

How Much Noise is there in Labels Generated by Humans? A Method to Validate Automatically Generated Bounding Boxes.

Mariusz Karol Nowak¹ Jacek Cyranka¹ Natalia Maślany^{1,3} Aleksander Kostuch^{1,2}
Jakub Derbisz¹ Mateusz Komorkiewicz^{1,2} Patryk Siwek^{1,2} Mateusz Jan Wójcik¹
Dariusz Marchewka^{1,2} Paweł Skruch^{1,2}

¹Aptiv, ²AGH University of Krakow, ³Jagiellonian University

Abstract

In order to train a model or evaluate its safety, high quality labels are necessary. Human labeling is considered gold standard in object detection and object classification problems. This approach is natural - humans do very well in finding cars or pedestrians in an image. However the answers to the same question, provided by different human experts, or even the same expert asked multiple times tend not to be completely identical. In this paper we show better performance of neural networks over humans in 2D object detection tasks by showing neural network labels are closer to human consensus than any particular human labeler. The method we present here may be used to validate labels generated using automated labeling methods, thereby decreasing the need for costly human labeling. For this task we created a dataset of 630 automotive images labeled by 10 different labelers each. Additionally we compare predictions of humans and networks given only single camera images to more accurate labels created using multiple sensors and sequences of images (from Waymo and nuImages datasets). Using the second method we again show better performance of the networks.

1. Introduction

In order to train a model or evaluate its safety, high quality labels are necessary. Human labeling is considered gold standard in object detection and object classification problems, for both general purpose (e.g., COCO [14], ImageNet [5]) and automotive (e.g., KITTI [7], Waymo [23], NuScenes [2]) datasets. This approach is natural – after all, humans do very well in finding cars or pedestrians in an image. However, the answers to the same question, provided by different human experts, or even the same expert asked multiple times tend not to be completely identical. Famous examples of noisy judgements in social sciences include widely variable crime sentences [24] or insurance claim assessments [12]. Kahneman [12] popularized the

idea of noise audits – estimating what is the distribution of answers to a specific question, rather than relying on a single point estimate.

In this paper, we propose to apply a similar noise audit to human generated 2D bounding boxes for objects detected in an image. Specifically, we want to check how the predictions made by object detection neural networks compare to the whole distribution of human-generated labels. In order to test our approach, we created a dataset consisting of 630 images labeled multiple times (using base images from KITTI, Waymo and nuImages). Each of the images was labeled by 10 distinct labelers, tasked with generating 2D bounding boxes for cars, pedestrians, cyclists and other vehicles. Output of each labeler was subsequently reviewed and improved by a different labeler. We used this dataset to assess how good are 15 versions of leading neural networks (11 with vanilla weights and 4 after transfer learning) at the object detection task when compared to human labelers, taking into account the differences in judgement between human labelers. Additionally we calculate classical KPIs comparing labels from humans and neural networks to official labels from authors of Waymo and nuImages. Our method is graphically represented in Figure 1.

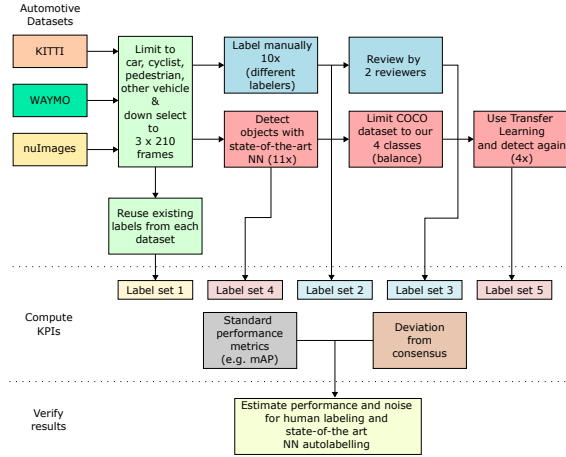
Contributions:

1. We introduced a novel method to assess the quality of bounding box annotations in the presence of multiple labelers based on the deviation from labeler consensus.
2. We created a dataset of 630 images, each labeled with 2D bounding boxes by 10 human labelers.
3. We performed comprehensive comparison of the quality of human generated and automatically generated labels using both our method and classical KPIs.

2. Related Work

Since 2015, neural networks are known to outperform humans in object classification tasks, with the seminal work of He et al. [9] showing above human performance on ImageNet [5]. Typically, in such comparisons, the neural network is trained on the train part of the dataset, and later

Figure 1. Diagram showing the idea of methodology implemented in the paper.



inference is performed on the validation part of the dataset. Then, a human labeler is asked to perform the classification task on the same images in the validation set. Finally, one compares the prediction accuracy between the neural network and the human labeler. Due to high cost of human labeling, such comparisons are done on relatively small datasets. For example, Russakovsky et al. [20] used 1.5k images to estimate human-level performance on ImageNet [5] object classification task.

As the neural networks surpassed the human performance benchmark, a natural question arose - is it the model making a mistake, or did the human labeler make a mistake when classifying an object. One reason for bad labels, may be lack of skill on the part of labelers. For example, Van Horn et al. [25] found that at least 4% of birds in ImageNet dataset are misclassified. Other papers exploring mistaken labels in ImageNet in a more general setting include Northcut et al. [17] and Lee et al. [13].

More recently, Vasudevan et al. [26] decided to explore whether it is the model making wrong predictions, or did the labelers misclassify examples. As a starting point they chose the ViT model [6], pretrained on JFT-3B [22] and finetuned on ImageNet-1K [5]. They manually reviewed each mistake made by the model by in the imagenet2012 multilabel dataset [21]. The model made 676 mistakes on a 20k image validation set. After manual review in 44% of the initial mistakes cases model was deemed to be correct in its prediction.

The problem of object detection (i.e., finding the smallest rectangular bounding boxes containing objects of particular class) is much higher dimensional and harder. While for object classification there is one major form of labeling error (image can be classified as incorrect class), object detection can fail in many more ways. A non exhaustive list of object

detection failure modalities include:

1. object classified as incorrect class,
2. false negative – labeler did not mark an existing object,
3. false positive – labeler marked a non-existing object,
4. inexact boundaries of a bounding box.

In case of the first three failure modalities, one can agree in most cases on a pretty clear definition of error. When it comes to the inexact boundaries of a bounding box, defining error is harder – multiple labelers will almost surely mark the boundaries of bounding boxes in slightly different positions.

Nassar et al. [16] tackle the object detection problem by using Krippendorff’s alpha coefficient, which mitigates this issue. Krippendorff’s alpha coefficient, is a statistical method of assessing the level of disagreement between different annotators on the dataset. The main difference between our method (defined in the Algorithm 1) and the methods relying on Krippendorff’s alpha coefficient is that the calculation of Krippendorff’s alpha requires estimating the expected disagreement if the labels were produced by chance. Our method does not require this step and consequently does not require somewhat arbitrary estimation of the probability distribution from which labels are sampled. We believe it to be a major strong point of our method, as estimating probability distribution of bounding box labels is highly nontrivial. For example, Nassar et al. [16] assume that a single pixel can be part of at most 1 bounding box of the same class for the purpose of estimating random distribution of labels. Experimentally, this is not a correct assumption, as bounding boxes can overlap (e.g. bounding boxes of cars in a traffic jam).

3. Dataset

In our experiments we aimed at a fair comparison between human labelers and neural networks. Hence, human labelers were given only images from a single camera without data from any other sensors, such as lidar or additional cameras. The resolution of camera images given to humans was the same as the resolution of images used by the networks. Since the chosen neural networks do not support sequential inputs, the frames selected for annotation are not sequential either - human labelers could not infer based on sequential data. While annotating the data, neither pre-labeling with a baseline neural network nor any metadata were used. Annotations and additional information about the labeling process are provided in the supplementary material.

3.1. Choice of Images

To prepare the dataset, we used the leading open data sets of automotive images with annotations: KITTI [7], Waymo [23], and nuImages [2]. We used images from a single, front-facing camera for each dataset. For Waymo and nuImages, we used the designated validation sets; for KITTI,

Algorithm 1 Calculating the Deviation from Consensus for a Single Labeler a (for a Given Class)

Require: \mathcal{S}, s_a

```
1: for each  $\mathcal{S}^j \in \mathcal{S}$  do
2:   for each  $s_k^j \in \mathcal{S}^j$  do
3:     if  $k \neq a$  then
4:       Match the bounding boxes in  $s_a^j$  and the bounding boxes in  $s_k^j$ 
5:       Save the mean of the IOU scores between the bounding boxes as  $IOU_{ak}^j$ 
6:     end if
7:   end for
8:    $IOU_a^j \leftarrow \sum_{k \neq a} \frac{IOU_{ak}^j}{n-1}$  ▷ Calculate the mean of the IOU scores in the scene  $j$ 
9:    $SCORE_a^j \leftarrow 1 - IOU_a^j$  ▷ Define the deviation of  $a$  from consensus as  $1 - IOU_a^j$ 
10: end for
11:  $SCORE_a \leftarrow \frac{\sum_{j=1}^m SCORE_a^j |s_a^j|}{\sum_{j=1}^m |s_a^j|}$  ▷ Calculate the mean score over all images, weighted by the number of true objects in the image
```

we used the test set as KITTI has no designated validation set. From each of the datasets, we took 210 images. We wanted to ensure that our dataset was diverse, so in each dataset, we picked 30 frames per pre-defined object count interval: three intervals for vehicles, three for pedestrians, and one for cyclists. We used only a single interval for cyclists because this class is significantly less represented in the datasets. The object count intervals for vehicles and pedestrians are 1-4, 5-9, and 10+. We decided to label 4 classes of objects - cars, pedestrians, cyclists, and other vehicles. We intend to publish the labels we generated for Waymo and nuImages, while we do not intend to publish the labels for KITTI, as they were generated for the test set.

3.2. Labeling Process

The manual labeling process was commissioned at an AGH University of Krakow labeling lab with 8 years of experience in labeling images, lidar point clouds, and radars point clouds for academic and commercial projects. The labelers were asked to generate 2D bounding boxes for four selected classes: *Car*, *Pedestrian*, *Cyclist* and *Other Vehicle*. We provided the labelers with a detailed labeling instruction and examples of objects from particular classes. We used the same 10 labelers to prepare 10 distinct sets of labels for each of the images. The labeling was done in 2 stages - preliminary labeling and review. During the preliminary stage, labeler marked the bounding boxes of objects of relevant classess. During the review process, a different labeler was tasked with correcting the preliminary labels. We present results for labels both before and after review.

