SegNBDT: Visual Decision Rules for Segmentation

Alvin Wan[1]*, **Daniel Ho**[1]*, **Younjin Song**[1], **Henk Tillman**[1],
Sarah Adel Bargal[2], **Joseph E. Gonzalez**[1]
UC Berkeley[1], Boston University[2]
{alvinwan, danielho, yjsong, henk, jegonzal}@berkeley.edu
sbargal@bu.edu

Abstract

The black-box nature of neural networks limits model decision interpretability, in particular for high-dimensional inputs in computer vision and for dense pixel prediction tasks like segmentation. To address this, prior work combines neural networks with decision trees. However, such models (1) perform poorly when compared to state-of-the-art segmentation models or (2) fail to produce decision rules with spatially-grounded semantic meaning. In this work, we build a hybrid neural-network and decision-tree model for segmentation that (1) attains neural network segmentation accuracy and (2) provides semi-automatically constructed visual decision rules such as "Is there a window?". We obtain semantic visual meaning by extending saliency methods to segmentation and attain accuracy by leveraging insights from neural-backed decision trees, a deep learning analog of decision trees for image classification. Our model SegNBDT attains accuracy within $\sim$2-4% of the state-of-the-art HRNetV2 segmentation model while also retaining explainability; we achieve state-of-the-art performance for explainable models on three benchmark datasets – Pascal-Context (49.12%), Cityscapes (79.01%), and Look Into Person (51.64%). Furthermore, user studies suggest visual decision rules are more interpretable, particularly for incorrect predictions. Code and pretrained models can be found at github.com/daniel-ho/SegNBDT.

1 Introduction

Neural networks are significantly more accurate than other models for most tasks in computer vision. Unfortunately, they are also significantly harder to analyze and reveal very little about the learned decision process, limiting widespread adoption in high stakes applications such as medical diagnosis. In response, some Explainable AI (XAI) methods work backwards: take a prediction, and conjure an explanation. One approach is to produce saliency maps that highlight pixels influencing predictions the most. These methods (1) focus on the *input* and (2) fail to address the model *prediction process*.

A parallel vein of XAI operates in the reverse direction: start with an interpretable model, and make its accuracy competitive with that of neural networks. In settings where interpretability is critical, decision trees enjoy widespread adoption. However, decision trees do not compete with neural network accuracy on modern segmentation datasets such as Cityscapes. While some [11, 22, 7] have tried to fuse decision-tree and deep-learning methods, the resulting models sacrifice both accuracy and interpretability. Furthermore, these hybrid approaches (1) focus on the model's *prediction process* (e.g., if input is furry, check if input is brown) and (2) fail to relate individual predictions with their *input* (e.g., this pixel is classified *Cat* because of adjacent furry patches in the input image).

In this work, we propose Neural-Backed Decision Trees for Segmentation (SegNBDT), a neural network and decision tree hybrid that constructs a tree in weight-space. Contrary to previous attempts

*Equal contribution

Preprint. Under review.

to produce interpretable segmentation models, SegNBDT produces verifiable *visual* decision rules with semantic meaning, such as "wheel", "sky", or "long vehicle", using three new insights: **(1)** SegNBDT saliency maps *may ignore the target class* to make an intermediate decision *e.g.* focusing on road to determine "not road", when classifying a car pixel. **(2)** Black-box and white-box saliency methods can leverage fine-grained segmentation labels from a *different* dataset to understand general model behavior. **(3)** Unlike post-hoc analyses, the product of our analysis – visual decision rules – are used *directly* for state-of-the-art segmentations.

To analyze visual decision rules, we propose a suite of explainability tools for segmentation: We propose **Minimum Required Context** to identify input to the decision tree for each pixel, refining effective receptive fields for segmentation; **Gradient-weighted Pixel Activation Mapping** (Grad-PAM), a spatially-aware Grad-CAM to support saliency queries for arbitrary pixels; and **Semantic Input Removal** (SIR), a semantically-aware variant of RISE. These methods produce (1) coarse visual decision rules splitting on sets of classes and (2) fine-grained visual decision rules splitting on specific objects and object parts. We summarize our contributions as follows:

1. We propose **SegNBDT**, the first decision-tree-based method to preserve interpretability *and* achieve competitive accuracy on modern, large-scale segmentation datasets – within $\sim$2% of state-of-the-art HRNetV2 on Cityscapes.
2. We identify both coarse and fine-grained **Visual Decision Rules** with semantic meaning, by adapting existing saliency techniques to segmentation.

