Flattening the Parent Bias: Hierarchical Semantic Segmentation in the Poincaré Ball

– Supplemental Material –

Simon Weber[1,2] Barış Zöngür[1] Nikita Araslanov[1,2] Daniel Cremers[1,2]

[1]Technical University of Munich [2]Munich Center for Machine Learning

This supplemental material is organized as follows:
Appendix A empirically complements our theoretical analysis of the hyperbolic distance in Proposition 2, and confirms the inter-class uniformity in the Poincaré ball.
Appendix B provides implementation details on calibrating hyperbolic networks.
Appendix C details the hierarchical two-level label structure used throughout our experiments.
Appendix D provides additional qualitative examples from other datasets (Cityscapes, Mappilary, IDD, ACDC, BDD and Wilddash).

A. Norms and concavity

Recall from Sec. 4.1 that during training, the hyperbolic likelihood,

$$p(\hat{y} := y \mid h) \propto \exp(\zeta_y(h))\,, \tag{14}$$

where

$$\zeta_y(h) := \frac{\lambda^c_{r_y}\|w_y\|}{\sqrt{c}}\operatorname{arcsinh}\left(\frac{2\sqrt{c}\langle -r_y \oplus_c h, w_y\rangle}{(1-c\|-r_y\oplus_c h\|^2)\|w_y\|}\right), \tag{15}$$

is minimized via the cross-entropy loss. Equivalently [23], Eq. (14) can be written as:

$$\log p(\hat{y} := y \mid h) \propto \operatorname{sign}\big(\langle -r_y \oplus_c h, a_y\rangle\big)\times \\ \times\sqrt{g^c_{r_y}(a_y,a_y)}\, d_{\mathbb{H}}(h, H^c_y)\,. \tag{16}$$

Thus, the closer a hyperbolic embedding is to the border (periphery) of the Poincaré ball, the higher its norm and the class posterior. Indeed, when the embedding is located between the gyroplane and the periphery of the Poincaré ball, *i.e.* the term $\langle -r_y \oplus_c h, a_y\rangle$ is strictly larger than zero, $p(\hat{y} := y \mid h)$ increases with the distance $d_{\mathbb{H}}(h, H^c_y)$ between the embedding and the gyroplane.

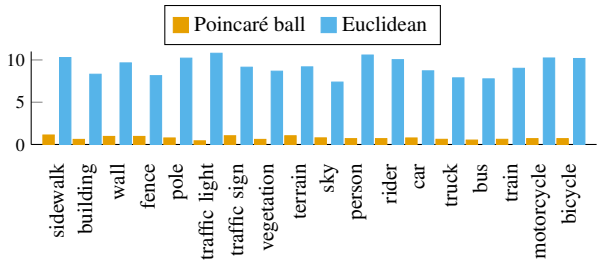
We empirically confirm that hyperbolic embeddings end up at the boundary of the Poincaré ball after model training. In Tab. 3, we compute the average norm of the embeddings for each class and find that the embedding norm in the Poincaré ball indeed tends to be close to 1, which indicates close proximity of the embeddings to the Poincaré ball's boundary. Let us now recall the hyperbolic distance from Eq. (8):

$$d_{\mathbb{H}}(h_1,h_2) = \operatorname{arcosh}\left(1 + 2\frac{\|h_1-h_2\|^2}{(1-\|h_1\|^2)(1-\|h_2\|^2)}\right). \tag{17}$$

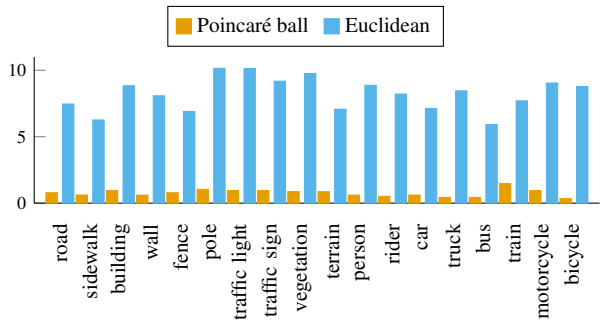
After network training, the class embeddings are well-separated. The denominator $(1-\|h_1\|^2)(1-\|h_2\|^2)$ becomes very small, hence the hyperbolic distance now operates at the high-end spectrum of the domain in Fig. 3 (*i.e.* when x in Fig. 3 is large). It follows that the hyperbolic dis-