4. Methodology

4.1. Overview

Broadly speaking, our method requires three steps:

1. Manually labeling the same image by multiple human labelers.
2. Calculating the Key Performance Indicators (KPI) on

how much the labels generated by each human labeler differ from the labels generated by other human labelers.

3. Performing the inference using a neural network and calculating how the quality of automatically-generated labels compares to the quality of human-generated labels. The manual labeling process is described in more detail in Section 3.2, the method for calculating how much the labels differ from each other is described in Section 4.2, and the neural networks we have used for inference are described in Section 4.3.

4.2. KPI - How Much the Labels Differ from Each Other

Suppose we ask n labelers to find a list of 2D bounding boxes of objects of a particular class in m images. Then, for $n \geq 3$ we can assess how good is a specific labeler by comparing his labels against the labels provided by all the other labelers.

Let \mathcal{S} be the set of sets of bounding boxes, representing an object of a particular class generated by each labeler. Let $\mathcal{S}^j \in \mathcal{S}$ be the set of sets of bounding boxes generated by each labeler for a particular image j . Let $s_k \in \mathcal{S}$ be the set of bounding boxes generated by the labeler k . Similarly, let $s_k^j \in \mathcal{S}^j$ be the set of all the bounding boxes generated by the labeler k for image j . Moreover, we follow a convention where by $|s_k^j|$ we denote the true number of objects of particular class in the image corresponding to s_k^j . For the purpose of calculations, we estimate the true number of objects as the penultimate order statistic of the number of objects detected by human labelers (at least 2 human labelers must mark at least this number of objects). Then, we can assess the quality of work of a single labeler (let's call him a) using Algorithm 1.

Algorithm 1 is defined in such a way, that we assess the quality of the work of specific labeler a , by treating the labels provided by all the other labelers as "Ground Truth". Therefore, we can very easily check whether a particular

Table 1. Scores for individual human labelers. The best performers are in bold.

Labeler ID	Pedestrian before review	Pedestrian after review	Car before review	Car after review	Cyclist before review	Cyclist after review	Other Vehicle before review	Other Vehicle after review
1	0.305	0.268	0.264	0.212	0.322	0.235	0.396	0.322
2	0.326	0.290	0.228	0.236	0.434	0.315	0.426	0.332
3	0.288	0.268	0.229	0.221	0.372	0.264	0.390	0.310
4	0.329	0.299	0.263	0.234	0.326	0.233	0.394	0.305
5	0.334	0.285	0.286	0.242	0.324	0.241	0.396	0.317
6	0.254	0.243	0.177	0.186	0.347	0.239	0.406	0.333
7	0.292	0.255	0.258	0.227	0.357	0.251	0.424	0.325
8	0.314	0.267	0.259	0.230	0.413	0.304	0.435	0.354
9	0.398	0.295	0.385	0.227	0.321	0.239	0.423	0.302
10	0.351	0.319	0.294	0.268	0.331	0.244	0.407	0.334
Mean	0.319	0.279	0.264	0.228	0.355	0.257	0.410	0.323

neural network provides predictions better or worse than the labeler a . We can simply run the algorithm using the neural network predictions as input and compare them against the $(n - 1)$ labelers other than a .

Please note, that the score is defined for a single labeler or a comparison between a single labeler and a neural network. Given that in the experimental part of the paper (Section 5) we present the results for 10 labelers and 15 versions of neural networks, we defined the mean version (lower is better) and normalized version (higher is better). The mean score for the human labelers is simply the mean of scores of all labelers. The mean score for a particular network is the mean of comparison scores against all the labelers. The normalized version of the score is defined for the neural networks as the fraction of human labelers who perform worse than the network.

4.3. Neural Networks

4.3.1. Choice of the Confidence Threshold

There is a qualitative difference between the labels generated by humans and the output of a neural network. When a human labeler is tasked with generating rectangular bounding box labels for objects of a particular class, (s)he simply generates a list of labels. On the other hand, a neural network outputs a list of bounding boxes, together with confidences that an object of particular class is within the bounding box. Therefore, a somewhat arbitrary decision has to be made regarding the minimal confidence that is necessary to output a prediction. We present the confidence thresholds for each of the networks in table 2, they were chosen to optimize our metric. We used the same confidence threshold for all classes.

4.3.2. Inference Setup and Class Mapping

We employed 15 neural networks to detect objects in the images from *KITTI*, *Waymo*, and *Nulmages* datasets. The target classes included *Pedestrians*, *Cars*, *Cyclists*, and *Other*

Table 2. Confidence thresholds and training datasets for each network.

Network	Threshold	Training Dataset
YOLOv8x	0.7	COCO
YOLOv9e	0.7	COCO
YOLOv10x	0.7	COCO
YOLOv11x	0.7	COCO
Co-DETR Swin-L	0.7	Objects365 + COCO
Co-DETR ViT-L	0.7	Objects365 + COCO
Detectron2	0.95	COCO
DDQ Detr 4	0.6	COCO
DDQ Detr 5	0.6	COCO
YOLOv8x TL	0.7	COCO (modified)
YOLOv9e TL	0.7	COCO (modified)
YOLOv10x TL	0.7	COCO (modified)
YOLOv11x TL	0.7	COCO (modified)
EPro-PNP (nuScenes)	0.5	nuScenes
DD3D (KITTI)	0.7	KITTI

Vehicles. For the models trained on COCO dataset or pre-trained on Objects365 and trained on COCO we mapped the *person* label to the *Pedestrian* class and retained the *car* label directly. Classes such as *motorcycle*, *bus*, *train*, and *truck* were grouped under *Other Vehicles*. The *Cyclist* class was synthetically created by merging the *person* and *bicycle* labels using a 0.2 Intersection over Union (*IoU*) threshold.

YOLO Networks We employed YOLOv8x, YOLOv9e, YOLOv10x, and YOLOv11x, the largest models from each generation [1, 10, 11, 27]. These models were pretrained on the COCO dataset. Minimal preprocessing was applied to the input images to ensure compatibility with YOLO family networks.

Co-DETR Our Co-Detr models [30] were pretrained on Objects365 and trained further on COCO. We have selected Co-Detr with ViT-L backbone (65.9 box AP) [30] and a ver-

sion with Swin-L backbone (64.1 box AP) [4].

DDQ We used two versions of Dense Distinct Queries models [29]: 5 scale with ResNet-50 backbone (12 epochs training on COCO, 52.1 box AP), and 4 scale with Swin-L backbone (30 epochs training on COCO, 58.7 box AP) [4].

EPro-PNP-PnP-Det_v2 We have used a model [3] that is based on ResNet-101 as a backbone. It was trained for 12 epochs on nuScenes dataset [2] and was able to achieve box AP of 42.3 in official nuScenes benchmark.

Detectron 2 We have used the Detectron 2 network [28] providing a generic detection and segmentation algorithm. We picked the Mask R-CNN checkpoint, which has the largest box AP (48.9) on COCO. It was trained using a longer training schedule and large-scale jitter [8].

DD3D For another reference, we have used the DD3D network [19] with the V2-99 backbone. We used weights pre-trained on half of the KITTI training dataset (exactly on 3712 samples) published by the authors. For nuImages and Waymo inference, the input resolution has been adjusted to more closely match the shape of the original data.

4.4. Transfer learning

To synchronize the model’s output space with human-defined classes and reduce discrepancies between machine and human labeling, we implemented transfer learning [18] using YOLO-family networks (v8x, v9e, v10x, and v11x). Unlike typical transfer learning, where the model is fine-tuned on new data, our experiment focused on re-adjusting the model’s output space utilizing the same dataset that was used for pretraining.

Using the FiftyOne framework [15] and COCO Train2017 dataset (on which YOLO models were trained), we selected a subset of 71k images that contained only instances of objects corresponding to the categories relevant to our study: *person*, *truck*, *bicycle*, *motorcycle*, and *car*. This selection focused specifically on images associated with our final target classes (*Pedestrian*, *Vehicle*, *Other Vehicle*, and *Cyclist*) to ensure relevance.

To align the model’s output with the human-defined classes in our experimental dataset, we mapped the COCO categories to our target categories as described in 4.3.2. This mapping ensured that the output classes of the YOLO models corresponded directly to the class definitions provided to human labelers. This step was crucial for reducing the semantic gap in class definitions and to enable comparison between human labelers directly with machine learning models without further output post-processing.

We fine-tuned each YOLO model variant over 50 epochs, using default training hyperparameters and without freezing any layers. This approach allowed the model to adjust fully to our target classes. For validation, we used the COCO Valid2017 dataset. We sampled images and mapped classess according to the strategy described earlier.

The fine-tuned models achieved the following mAP@0.5: YOLOv8x 0.767, YOLOv9e 0.749, YOLOv10x 0.762, YOLOv11x 0.769. These mAP scores indicate that the fine-tuning process successfully aligned the models with our adjusted class mappings, achieving high performance on the selected COCO validation subset.

5. Experiments

5.1. Design of Experiments

We performed three kinds of experiments comparing the performance of neural networks and human labelers:

1. Experiments with vanilla object detection networks, using literature weights.
2. Experiments with vanilla object detection networks, using transfer learning to decrease the number of predicted classess to better suit our task (finetuning process is described in Section 4.4).
3. Experiments using object detection neural networks trained specifically in the automotive domain - we used the networks from KITTI and NuScenes leaderboards (Waymo forbids publication of neural networks trained on their dataset).

For the experiments using networks trained on KITTI and nuScenes, we used the top performing, open source, vision-only networks from the respective leaderboards.

5.2. Results for Combined Dataset

In the Table 1 we present the scores for each of the labelers, as well as the mean scores of all the labelers. As the reader can see, the review process increases the quality of labels, however the final labels still differ widely among the labelers even after the review. Another important fact to note is that while there is variation in the quality of output of different labelers, it would be hard to point to the best performing labeler, with different people performing best for different object classess.

In Tables 3 and 4 we present the scores of the neural networks and compare them to human labelers. The neural networks outperform humans for the car and pedestrian classes, while underperforming for cyclists and other vehicles. Co-DETR ViT-L network is best at approximating the consensus of human labelers, which is consistent with the same network achieving best results using classical KPIs (Tables 5, 6).

