$, get into the second stage $\hat{\mathcal{B}}_{1,z} = \{(\hat{x}^{(i)}, \hat{y}^{(i)}, \hat{w}^{(i)}, \hat{h}^{(i)}, \hat{s}_0^{(i)}) : i = 1, \ldots, N_1\}$. After box refinement and classification as well as NMS on the detection score $\hat{s}_2^{(i)}$, N_2 label error proposals remain. Here, $\hat{s}_2^{(i)}$ is the detection score of the detection head and unlike $\hat{s}_0^{(i)}$, $\hat{s}_2^{(i)}$ represents not only the presence of an object, but also takes the class probabilities of the predicted object into account. The remaining N_2 label error proposals $\hat{\mathcal{B}}_{2,z} = \{(\hat{x}^{(i)}, \hat{y}^{(i)}, \hat{w}^{(i)}, \hat{h}^{(i)}, \hat{s}_2^{(i)}, \hat{p}_1^{(i)}, \ldots, \hat{p}_C^{(i)}) : i = 1, \ldots, N_2\}$, are defined by the localization $(\hat{x}^{(i)}, \hat{y}^{(i)}, \hat{w}^{(i)}, \hat{h}^{(i)})$, the detection score $\hat{s}_2^{(i)}$ and the class probabilities $\hat{p}_1^{(i)}, \ldots, \hat{p}_C^{(i)}$. The predicted class is given by $\hat{c}^{(i)} = \arg\max_{k=1,\ldots,C} \hat{p}_k^{(i)}$. Score thresholding is omitted here, or the score τ used for this is equal to 0, since $\tau > 10^{-4}$ would suppress most of the label error proposals that detect *spawns*. The detection score of these proposals is mostly very close to zero unless

a second true label is nearby. After inferring each image $z \in Z$ as described above, we get label error proposals for the whole dataset by $\bigcup_{z \in Z} \hat{\mathcal{B}}_{2,z}$.

Entropy Baseline The entropy baseline follows the same procedure, only the NMS in the first and second stage are based on the respective box-wise entropy rather than the detection score.

Probability Differential Baseline For the PD baseline from Hu *et al*. [16], we do not add the boxes of the labels as proposals. Furthermore, score thresholding and NMS is not applied, such that every box $\hat{b} \in \hat{\mathcal{B}}_{0,z}$ also remains in $\hat{\mathcal{B}}_{2,z}$. After assigning every label with sufficiently overlapping predictions, the probability differential (PD) for every label $b \in \mathcal{Y}$ with class c and the m assigned predictions $\hat{b}^{(i)}$ $(i = 1, \ldots, m)$ is defined as: $PD(b) = \frac{\sum_{i=1}^{m} \left(1 + \max_{k \in \{1,\ldots,C\} \setminus \{c\}} \hat{p}_k^{(i)} - \hat{p}_c^{(i)}\right)}{2m}$. The PD of a label is in $[0, 1]$ and intuitively, the more the probabilities of the predictions and the class of the label differ (higher PD) the more likely a label error is present. Note that *drops* are always overlooked.

3.3. Instance-wise Loss Method

Our method to detect the introduced label error types (Sec. 3.1) is based on an instance-wise loss for two-stage object detectors. The NMS is no longer based on the detection score or the entropy, but on the box-wise loss of the respective stage. Every prediction $\hat{b}_0 \in \hat{\mathcal{B}}_{0,z}$ is assigned with a region proposal loss ($\mathcal{L}_{RPN}$), which is the sum of a classification (binary cross-entropy) and regression (smooth-L1) loss for the labels and the prediction itself. The computation of the loss is identical to the one in training. Since not all labels are associated with a proposal after the first stage, *i.e*. the model may predict only background near a label, we add the labels themselves to the set of label error proposals. After box refinement and classification, every box $\hat{b}_1 \in \hat{\mathcal{B}}_{1,z}$ is assigned with a region of interest loss ($\mathcal{L}_{ROI}$), which is the sum of a classification (cross entropy) and regression (smooth-L1) loss for the labels and the prediction itself. Then $\mathcal{L}_{RPN}$ and $\mathcal{L}_{ROI}$ are summed up to obtain an instance-wise loss score. A sketch of our method is shown in Fig. 3. We can find the *dropped* blue label for "N" since the predictions near the object should have a high detection score, resulting in a high first stage classification loss. The *spawned* red label is assigned with a high classification loss from the first and second stage, since the assigned predictions should have a score close to zero in the first stage and an almost uniform class distribution in the second stage. Whether the yellow label is a *flip* is irrelevant for the first stage, since the loss should be small either way. If the box is classified correctly according to the associated label, there is a large classification loss for a *flip* and a small one otherwise. The *shifts* are addressed by the first and second stage regression loss.

The intuition behind our method is that a sufficiently well-specified and fitted model has small expected loss on data sampled during training. Sufficient data sampling and moderate label error rates lead to label errors giving rise to outlier losses which are identified as proposals. We show that our method separates correct from incorrect labels for a classification model $\widehat{p}$ trained with the cross entropy loss ℓ_{CE}.

Proposition 1 (Statistical Separation of the Cross Entropy Loss)**.** *Let training and testing labels be given under a stochastic flip in $p(\cdot|x)$ with probability p_{F}. A correct label $y = f(x)$ is given by a true labeling function f and has probability $p(f(x)|x) = 1 - p_{\mathrm{F}}$. Incorrect labels $\widetilde{y} \neq f(x)$ are drawn with probability $p(\widetilde{y}|x) = p_{\mathrm{F}}/(C-1)$. Let the label distribution $p(\cdot|x)$ be PAC-learnable by the hypothesis space of $\widehat{p}(\cdot|x)$ w.r.t. D_{KL} (to precision ε and confidence $1-\delta$) and let $\kappa > 0$. If $p_{\mathrm{F}} < \frac{C-1}{C}(1-2\kappa)$, we obtain strict separation of the loss function*

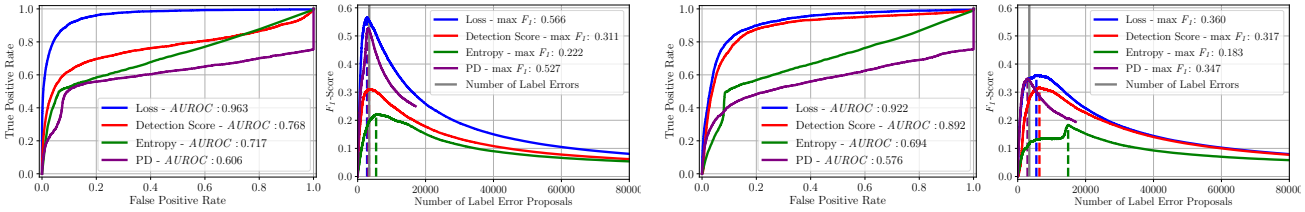
$$\begin{aligned} \ell_{\mathrm{CE}}(\widehat{p}(x) \| f(x)) &< -\log(1 - p_{\mathrm{F}} - \kappa) \\ &< -\log(\kappa + \tfrac{p_{\mathrm{F}}}{C-1}) < \ell_{\mathrm{CE}}(\widehat{p}(x) \| \widetilde{y}) \end{aligned} \tag{1}$$

for any incorrect label $\widetilde{y} \neq f(x)$ with probability $1 - \delta$ over chosen training data and with probability $1 - \frac{2\varepsilon}{\kappa^2}$ over the choice of x.