Class	Average norm
Road	0.995 941 57
Sidewalk	0.995 935 76
Building	0.995 967 34
Wall	0.995 288 85
Fence	0.995 685 49
Pole	0.995 800 16
Traffic light	0.995 988 23
Traffic sign	0.995 781 03
Vegetation	0.995 964 70
Terrain	0.995 886 26
Sky	0.995 967 04
Person	0.995 909 56
Rider	0.995 988 13
Car	0.995 959 47
Truck	0.995 816 33
Bus	0.995 965 89
Train	0.995 361 07
Motorcycle	0.995 836 02
Bicycle	0.995 963 11

Table 3. The average norm of hyperbolic embeddings in the Poincaré ball for each class. Note that the norms are close to 1. Geometrically, this means that the embeddings reside at the periphery of the ball. Linking this observation to the concavity of the hyperbolic distance (*cf*. Sec. 4.3) explains the uniformity in the inter-classes distances, as a change in the embedding of a class, as long as its norm is preserved, does not have a significant effect on its relative distance to the embeddings of other classes.

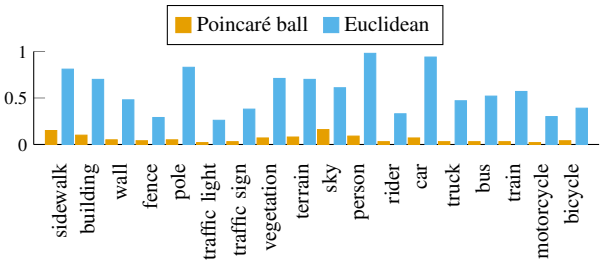


(a) Measuring the coefficient of variation w.r.t. "Road" embeddings.

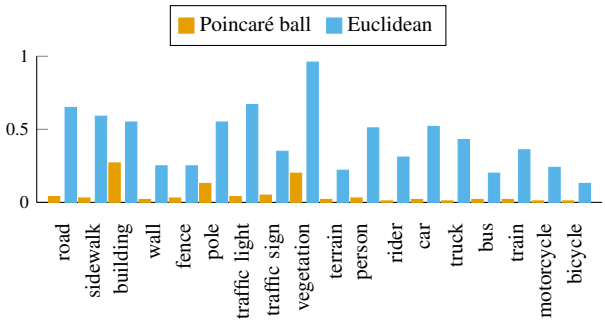


(b) Measuring the coefficient of variation w.r.t. "Sky" embeddings.

Figure 5. The coefficient of variation of the distance between (a) 'Road' embeddings (resp. (b) 'Sky' embeddings) and the embeddings of other classes. The hyperbolic distance between the embeddings of a given class and the embeddings of other classes is much more uniform than the respective Euclidean distance.



(a) Measuring the coefficient of variation of hyper-/gyroplanes distance w.r.t. "Road" embeddings.



(b) Measuring the coefficient of variation of hyper-/gyroplanes distance w.r.t. "Sky" embeddings.

Figure 6. The coefficient of variation of the distance between (a) 'Road' embeddings (resp. (b) 'Sky' embeddings) and gyroplanes/hyperplanes of other classes. The hyperbolic distance between a given class and the gyroplanes of other classes is much more uniform than the respective Euclidean distance.

tance between embeddings of different classes tends to be uniform. To demonstrate this empirically, we provide an example analysis of the inter-classes distance for two classes, 'Road' and 'Sky'. We compare the Poincaré ball model and the Euclidean representation. For all embeddings in the Poincaré ball (resp. Euclidean space) corresponding to these two classes, we derive the distance to the embeddings and to the gyroplanes (resp. hyperplanes) associated with the other classes. Fig. 5a (resp. Fig. 5b) illustrates the coefficient of variation (standard deviation divided by the mean) of the distance between 'Road' (resp. 'Sky') embeddings and the embeddings of other classes, grouped by each class. Fig. 6a (resp. Figure 6b) compares the coefficient of variation of the distance between 'Road' (resp. 'Sky') embeddings and the gyroplanes (for a hyperbolic space) and hyperplanes (for a Euclidean space) associated to the other classes. Observe that the coefficient of variation in the hyperbolic case is substantially lower across the board, which supports our analysis in the main text: The inter-class distance between the Poincaré embeddings is substantially more uniform than the embeddings in the Euclidean space. The same conclusion holds w.r.t. other semantic categories.