When comparing YOLO family networks before and after transfer learning we can notice significant improvement for the cyclist class, moderate improvement for cars and pedestrians, and slight deterioration for the other vehicle class.

Somewhat surprisingly, the networks trained on automotive datasets (epro trained on Nuscenes and DD3D trained on KITTI) performed the worst among neural networks and worse than human labelers. It strongly suggests that for la-

Table 3. Labels before review. Using (norm) instead of (normalized) to fit the table on the page. The best performers are in bold. If no network is better than 50% of labelers, then we mark human labelers as the best in normalized score.

Source of Labels	Car (mean)	Car (norm)	Pedestrian (mean)	Pedestrian (norm)	Cyclist (mean)	Cyclist (norm)	Other Vehicle (mean)	Other Vehicle (norm)
Human Labelers	0.264	N/A	0.319	N/A	0.355	N/A	0.410	N/A
YOLOv8x	0.211	0.9	0.274	0.9	0.694	0.0	0.510	0.0
YOLOv9e	0.208	0.9	0.271	0.9	0.710	0.0	0.532	0.0
YOLOv10x	0.228	0.9	0.292	0.8	0.698	0.0	0.571	0.0
YOLOv11x	0.234	0.8	0.307	0.6	0.720	0.0	0.587	0.0
Detectron2	0.231	0.9	0.301	0.7	0.664	0.0	0.543	0.0
Co-DETR Swin-L	0.176	1.0	0.261	0.9	0.625	0.0	0.517	0.0
Co-DETR ViT-L	0.169	1.0	0.253	1.0	0.652	0.0	0.516	0.0
DDQ Detr 4	0.213	0.9	0.282	0.9	0.620	0.0	0.564	0.0
DDQ Detr 5	0.195	0.9	0.304	0.6	0.763	0.0	0.636	0.0
YOLOv8x TL	0.196	0.9	0.270	0.9	0.519	0.0	0.513	0.0
YOLOv9e TL	0.197	0.9	0.264	0.9	0.571	0.0	0.541	0.0
YOLOv10x TL	0.225	0.9	0.272	0.9	0.534	0.0	0.529	0.0
YOLOv11x TL	0.217	0.9	0.272	0.9	0.542	0.0	0.536	0.0
EPro-PNP (NuScenes)	0.532	0.0	0.705	0.0	0.950	0.0	0.659	0.0
DD3D (KITTI)	0.291	0.2	0.520	0.0	0.724	0.0	0.812	0.0

Table 4. Labels after review. Using (norm) instead of (normalized) to fit the table on the page. The best performers are in bold. If no network is better than 50% of labelers, then we mark human labelers as the best in normalized score.

Source of Labels	Car (mean)	Car (norm)	Pedestrian (mean)	Pedestrian (norm)	Cyclist (mean)	Cyclist (norm)	Other Vehicle (mean)	Other Vehicle (norm)
Human Labelers	0.228	N/A	0.279	N/A	0.257	N/A	0.323	N/A
YOLOv8x	0.208	0.9	0.264	0.8	0.644	0.0	0.477	0.0
YOLOv9e	0.206	0.9	0.261	0.8	0.662	0.0	0.499	0.0
YOLOv10x	0.228	0.5	0.282	0.5	0.655	0.0	0.545	0.0
YOLOv11x	0.234	0.4	0.299	0.1	0.679	0.0	0.560	0.0
Detectron2	0.227	0.6	0.293	0.3	0.610	0.0	0.512	0.0
Co-DETR Swin-L	0.171	1.0	0.251	0.9	0.564	0.0	0.481	0.0
Co-DETR ViT-L	0.163	1.0	0.241	1.0	0.604	0.0	0.477	0.0
DDQ Detr 4	0.210	0.9	0.270	0.7	0.568	0.0	0.527	0.0
DDQ Detr 5	0.192	0.9	0.295	0.2	0.727	0.0	0.606	0.0
YOLOv8x TL	0.193	0.9	0.260	0.8	0.443	0.0	0.484	0.0
YOLOv9e TL	0.194	0.9	0.253	0.8	0.501	0.0	0.510	0.0
YOLOv10x TL	0.224	0.6	0.262	0.8	0.464	0.0	0.501	0.0
YOLOv11x TL	0.216	0.8	0.264	0.8	0.468	0.0	0.506	0.0
EPro-PNP (NuScenes)	0.525	0.0	0.711	0.0	0.941	0.0	0.634	0.0
DD3D (KITTI)	0.283	0.0	0.511	0.0	0.684	0.0	0.801	0.0

being tasks on new datasets general purpose object detection networks are a better choice than models trained solely on much smaller, task specific datasets. In the supplementary material we publish the results of different networks on each of the subdatasets we used (KITTI, Waymo and nuImages).

6. Comparison to Official Labels

To provide a comprehensive comparison of the performance of human labelers (after review) versus neural networks in automotive object detection, we calculated sev-

eral standard key performance indicators (KPIs), including mAP, micro-F1, macro-F1, and the counts of true positives (TP), false positives (FP), and false negatives (FN). mAp is calculated as in COCO, i.e., @[IoU=0.50:0.95, area=all, maxDets=100]. Other metrics used 0.2 IoU threshold. These metrics were computed with respect to the original labels provided by the respective dataset creators. It is important to note that the original 2D object detection labels were generated on a best-effort basis, using available lidar and object tracking data in addition to the images themselves. Therefore, we treat these labels as the ground truth (GT) for our experiments.

We present the general KPI metrics for the data sampled from the Waymo and nuImages in Tables 5 and 6. We mapped original nuImages classes to ours (Car, Cyclist, Pedestrian, Other Vehicle). For Waymo, since it has only three 2D detection classes (Vehicle, Cyclist, Pedestrian) we applied appropriate mapping of our classes.

A key consideration when interpreting these results is the selection of detection thresholds for the neural networks. Since our goal was to evaluate the use of neural networks for auto-labeling unseen data, we did not tune the confidence thresholds for individual classes or networks. Instead, we used the default thresholds found in the code samples in official repositories of the respective networks: 0.25 for all YOLO models, 0.3 for Co-Detr, DDQ, and EPro-PNP, 0.4 for DD3D, and 0.5 for Detectron2, applied uniformly across all classes.

Several key insights emerge from these results. First, the top-performing neural networks (for the chosen threshold values) generally outperform human labelers, as evidenced by higher F1 scores. The performance gap is particularly noticeable in the number of false negatives, with neural networks leaving fewer GT objects unlabeled compared to human labelers. However, human labelers exhibit a significantly lower number of false positives, by an order of magnitude. Second, the human labelers show considerable variability in performance, as reflected by the variance in their individual F1 scores. This variability could stem from various factors, including differences in experience, tiredness or interpretation of the labeling guidelines.

We also observe that while higher inference thresholds resemble human consensus more closely, the default confidence thresholds lead to better performance of the networks using the classical KPIs. In the supplementary material we publish extended standard KPIs results.

Examples of labels from a neural network, human labeler and official labels are presented in the Figure 3. For the selected image, CO-DETR VIT-L neural network gets 22 false negatives and 5 false positives, whereas the selected human labeler gets 34 false negatives and 2 false positives (average values of false negatives and false positives among all the human labelers are 39.9 and 1.2). Interestingly, Waymo’s official labeling missed one pedestrian who was captured by both the neural network and the selected human labeler (80% of the human labelers found him).

6.1. Recall and Precision Comparison

Figure 2 compares precision and recall between neural networks and human labelers. Neural networks show a slight advantage in recall. For the nuImages dataset (a, top left), neural networks achieve a median recall close to 0.72, while human labelers are slightly lower, around 0.68. On the Waymo dataset (b, top right), neural networks reach a median recall of about 0.6 compared to 0.44 for human label-

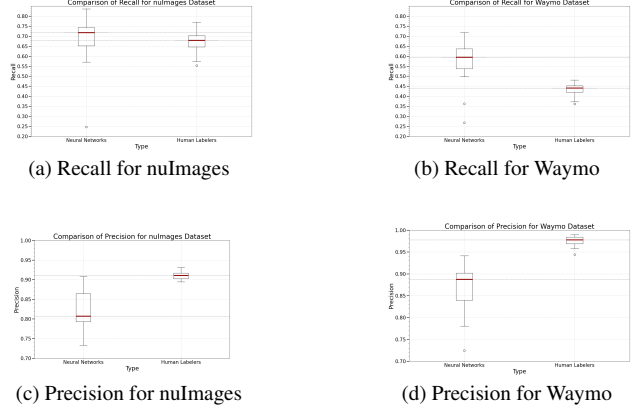


Figure 2. Comparison of *recall* (top row) and *precision* (bottom row) between the neural networks and human labelers for the NuImages (left) and Waymo (right) datasets. Median is marked in red, boxes represent the interquartile range (IQR), while whiskers extend to 1.5 times the IQR from the quartiles; points beyond the whiskers are marked as outliers.

ers. Neural networks display more variability and outliers in recall for both datasets.

For precision (bottom row), human labelers maintain higher medians. In nuImages (c, bottom left), human labelers have a median precision around 0.91, while neural networks are closer to 0.81. For Waymo (d, bottom right), human labelers reach a median precision of approximately 0.98, while neural networks achieve a high median around 0.89.

In summary, neural networks hold a slight recall advantage, while human labelers achieve superior precision. Nonetheless, neural networks demonstrate a high median precision, showing robust performance across datasets.

7. Conclusions

Our experiments have shown that for clearly defined classes, which were present in the training datasets (car and pedestrian classes), the image processing neural networks outperform majority or all of the tested human labelers. For other classes (cyclist and other vehicle), neural networks performed significantly worse than human labelers. We believe, that this underperformance is due to the fact that cyclist and other vehicle classes were improperly defined from the perspective of the networks - for majority of networks cyclists were synthetically constructed, by concatenating pedestrians and bicycles with sufficient overlap (described in section 4.3.2) and other vehicle is a catch-all class for anything that moves on the road and is neither a car, nor a cyclist. We believe that this issue can be fixed by training neural networks on datasets whose set of classes matches the final prediction problem.