We include a proof of this statement in the supplementary material. The PAC-learnability assumption [33] yields rigorous bounds for the deviation of the model $\widehat{p}$ from the label distribution p which contains label flips. Conditioned to the events of drawing correct versus drawing incorrect labels, these bounds carry over to the cross entropy. These bounds separate the two events with certain probability given in the statement above.

3.4. Evaluation Metrics

Ignoring that natural label errors exist in EMNIST-Det and BDD, we benchmark the five methods introduced in above by means of our label error simulation. To this end, we take the label error proposals of the respective method and the set of original labels $\mathcal{Y}$ and decide for every proposal whether it is a label error, which corresponds to a true positive (TP_l), or no label error, which corresponds to a false positive (FP_l). Label errors that are not detected are called false negatives (FN_l). A proposal of a label error detector is a TP_l if the IoU between the proposal under consideration and a noisy label on the image is greater or equal to a threshold $1 \geq \alpha > 0$. Here, the noisy label categorizes what type of label error is detected by the proposal. If the IoU is less than α, the proposal is a FP_l. After determining this for each proposal



(a) Original training data

(b) Noisy training data with $\gamma = 0.2$

Figure 4. The two left plots in (a) show evaluations based on the predictions of a model trained on original training data and the two right ones in (b) based on noisy training data with $\gamma = 0.2$. The number of considered label error proposals depends on threshold τ.

Dataset	Backbone	mAP_{50}	$mAP_{50}^{(*)}$
EMNIST-Det	Swin-T	98.2	98.0
EMNIST-Det	ResNeSt101	96.4	95.2
BDD	Swin-T	52.1	50.3
BDD	ResNeSt101	56.8	52.9
COCO	Swin-T	54.1	
Kitti	Swin-T	38.6	
VOC	Swin-T	83.3	
CE	Swin-T	70.0	

Table 1. Validation of object detection performance on our datasets. $^{(*)}$ indicates learning with simulated label errors ($\gamma = 0.2$).

from the dataset, the area under the receiver operator characteristic curve ($AUROC$, see [6]) and F_1 values, which is the harmonic mean of precision and recall (see [7]), is calculated according to the decision between TP_l and FP_l. F_1 values are determined with thresholding on the score of the respective method (loss/detection score/entropy/PD). We always choose the optimal threshold, *i.e.* the threshold at which the F_1 value is maximized ($\max F_1$). Note, since the naive baseline considers images and thus label error proposals in random order, the associated $AUROC$ values are always 0.5.

3.5. Detection of Real Label Errors

For commonly used datasets we proceed as follows. We consider for each dataset 200 proposals of our method with highest loss and manually flag them as TP_l or FP_l, based on the label policy corresponding to the given dataset. Note that we can still compute precision values but we are not able to determine $AUROC$ or $\max F_1$ values as the number of total label errors is unknown. Since several label errors can be detected with one proposal, precision describes the ratio of proposals with at least one label error and the total number of proposals considered, i.e. 200.

4. Numerical Results

In this section we study label error detection performance on our label error benchmark as well as for real label errors in BDD, VOC, MS-COCO (COCO), Kitti and the proprietary dataset (CE). The benchmark results are presented in terms of $AUROC$ and $\max F_1$ values for the joint evaluation of all label error types, *i.e.* when all label error types are present simultaneously, in Sec. 4.2. For the latter, we show how many real label errors we can detect among the top-200 proposals for each real-world dataset in Sec. 4.3.

4.1. Implementation Details

We implemented our benchmark and methods in the open source MMDetection toolbox [3]. Our models are based on a Swin-T transformer and a ResNeSt101 backbone, both with a CascadeRoIHead as the object detection head, with a total number of trainable parameters of approx. 72M and 95M. As hyperparameters for the label error benchmark we choose relative frequency of label errors $\gamma = 0.2$, the value for score thresholding after the first stage $s_\epsilon = 0.25$, the value for score thresholding after the second stage $\tau = 0$ and the IoU-value $\alpha = 0.3$ from which a proposal for a label error is considered a TP_l. We show performance results for the respective models and for each dataset in Tab. 1. The upper half shows results on original (mAP_{50}) and noisy training data (mAP_{50}^*), for which $\gamma = 0.2$ also holds. With sufficient training data and a moderate label error rate ($\gamma = 0.2$), the models still generalize well, resulting in mAP values comparable to models trained without simulated label errors. Thus, to make benchmarks evaluations trustworthy, whether already published or still under development, in particular the underlying test datasets should contain as few label errors as possible. The bottom half of Tab. 1 presents the performance of the models that we use for predicting label errors on real datasets. All models have been trained and evaluated on the original datasets (without any label modification).

Datasets For the detection of real label errors we use the same split for BDD as introduced in Sec. 3.1 as well as VOC, COCO, Kitti and CE. The training data for VOC consists of "2007 train" + "2012 trainval" and we predict label errors on the "2007 test"-split. COCO is trained on the train split and label errors are predicted on the validation split from 2017. For Kitti we use a scene-wise split, resulting in 5 scenes

Dataset	Backbone	Train Labels	$AUROC$ Loss	$AUROC$ Detection Score	$AUROC$ Entropy	$AUROC$ PD	max F_1 Loss	max F_1 Detection Score	max F_1 Entropy	max F_1 PD
EMNIST-Det	Swin-T	Original	**99.46**	73.24	71.49	59.67	**95.54**	64.74	49.58	62.32
EMNIST-Det	Swin-T	Noisy	**99.40**	82.44	77.32	62.26	**93.43**	62.37	45.25	62.24
EMNIST-Det	ResNeSt101	Original	**99.84**	88.45	86.70	60.59	**94.31**	62.56	38.81	60.82
EMNIST-Det	ResNeSt101	Noisy	**99.87**	93.11	86.40	61.82	**90.74**	59.50	34.53	59.01
BDD	Swin-T	Original	**96.30**	76.82	71.73	60.59	**56.59**	31.14	22.21	52.66
BDD	Swin-T	Noisy	**92.16**	89.21	69.42	57.58	**35.97**	31.68	18.33	34.72
BDD	ResNeSt101	Original	**95.79**	87.47	83.58	60.31	**54.62**	31.99	20.37	47.16
BDD	ResNeSt101	Noisy	**92.97**	90.76	78.18	56.79	**27.85**	25.65	18.10	27.74

Table 2. Label error detection experiments with two different backbones; higher values are better. Bold numbers indicate the highest $AUROC$ or max F_1 per experiment and underlined numbers are the second highest.

Dataset	Label Error	Prec.	Spawn	Drop	Flip	Shift
BDD	34	15.5	3	2	26	3
Kitti	96	47.5	75	0	4	17
COCO	50	24.5	14	1	18	17
COCO$^{(*)}$	125	61.0	0	125	0	0
VOC	23	11.5	13	0	10	0
VOC$^{(*)}$	175	71.5	0	175	0	0
CE$^{(*)}$	194	97.0	0	0	0	0

Table 3. Categorization of the top-200 proposals for real label errors with the loss method for the Swin-T backbone. $^{(*)}$ indicates the evaluation of proposals based on the detection of *drops*.

($S = \{2, 8, 10, 13, 17\}$) and 1,402 images for evaluation as well as 16 scenes ($\{0, 1, \ldots, 20\} \setminus S$) and 6,407 images for training. The subset of CE data used includes 20,100 images for training and 1,070 images for evaluation. In the images, a car is in focus and the task is to do a car part detection. The labels consist of 29 different classes and divide the car into different parts, *i.e*. the four wheels, doors, number plate, mirrors, bumper, etc. Compared to the static academic datasets, the CE dataset is dynamic and thus of heterogeneous quality.