B. Calibration in the Poincaré ball

A model is said to be *calibrated* when the predictive probabilities correspond to the true probabilities [26]. Following the geometrical interpretation of Ganea et al. [23], the predictive probabilities in hyperbolic networks correspond to the logits after the hyperbolic multinominal logistic regression. Consequently, this allows us to extend popular calibration methods in Euclidean space to the Poincaré ball in a straightforward manner: Instead of considering the Euclidean predictions – a softmax applied to Euclidean logits – we apply a softmax to hyperbolic logits. Formally, in a hyperbolic space, the probability of class $y \in \{1, ..., k\}$ for hyperbolic output h is given by the softmax:

$$p(\hat{y} := y \mid h) = \frac{\exp(\zeta_y(h))}{\sum_{i=1}^{k} \exp(\zeta_i(h))}\,, \tag{18}$$

where k is the number of classes. Training is performed via the cross-entropy loss, as in the Euclidean networks.

To evaluate the calibration quality, we follow Kull et al. [33] to derive the class-wise Expected Calibration Error

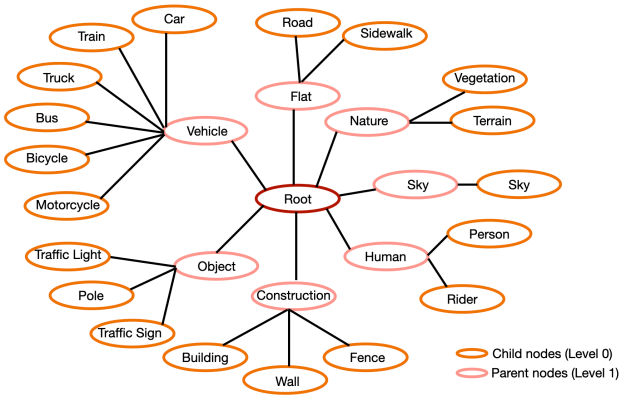


Figure 7. Hierarchical taxonomy derived from Cityscapes [15], divided in 7 parents (in pink) and 19 children (in orange). Same hierarchy is used when testing on Mapillary [42], IDD [57], ACDC [47], BDD [64] and Wilddash [66].

(cwECE). We partition predictions into M equally spaced bins for each class $\{B_{y,m}\}_{m=1,\dots,M}^{y=1,\dots,k}$ and take a weighted average of the difference between the accuracy and confidence for each bin:

$$\text{cwECE} = \frac{1}{k}\sum_{y=1}^{k}\sum_{m=1}^{M}\frac{|B_{y,m}|}{n}|\text{acc}_y(B_{y,m}) - \text{conf}_y(B_{y,m})|\,, \tag{19}$$

where n is the total number of samples of a given class across all bins. $\text{conf}_y(B_{y,m})$ and $\text{acc}_y(B_{y,m})$ are, respectively, the average prediction of class y probability and the actual ratio of class y in bin $B_{y,m}$. cwECE represents the average gap between the predicted confidence and the expected accuracy, averaged over all classes.

C. Hierarchical taxonomy

For reference, in Fig. 7 we provide the hierarchical taxonomy used in our experiments. It follows the same structure from previous work [34] and comprises 7 parent categories (Level 1) and 19 child categories (Level 0).

D. Qualitative results

Here, we provide additional qualitative analysis. Following the same layout as in Fig. 4, the additional examples here illustrate (top-down): label predictions, pixel-level accuracy, and confidence maps. Results on Cityscapes (in-domain) in Fig. 8 show that each model has generally high accuracy and calibration quality. Recall from our evaluation in the main text that parent-level predictions for flat models are on par with hierarchical models (HSSN) even when the hierarchical model has better accuracy on child nodes. Moreover, all models are well calibrated in the in-domain settings both on the parent and child levels.

Under the domain shift, even of a moderate nature, we already observe the advantage of flat models to HSSN. In Fig. 9 for Mappilary, the HSSN model misclassifies "Construction" with "Flat" in both examples, while flat models have an overall higher accuracy in the respective area.

In a scenario with occlusion, flat models are less likely to misclassify compared to the HSSN model. This becomes evident while inspecting the examples in Fig. 10 for the IDD dataset. In the first example, we observe that the HSSN model misclassifies the "Construction" class with "Vehicle", and in the second example, the "Nature" class with "Construction". By contrast, a flat classifier exhibits higher accuracy in the occluded area.