As described in the section 3.2, manual labeling process typically involves 2 stages - initial labeling and review. We

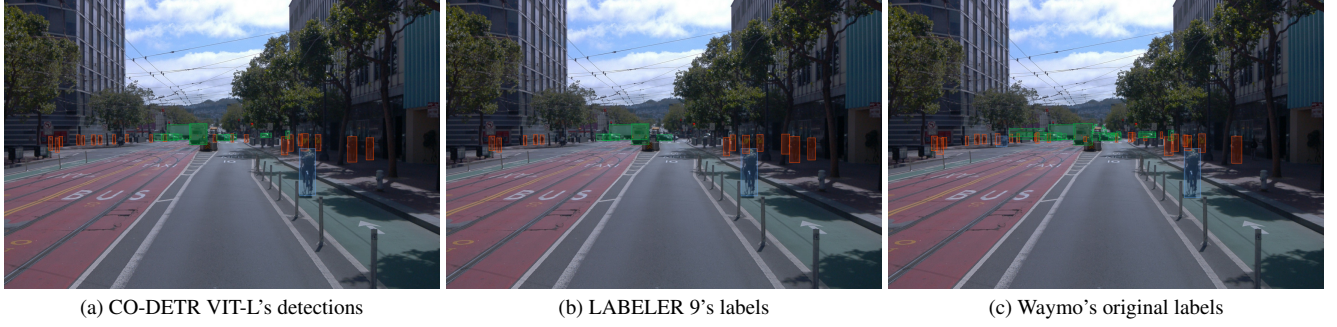


Figure 3. Examples of object detections provided by a selected neural network (a), a selected human labeler (b) and the Waymo's labels (c). Vehicle objects are marked in green, Pedestrian objects are marked in red while Cyclist objects are marked in blue.

Table 5. Standard KPI indicators computed for the Waymo samples. The results are ordered according to mAP and micro-F1. The best result marked in bold.

NAME	MAP	MICROF1	MACROF1	TP	FP	FN
Co-DETR ViT-L	40.10	0.72	0.70	2958	512	1779
Co-DETR SWIN-L	39.40	0.72	0.69	2967	514	1770
YOLOv8x	35.30	0.72	0.66	2814	284	1923
YOLOv9E	35.00	0.70	0.65	2701	231	2036
YOLOv11x	34.50	0.75	0.66	3068	387	1669
YOLOv11x TL	34.30	0.74	0.67	2972	302	1765
YOLOv10x TL	34.10	0.72	0.65	2858	333	1879
YOLOv10x	34.10	0.71	0.65	2784	278	1953
YOLOv8x TL	33.80	0.70	0.64	2645	209	2092
YOLOv9E TL	33.50	0.70	0.65	2661	193	2076
DETECTRON2	31.90	0.72	0.64	2826	336	1911
DDQ DETR 4	31.50	0.68	0.60	2581	302	2156
DDQ DETR 5	29.70	0.64	0.55	2358	310	2379
DD3D	11.70	0.41	0.38	1266	161	3471
EPro-PNP	6.10	0.48	0.38	1719	654	3018
LABELER 4	N/A	0.64	0.65	2278	74	2459
LABELER 10	N/A	0.64	0.64	2236	69	2501
LABELER 5	N/A	0.62	0.62	2146	49	2591
LABELER 3	N/A	0.61	0.60	2122	126	2615
LABELER 9	N/A	0.61	0.61	2079	48	2658
LABELER 2	N/A	0.61	0.58	2097	92	2640
LABELER 7	N/A	0.59	0.58	2012	29	2725
LABELER 1	N/A	0.59	0.59	1979	40	2758
LABELER 6	N/A	0.54	0.55	1764	19	2973
LABELER 8	N/A	0.53	0.53	1713	22	3024

Table 6. Standard KPI indicators computed for the nuImages samples. The results are ordered according to mAP and micro-F1. The best result marked in bold.

NAME	MAP	MICROF1	MACROF1	TP	FP	FN
Co-DETR ViT-L	54.50	0.81	0.81	2238	594	439
Co-DETR SWIN-L	53.00	0.80	0.80	2194	609	483
DDQ DETR 4	44.80	0.76	0.75	1958	503	719
YOLOv10x TL	42.80	0.78	0.74	1898	321	779
YOLOv8x TL	42.70	0.77	0.75	1831	242	846
YOLOv9E TL	42.30	0.76	0.75	1798	228	879
DETECTRON2	42.00	0.78	0.75	1904	330	773
YOLOv11x TL	41.90	0.78	0.77	1921	298	756
YOLOv9E	41.90	0.78	0.77	1921	298	756
YOLOv11x	41.70	0.75	0.73	1786	282	891
DDQ DETR 5	41.40	0.72	0.68	1743	424	934
YOLOv8x	41.20	0.77	0.74	1848	280	829
YOLOv10x	41.20	0.75	0.71	1743	245	934
EPro-PNP	21.10	0.64	0.57	1527	557	1150
DD3D	12.90	0.38	0.37	660	150	2017
LABELER 10	N/A	0.83	0.84	2058	225	619
LABELER 7	N/A	0.81	0.82	1933	167	744
LABELER 4	N/A	0.80	0.81	1893	179	784
LABELER 5	N/A	0.78	0.79	1822	179	855
LABELER 8	N/A	0.78	0.78	1841	217	836
LABELER 1	N/A	0.78	0.79	1812	177	865
LABELER 9	N/A	0.77	0.80	1760	157	917
LABELER 6	N/A	0.76	0.79	1720	128	957
LABELER 3	N/A	0.70	0.73	1534	167	1143
LABELER 2	N/A	0.69	0.71	1480	156	1197

have shown that the review process has clear advantages - the scores of labelers, as well as the scores of neural networks show that all predictions are closer to each other after review (Tables 3 and 4). Given that the neural networks tend to outperform humans on the well defined classes, we believe that the initial labeling should always be done using automated labeling (i.e. neural networks). The consequent review process should be performed by humans, to leverage higher precision of human labelers.

We found the official labels from Waymo and Nuscenes to be significantly better than the labels generated by our hu-

man labelers. We believe that this is caused by the fact, that while our labelers had access only to a set of random camera images (not in a sequence), the official labels for those datasets were prepared using sequences of consecutive frames, as well as data from additional sensors (like lidars). We believe that it makes a strong argument to utilize as much data as possible during labeling. In particular, it suggests that in automotive setting, labeling sequences of frames should be strongly preferred over labeling single frames, even if the final system is supposed to operate on static images.

How Much Noise is there in Labels Generated by Humans? A Method to Validate Automatically Generated Bounding Boxes.

Supplementary Material

8. Labelling process

Labeling was performed in a professional lab experienced with similar labeling tasks for major automotive producers, using Label Studio (<https://labelstud.io/>) software. Labelers were experienced at their task, so it is highly unlikely that they might have misunderstood the labeling specifications. In the final paper we intend to reveal the name of the labeling company. Label studio software enables zooming in on the images as well as brightness manipulation by the labeler. The labelers did not have any explicit time limit for the task and were paid by the hour. On average, a single labeler spent around 4 minutes on a single image (with huge variation among labelers). Human labeling could certainly be improved (e.g. by utilizing significant cash bonuses for high quality labels), but the goal of the paper was to examine the quality of typical labeling by an experienced group of labelers.

9. Deviation from Consensus on Subdatasets

To construct our dataset we used images from KITTI [7], Waymo [23] and nuImages [2], taking 210 examples from each of them. Each of those subdatasets has different characteristics. Therefore, in addition to combined deviation from consensus KPIs presented in Section 5, here (Tables 7-12) we present the KPIs calculated separately for each subdataset. The general conclusions (top neural networks are better than humans for cars and pedestrians) from the paper hold for all the subdatasets.

10. Extended Classical KPIs

We present supplementary metrics with respect to the KPIs calculated in tables 5 and 6. In tables 13, 14 we show additional mAP metrics for the models for different IoU thresholds and bounding boxes sizes (understood as in COCO evaluation). We don't include mAP metrics of each of the human labelers. This is because we cannot use different confidence levels for humans.

Tables 15-23 show classical KPI (F1, TP, FP, FN, Precision and Recall) computed for the Waymo and nuImages datasets with detailed breakdown per label class. In case of Waymo data for all classes except Cyclist the networks exhibit significantly better performance than the labelers. In case of nuImages data the best performance of Labeler 10 is due to the significantly better performance for the Other Vehicle class, in the case of this class we see that in general labelers are outperforming networks.

Figures 4 and 5 show labels of all labelers and predictions of all considered neural networks on selected images from the Waymo and nuImages datasets respectively.

11. Acknowledgements

This work was partially supported by the project AI4PL - Integrating Reasoning, Learning, Optimization and Interpretability for accelerated commercialization of next-generation intelligent software systems

12. Author Contact Information

Mariusz Karol Nowak ,
Jacek Cyranka ,
Natalia Maślany ,
Aleksander Kostuch ,
Jakub Derbisz ,
Mateusz Komorkiewicz , Patryk Siwek ,
Mateusz Jan Wójcik ,
Dariusz Marchewka ,
Paweł Skruch 

Table 7. Deviation from consensus scores for KITTI dataset. Labels before review. Using (norm) instead of (normalized) to fit the table on the page. The best performers are in bold. If no network is better than 50% of labelers, then we mark human labelers as the best in normalized score.

Source of Labels	Car (mean)	Car (norm)	Pedestrian (mean)	Pedestrian (norm)	Cyclist (mean)	Cyclist (norm)	Other Vehicle (mean)	Other Vehicle (norm)
Human Labelers	0.245	N/A	0.245	N/A	0.262	N/A	0.486	N/A
YOLOv8x	0.175	1.000	0.209	0.900	0.604	0.000	0.712	0.000
YOLOv9e	0.177	1.000	0.198	0.900	0.655	0.000	0.742	0.000
YOLOv10x	0.184	1.000	0.207	0.900	0.615	0.000	0.771	0.000
YOLOv11x	0.177	1.000	0.214	0.800	0.669	0.000	0.723	0.000
Detectron2	0.221	0.700	0.223	0.700	0.564	0.000	0.817	0.000
Co-DETR Swin-L	0.168	1.000	0.170	1.000	0.494	0.000	0.700	0.000
Co-DETR ViT-L	0.172	1.000	0.166	1.000	0.528	0.000	0.703	0.000
DDQ Detr 4	0.201	0.900	0.201	0.900	0.510	0.000	0.692	0.000
DDQ Detr 5	0.218	0.800	0.212	0.800	0.630	0.000	0.779	0.000
YOLOv8x TL	0.172	1.000	0.177	1.000	0.374	0.000	0.704	0.000
YOLOv9e TL	0.172	1.000	0.170	1.000	0.470	0.000	0.748	0.000
YOLOv10x TL	0.194	0.900	0.190	1.000	0.409	0.000	0.724	0.000
YOLOv11x TL	0.177	1.000	0.182	1.000	0.397	0.000	0.731	0.000
EPro-PNP (NuScenes)	0.759	0.000	0.964	0.000	0.995	0.000	0.962	0.000
DD3D (KITTI)	0.216	0.800	0.346	0.000	0.536	0.000	0.895	0.000

Table 8. Deviation from consensus scores for KITTI dataset. Labels after review. Using (norm) instead of (normalized) to fit the table on the page. The best performers are in bold. If no network is better than 50% of labelers, then we mark human labelers as the best in normalized score.