4.2. Benchmark Results for Simulated Label Errors

Table 1 shows that although 20% of the training labels are modified, the performance in terms of mAP_{50} to mAP^*_{50} only decreases by a maximum of 1.2 percent points (pp) for EMNIST-Det and 3.9 pp for BDD. In both cases, the performance decreases more for the backbone containing more trainable parameters (ResNeSt101). This is consistent with the results for image classification from Northcutt *et al*. [28]. Architectures with fewer trainable parameters seem more suitable for handling label errors in the training data, possibly due to the network having less capacity to overfit the label errors. Figure 4 shows exemplary plots for $AUROC$ and F_1 curves for the Swin-T backbone and BDD. On the two left plots we show results based on original training data and the two right plots based on noisy training data. The ranking of the methods is not identical everywhere: in terms of $AUROC$, loss (our method) is superior, followed by detection score, then entropy (our baselines) and finally PD. In terms of max F_1, PD outperforms the detection score and the entropy but is inferior compared to the loss. Because $AUROC$ considers rates and (max) F_1 considers absolute values and the number of label error proposals varies widely (PD = number of labels G, here 17,064; others > 80,000), the methods behave very differently with respect to $AUROC$ and max F_1. However, our loss method outperforms all other methods on both metrics. Note, that the small step in the upper right of each of the $AUROC$ plots are the false negatives according to the label errors (FN_l), *i.e*. the simulated label errors that are not found by the methods. This number of FN_l is vanishingly small in relation to all simulated label errors, with the exception of PD as the method is not able to detect *drops*. The generally observed behavior for BDD also does not change when looking at the results for the ResNeSt101 backbone in Tab. 2. When comparing the results for the different backbones with each other the $AUROC$ for the loss and PD seems to remain similar, whereas the $AUROC$ for detection score/entropy increases by 10.65/11.85 pp for original training data and 1.55/8.76 pp for noisy training data. The situation is different for the max F_1 values. For label error detection, loss/entropy/PD performs superior with the Swin-T backbone for original training data (1.97/1.84/5.50 pp). In particular, the loss and PD seem to handle the noisy training data more effectively, resulting in 8.12 pp max F_1 difference between Swin-T and ResNeSt for the loss and 6.98 pp difference for PD. The detection score increases by 0.85 pp with the ResNeSt101 backbone on original training data, but on noisy data the Swin-T outperforms the ResNeSt101 by 6.03 pp. Also for EMNIST-Det it holds that the loss outperforms the detection score and both outperform the entropy. In contrast to the results of BDD, the detection score slightly outperforms PD in all EMNIST-Det experiments also in terms of max F_1. The $AUROC$ for loss appears to be stable across backbone and training data quality with only a maximum 0.47 pp difference overall. The $AUROC$ values for the detection score and entropy are superior with the ResNeSt101 backbone, but inferior in terms of max F_1 and the detection score performs superior in terms of $AUROC$ based on noisy training

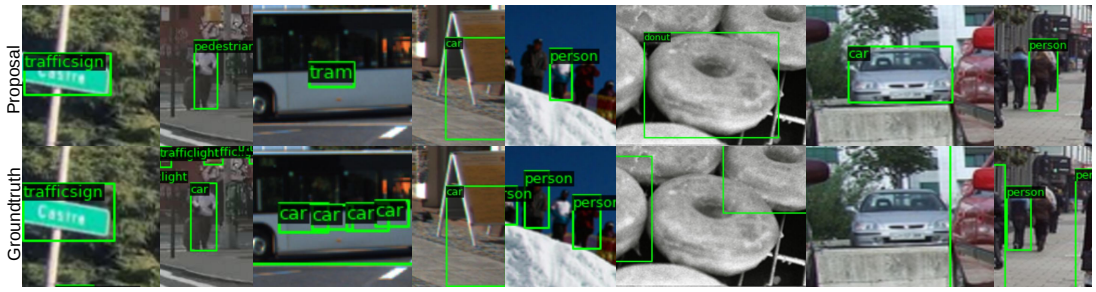


Figure 5. Visualization of detected label errors in real test datasets. The top row of images depicts the label error proposals and the bottom row the corresponding labels from the dataset. The image pairs belong from left to right in steps of two to BDD, Kitti, COCO and VOC.

data, but inferior in terms of $\max\ F_1$. For PD, the $AUROC$ seems to be rather stable comparing the two backbones, but the $\max\ F_1$ is superior for Swin-T.

4.3. Evaluation for Real Label Errors

We now aim at detecting real instead of simulated label errors. The considered real-world datasets apart from BDD (VOC, COCO, Kitti, CE) are more similar in complexity to BDD than to EMNIST-Det. For BDD we observed in Sec. 4.2 that the loss method for the Swin-T backbone seems to be more stable according to label errors in the training data, as especially the $\max\ F_1$ values for the loss and noisy labels are superior for Swin-T than for ResNeSt101. As we suspect label errors in the VOC, COCO, Kitti and CE training datasets, we use the Swin-T backbone to detect as many label errors as possible. Furthermore, we showed in Tab. 2 that the loss method outperforms the detection score, entropy and PD in each presented experiment, hence we detect label errors using only the loss method in the following. Since we manually look at all proposals individually and we are not able to look at all proposals (*i.e.* about 265,000 for VOC), we categorize the top-200 proposals into TP_l or FP_l. If a TP_l is found we also note which type of label error is present and if we are not sure whether the proposal is TP_l or FP_l, we conservatively interpret it as FP_l. The results are summarized in Tab. 3. For BDD, there are at least 34 label errors, which mostly consist of *flips*. Since Kitti consists of image sequences, it happens that one label error appears on several consecutive frames. When this happens, it usually affects objects that are visible on previous frames but are covered by, for instance, a bus for several frames but are still labeled. Label error proposals that fall into "Don't Care" areas are not considered. In total, we find 96 label errors with a precision of 47.5% on Kitti. As COCO and VOC consist of images of different everyday scenes that really differ from image to image, the variability of the representation of objects is very high in these two datasets. Since a label error proposal is enforced for each label, this also applies to the labels that are classified as background. In a usual test setting, these labels would have been false negatives of the model, *i.e.* overlooked labels. The resulting loss is so high that these proposals end up in the reviewed top-200 proposals. Nevertheless, 50 label errors can be detected on COCO and 23 on VOC. When dealing with these two datasets, we noticed that *drops* are the most present label error type, although we did not find any among the top-200 proposals. We use this knowledge to restrict the proposals to those that have a class-independent IoU with the labels of the image of less than α. Using this subset and re-reviewing the top-200 proposals, we are able to find 125 *drops* with a precision of 61.0% for COCO and 175 *drops* with a precision of 71.5% for VOC. For the calculation of the precision see Sec. 3.5. Prior knowledge about the label quality of the dataset and the types of label errors that occur helps to detect a specific type of label error. From the high precisions for VOC and COCO, we conclude that our method can help to correct the label errors resulting in cleaner benchmarks. Exemplary label errors for the above datasets are shown in Fig. 5. The first proposal detects a *shift*, the second a *flip*, the third and fourth a *spawn* and the remaining proposals detect *drops*. For CE, we filter the proposals by *drops*, resulting in 194 detected *drops* with a precision of 97%.