When the evaluated models perform comparably in terms of accuracy, we observe that flat models are better calibrated. In the first example of Fig. 11 for ACDC, we show that the HSSN model has low confidence in the "Construction" label (side of a building) even though the pixel accuracy is high in that area. In the second example, HSSN misclassifies "Construction" with "Road", but has higher confidence than flat models in the area, even though the accuracy is lower.

In more challenging scenarios, hyperbolic models exhibit increased accuracy compared to Euclidean models. In Fig. 12 for the BDD dataset, in both examples, the flat models show improved accuracy over the HSSN model and the hyperbolic model outperforms the Euclidean model. In the first example, in the area where the HSSN model misclassifies "Nature" and "Sky", we see an increase in accuracy as we move from the leftmost example (HSSN) to the rightmost example (flat hyperbolic network). We can observe a similar increase in the accuracy where the HSSN model misclassifies "Construction" with "Vehicle" as we move from left to right. The HSSN model has high confidence in areas that it misclassified, indicating a higher degree is miscalibration than the flat models.

By observing the pixels around the synthetic occlusion on the first example in Fig. 13 (top left corner) for Wilddash, we observe superior accuracy of the hyperbolic model compared to Euclidean and HSSN models. In terms of calibration, we observe that the Euclidean model has an overall higher confidence around the occluded area compared to the hyperbolic model, even though the accuracy is lower in this area. It illustrates our empirical findings that the hyperbolic model has smoother confidence maps. This is also true for more challenging cases, such as the second example in Fig. 13. HSSN and the Euclidean network have an overall lower accuracy than the hyperbolic model. HSSN also has high confidence, despite misclassifying the "Flat" and "Vehicle" classes, while the Euclidean model has low confidence in the correctly predicted areas. By contrast, the hyperbolic model has low certainty in the areas with lower pixel accuracy, suggesting good calibration of these models.

Figure 8. **Two qualitative examples from Cityscapes.** See Appendix D for analysis.