Source of Labels	Car (mean)	Car (norm)	Pedestrian (mean)	Pedestrian (norm)	Cyclist (mean)	Cyclist (norm)	Other Vehicle (mean)	Other Vehicle (norm)
Human Labelers	0.216	N/A	0.225	N/A	0.222	N/A	0.339	N/A
YOLOv8x	0.167	1.000	0.205	0.600	0.596	0.000	0.701	0.000
YOLOv9e	0.171	1.000	0.195	0.900	0.648	0.000	0.732	0.000
YOLOv10x	0.177	1.000	0.203	0.700	0.607	0.000	0.758	0.000
YOLOv11x	0.171	1.000	0.211	0.600	0.662	0.000	0.704	0.000
Detectron2	0.216	0.500	0.217	0.500	0.555	0.000	0.807	0.000
Co-DETR Swin-L	0.162	1.000	0.165	1.000	0.484	0.000	0.682	0.000
Co-DETR ViT-L	0.165	1.000	0.162	1.000	0.518	0.000	0.673	0.000
DDQ Detr 4	0.195	0.900	0.194	0.900	0.500	0.000	0.675	0.000
DDQ Detr 5	0.217	0.500	0.208	0.600	0.622	0.000	0.766	0.000
YOLOv8x TL	0.166	1.000	0.174	1.000	0.360	0.000	0.696	0.000
YOLOv9e TL	0.166	1.000	0.166	1.000	0.458	0.000	0.734	0.000
YOLOv10x TL	0.189	0.900	0.185	0.900	0.396	0.000	0.712	0.000
YOLOv11x TL	0.171	1.000	0.178	0.900	0.384	0.000	0.720	0.000
EPro-PNP (NuScenes)	0.758	0.000	0.964	0.000	0.995	0.000	0.967	0.000
DD3D (KITTI)	0.209	0.600	0.344	0.000	0.526	0.000	0.888	0.000

Table 9. Deviation from consensus scores for Waymo dataset. Labels before review. Using (norm) instead of (normalized) to fit the table on the page. The best performers are in bold. If no network is better than 50% of labelers, then we mark human labelers as the best in normalized score.

Source of Labels	Car (mean)	Car (norm)	Pedestrian (mean)	Pedestrian (norm)	Cyclist (mean)	Cyclist (norm)	Other Vehicle (mean)	Other Vehicle (norm)
Human Labelers	0.280	N/A	0.341	N/A	0.505	N/A	0.430	N/A
YOLOv8x	0.211	0.900	0.269	0.900	0.740	0.000	0.511	0.000
YOLOv9e	0.208	0.900	0.262	0.900	0.741	0.000	0.544	0.000
YOLOv10x	0.237	0.900	0.295	0.900	0.738	0.000	0.604	0.000
YOLOv11x	0.247	0.800	0.295	0.900	0.727	0.000	0.573	0.000
Detectron2	0.231	0.900	0.289	0.900	0.757	0.000	0.524	0.000
Co-DETR Swin-L	0.165	1.000	0.270	0.900	0.743	0.000	0.574	0.000
Co-DETR ViT-L	0.159	1.000	0.270	0.900	0.786	0.000	0.552	0.000
DDQ Detr 4	0.195	0.900	0.288	0.900	0.754	0.000	0.616	0.000
DDQ Detr 5	0.164	1.000	0.281	0.900	0.919	0.000	0.643	0.000
YOLOv8x TL	0.185	0.900	0.252	1.000	0.657	0.000	0.496	0.000
YOLOv9e TL	0.190	0.900	0.257	1.000	0.666	0.000	0.532	0.000
YOLOv10x TL	0.231	0.900	0.285	0.900	0.651	0.000	0.509	0.000
YOLOv11x TL	0.225	0.900	0.277	0.900	0.666	0.000	0.536	0.000
EPro-PNP (NuScenes)	0.501	0.000	0.601	0.000	0.972	0.000	0.704	0.000
DD3D (KITTI)	0.310	0.100	0.487	0.000	0.860	0.000	0.790	0.000

Table 10. Deviation from consensus scores for Waymo dataset. Labels after review. Using (norm) instead of (normalized) to fit the table on the page. The best performers are in bold. If no network is better than 50% of labelers, then we mark human labelers as the best in normalized score.

Source of Labels	Car (mean)	Car (norm)	Pedestrian (mean)	Pedestrian (norm)	Cyclist (mean)	Cyclist (norm)	Other Vehicle (mean)	Other Vehicle (norm)
Human Labelers	0.234	N/A	0.284	N/A	0.343	N/A	0.354	N/A
YOLOv8x	0.214	0.700	0.266	0.600	0.617	0.000	0.475	0.000
YOLOv9e	0.210	0.700	0.256	0.700	0.617	0.000	0.512	0.000
YOLOv10x	0.246	0.500	0.294	0.400	0.633	0.000	0.582	0.000
YOLOv11x	0.257	0.300	0.297	0.400	0.617	0.000	0.545	0.000
Detectron2	0.229	0.600	0.289	0.500	0.642	0.000	0.498	0.000
Co-DETR Swin-L	0.161	1.000	0.273	0.600	0.614	0.000	0.538	0.000
Co-DETR ViT-L	0.152	1.000	0.265	0.600	0.711	0.000	0.516	0.000
DDQ Detr 4	0.191	0.900	0.286	0.600	0.664	0.000	0.590	0.000
DDQ Detr 5	0.158	1.000	0.278	0.600	0.883	0.000	0.621	0.000
YOLOv8x TL	0.186	0.900	0.243	0.900	0.498	0.000	0.468	0.000
YOLOv9e TL	0.189	0.900	0.249	0.900	0.501	0.000	0.501	0.000
YOLOv10x TL	0.236	0.500	0.285	0.600	0.503	0.000	0.488	0.000
YOLOv11x TL	0.230	0.600	0.281	0.600	0.503	0.000	0.511	0.000
EPro-PNP (NuScenes)	0.488	0.000	0.611	0.000	0.959	0.000	0.684	0.000
DD3D (KITTI)	0.301	0.000	0.471	0.000	0.804	0.000	0.784	0.000

Table 11. Deviation from consensus scores for nuImages dataset. Labels before review. Using (norm) instead of (normalized) to fit the table on the page. The best performers are in bold. If no network is better than 50% of labelers, then we mark human labelers as the best in normalized score.

Source of Labels	Car (mean)	Car (norm)	Pedestrian (mean)	Pedestrian (norm)	Cyclist (mean)	Cyclist (norm)	Other Vehicle (mean)	Other Vehicle (norm)
Human Labelers	0.259	N/A	0.368	N/A	0.310	N/A	0.371	N/A
YOLOv8x	0.252	0.500	0.338	0.800	0.832	0.000	0.441	0.000
YOLOv9e	0.243	0.600	0.345	0.800	0.788	0.000	0.452	0.000
YOLOv10x	0.262	0.300	0.368	0.400	0.833	0.000	0.483	0.000
YOLOv11x	0.275	0.300	0.402	0.300	0.833	0.000	0.550	0.000
Detectron2	0.243	0.600	0.382	0.400	0.741	0.000	0.463	0.000
Co-DETR Swin-L	0.204	0.900	0.337	0.800	0.735	0.000	0.418	0.000
Co-DETR ViT-L	0.182	0.900	0.317	0.800	0.716	0.000	0.429	0.000
DDQ Detr 4	0.258	0.400	0.350	0.800	0.650	0.000	0.486	0.000
DDQ Detr 5	0.224	0.800	0.408	0.100	0.809	0.000	0.583	0.000
YOLOv8x TL	0.240	0.600	0.372	0.400	0.626	0.000	0.459	0.000
YOLOv9e TL	0.238	0.600	0.355	0.700	0.650	0.000	0.476	0.000
YOLOv10x TL	0.248	0.600	0.337	0.800	0.630	0.000	0.475	0.000
YOLOv11x TL	0.247	0.600	0.349	0.800	0.675	0.000	0.469	0.000
EPro-PNP (NuScenes)	0.336	0.100	0.559	0.000	0.794	0.000	0.527	0.000
DD3D (KITTI)	0.340	0.100	0.706	0.000	0.945	0.000	0.799	0.000

Table 12. Deviation from consensus scores for nuImages dataset. Labels after review. Using (norm) instead of (normalized) to fit the table on the page. The best performers are in bold. If no network is better than 50% of labelers, then we mark human labelers as the best in normalized score.