5. Conclusion

In this work, we introduced a benchmark for label error detection for object detection datasets. We for the first time simulated and evaluated four different types of label errors on two selected datasets. We also developed a novel method based on instance-wise loss scoring and compare it with four baselines. Our method prevails by a significant margin in experiments on our simulated label error benchmark. In our experiments with real label errors, we found a number of label errors in prominent datasets as well as in a proprietary production-level dataset. With the evaluation for individual label error types we can detect real label errors on commonly used test datasets in object detection with a precision of up to 71.5%. Furthermore, we presented additional findings. Models with less parameters are more robust to label errors in training sets while models with more parameters suffer more. We make our code publicly available at *GitHub*.

References

[1] Martha Büttner, Lisa Schneider, Aleksander Krasowski, Joachim Krois, Ben Feldberg, and Falk Schwendicke. Impact of noisy labels on dental deep learning—calculus detection on bitewing radiographs. *Journal of Clinical Medicine*, 12(9):3058, 2023. 2

[2] Simon Chadwick and Paul Newman. Training object detectors with noisy data. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1319–1325. IEEE, 2019. 2

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[4] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pages 1062–1070. PMLR, 2019. 2

[5] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017. 2

[6] Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, pages 233–240, 2006. 6

[7] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945. 6

[8] Markus Dickinson and Detmar Meurers. Detecting errors in part-of-speech annotation. In *10th conference of the European chapter of the association for computational linguistics*, 2003. 2

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 1, 2

[10] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020. 1

[11] Di Feng, Zining Wang, Yiyang Zhou, Lars Rosenbaum, Fabian Timm, Klaus Dietmayer, Masayoshi Tomizuka, and Wei Zhan. Labels are not perfect: Inferring spatial uncertainty in object detection. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 2

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2

[13] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *International Conference on Machine Learning*, 2017. 2

[14] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018. 2

[15] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. *Advances in neural information processing systems*, 31, 2018. 2

[16] Zibo Hu, Kun Gao, Xiaodian Zhang, Junwei Wang, Hong Wang, and Jiawei Han. Probability differential-based class label noise purification for object detection in aerial images. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. 2, 4, 5

[17] Rasheed Hussain and Sherali Zeadally. Autonomous cars: Research results, issues, and future challenges. *IEEE Communications Surveys & Tutorials*, 21(2):1275–1313, 2018. 1

[18] Paul F Jaeger, Simon AA Kohl, Sebastian Bickelhaupt, Fabian Isensee, Tristan Anselm Kuder, Heinz-Peter Schlemmer, and Klaus H Maier-Hein. Retina u-net: Embarrassingly simple exploitation of segmentation supervision for medical object detection. In *Machine Learning for Health Workshop*, pages 171–183. PMLR, 2020. 1

[19] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pages 2304–2313. PMLR, 2018. 2

[20] Daniel Kang, Nikos Arechiga, Sudeep Pillai, Peter D Bailis, and Matei Zaharia. Finding label and model errors in perception data with learned observation assertions. In *Proceedings of the 2022 International Conference on Management of Data*, pages 496–505, 2022. 2

[21] Amrita Kaur, Yadwinder Singh, Nirvair Neeru, Lakhwinder Kaur, and Ashima Singh. A survey on deep learning approaches to medical images and a systematic look up into real-time object detection. *Archives of Computational Methods in Engineering*, pages 1–41, 2021. 1

[22] Aybora Koksal, Kutalmis Gokalp Ince, and Aydin Alatan. Effect of annotation errors on drone detection with yolov3. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1030–1031, 2020. 2

[23] Sampo Kuutti, Richard Bowden, Yaochu Jin, Phil Barber, and Saber Fallah. A survey of deep learning applications to autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):712–733, 2020. 1

[24] Hengduo Li, Zuxuan Wu, Chen Zhu, Caiming Xiong, Richard Socher, and Larry S Davis. Learning from noisy anchors for one-stage object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10588–10597, 2020. 2

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[26] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3243–3249. IEEE, 2018. 2

[27] Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021. 2

[28] Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*, 2021. 2, 7

[29] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014. 2

[30] Tobias Riedlinger, Marius Schubert, Karsten Kahl, Hanno Gottschalk, and Matthias Rottmann. Towards rapid prototyping and comparability in active learning for deep object detection, 2022. 1, 2, 3

[31] Tobias Riedlinger, Marius Schubert, Karsten Kahl, and Matthias Rottmann. Uncertainty quantification for object detection: output-and gradient-based approaches. In *Deep Neural Networks and Data for Automated Driving*, pages 251–275. Springer, Cham, 2022. 2

[32] Matthias Rottmann and Marco Reese. Automated detection of label errors in semantic segmentation datasets via deep learning and uncertainty quantification. *arXiv preprint arXiv:2207.06104*, 2022. 2

[33] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014. 5, 15

[34] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 1

[35] Aditya Thyagarajan, Elías Snorrason, Curtis Northcutt, and Jonas Mueller. Identifying incorrect annotations in multi-label classification data, 2022. 2

[36] Ramon Van Handel. Probability in high dimension. Technical report, PRINCETON UNIV NJ, 2014. 15

[37] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019. 2

[38] Zhe Wu, Navaneeth Bodla, Bharat Singh, Mahyar Najibi, Rama Chellappa, and Larry S Davis. Soft sampling for robust object detection. *arXiv preprint arXiv:1806.06986*, 2018. 2

[39] Mengmeng Xu, Yancheng Bai, Bernard Ghanem, Boxiao Liu, Yan Gao, Nan Guo, Xiaochun Ye, Fang Wan, Haihang You, Dongrui Fan, et al. Missing labels in object detection. In *CVPR Workshops*, volume 3, page 5, 2019. 2

[40] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. *Advances in neural information processing systems*, 32, 2019. 2

[41] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 2

[42] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

[43] Haichao Zhang and Jianyu Wang. Towards adversarially robust object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 421–430, 2019. 2

Supplementary Material

Proprietary Dataset An exemplary test image including labels for the proprietary dataset (CE) is shown in Fig. 6. The labels divide the car into parts, such as the two wheels "WheelFrontRight" and "WheelRearRight" as well as doors, roof, etc. The example also includes a *drop* with the missing mirror "MirrorRight".

Dataset Dependent Parameters for Training The dataset-dependent hyperparameters for training are stated in Tab. 4. The original images from EMNIST-Det have an image resolution of 320×320 pixels, i.e., we do not artificially scale them to a higher image resolution. The BDD images also contain many small labels while having a high original resolution (1280×720), which is a challenging setup. To get the best possible label error detection, we keep this high resolution and rescale the images to 1333×800 pixels. Kitti, COCO, VOC and CE are each rescaled to an image resolution of 1000×600 pixels. The batch size for all datasets is in the range of 4-24, the initial learning rate is either 0.02 or 0.01 depending on the dataset, and the number of training iterations is in the range of 24,000-250,000. All numbers apply to the Swin-T backbone except the numbers $^{(*)}$ for the training iterations of EMNIST-Det and BDD, which apply to the ResNeSt101 backbone. All other hyperparameters are identical for the different architectures. The files for the configurations used in training, also containing the precise values of the above hyperparameters, are published with the code on *GitHub*.

Benchmark Results for Individual Simulated Label Error Types In our experiments, all label errors occur simultaneously, but the evaluation can also be conditioned on the individual label error types. For *drops* or *flips* we consider only the false positives according to $\tilde{y}$, *i.e.* all boxes that have a maximum class-wise IoU of less than $\alpha (= 0.3)$ with all noisy labels of the associated image. Then, we can calculate $AUROC$ and $\max\ F_1$ values on this subset. We do the same for the *shifts*, except that we only consider the true positives according to $\tilde{\mathcal{Y}}$. For the *spawns*, we must consider both true positives and false positives according to $\tilde{\mathcal{Y}}$, since the predicted class, that overlaps sufficiently with the *spawned* label, can be the same as the class of the *spawn* itself.