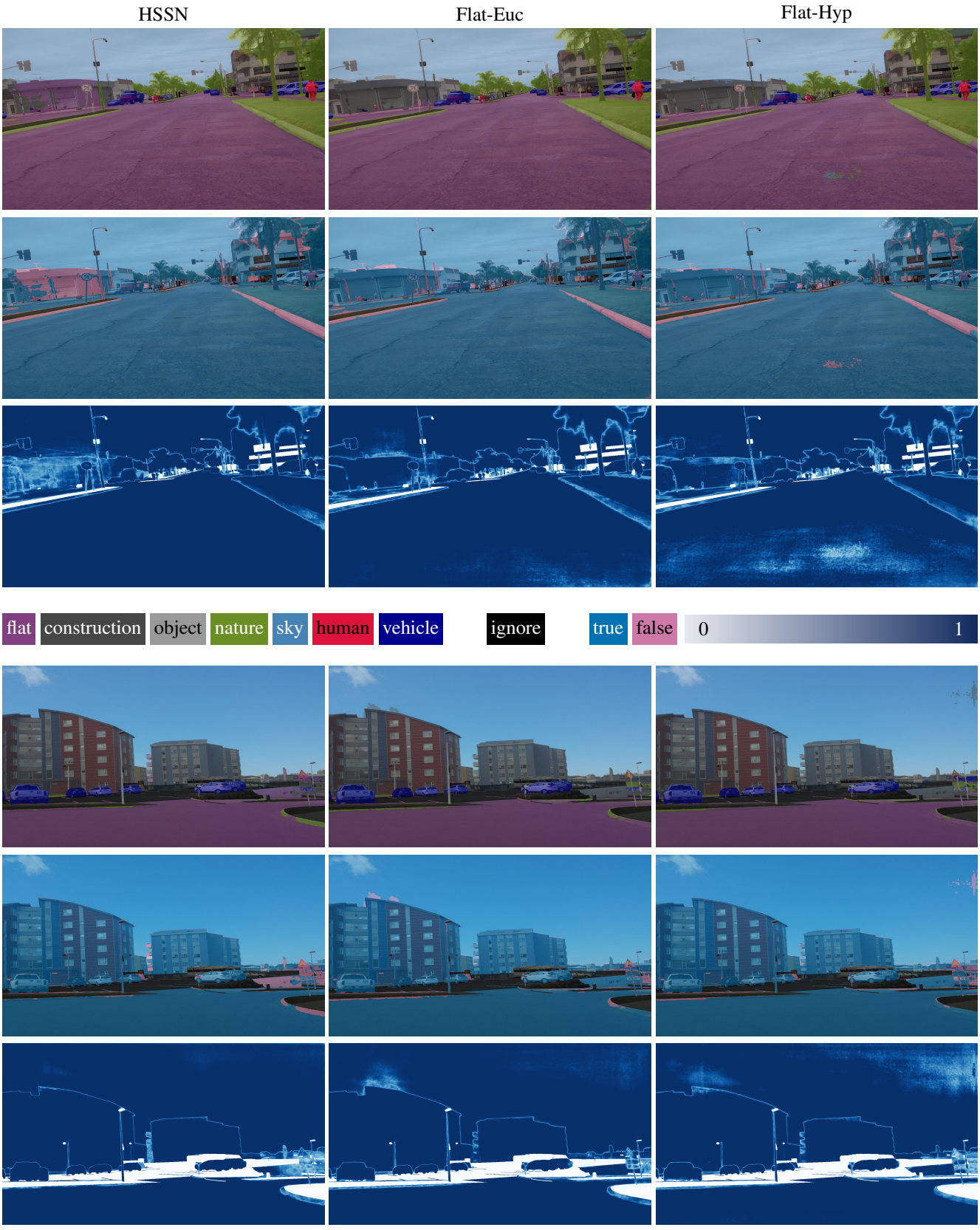


Figure 9. **Two qualitative examples from Mappilary.** See Appendix D for analysis.