2 Related Works

Post-hoc Segmentation Explanations: Recent work addresses the lack of interpretability in deep learning methods by supplementing post-hoc explanations like *saliency maps*, which identify pixels in the input that most influenced model predictions [29, 34, 28, 35, 24, 20, 19, 31]. These methods are predominantly for image classification, with only recent interest [9] in saliency maps for dense prediction tasks like segmentation. Regardless of the task, all such methods focus on the *input* and ignore the model's *decision process*.

Interpretable Segmentation Models: Likwise, recent XAI work tackles directly interpretable classification models [32, 18, 23, 1, 4], but analogous work in segmentation is sparse. Such works involve numerous decision trees for segmentation, including a decision-tree-k-means hybrid [33], a decision-tree-SVM hybrid [8, 33] and guided bagging with decision forests [14]. The corresponding deep learning reincarnations for segmentation integrate convolutional filters [11] or whole neural networks into each decision rule [22, 7], and other approaches define hierarchies spatially, hierarchically merging pixels or superpixels [26, 13, 30]. However, all such decision-tree-based approaches to segmentation lose interpretable properties by making leaves impure[27, 38] or employing ensembles [22, 21, 27]. Furthermore, such works operate on well-structured images – medical [14, 7, 21] or satellite imagery [8] – or test on as few as 6 natural images, each with one centered object [33]. Furthermore, these methods interpret the model's *decision process* but ignore the prediction's relationship with the *input*.

Visual Decision Rules for Decision Trees address both the *input* and the model's *decision process*. Work in this area is limited to [12, 36] with complementary downfalls: Li *et al.* [12] analyze a prior deep learning decision tree (dNDF) [10] for MNIST and CIFAR-10. dNDF (a) properly breaks down predictions into sequential decisions, but (b) Li *et al.* 's visualizations are noisy saliency maps that lack perceptible semantic meaning. Zhang *et al.* [36] have the opposite problem: (a) Each decision rule has clear semantic meaning, but (b) the decision tree is not used directly in a sequential, discrete process to make predictions; instead, the decision tree is inferred from a random selection of convolutional filters. Furthermore, both methods are designed for image classification. In contrast, our SegNBDT (a) produces visual decision rules with semantic meaning for segmentation, using a neural-backed decision tree that makes (b) sequential, discrete decisions for its predictions.

3 Method

We introduce our Neural-Backed Decision Tree for semantic segmentation (Sec. 3.1) then present an analysis to produce a visually-interpretable decision tree. First, to determine the input for each

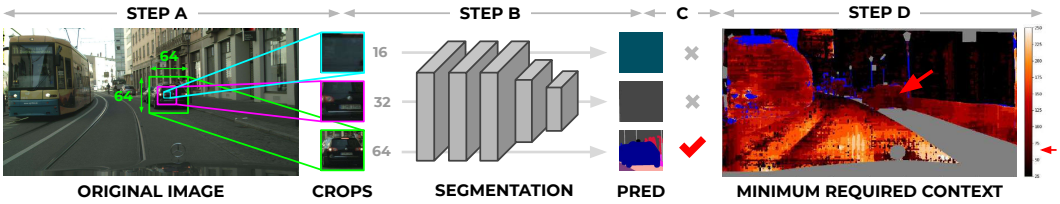

Figure 1: **How to compute decision tree inputs for segmentation**: First, assume we have per-pixel labels; our goal is to find the *smallest* crop such that the pixel prediction is correct. **Step A**: Crop the image. **Step B**: Segment the crop. **Step C**: Repeat until the center pixel prediction is correct, with incrementally larger crops. **Step D**: Repeat for all pixels and obtain a plot for minimum required context. This is useful for analyzing model behavior. However, during inference, we still need to assess decision tree input without labels. To accomplish this, find the *largest* crop such that the pixel prediction is unchanged. The process is otherwise the identical to the above.

decision tree, we define a procedure to determine the spatially-varying receptive field for each pixel (Sec. 3.1). Second, to determine which classes a decision rule looks at (coarse visual decision rule in Sec. 3.3), we present a spatially-aware saliency. Finally, to determine what object part a decision rule looks at (fine-grained visual decision rule in Sec. 3.4), we present a semantically-aware saliency. This produces a state-of-the-art interpretable segmentation model with visual decision rules.