Source of Labels	Car (mean)	Car (norm)	Pedestrian (mean)	Pedestrian (norm)	Cyclist (mean)	Cyclist (norm)	Other Vehicle (mean)	Other Vehicle (norm)
Human Labelers	0.233	N/A	0.326	N/A	0.227	N/A	0.288	N/A
YOLOv8x	0.242	0.300	0.318	0.500	0.812	0.000	0.404	0.000
YOLOv9e	0.238	0.300	0.328	0.500	0.763	0.000	0.415	0.000
YOLOv10x	0.255	0.200	0.349	0.300	0.812	0.000	0.452	0.000
YOLOv11x	0.267	0.200	0.385	0.000	0.812	0.000	0.521	0.000
Detectron2	0.237	0.300	0.368	0.100	0.711	0.000	0.423	0.000
Co-DETR Swin-L	0.197	1.000	0.318	0.500	0.706	0.000	0.380	0.000
Co-DETR ViT-L	0.177	1.000	0.301	0.800	0.683	0.000	0.389	0.000
DDQ Detr 4	0.254	0.200	0.331	0.500	0.615	0.000	0.442	0.000
DDQ Detr 5	0.218	0.700	0.390	0.000	0.788	0.000	0.545	0.000
YOLOv8x TL	0.234	0.300	0.354	0.200	0.587	0.000	0.423	0.000
YOLOv9e TL	0.232	0.300	0.340	0.300	0.613	0.000	0.441	0.000
YOLOv10x TL	0.244	0.300	0.319	0.500	0.591	0.000	0.439	0.000
YOLOv11x TL	0.243	0.300	0.332	0.400	0.640	0.000	0.432	0.000
EPro-PNP (NuScenes)	0.329	0.000	0.541	0.000	0.772	0.000	0.495	0.000
DD3D (KITTI)	0.337	0.000	0.700	0.000	0.938	0.000	0.782	0.000

Table 13. mAP results of Neural Networks for Waymo dataset

Name	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
CO-DETR VIT-L	40.10	65.80	42.30	15.30	47.20	78.50
CO-DETR SWIN-L	39.40	65.50	40.40	14.40	47.20	77.60
YOLOV8X	35.30	56.20	38.30	9.80	42.80	76.60
YOLOV9E	35.00	55.40	38.10	9.20	42.40	76.50
YOLOV11X	34.50	55.70	37.30	9.90	41.40	75.40
YOLOV11X TL	34.30	55.50	36.60	9.00	41.70	75.20
YOLOV10X TL	34.10	54.80	37.00	9.30	41.40	74.90
YOLOV10X	34.10	54.20	36.90	9.50	41.00	76.20
YOLOV8X TL	33.80	53.20	36.30	8.70	41.70	73.30
YOLOV9E TL	33.50	53.30	36.50	7.80	41.60	74.90
DETECTRON2	31.90	53.40	32.50	7.60	38.10	73.70
DDQ DETR 4	31.50	53.10	33.10	8.20	36.00	76.50
DDQ DETR 5	29.70	50.00	29.90	6.50	35.10	71.30
DD3D	11.70	24.90	9.00	0.70	15.30	32.60
EPRO-PNP	6.10	19.10	2.60	2.00	10.00	10.00

Table 14. mAP results of Neural Networks for nuImages dataset

Name	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
CO-DETR VIT-L	54.50	77.60	57.00	24.30	50.80	75.20
CO-DETR SWIN-L	53.00	75.00	55.90	24.30	49.10	73.70
DDQ DETR 4	44.80	68.60	45.40	16.80	41.30	68.00
YOLOV10X TL	42.80	63.80	46.30	18.20	39.30	65.60
YOLOV8X TL	42.70	62.10	45.70	18.10	39.30	65.30
YOLOV9E TL	42.30	61.80	45.10	18.50	38.80	65.10
DETECTRON2	42.00	64.50	43.30	17.30	37.20	65.80
YOLOV9E	41.90	62.60	45.60	18.10	38.40	64.70
YOLOV11X TL	41.90	62.60	45.60	18.10	38.40	64.70
YOLOV11X	41.70	61.30	44.30	16.00	37.70	66.40
DDQ DETR 5	41.40	61.90	43.40	16.60	37.00	67.50
YOLOV10X	41.20	60.30	44.10	14.80	38.40	64.40
YOLOV8X	41.20	60.00	44.80	17.00	36.60	66.90
EPRO-PNP	21.10	49.50	13.70	10.50	21.90	31.80
DD3D	12.90	23.60	12.50	1.10	12.10	29.10

Table 15. Classical KPI for the Waymo dataset, All classes.

NAME	MICROF1	MACROF1	TP	FP	FN	PRECISION	RECALL
YOLOv11x	0.749	0.664	3068	387	1669	0.888	0.648
YOLOv11x TL	0.742	0.666	2972	302	1765	0.908	0.627
Co-DETR SWIN-L	0.722	0.687	2967	514	1770	0.852	0.626
YOLOv10x TL	0.721	0.649	2858	333	1879	0.896	0.603
Co-DETR ViT-L	0.721	0.700	2958	512	1779	0.852	0.624
YOLOv8x	0.718	0.664	2814	284	1923	0.908	0.594
DETECTRON2	0.716	0.644	2826	336	1911	0.894	0.597
YOLOv10x	0.714	0.648	2784	278	1953	0.909	0.588
YOLOv9E	0.704	0.652	2701	231	2036	0.921	0.570
YOLOv9E TL	0.701	0.648	2661	193	2076	0.932	0.562
YOLOv8x TL	0.697	0.643	2645	209	2092	0.927	0.558
DDQ DETR 4	0.677	0.599	2581	302	2156	0.895	0.545
LABELER 4	0.643	0.652	2278	74	2459	0.969	0.481
DDQ DETR 5	0.637	0.554	2358	310	2379	0.884	0.498
LABELER 10	0.635	0.638	2236	69	2501	0.970	0.472
LABELER 5	0.619	0.623	2146	49	2591	0.978	0.453
LABELER 3	0.608	0.604	2122	126	2615	0.944	0.448
LABELER 9	0.606	0.614	2079	48	2658	0.977	0.439
LABELER 2	0.606	0.579	2097	92	2640	0.958	0.443
LABELER 7	0.594	0.579	2012	29	2725	0.986	0.425
LABELER 1	0.586	0.591	1979	40	2758	0.980	0.418
LABELER 6	0.541	0.548	1764	19	2973	0.989	0.372
LABELER 8	0.529	0.525	1713	22	3024	0.987	0.362
EPro-PNP	0.484	0.384	1719	654	3018	0.724	0.363
DD3D	0.411	0.381	1266	161	3471	0.887	0.267

Table 16. Classical KPI for the Waymo dataset, the Pedestrian class.

NAME	F1	TP	FP	FN	PRECISION	RECALL
Co-DETR SWIN-L	0.727	815	168	444	0.829	0.647
Co-DETR ViT-L	0.721	800	159	459	0.834	0.635
YOLOv8x	0.707	753	117	506	0.866	0.598
YOLOv11x TL	0.704	745	112	514	0.869	0.592
YOLOv11x	0.702	753	134	506	0.849	0.598
YOLOv9E	0.702	736	103	523	0.877	0.585
YOLOv10x TL	0.694	732	119	527	0.860	0.581
YOLOv8x TL	0.683	698	87	561	0.889	0.554
YOLOv10x	0.680	695	90	564	0.885	0.552
YOLOv9E TL	0.679	690	83	569	0.893	0.548
DETECTRON2	0.679	715	132	544	0.844	0.568
DDQ DETR 4	0.644	652	113	607	0.852	0.518
LABELER 4	0.619	571	15	688	0.974	0.454
DDQ DETR 5	0.615	599	89	660	0.871	0.476
LABELER 10	0.607	555	16	704	0.972	0.441
LABELER 9	0.591	534	14	725	0.974	0.424
LABELER 3	0.586	535	31	724	0.945	0.425
LABELER 5	0.574	510	8	749	0.985	0.405
LABELER 2	0.566	503	16	756	0.969	0.400
LABELER 1	0.558	492	14	767	0.972	0.391
LABELER 7	0.541	470	7	789	0.985	0.373
LABELER 6	0.497	418	4	841	0.991	0.332
EPro-PNP	0.497	544	385	715	0.586	0.432
LABELER 8	0.490	411	9	848	0.979	0.326
DD3D	0.362	289	47	970	0.860	0.230

Table 17. Classical KPI for the Waymo dataset, the Cyclist class.

NAME	F1	TP	FP	FN	PRECISION	RECALL
LABELER 4	0.687	46	2	40	0.958	0.535
LABELER 5	0.662	43	1	43	0.977	0.500
LABELER 10	0.662	43	1	43	0.977	0.500
Co-DETR ViT-L	0.657	44	4	42	0.917	0.512
LABELER 9	0.641	41	1	45	0.976	0.477
LABELER 1	0.619	39	1	47	0.975	0.453
Co-DETR SWIN-L	0.612	41	7	45	0.854	0.477
LABELER 3	0.609	39	3	47	0.929	0.453
LABELER 6	0.590	36	0	50	1.000	0.419
LABELER 7	0.583	37	4	49	0.902	0.430
YOLOv8x	0.560	35	4	51	0.897	0.407
YOLOv9E TL	0.553	34	3	52	0.919	0.395
LABELER 2	0.550	33	1	53	0.971	0.384
YOLOv9E	0.544	34	5	52	0.872	0.395
LABELER 8	0.542	32	0	54	1.000	0.372
YOLOv8x TL	0.540	34	6	52	0.850	0.395
YOLOv10x	0.533	32	2	54	0.941	0.372
YOLOv11x TL	0.533	32	2	54	0.941	0.372
YOLOv11x	0.521	31	2	55	0.939	0.360
DETECTRON2	0.520	32	5	54	0.865	0.372
YOLOv10x TL	0.517	31	3	55	0.912	0.360
DDQ DETR 4	0.458	27	5	59	0.844	0.314
DDQ DETR 5	0.397	23	7	63	0.767	0.267
DD3D	0.350	21	13	65	0.618	0.244
EPro-PNP	0.171	9	10	77	0.474	0.105

Table 18. Classical KPI for the Waymo dataset, the Vehicle class.

NAME	F1	TP	FP	FN	PRECISION	RECALL
YOLOv11x	0.771	2284	251	1108	0.901	0.673
YOLOv11x TL	0.760	2195	188	1197	0.921	0.647
YOLOv10x TL	0.735	2095	211	1297	0.908	0.618
DETECTRON2	0.733	2079	199	1313	0.913	0.613
YOLOv10x	0.730	2057	186	1335	0.917	0.606
YOLOv8x	0.726	2026	163	1366	0.926	0.597
Co-DETR SWIN-L	0.723	2111	339	1281	0.862	0.622
Co-DETR ViT-L	0.722	2114	349	1278	0.858	0.623
YOLOv9E TL	0.713	1937	107	1455	0.948	0.571
YOLOv9E	0.709	1931	123	1461	0.940	0.569
YOLOv8x TL	0.706	1913	116	1479	0.943	0.564
DDQ DETR 4	0.694	1902	184	1490	0.912	0.561
LABELER 4	0.650	1661	57	1731	0.967	0.490
DDQ DETR 5	0.650	1736	214	1656	0.890	0.512
LABELER 10	0.645	1638	52	1754	0.969	0.483
LABELER 5	0.634	1593	40	1799	0.976	0.470
LABELER 2	0.621	1561	75	1831	0.954	0.460
LABELER 3	0.615	1548	92	1844	0.944	0.456
LABELER 7	0.612	1505	18	1887	0.988	0.444
LABELER 9	0.610	1504	33	1888	0.979	0.443
LABELER 1	0.595	1448	25	1944	0.983	0.427
LABELER 6	0.555	1310	15	2082	0.989	0.386
LABELER 8	0.543	1270	13	2122	0.990	0.374
EPro-PNP	0.484	1166	259	2226	0.818	0.344
DD3D	0.430	956	101	2436	0.904	0.282

Table 19. Classical KPI for the nuImages dataset, All classes.