The benchmark results for individual simulated label errors are stated in Tab. 5. For *drops*, the detection score and instance-wise loss perform similarly well, with the $AUROC$ values differing by at most 1.92 pp and a minimal $AUROC$ of 91.72%. The difference in the $\max\ F_1$ values is more pronounced, with the loss at EMNIST-Det outperforming the detection score by 3.03 to 6.56 pp. For BDD, the detection score of Swin-T is superior to the loss by up to 9 pp, whereas the loss for ResNeSt101 outperforms the detection score by up to 10.22 pp. The entropy reaches a maximum of $88.22\%/56.70\%$ $AUROC/\max\ F_1$ for EMNIST-Det and $73.14\%/7.11\%$ for BDD, which is far from the numbers achieved for the loss and the detection score. PD is not able to detect *drops*, as the bounding boxes of the labels are also the label error proposals itself.

A similar behavior can be observed for the *flips*, where the $AUROC$ values for loss and detection score only differ by a maximum of 1.64 pp. In terms of $\max\ F_1$ the loss outperforms the detection score and entropy in every case. PD performs inferior in terms of $AUROC$ compared to the loss, but in terms of $\max\ F_1$ PD outperforms the loss for BDD based on both backbones trained on noisy data.

For the *shifts*, the detection score and PD have similar performance as the naive baseline in terms of $AUROC$ and all $\max\ F_1$ values are $< 14\%$. Except for BDD trained on noisy data, where entropy performs superior to the loss, loss outperforms all baselines.

For the *spawns*, the detection score performs similar compared to the *shifts*. PD performs well especially in terms of $\max\ F_1$, where PD even outperforms the loss for RestNeSt101 on BDD with noisy training data by 20.16 pp, otherwise loss is superior to PD. In the cases where entropy outperforms loss, the difference is at most 5.44 pp in terms of $AUROC$ and 3.69 pp in terms of $\max\ F_1$.

The detection score can neither reliably detect the *shifts* nor the *spawns*, whereas the entropy cannot detect the *drops* and *flips* well, especially for complicated problems such as BDD. PD cannot reliably detect the *shifts* and is not able to detect *drops* by design. All in all, the loss method is the only one of those presented that can detect all four different types of label errors efficiently.

Different Noise Intensity in Training Table 6 shows mAP, $AUROC$ and $\max\ F_1$ values for different noise intensities for Swin-T on the BDD training dataset. In our experiments, it makes no difference whether the labels of the training data contain 5% or 20% noise, the mAP is between 50.2% and 50.4%, where the model has an mAP of 52.1% due to training on the original training data. All mAP evaluations are based on the test data with original and thus unmodified labels.

On the one hand, the $AUROC/\max\ F_1$ values decrease with increasing noise intensity by $3.69/20.62$ pp for loss, by $2.31/3.88$ pp for entropy and by $3.01/17.94$ pp for PD, respectively. For the detection score, on the other hand, the $AUROC$ value increases by 12.39 pp from 76.82% to 89.21% and the $\max\ F_1$ value increases only marginally by 0.54 pp to 31.68%. Nevertheless, the loss outperforms the detection score/entropy/PD in every case by at least $3.40/21.68/35.71$ pp in terms of $AUROC$ and by at least $4.29/17.64/0.38$ pp in terms of $\max\ F_1$. All $AUROC/\max\ F_1$ evaluations are based on the test data with

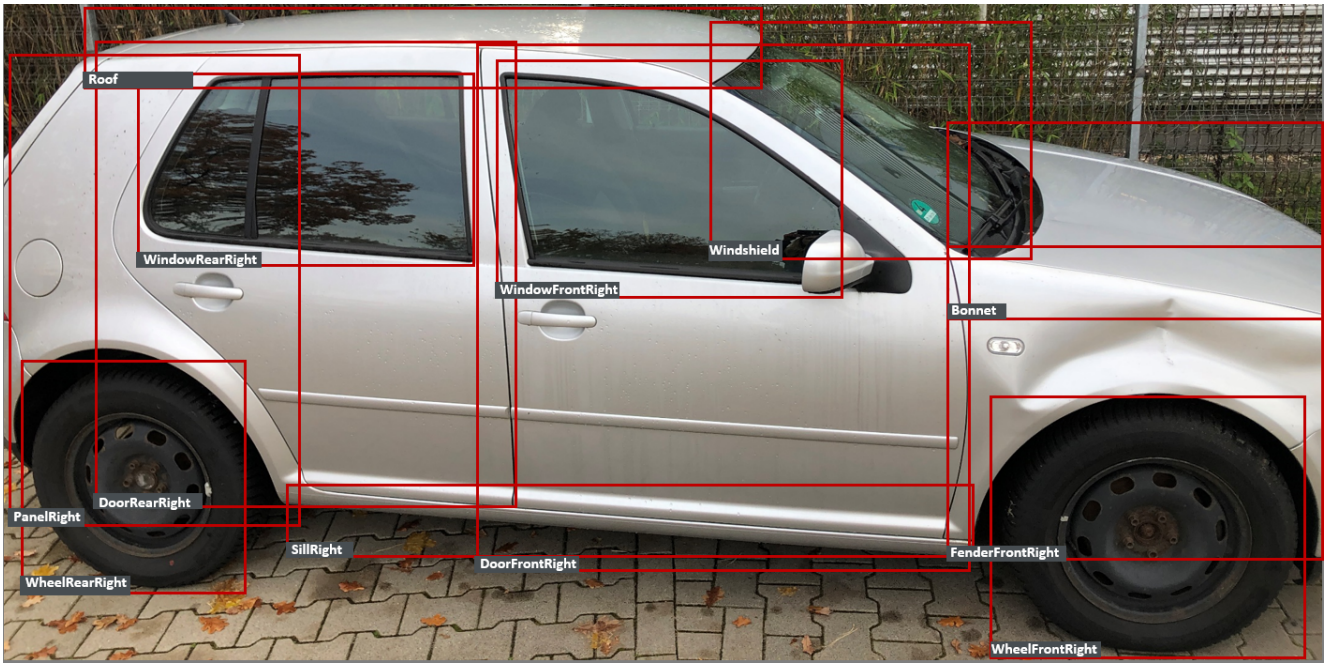


Figure 6. Example image from the CE test data with labels and a missing "MirrorRight".

Dataset	Batch Size	Image Resolution	# Training Iterations	Learning Rate
EMNIST-Det	24	300×300	$24{,}000/48{,}000^{(*)}$	0.02
BDD	4	1333×800	$150{,}000/250{,}000^{(*)}$	0.01
Kitti	6	1000×600	70,000	0.01
COCO	12	1000×600	250,000	0.02
VOC	6	1000×600	70,000	0.02
CE	4	1000×600	200,000	0.01

Table 4. Training hyperparameters for the Swin-T and the ResNeSt101 ($^{(*)}$) backbone.

$\gamma = 0.2$ and thus on the identical label basis as for Tab. 2.

Different Amount of Training Images Table 7 shows mAP, $AUROC$ and $\max\ F_1$ values for different amounts of training images for Swin-T on BDD. Therefore, the subsets with fewer images are always included in the subsets with more images and the identically sized subsets with different noise intensities contain the same images.

The mAP increases the more images are used for training and the less label errors exist in the training data. Here, the model trained on 6,227 and unmodified labels ($\gamma = 0$) has a 0.8 points higher mAP than the model trained on 12,454 images with $\gamma = 0.2$. In this case, after comparing the performances, it is worth to review and improve the underlying labels instead of labeling new images and add them to the training set.