3.1 Neural-Backed Decision Trees for Segmentation

Neural-backed decision trees (NBDT) [32] are decision trees that accept neural features as input and use inner-products for each decision rule [17]. The tree structure is built from the weight vectors of the final fully-connected layer, and the model is trained end-to-end using a surrogate loss, for image classification.

We extend the NBDT training procedure to segmentation, using the fact that a collection of 1×1 convolutions is simply a fully-connected layer: Each row of the fully-connected layer corresponds to a class. In the same way, each 1×1 filter corresponds to a class. This allows SegNBDT to inherit (1) interpretable properties – pure leaves and sequential decisions – and (2) accuracy competitive with state-of-the-art neural networks.

However, NBDTs exhibit one major limitation: NBDTs do not ground decision rules in the input, yielding only hypothesized decision rules such as *Animal* vs. *Vehicle*. In contrast, our method diagnoses *visual decision rules* that split on sets of objects such as *long vehicle* or object parts such as *wheel* (Fig. 7). We present methods of diagnosing coarse and fine-grained visual decision rules in Sec. 3.3, 3.4. In this work, we focus on visual decision rules for segmentation.

3.2 Diagnosing Decision Tree Input

To understand *how* the image is used for decisions, we need to first understand *which part* of the image is actually used for decisions. This informs our decision rule analysis, understanding whether a decision rule uses texture from a small 5×5 patch or an entire vehicle in a 400×400 patch.

Effective receptive fields (ERF) [15] claim that the receptive field for all pixels is the same size. To remedy this, we instead diagnose decision tree inputs by computing *the smallest image crop such that the model's prediction is correct*. This yields the minimum amount of context required for each pixel's prediction i.e., the pixel's decision tree input. To compute **Minimum Required Context (MRC)**, we use the following algorithm (also depicted in Figure 1):

1. **Step A**: Pick a pixel (x, y) in the input, and take crop of size $m \times m$ around that pixel, $I^m_{x,y}$. We use superscripts for crop sizes and subscripts for crop center position.
2. **Step B**: Run inference on that crop $J(I^m_{x,y})$. This is possible due to the fully-convolutional nature of the base neural network, J.
3. **Step C**: Keep only the center pixel prediction $J(I^m_{x,y})^1_{x,y}$. Repeat with incrementally larger crops until the center pixel prediction is correct, *i.e.* $J(I^m_{x,y})^1_{x,y} = \ell^1_{x,y}$ for the label ℓ. We use n different crop sizes $m \in \{\beta i\}^n_{i=1}$. In our experiment, we use $n = 10$ and $\beta = 25$.

Figure 2: **Coarse Visual Decision Rules**: Take a decision tree (left) that has two leaves – *Car* and *Person*. There are two options for the decision rule at the root: The decision rule can ask "Is there a car?" (Rule Option A), selecting *Car* if so and *Person* otherwise. The decision rule could also ask "Is there a person?" (Rule Option B). For each node in the decision tree, our method (Sec. 3.3) picks the object that the decision rule splits on (Fig. 3), based on saliency. In this example, our analysis would pick either *Car* or *Person*. Fine-grained visual decision rules (Sec. 3.4) then determine object *parts* that the decision rule splits on.

4. **Step D**: The smallest crop size with a correct prediction is the *minimum required context* for that pixel. Repeat for all pixels. We then plot minimum required context for all pixels.

MRC is defined as the following, for crop sizes m, indices (x, y), model J, image I, and label ℓ:

$$\text{MRC}(x, y) = \text{argmin}_{m \in \{\beta i\}_{i=1}^{n}} \mathbb{1}\left\{J(I^m_{x,y})^1_{x,y} = \ell^c_{x,y}\right\}. \tag{1}$$

We use the above analysis to understand model behavior. Using MRC for all pixels in the input, we can correlate aggregate statistics with other properties like average object size.

However, when running inference on a new image, we do not have ground truth labels to compare against. As a result, during inferenece, *without labels*, we compute input size with a slightly modified objective: Our goal is to incrementally try smaller crop sizes until the prediction changes. We refer to this as the **Unsupervised Minimum Required Context (UMRC)**, defined as:

$$\text{UMRC}(x, y) = \text{argmin}_{c \in \{\beta i\}_{i=1}^{n}} \mathbb{1}\left\{J(I^m_{x,y})^1_{x,y} = J(I_{x,y})^m_{x,y}\right\}. \tag{2}$$

3.3 Coarse Visual Decision Rules: Spatially-Aware Saliency for Segmentation

Our goal is to procure a coarse visual decision rule that answers the following question: Given two child nodes, one with classes from set A and the other from set B, which set does the decision rule look for in the input (Fig. 2)? To answer this, we visualize saliency for each decision rule (Fig. 3).