NAME	MICROF1	MACROF1	TP	FP	FN	PRECISION	RECALL
LABELER 10	0.830	0.838	2058	225	619	0.901	0.769
Co-DETR ViT-L	0.812	0.814	2238	594	439	0.790	0.836
LABELER 7	0.809	0.823	1933	167	744	0.920	0.722
Co-DETR SWIN-L	0.801	0.797	2194	609	483	0.783	0.820
LABELER 4	0.797	0.813	1893	179	784	0.914	0.707
YOLOv11x TL	0.785	0.770	1921	298	756	0.866	0.718
YOLOv9E	0.785	0.770	1921	298	756	0.866	0.718
LABELER 5	0.779	0.791	1822	179	855	0.911	0.681
LABELER 8	0.778	0.781	1841	217	836	0.895	0.688
LABELER 1	0.777	0.790	1812	177	865	0.911	0.677
DETECTRON2	0.775	0.748	1904	330	773	0.852	0.711
YOLOv10x TL	0.775	0.741	1898	321	779	0.855	0.709
YOLOv8x TL	0.771	0.749	1831	242	846	0.883	0.684
YOLOv8x	0.769	0.739	1848	280	829	0.868	0.690
LABELER 9	0.766	0.796	1760	157	917	0.918	0.657
YOLOv9E TL	0.765	0.752	1798	228	879	0.887	0.672
DDQ DETR 4	0.762	0.751	1958	503	719	0.796	0.731
LABELER 6	0.760	0.787	1720	128	957	0.931	0.643
YOLOv11x	0.753	0.726	1786	282	891	0.864	0.667
YOLOv10x	0.747	0.714	1743	245	934	0.877	0.651
DDQ DETR 5	0.720	0.680	1743	424	934	0.804	0.651
LABELER 3	0.701	0.732	1534	167	1143	0.902	0.573
LABELER 2	0.686	0.712	1480	156	1197	0.905	0.553
EPro-PNP	0.641	0.574	1527	557	1150	0.733	0.570
DD3D	0.379	0.367	660	150	2017	0.815	0.247

Table 20. Classical KPI for the nuImages dataset, the Pedestrian class.

NAME	F1	TP	FP	FN	PRECISION	RECALL
Co-DETR SWIN-L	0.832	923	168	204	0.846	0.819
Co-DETR ViT-L	0.830	928	182	199	0.836	0.823
LABELER 10	0.823	828	58	299	0.935	0.735
LABELER 4	0.789	766	48	361	0.941	0.680
YOLOv11x TL	0.775	754	64	373	0.922	0.669
YOLOv9E	0.775	754	64	373	0.922	0.669
LABELER 7	0.773	734	38	393	0.951	0.651
DETECTRON2	0.769	741	59	386	0.926	0.657
YOLOv10x TL	0.764	739	69	388	0.915	0.656
YOLOv8x	0.762	727	54	400	0.931	0.645
DDQ DETR 4	0.762	791	159	336	0.833	0.702
LABELER 5	0.752	704	41	423	0.945	0.625
LABELER 1	0.751	702	41	425	0.945	0.623
YOLOv9E TL	0.747	700	47	427	0.937	0.621
YOLOv8x TL	0.747	699	46	428	0.938	0.620
LABELER 8	0.746	699	47	428	0.937	0.620
LABELER 9	0.732	675	43	452	0.940	0.599
YOLOv11x	0.724	663	42	464	0.940	0.588
YOLOv10x	0.716	657	50	470	0.929	0.583
LABELER 6	0.708	636	33	491	0.951	0.564
DDQ DETR 5	0.700	663	105	464	0.863	0.588
LABELER 3	0.659	570	33	557	0.945	0.506
LABELER 2	0.644	552	35	575	0.940	0.490
EPro-PNP	0.620	686	400	441	0.632	0.609
DD3D	0.292	199	39	928	0.836	0.177

Table 21. Classical KPI for the nuImages dataset, the Cyclist class.

NAME	F1	TP	FP	FN	PRECISION	RECALL
LABELER 4	0.941	32	4	0	0.889	1.000
LABELER 9	0.941	32	4	0	0.889	1.000
Co-DETR ViT-L	0.939	31	3	1	0.912	0.969
LABELER 10	0.928	32	5	0	0.865	1.000
LABELER 7	0.928	32	5	0	0.865	1.000
LABELER 6	0.928	32	5	0	0.865	1.000
Co-DETR SWIN-L	0.923	30	3	2	0.909	0.938
LABELER 1	0.914	32	6	0	0.842	1.000
LABELER 3	0.909	30	4	2	0.882	0.938
LABELER 5	0.901	32	7	0	0.821	1.000
LABELER 8	0.873	31	8	1	0.795	0.969
LABELER 2	0.862	25	1	7	0.962	0.781
YOLOv9E TL	0.842	24	1	8	0.960	0.750
YOLOv11x TL	0.836	23	0	9	1.000	0.719
YOLOv9E	0.836	23	0	9	1.000	0.719
DDQ DETR 4	0.818	27	7	5	0.794	0.844
YOLOv8x TL	0.786	22	2	10	0.917	0.688
DETECTRON2	0.778	21	1	11	0.955	0.656
YOLOv11x	0.778	21	1	11	0.955	0.656
YOLOv8x	0.750	21	3	11	0.875	0.656
YOLOv10x TL	0.741	20	2	12	0.909	0.625
YOLOv10x	0.731	19	1	13	0.950	0.594
DDQ DETR 5	0.692	18	2	14	0.900	0.562
DD3D	0.414	12	14	20	0.462	0.375
EPro-PNP	0.360	16	41	16	0.281	0.500

Table 22. Classical KPI for the nuImages dataset, the Car class.

NAME	F1	TP	FP	FN	PRECISION	RECALL
LABELER 7	0.853	965	103	229	0.904	0.808
LABELER 10	0.852	991	141	203	0.875	0.830
Co-DETR ViT-L	0.840	1052	259	142	0.802	0.881
Co-DETR SWIN-L	0.826	1023	261	171	0.797	0.857
YOLOv11x TL	0.824	968	187	226	0.838	0.811
YOLOv9E	0.824	968	187	226	0.838	0.811
LABELER 8	0.823	933	140	261	0.870	0.781
LABELER 4	0.823	914	113	280	0.890	0.765
YOLOv8x TL	0.822	932	141	262	0.869	0.781
LABELER 5	0.818	903	110	291	0.891	0.756
YOLOv11x	0.817	935	159	259	0.855	0.783
LABELER 1	0.817	903	113	291	0.889	0.756
YOLOv10x TL	0.816	962	201	232	0.827	0.806
DETECTRON2	0.816	963	204	231	0.825	0.807
YOLOv9E TL	0.815	916	137	278	0.870	0.767
LABELER 6	0.814	870	74	324	0.922	0.729
YOLOv10x	0.812	897	119	297	0.883	0.751
YOLOv8x	0.811	914	147	280	0.861	0.765
LABELER 9	0.805	870	98	324	0.899	0.729
DDQ DETR 4	0.794	951	249	243	0.792	0.796
DDQ DETR 5	0.776	909	241	285	0.790	0.761
LABELER 3	0.750	784	114	410	0.873	0.657
LABELER 2	0.736	754	101	440	0.882	0.631
EPro-PNP	0.681	646	57	548	0.919	0.541
DD3D	0.476	384	34	810	0.919	0.322

Table 23. Classical KPI for the nuImages dataset, the Other Vehicle class.

NAME	F1	TP	FP	FN	PRECISION	RECALL
LABELER 10	0.750	207	21	117	0.908	0.639
LABELER 7	0.739	202	21	122	0.906	0.623
LABELER 9	0.705	183	12	141	0.938	0.565
LABELER 4	0.697	181	14	143	0.928	0.559
LABELER 6	0.697	182	16	142	0.919	0.562
LABELER 5	0.693	183	21	141	0.897	0.565
LABELER 8	0.679	178	22	146	0.890	0.549
LABELER 1	0.678	175	17	149	0.911	0.540
Co-DETR ViT-L	0.648	227	150	97	0.602	0.701
YOLOv10x TL	0.644	177	49	147	0.783	0.546
YOLOv11x TL	0.644	176	47	148	0.789	0.543
YOLOv9E	0.644	176	47	148	0.789	0.543
YOLOv8x TL	0.641	178	53	146	0.771	0.549
EPro-PNP	0.637	179	59	145	0.752	0.552
YOLOv8x	0.635	186	76	138	0.710	0.574
DETECTRON2	0.629	179	66	145	0.731	0.552
DDQ DETR 4	0.629	189	88	135	0.682	0.583
LABELER 3	0.612	150	16	174	0.904	0.463
Co-DETR SWIN-L	0.606	218	177	106	0.552	0.673
LABELER 2	0.606	149	19	175	0.887	0.460
YOLOv9E TL	0.602	158	43	166	0.786	0.488
YOLOv10x	0.598	170	75	154	0.694	0.525
YOLOv11x	0.585	167	80	157	0.676	0.515
DDQ DETR 5	0.553	153	76	171	0.668	0.472
DD3D	0.288	65	63	259	0.508	0.201