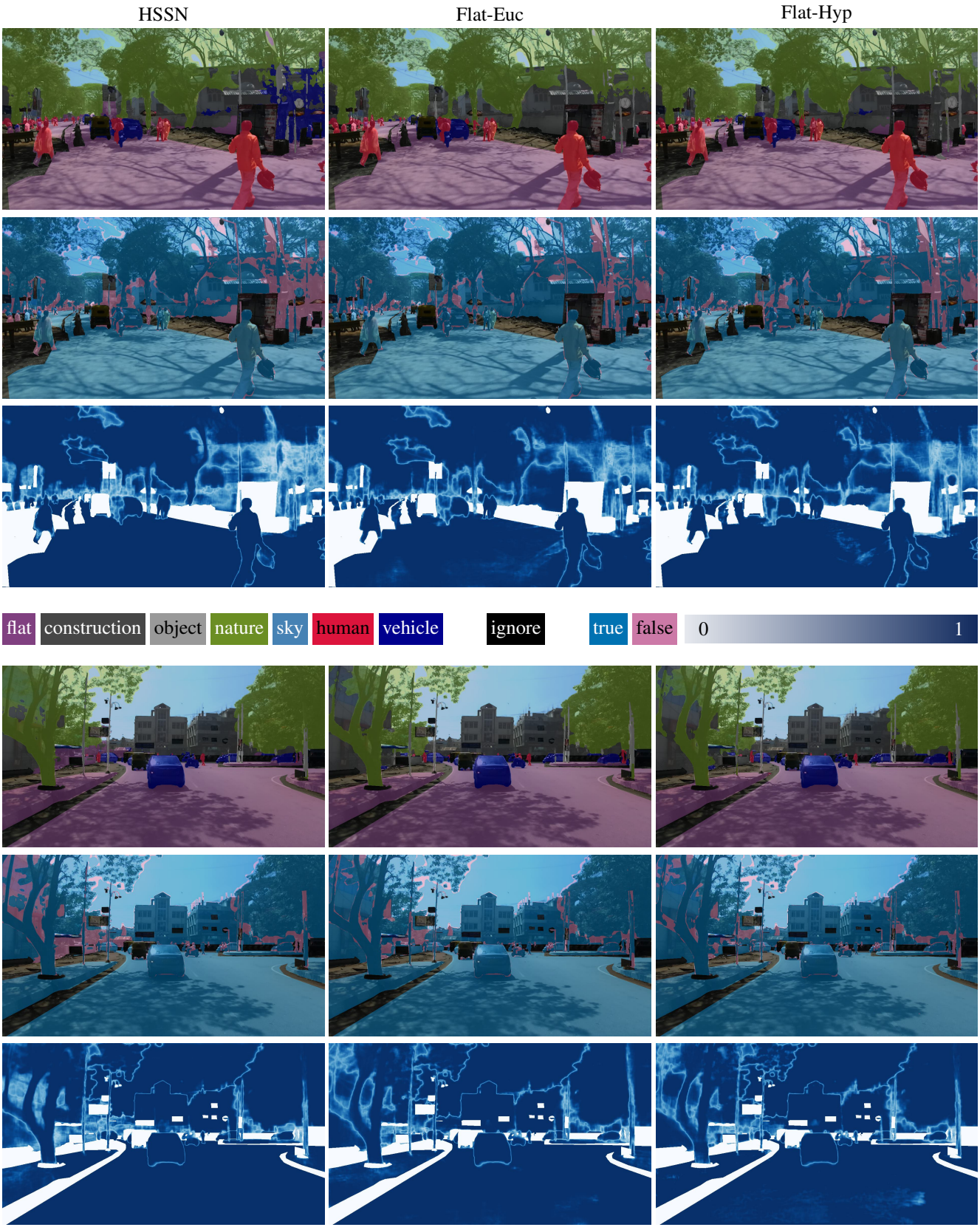


Figure 10. **Two qualitative examples from IDD.** See Appendix D for analysis.