However, saliency methods such as Grad-CAM are not designed to preserve spatial information. Recall Grad-CAM [24] takes a weighted average of the last convolution's k feature maps A^k. These weights $\alpha_k^{(c)}$ are spatially-averaged gradients of the classification model's output $\hat{y} = J(I)$ for class c ($\hat{y}^{(c)}$). More formally, with spatial indices (x, y):

$$\mathcal{L}^{(c)}_{\text{CAM}} = ReLU\left(\sum_k \alpha_k^{(c)} A^k\right) \quad \text{where} \quad \alpha_k^{(c)} = \overbrace{\frac{1}{N}\sum_{x,y}}^{\text{avg pool}} \frac{\partial \hat{y}^{(c)}}{\partial A^k_{x,y}}. \tag{3}$$

When computing $\alpha_k^{(c)}$, the global average pool discards all spatial information. This is problematic for segmentation model saliency maps, as Grad-CAM for all car pixels will look largely identical, even for predictions 1000 pixels apart. Our **Gradient-weighted Pixel Activation Mapping (Grad-PAM)** introduces a simple fix by removing the global average pool, for a matrix-valued importance weight G_k^c, hadamard product $\circ$, and a segmentation score for class c at index (x, y) ($\hat{Y}^{(c)}_{x,y}$):

$$\mathcal{L}^{(c)}_{\text{PAM}}(x, y) = ReLU\left(\sum_k G_k^{(c)} \circ A^k\right) \quad \text{where} \quad G_k^{(c)} = \frac{\partial \hat{Y}^{(c)}_{x,y}}{\partial A^k}. \tag{4}$$

Critically, in contrast to grid saliency [9], we inherit the Grad-CAM $ReLU$ instead of applying an absolute value, as the former qualitatively discriminates between relevant and irrelevant pixels more effectively (Fig 4). To support Grad-PAM over groups of pixels, we simply sum saliency maps across all pixels of interest, S: $\mathcal{L}^{(c)}_S = \sum^n_{(x,y)\in S} \mathcal{L}^{(c)}_{\text{PAM}}(x, y)$.

Using spatially-aware saliency Grad-PAM, our goal is now to understand what each decision rule in our SegNBDT model is splitting on, visually. We first pick a node's decision rule to diagnose. Note

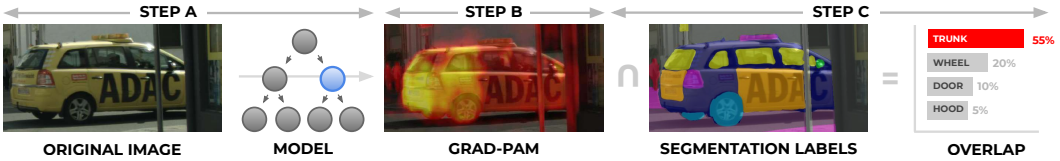

Figure 3: **How to visually-ground decision rules**: At a high level, compute the overlap between salient portions of the image with segmentation labels. Specifically, start by picking a node to analyze. In **Step A**, run inference on the image, using either NBDTs for classification or segmentation. **Step B**, Obtain Grad-PAMs for the node's output. **Step C**, Compute overlap between Grad-PAM and segmentation labels. The segmentation class with the highest overlap (which is *Wheel* in the above example) is the semantic, visual feature responsible for the decision rule.