The $AUROC$ values increase as the number of images increases with $\gamma = 0$. With $\gamma = 0.2$, the values for loss and entropy decrease as the number of images increases. The $\max\ F_1$ values decrease independently of γ with increasing number of images for loss and entropy, whereas the values increase for detection score. The decrease in $AUROC$ and $\max\ F_1$ values for loss and entropy could be due to overfitting of the model. For PD, $AUROC$ and $\max\ F_1$ values remain almost constant for the respective datasets. However, the loss always outperforms all baselines in terms of $AUROC$ and $\max\ F_1$. All $AUROC$/$\max\ F_1$ evaluations are based on the test data with $\gamma = 0.2$ and thus on the identical label basis as for Tab. 2.

Real Label Errors Further detected real label errors are presented in Fig. 7. The top row shows examples for BDD, where all found label errors are *flips*, except for the third proposal from the right. This proposals can be interpreted as two label errors. Either the "car" label on the "bus" is wrong (*spawn*) and the bus was forgotten to be labeled (*drop*), or the

Label Error Type	Dataset	Backbone	Train Labels	$AUROC$				max F_1			
				Loss	Score	Entropy	PD	Loss	Score	Entropy	PD
Drop	EMNIST-Det	Swin-T	Original	98.94	**99.12**	88.16	0.00	**94.91**	89.63	56.70	0.00
	EMNIST-Det	Swin-T	Noisy	98.85	**99.19**	88.22	0.00	**93.27**	90.24	48.97	0.00
	EMNIST-Det	ResNeSt101	Original	**99.66**	99.65	78.33	0.00	**93.58**	87.02	32.82	0.00
	EMNIST-Det	ResNeSt101	Noisy	99.91	**99.94**	78.79	0.00	**86.42**	81.03	19.21	0.00
	BDD	Swin-T	Original	94.92	**96.05**	51.48	0.00	41.80	**48.38**	2.37	0.00
	BDD	Swin-T	Noisy	91.72	**93.64**	52.88	0.00	37.93	**46.93**	1.45	0.00
	BDD	ResNeSt101	Original	**94.52**	93.61	73.14	0.00	**45.89**	35.67	7.11	0.00
	BDD	ResNeSt101	Noisy	**91.89**	91.84	62.25	0.00	**26.29**	22.62	1.75	0.00
Flip	EMNIST-Det	Swin-T	Original	99.74	**99.78**	91.09	99.34	**92.89**	90.08	59.51	86.79
	EMNIST-Det	Swin-T	Noisy	99.62	**99.83**	90.79	99.51	**89.42**	88.70	49.32	87.44
	EMNIST-Det	ResNeSt101	Original	99.96	**99.97**	78.95	99.07	**90.77**	86.70	31.65	82.93
	EMNIST-Det	ResNeSt101	Noisy	99.89	**99.94**	78.50	98.83	**81.49**	80.35	18.98	80.49
	BDD	Swin-T	Original	**99.68**	98.36	50.63	98.53	**74.54**	58.79	2.75	73.86
	BDD	Swin-T	Noisy	**99.56**	98.12	50.06	98.32	60.31	58.91	2.13	**71.23**
	BDD	ResNeSt101	Original	**99.80**	98.16	75.93	97.96	**72.81**	54.38	7.12	69.95
	BDD	ResNeSt101	Noisy	**99.31**	97.24	64.34	97.13	44.94	40.15	2.18	**61.75**
Shift	EMNIST-Det	Swin-T	Original	**99.80**	51.52	93.55	40.71	**91.76**	11.14	49.41	10.61
	EMNIST-Det	Swin-T	Noisy	**99.56**	50.26	88.01	50.70	**87.86**	10.92	40.71	10.88
	EMNIST-Det	ResNeSt101	Original	**99.67**	51.28	86.14	45.54	**88.65**	11.28	30.32	10.56
	EMNIST-Det	ResNeSt101	Noisy	**99.30**	53.73	80.52	51.91	**85.97**	13.99	25.65	10.95
	BDD	Swin-T	Original	**65.49**	51.76	61.17	50.24	**16.78**	11.22	14.99	10.55
	BDD	Swin-T	Noisy	57.23	52.57	**57.91**	52.85	12.73	11.44	**12.88**	10.94
	BDD	ResNeSt101	Original	**65.84**	51.51	63.37	54.19	**17.56**	11.40	14.76	11.86
	BDD	ResNeSt101	Noisy	55.92	50.85	**56.18**	52.54	**12.58**	10.87	12.17	11.13
Spawn	EMNIST-Det	Swin-T	Original	**99.37**	75.62	97.92	97.04	**98.87**	19.89	65.08	78.97
	EMNIST-Det	Swin-T	Noisy	**99.68**	50.95	98.48	97.16	**97.77**	19.26	59.18	79.33
	EMNIST-Det	ResNeSt101	Original	**99.84**	57.98	99.40	96.12	**98.06**	18.96	37.84	74.19
	EMNIST-Det	ResNeSt101	Noisy	**99.93**	76.31	99.33	94.89	**94.93**	15.89	35.39	67.92
	BDD	Swin-T	Original	**98.48**	66.33	98.07	92.09	**74.97**	2.23	20.24	50.81
	BDD	Swin-T	Noisy	90.55	78.13	**92.98**	78.00	**17.98**	9.21	11.32	10.94
	BDD	ResNeSt101	Original	95.80	79.79	**97.00**	87.57	**60.38**	5.04	13.75	38.71
	BDD	ResNeSt101	Noisy	90.30	89.19	**95.74**	76.07	7.39	6.92	11.08	**28.55**

Table 5. $AUROC$ and $\max\ F_1$ values for loss, detection score (Score), entropy and PD for all dataset-backbone-training label combinations; higher values are better. Bold numbers indicate the highest $AUROC$ or $\max\ F_1$ per experiment and underlined numbers are the second highest.

γ	# train images	mAP_{50}	$AUROC$				max F_1			
			Loss	Detection Score	Entropy	PD	Loss	Detection Score	Entropy	PD
0	12,454	52.1	**96.30**	76.82	71.73	60.59	**56.59**	31.14	22.21	52.66
0.05	12,454	50.4	**93.44**	88.09	71.76	59.16	**43.36**	30.78	18.54	42.98
0.1	12,454	50.2	**93.21**	89.05	70.98	58.56	**39.36**	30.79	18.53	37.92
0.2	12,454	50.3	**92.61**	89.21	69.42	57.58	**35.97**	31.68	18.33	34.72

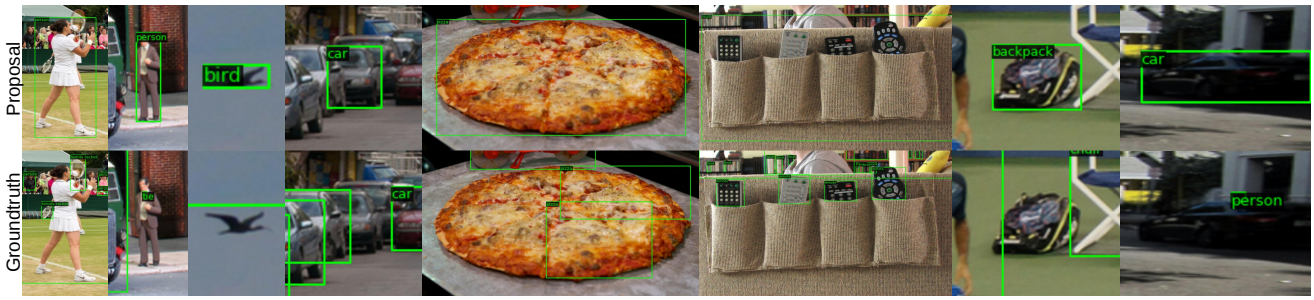
Table 6. Validation of object detection performance and label error detection experiments for different noise for training Swin-T on BDD; higher values are better. Bold numbers indicate the highest $AUROC$ or $\max\ F_1$ per experiment and underlined numbers are the second highest.

localization is inaccurate (*shift*) and the label has a wrongly assigned class (*flip*). The middle and bottom rows represent detected real label errors on COCO and VOC. All proposals show *drops* and at the fourth proposal from the left in the middle row "pizza", the two small labels "pizza" are also count as *spawns* resulting in three label errors for this single proposal.