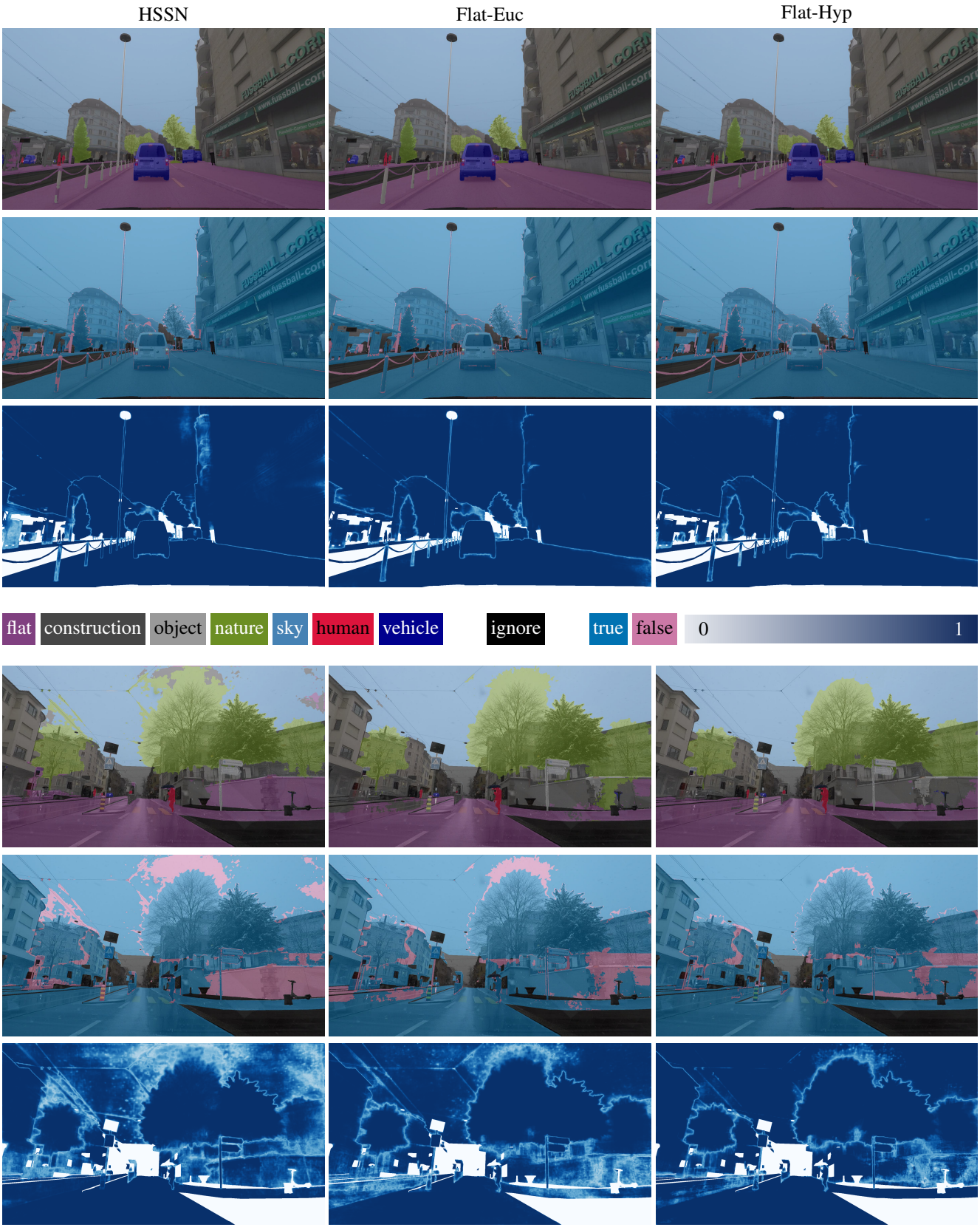


Figure 11. **Two qualitative examples from ACDC.** See Appendix D for analysis.

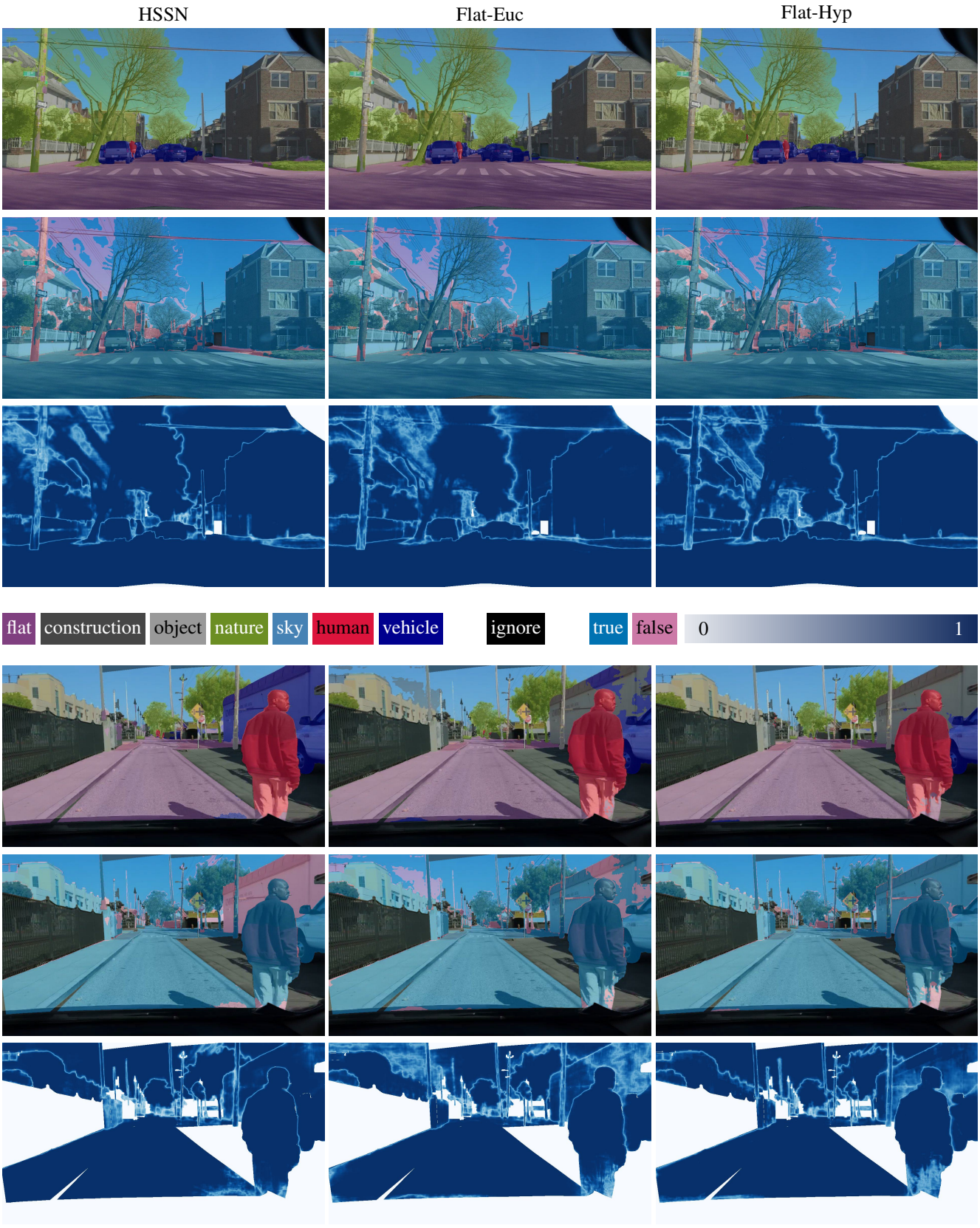


Figure 12. **Two qualitative examples from BDD.** See Appendix D for analysis.