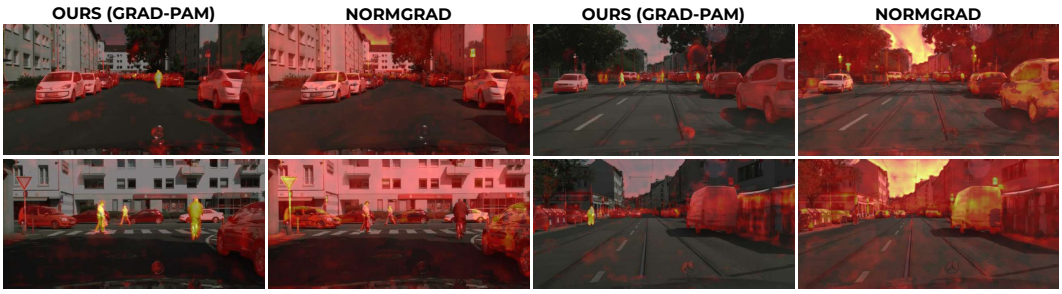

Figure 4: **Segmentation Saliency Techniques**: A saliency map, as prescribed in previous explainability work [24, 25] should be class discriminative. The above visualizes a *Person* decision rule in SegNBDT. To understand global class discrimination, we take the average of saliency maps for all pixels. Notice that (NormGrad) roughly highlights all pixels. However, sticking with $ReLU$ (Grad-PAM, Ours) produces far more class discriminative saliency maps that properly highlight people. Note that grid saliency, which uses absolute values, would suffer from the same issue.

that instead of deciding between classes, each node i is deciding between child nodes j, where each child node's subtree contains classes C_j. Define node i's score for child node j to be $y(i)^{(j)}$ and the corresponding Grad-PAM to be $\mathcal{L}(i)^{(j)}$. As before, we use a superscript with parentheses to denote the output's score for a particular class. Second, we compute overlap between Grad-PAM and the dataset's provided segmentation labels, taking the set of classes with the highest Grad-PAM weight:

$$\text{argmax}_j \sum_{c \in C_j} \sum_{x,y} \mathbb{1}\{\mathcal{L}(i)^{(j)}_{x,y} = c\}. \tag{5}$$

Done naively, this means large objects *e.g.* road will almost certainly receive the most total saliency. As a result, third, we normalize by the number of pixels belonging to each class c: $\sum_{x,y} \mathbb{1}\{\ell_{x,y} = c\}$. The coarse visual decision rule then outputs a set of classes C_j, which we formally define to be:

$$C(i) = \text{argmax}_j \sum_{c \in C_j} \frac{\sum_{x,y} \mathbb{1}\left\{\mathcal{L}(i)^{(j)}_{x,y} = c\right\}}{\sum_{x,y} \mathbb{1}\{\ell_{x,y} = c\}}. \tag{6}$$

3.4 Fine-Grained Visual Decision Rules: Semantically-Aware Saliency for Segmentation

While Grad-PAM provides node-level and class-level discrimination, our goal is to procure a finer-grained decision rule based on specific object parts. To find the most salient object parts, we use intuition from other black-box saliency techniques like RISE [19] and LIME [20]: removing the most important portions of the image will degrade accuracy the most. Unlike Grad-CAM and other saliency techniques, our goal is not to find the most salient *pixels* but the most salient *semantic* image parts. There are two challenges in applying existing black-box techniques:

Problem 1: Prohibitive Inefficiency of Sampling Subsets: RISE must test a large number of image subsets. This is possible with classification, which RISE was originally designed for, but adequately

Table 1: **SegNBDT Accuracy** remains within ∼2-4% of the base neural network's on 3 popular segmentation benchmarks – Pascal-Context, Cityscapes, and Look Into Person(LIP). We use a state-of-the-art neural network HRNetV2 and the corresponding SegNBDT models attain state-of-the-art accuracy for decision-tree-based methods. We note that all previous decision-tree-based methods are run on highly-specialized (satellite) or sparse datasets of as little as 6 natural images. SegNBDT models are the first decision-tree-based models to be run on modern computer vision segmentation datasets.

Dataset	SegNBDT-S (Ours)	SegNBDT-H (Ours)	HRNetV2 Size	NN Acc	Δ
Pascal-Context	49.12%	–	W48	52.54%	3.42%
Cityscapes	67.53%	67.33%	W18-Small	70.3%	2.77%
Cityscapes	79.01%	–	W48	81.12%	2.11%
Look Into Person	51.64%	–	W48	55.37%	3.73%

Table 2: **Average Minimum Required Context** (MRC) for each class in Cityscapes, compared against object sizes and class frequency. Note that MRC is not strictly lower with more frequent classes. In fact, the most frequent class *Road* is one of the classes requiring most context.

Metric	Building	Vegetation	Sky	Motorcycle	Traffic Light	Traffic Sign	Terrain	Person	Bus
Average MRC	30.662	33.734	41.206	52.348	54.235	55.003	59.52	76.837	78.479
Average Height	415.55	242.31	295.78	91.48	42.62	36.16	83.89	106.35	151.89
Class Frequency	21.92%	17.32%	3.35%	0.08%	0.20%	0.67%	0.83%	1.30%	0.39%

Bicycle	Car	Truck	Rider	Pole	Train	Sidewalk	Road	Fence	Wall
83.183	86.928	97.129	101.626	102.957	107.967	113.968	124.137	130.153	134.355
90.69	94.55	137.8	110.69	139.12	172.96	179.95	621.09	109.75	100.14
0.71%	6.51%	0.30%	0.22%	1.48%	0.11%	5.41%	37.65%	0.82%	0.73%

covering the space of all possibilities is far less likely for an $\sim 8\times$ larger image: 1080×512 for segmentation as opposed to 224×224 for classification.