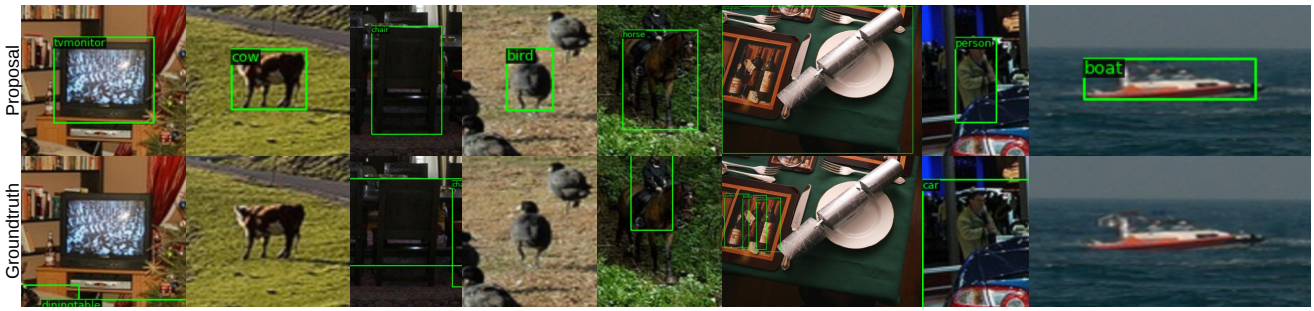
γ	# train images	mAP_{50}	$AUROC$ Loss	$AUROC$ Detection Score	$AUROC$ Entropy	$AUROC$ PD	max F_1 Loss	max F_1 Detection Score	max F_1 Entropy	max F_1 PD
0	1,556	45.1	**94.79**	69.56	<u>72.61</u>	59.86	**58.67**	30.59	25.95	<u>50.77</u>
0	3,113	49.7	**95.25**	<u>73.69</u>	72.70	59.93	**56.48**	31.38	24.64	<u>50.13</u>
0	6,227	51.3	**95.18**	<u>74.95</u>	73.21	60.12	**55.79**	31.55	24.52	<u>49.78</u>
0	12,454	52.1	**96.30**	<u>76.82</u>	71.73	60.59	**56.59**	31.14	22.21	<u>52.66</u>
0.2	1,556	40.7	**94.82**	<u>84.98</u>	73.53	58.73	**44.47**	27.07	18.72	<u>33.87</u>
0.2	3,113	46.9	**93.35**	<u>88.90</u>	70.31	58.98	**35.45**	28.38	18.28	<u>33.45</u>
0.2	6,227	49.1	**93.08**	<u>90.31</u>	70.80	58.37	**33.60**	30.38	18.18	<u>33.43</u>
0.2	12,454	50.3	**92.61**	<u>89.21</u>	69.42	57.58	**35.97**	31.68	18.33	<u>34.72</u>

Table 7. Validation of object detection performance and label error detection experiments for different noise and number of images for training Swin-T on BDD; higher values are better. Bold numbers indicate the highest $AUROC$ or max F_1 per experiment and underlined numbers are the second highest.

(a) BDD



(b) COCO



(c) VOC

Figure 7. Visualization of further detected real label errors in test datasets for BDD (top), COCO (center) and VOC (below).

Theoretical Justification of our Instance-Wise Loss Method Our goal is to show that the flip of a test label is statistically captured by the cross entropy loss evaluated at a deep neural network's (DNN[1]) prediction on a test sample x

[1]Technically, it is not required that the model is a DNN as long as PAC-learnability is fulfilled.

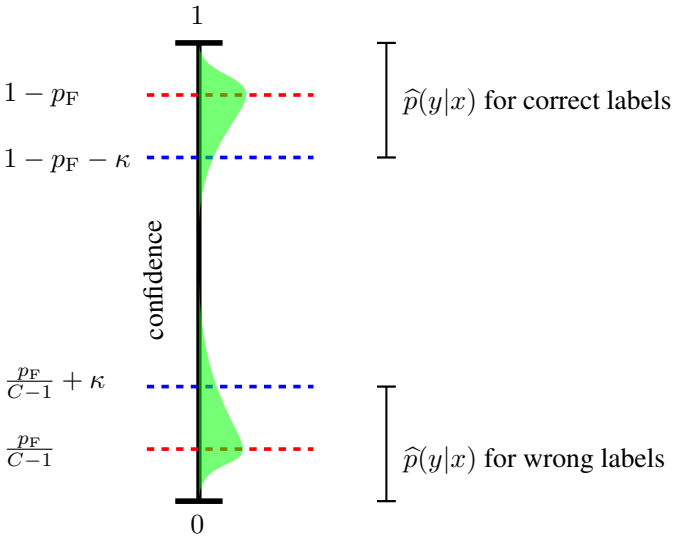


Figure 8. Illustration of the probabilistic statement about predicted confidences conditioned to correct and incorrect given labels. PAC learning leads to concentration of the confidences around $1 - p_F$ and $\frac{p_F}{C-1}$, respectively. The separation on the confidences carries over to the cross entropy loss.

and the corresponding label y.

The rough intuition for this statement is that a probably approximate correct learner [33] (PAC-learner) $\widehat{p}$ has probabilistic bounds for having predictive distribution close to the data-generating distribution p. Therefore, sufficient data sampling and empirical loss minimization will lead to statistical concentration of confidences $\widehat{p}$ around p. If p does not suffer from too strong label noise, we obtain separation between confidences on incorrect and confidences on correct labels. This separation then carries over to the negative log-likelihood (i.e., cross entropy) loss by monotony.

We assume data points $(x, y) \sim p$ following some noisy data generating distribution p, where $x \sim P_x$ follows a marginal distribution P_x. In practice, training and test data originate from the same data pool and we do not see any reason to assume that they follow different labeling procedures. However, it is sufficient to require that for testing data (x, y), x follows the same marginal distribution $x \sim P_x$. Our proof builds on the existence of a true labeling function $f : x \mapsto y$ and the assumption that the data distribution p introduces stochastic flips of labels that occur with a fixed uniform rate $p_F \in [0, 1)$. This flip probability p_F uniformly distributed over all $C - 1$ incorrect classes which are not $f(x)$ leads to the following constraints on p when conditioned to x:

$$p(f(x)|x) := 1 - p_F, \quad p(y|x) := p_F/(C - 1) \tag{2}$$

$\forall y \neq f(x)$. A statistical model $\widehat{p} = \widehat{p}(y|x)$ PAC-learns classification on samples of the (noisy) data generating distribution $p = p(y|x)$. In the present treatment, we assume PAC-learning with respect to the Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}(p(\cdot|x)\|\widehat{p}(\cdot|x)) = -\int \log\left(\frac{\mathrm{d}\widehat{p}(y|x)}{\mathrm{d}p(y|x)}\right) p(y|x)\,\mathrm{d}y$. In the following our goal is to show probabilistic statements about the cross entropy loss

$$\ell_{\mathrm{CE}}(\widehat{p}(x)\|y) := -\sum_{c=1}^{C} y_c \cdot \log(\widehat{p}_c(x)) \tag{3}$$

on test data pairs (x, y). We show that the loss is above a certain threshold if an incorrect label is given and below some threshold in case of a correct, non-flipped label. Non-overlapping intervals indicate that the statistical separation between losses given correct and false labels seen in our experiments can be explained theoretically.