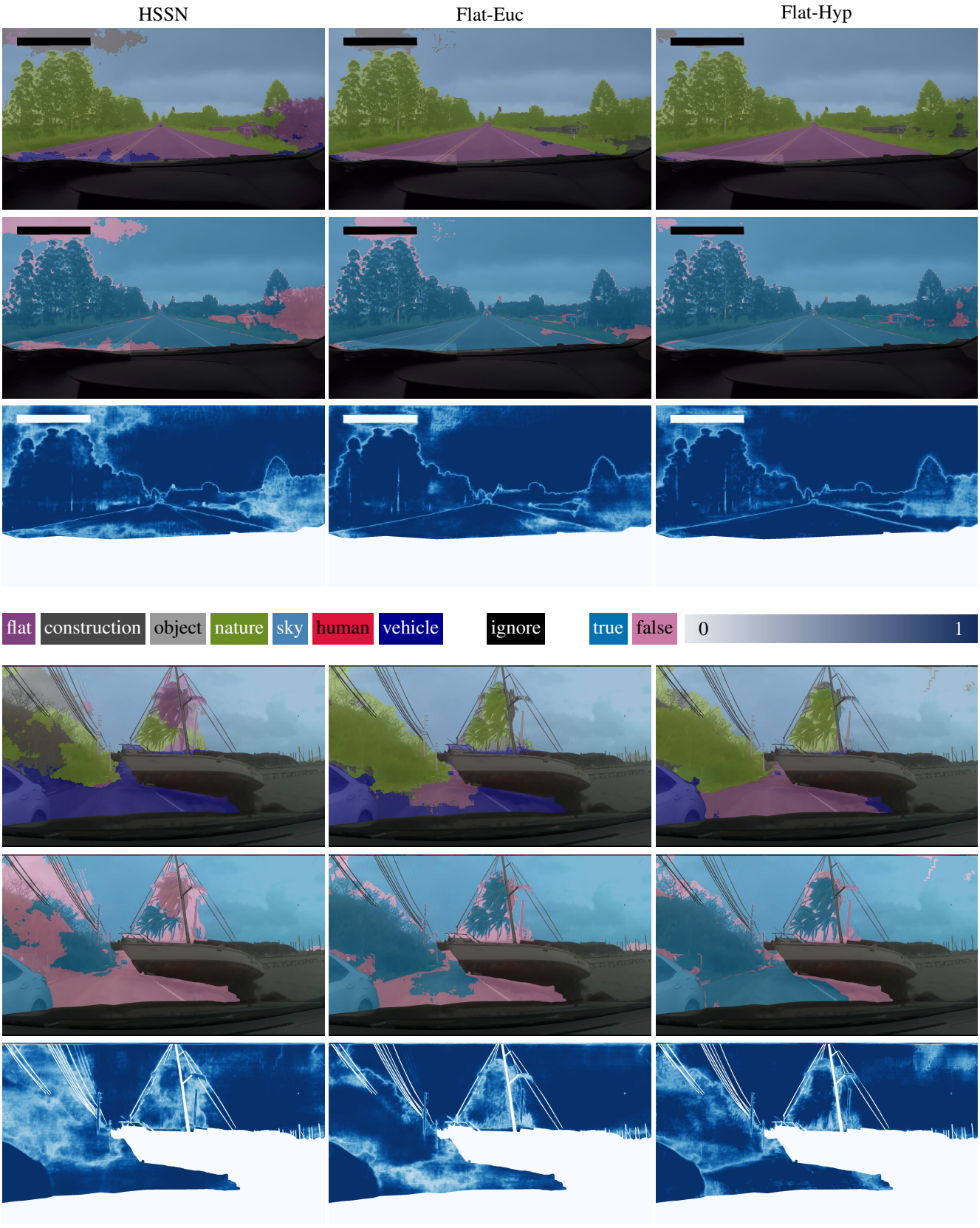


Figure 13. **Two qualitative examples from Wilddash.** See Appendix D for analysis.