Instead of applying random masks, we can use an auxiliary dataset with object part segmentation labels: simply use the segmentation labels as masks.

Problem 2: Zero Masks are Ineffective: RISE removes objects from the image by replacing pixel values with zeros. However, these zero masks do not properly remove objects: Neural networks can both segment and predict the masked-out object using (a) the shape, (b) the discontinuity between zero and non-zero pixels, and (c) mis-aligned local image statistics like mean.

To ameliorate zero-masks, we thus shuffle pixels to preserve image statistics and lessen discontinuity between the "removed" object and its context. In general, shuffling values to ascertain feature importance is not new: this is known as permutation feature importance [2] or model reliance [5]; we simply apply this to images in pixel-space. See Appendix C for empirical support for pixel shuffling.

We refer to this semantic, modified RISE as **Semantic Input Removal** (SIR): for each *object part* (instead of random mask) in an auxiliary segmentation dataset, remove the object part (via pixel shuffling) and gauge accuracy damage. The object parts that most damage the decision rule's accuracy are the object parts this decision rule splits on.

4 Experiments

In this section, we demonstrate state-of-the-art results for interpretable models, with SegNBDT. Furthermore, we show the first set of visually-interpretable decision rules for segmentation, which we describe as visual decision rules, and validate their efficacy with user studies.

Our **SegNBDT** model attains state-of-the-art performance for decision-tree-based models on three segmentation benchmark datasets, achieving accuracy within ∼2-4% of the base state-of-the-art neural network on Pascal Context [16], Cityscapes [3], and Look Into Person [6] (Table 1).

We show that **Minimum Required Context (MRC)** varies drastically for different pixels in the image, from 25×25 windows to 250×250 windows (Figure 5). To reduce computational cost, we use the center 11×11 pixels so that the MRC condition becomes $J(I^m_{x,y})^{11}_{x,y} = \ell^c_{x,y}$. Objects with less distinctive texture require more context to correctly segment e.g., sidewalk, road, wall. In contrast, image parts with more distinctive texture require less context e.g., sky, vegetation (Table 2).

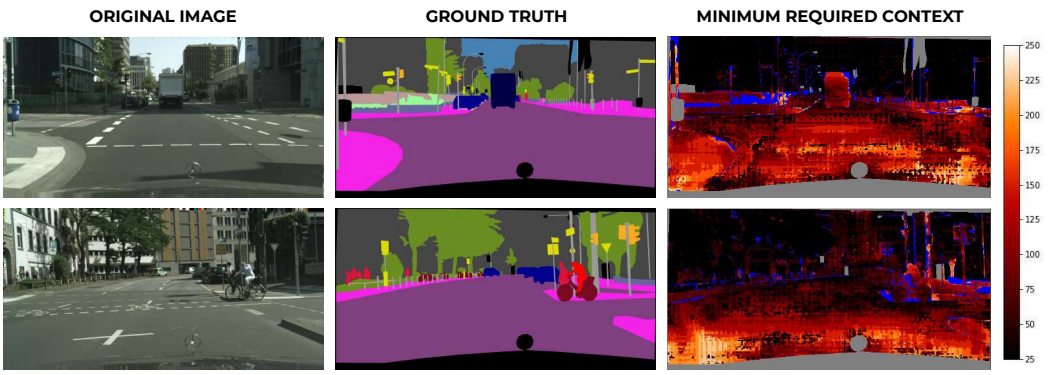

Figure 5: **Minimum Required Context**: The left-most column contains the original image, the middle column contains the ground truth labels, and the right-most column shows minimum required context for each pixel. Lighter colors indicate more context is required to correctly classify that pixel, with white denoting a 250×250 window and black denoting a 25×25 window. Note that portions of the image with less distinctive texture e.g., road require more context than portions of the image with highly-distinctive texture e.g., vegetation. Gray are "ignore" pixels, per Cityscapes, and blue pixels are those that network mis-classified for all possible contexts.