We assume PAC-learnability for the proof. This assumption can be justified via the error decomposition of empirical risk minimization for the KL divergence over the hypothesis space $\mathscr{H}$ with training data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$:

$$\begin{aligned}
&\mathsf{D}(p\|\widehat{p}) := \mathbb{E}_{x\sim p_x}[D_{\mathrm{KL}}(p(\cdot|x)\|\widehat{p}(\cdot|x))] \\
&\leq \inf_{h\in\mathscr{H}} \mathsf{D}(p\|h) \\
&+ \left(\frac{1}{n}\sum_{j=1}^{n} \ell_{\mathrm{CE}}(\widehat{p}(x_j)\|y_j) - \inf_{h\in\mathscr{H}} \ell_{\mathrm{CE}}(h(x_j)\|y_j)\right) \\
&+ 2 \cdot \sup_{h\in\mathscr{H}} \left|\mathsf{D}(p\|h) - \frac{1}{n}\sum_{j=1}^{n} \ell_{\mathrm{CE}}(h(x_j)\|y_j) - H(p(\cdot|x))\right| \\
&< \varepsilon
\end{aligned} \tag{4}$$

where $H = -\sum_{c=1}^{C} p(c|x) \cdot \log(p(c|x))$ is the entropy of the data generating distribution[2]. The first term is the model misspecification error given by $\mathscr{H}$. In practice, we assume an expressive DNN with a large amount of capacity (appealing to universal approximation) which allows for this error to be negligible. In particular, in this case, no restrictions need to be made in the choice of $\mathscr{H}$. The second term measures the error of the learning algorithm w.r.t. an empirical risk minimizer h. Similarly to the term, an expressive DNN trained to convergence leads to small contributions by this term. Lastly, the third term is the sampling error made as compared to the loss $\mathsf{D}(p\|h)$ in the true distribution. The third term can be controlled by application of concentration inequalities and chaining under certain assumptions (see [36]) which is why the sum of the three terms can be made smaller than some fixed $\varepsilon > 0$ given sufficient amount of data.

[2]Together with the cross entropy ℓ_{CE}, the entropy H yields an unbiased risk function for D_{KL}.

Proposition 1 (Statistical Separation of the Cross Entropy Loss)**.** *Let training and testing labels be given under a stochastic flip in $p(\cdot|x)$ with probability p_{F} as above, let the label distribution $p(\cdot|x)$ be PAC-learnable by the hypothesis space of $\widehat{p}(\cdot|x)$ w.r.t. D_{KL} (to precision ε and confidence $1-\delta$) and let $\kappa > 0$. If $p_{\mathrm{F}} < \frac{C-1}{C}(1-2\kappa)$, we obtain strict separation of the loss function*

$$\begin{aligned}\ell_{\mathrm{CE}}(\widehat{p}(x)\|f(x)) &< -\log(1-p_{\mathrm{F}}-\kappa)\\ &< -\log(\kappa+\tfrac{p_{\mathrm{F}}}{C-1}) < \ell_{\mathrm{CE}}(\widehat{p}(x)\|\widetilde{y})\end{aligned} \tag{5}$$

for any incorrect label $\widetilde{y} \neq f(x)$ with probability $1-\delta$ over chosen training data and with probability $1-\frac{2\varepsilon}{\kappa^2}$ over the choice of x.

Proof. We aim at bounding $\max_{y=1,\dots,C}|p(y|x)-\widehat{p}(y|x)|$ by the total variation distance. PAC-learnability asserts that given enough data, the $\widehat{p}$-distributions illustrated in Fig. 8 are concentrated around $1-p_{\mathrm{F}}$ for true labels and $\frac{p_{\mathrm{F}}}{C-1}$ for incorrect labels. In particular, PAC-learnability implies

$$\mathbb{E}_{x\sim p_x}[D_{\mathrm{KL}}(p(\cdot|x)\|\widehat{p}(\cdot|x))] < \varepsilon \tag{6}$$

with probability $1-\delta$ over the choice of training data. Let $\kappa > 0$. From this PAC result, we derive bounds for the probability of $\max_{y=1,\dots,C}|p(y|x)-\widehat{p}(y|x)|$ exceeding κ via the total variation distance. We have

$$\begin{aligned}&P_x(\|p(\cdot|x)-\widehat{p}(\cdot|x)\|_{\mathrm{TV}} \geq \kappa)\\ &\leq P_x(\sqrt{2D_{\mathrm{KL}}(p(\cdot|x)\|\widehat{p}(\cdot|x))} \geq \kappa)\\ &\leq P_x\left(D_{\mathrm{KL}}(p(\cdot|x)\|\widehat{p}(\cdot|x)) \geq \frac{\kappa^2}{2}\right)\\ &\leq \frac{2}{\kappa^2}\mathbb{E}_{x\sim p_x}[D_{\mathrm{KL}}(p(\cdot|x)\|\widehat{p}(\cdot|x))] < \frac{2\varepsilon}{\kappa^2}\end{aligned} \tag{7}$$

with probability $1-\delta$ over the choice of training data. Here, the first inequality is the application of Pinsker's inequality and the third due to the Markov inequality.

Assume that we are given a correct label y for x, then with probability $1-\delta$ over training data and with probability $1-\frac{2\varepsilon}{\kappa^2}$ over sampling x, we have that

$$\begin{aligned}|p(y|x)-\widehat{p}(y|x)| &= |(1-p_{\mathrm{F}})-\widehat{p}(y|x)|\\ &\leq \max_y |p(y|x)-\widehat{p}(y|x)|\\ &\leq \|p(\cdot|x)-\widehat{p}(\cdot|x)\|_{\mathrm{TV}} < \kappa.\end{aligned} \tag{8}$$

This implies $\widehat{p}(y|x) > 1-p_{\mathrm{F}}-\kappa$ and therefore, by monotony of the logarithm $\ell_{\mathrm{CE}}(\widehat{p}(y|x)\|y) < -\log(1-p_{\mathrm{F}}-\kappa)$. Similarly, if y is any incorrect label, we have the probabilistic statement

$$\begin{aligned}|p(y|x)-\widehat{p}(y|x)| &= \left|\frac{p_{\mathrm{F}}}{C-1}-\widehat{p}(y|x)\right|\\ &\leq \max_y |p(y|x)-\widehat{p}(y|x)| < \kappa,\end{aligned} \tag{9}$$

i.e., $\widehat{p}(y|x) < \kappa + \frac{p_{\mathrm{F}}}{C-1}$ and we have $\ell_{\mathrm{CE}}(\widehat{p}(x)\|y) > -\log\left(\kappa+\frac{p_{\mathrm{F}}}{C-1}\right)$. Finally, we obtain separability of losses with true versus false labels in probability if

$$1-p_{\mathrm{F}}-\kappa > \kappa+\frac{p_{\mathrm{F}}}{C-1} \iff p_{\mathrm{F}} < \frac{C-1}{C}(1-2\kappa). \tag{10}$$

□