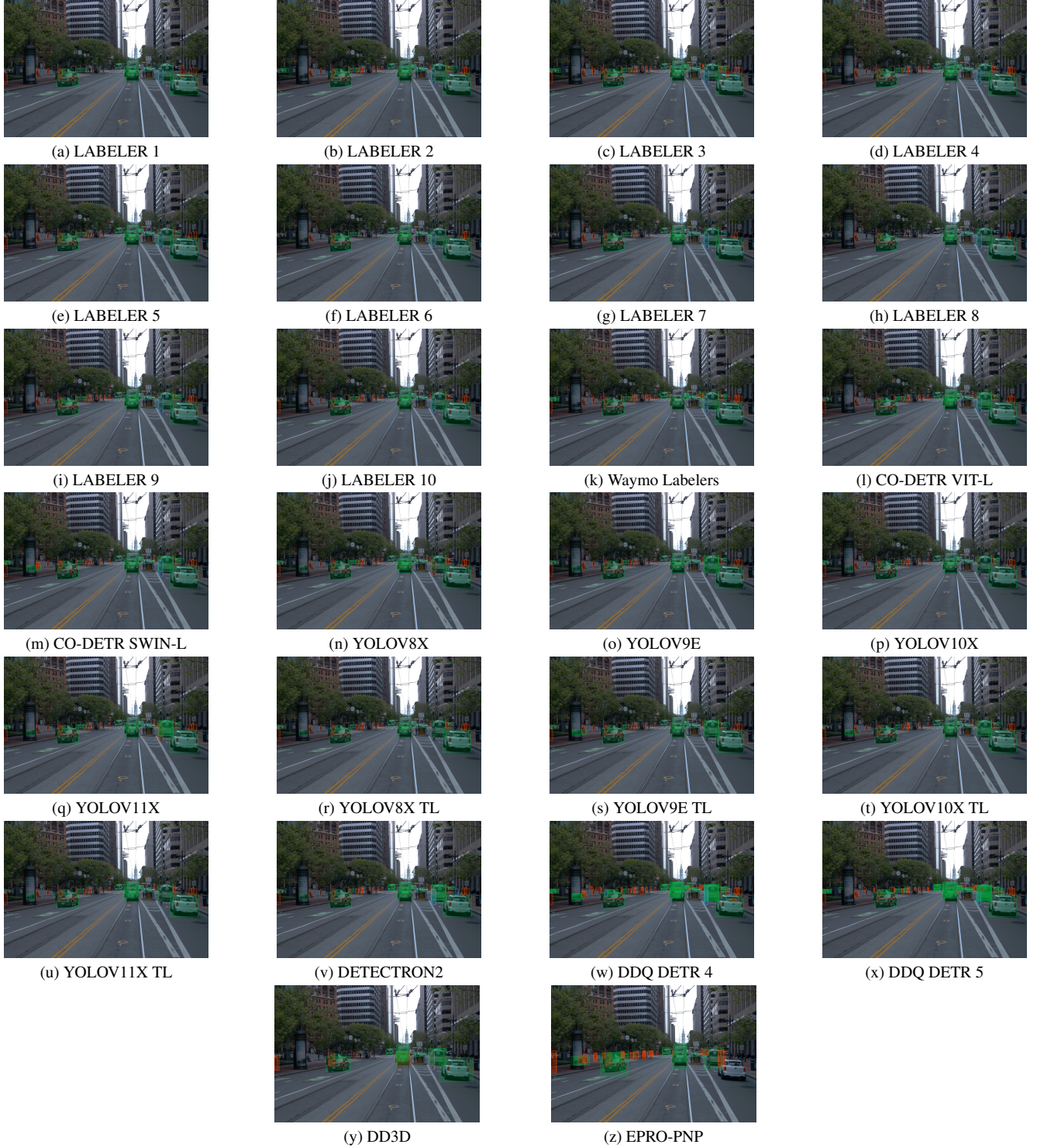


Figure 4. Examples of object detections on the Waymo dataset provided by human labelers and all selected neural networks. Vehicle objects are marked in green, Pedestrian objects are marked in red while Cyclist objects are marked in blue.

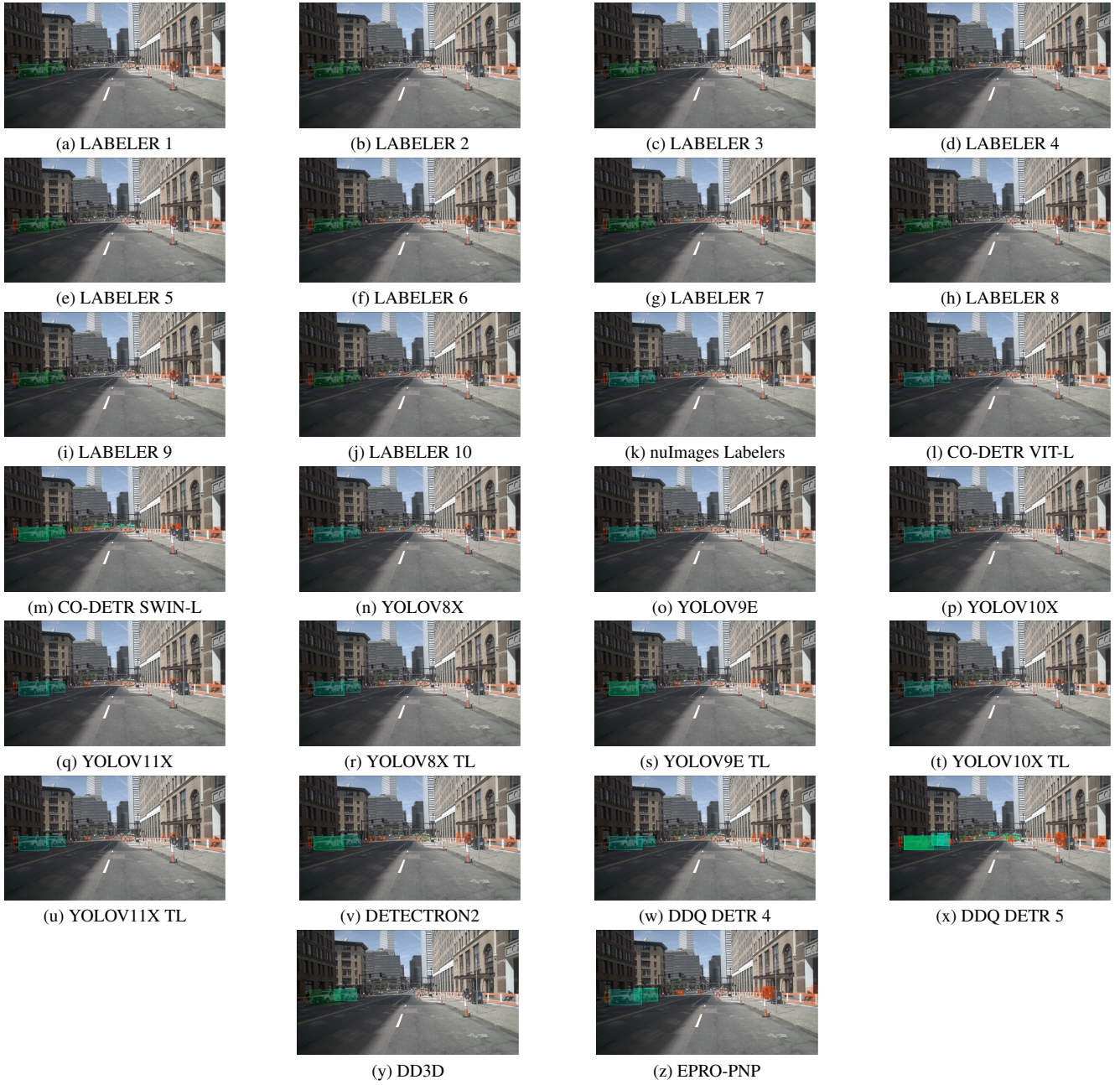


Figure 5. Examples of object detections on the nuImages dataset provided by human labelers and all selected neural networks. Vehicle objects are marked in green, Pedestrian objects are marked in red while Cyclist objects are marked in blue.

References

- [1] Lihao Liu, Ao Wang, Hui Chen et al. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024. Accessed: 2024-11-13.
- [2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.
- [3] Hansheng Chen, Pichao Wang, Fan Wang, Wei Tian, Lu Xiong, and Hao Li. Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [4] 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, and Dahua Lin. Mmdetection: Open mmlab detection toolbox and benchmark, 2019.
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.
- [7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.
- [8] Golnaz Ghiasi, Yin Cui, A. Srinivas, Rui Qian, Tsung-Yi Lin, Ekin Dogus Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2917–2927, 2020.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.
- [10] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. Accessed: 2024-11-13.
- [11] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. Accessed: 2024-11-13.
- [12] D. Kahneman, O. Sibony, and C.R. Sunstein. *Noise: A Flaw in Human Judgment*. William Collins, 2021.
- [13] Han S. Lee, Alex A. Agarwal, and Junmo Kim. Why do deep neural networks still not recognize these images?: A qualitative analysis on failure cases of imagenet classification, 2017.
- [14] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.
- [15] B. E. Moore and J. J. Corso. Fiftyone. *GitHub. Note: https://github.com/voxel51/fiftyone*, 2020.
- [16] J Nassar, V Pavon-Harr, M Bosch, and I McCulloh. Assessing data quality of annotations with krippendorff alpha for applications in computer vision. *arxiv 2019. arXiv preprint arXiv:1912.10107*, 1912.
- [17] Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks, 2021.

- [18] Aaryan Panda, Damodar Panigrahi, Shaswata Mitra, Sudip Mittal, and Shahram Rahimi. Transfer learning applied to computer vision problems: Survey on current progress, limitations, and opportunities, 2024.
- [19] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.
- [20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015.
- [21] Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. Evaluating machine accuracy on imagenet. In *International Conference on Machine Learning*, pages 8634–8644. PMLR, 2020.
- [22] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. *CoRR*, abs/1707.02968, 2017.
- [23] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [24] James R. Thompson and Gary L. Starkman. *Columbia Law Review*, 74(1):152–158, 1974.
- [25] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 595–604, 2015.
- [26] Vijay Vasudevan, Benjamin Caine, Raphael Gontijo Lopes, Sara Fridovich-Keil, and Rebecca Roelofs. When does dough become a bagel? analyzing the remaining mistakes on imagenet. In *Advances in Neural Information Processing Systems*, pages 6720–6734. Curran Associates, Inc., 2022.
- [27] Chien-Yao Wang and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. 2024. Accessed: 2024-11-13.
- [28] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [29] Shilong Zhang, Xinjiang Wang, Jiaqi Wang, Jiangmiao Pang, Chengqi Lyu, Wenwei Zhang, Ping Luo, and Kai Chen. Dense distinct query for end-to-end object detection. pages 7329–7338, 2023.
- [30] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758, 2023.