Table 3: **User studies** We distribute surveys to two groups of people: those knowledgeable about machine learning ("Pre-qualified") and those not ("Non pre-qualified"). Note mechanical turks in the latter category did not pick "Neither". Both groups prefer our SegNBDT visual decision rules to Grad-CAM (59.9% vs. 33.7%), especially when the segmentation prediction is incorrect (second row, 80.0% vs. 12.5%).

	Pre-qualified				Non-pre-qualified			Total			
	#	SegNBDT	Grad-CAM	Neither	#	SegNBDT	Grad-CAM	#	SegNBDT	Grad-CAM	Neither
All	190	50%	35.8%	14.2%	239	67.78%	32.2%	429	59.9%	33.7%	6.3%
Miss	47	78.7%	8.5 %	12.8%	33	81.8%	18.2%	80	80.0%	12.5%	7.52%

Coarse Visual Decision Rules are highly-discriminative with Grad-PAM: In particular, (a) along a single path from a leaf to the root, each node focuses on successively more relevant image portions (*e.g.* Fig 6, from *Person* 1 to *Cyclist* 5) and (b) for paths to different leaves, each node focuses on drastically different portions of the same image (*e.g.* Fig 6, from *Person* 1 to *Long Vehicle* 3). Note that saliency does not necessarily shrink with deeper nodes, leveraging context as needed (Fig. 8).

For **Fine-grained Visual Decision Rules** with SIR, we use ADE20k [37] for our source of fine-grained object part annotations. As demonstrated in Fig. 7, removing car door appears to decimate car accuracy the most, but after normalizing by object size, it becomes apparent that car parts *specific to a car* are more critical – *e.g. Headlight*. Objects shared with other classes, such as *Window* or *Wheel*, lag behind. Our final series of visual decision rules is shown in Fig. 8.

We conduct **user studies** on 1000 randomly-selected SegNBDT visualizations with visual decision rules. Between SegNBDT and Grad-CAM, we ask users to pick an explanation that better explains *why* the model makes a certain pixel prediction, from "A", "B", or "Neither is more interpretable". Survey details can be found in Appendix D. We survey two groups: participants in the first group are pre-qualified as having machine learning knowledge. Of 429 total evaluations across both groups (Table 3), 59.9% favor SegNBDT and 35.8% favor the baseline. The gap widens when considering only incorrect segmentation predictions: 80.0% favor SegNBDT and only 12.5% favor the baseline. This suggests our visual decision rules are more interpretable, particularly for incorrect predictions.

5 Conclusion

We present SegNBDT, an interpretable segmentation model that establishes competitive accuracy with state-of-the-art neural networks on modern, large-scale segmentation datasets. We furthermore propose extensions for saliency methods – the spatially-aware Grad-PAM and semantically-aware SIR – to uncover semantic, visual decision rules in our neural-backed decision tree for segmentation. This culminates in the first high-accuracy and visually-interpretable model for segmentation.

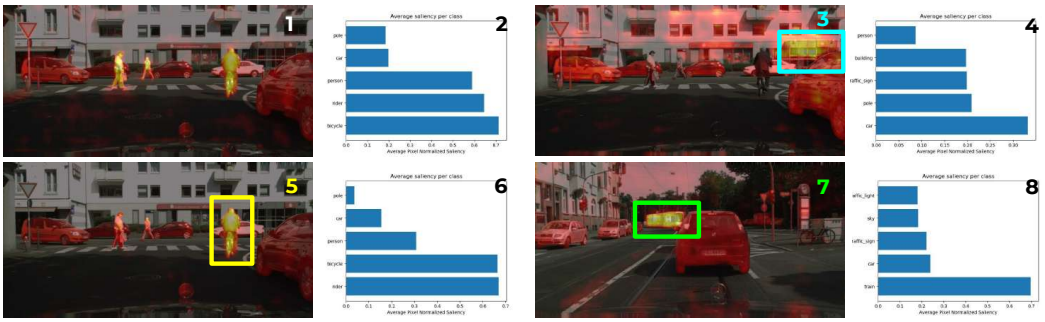

Figure 6: Note that Grad-PAM for nodes on the same path feature perceptible qualitative and quantitative differences: (1) highlights people-related classes broadly, with high overlap (2) across *Person*, *Rider*, *Bicycle*. However, the node for (5) is deeper than the node for (1), focusing more specifically on cyclists (yellow), with high overlap (6) for *Bicycle*, *Rider*. Furthermore, nodes on different paths focus on drastically different items – in contrast to (1) and (5), *e.g.* the *Long Vehicle* node looks for series of windows (3, blue) in the absence of *Truck*, *Bus*, or *Train*. To double-check the *Long Vehicle* node saliency, note saliency shifts focus correctly (7, 8) when a train (green) appears. Lighter colors (closer to white) indicate higher saliency values.

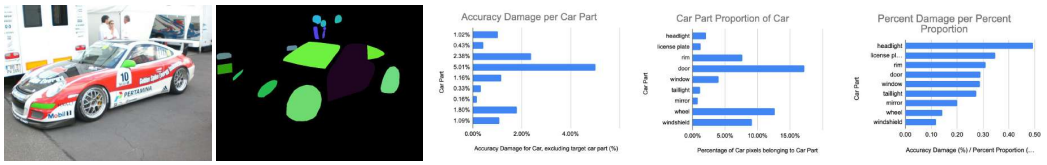

Figure 7: Our semantic-aware black-box saliency method SIR identifies car parts most critical for SegNBDT's *Car* vs. *Not Car* node, featuring *Headlight*. Parts shared with other classes such as *Window* are assigned far less importance.

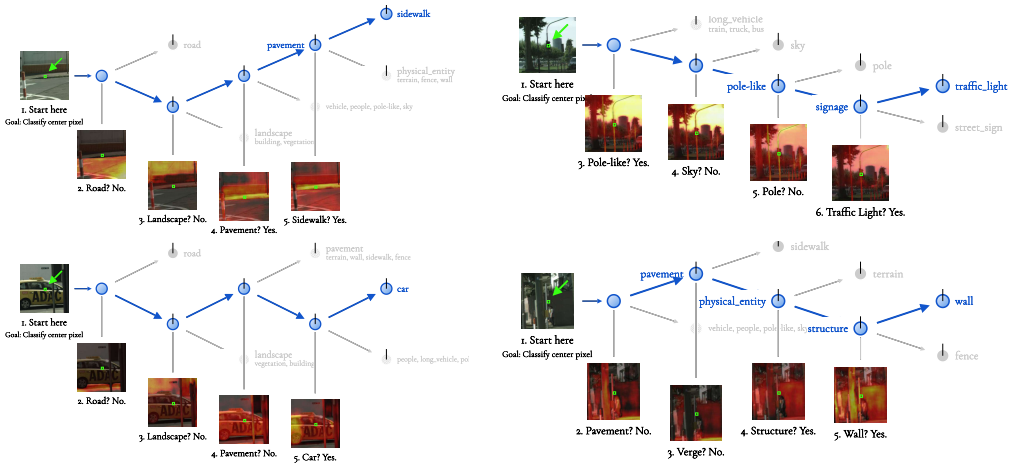

Figure 8: Examples of our final SegNBDT's visual decision rules. Saliency maps for successively deeper decisions do not necessarily shrink: **(Top Left)** To segment pavement, the model focuses on only *Building* and *Sidewalk*. However, for the final decision rule, the model percetibly highlights *Road* as well to determine the sidewalk class. **(Bottom Left)**: The same phenomeon occurs: To discern *Car* from pavement, saliency only lightly focuses on the car. To finally distinguish *Car* from other vehicles and people, saliency focuses on the trunk and wheel. **(Right)**: In other cases, saliency maps grow successively smaller as SegNBDT rules out other classes, like *Sky*. More examples can be found in the Appendix. Lighter colors (closer to white) indicate higher saliency values.

Acknowledgments and Disclosure of Funding

In addition to NSF CISE Expeditions Award CCF-1730628, UC Berkeley research is supported by gifts from Alibaba, Amazon Web Services, Ant Financial, CapitalOne, Ericsson, Facebook, Futurewei, Google, Intel, Microsoft, Nvidia, Scotiabank, Splunk and VMware. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 1752814.

Broader Impact

A large number of machine learning applications, even those beyond the canon high-stakes applications, can benefit from explainable models. In this work, we extend explainability beyond image classification to a real-world task with higher-resolution inputs and dense pixel predictions – namely, segmentation, constructing an interpretable and accurate neural-network-and-decision-tree hybrid. In particular, these new explainability techninques are used to diagnose each decision rule, asking: "What does the decision rule split on, in the image?" This allows practitioners to better understand model misclassification – as shown by our user studies – and for users at large to better understand model predictions in everyday life, be it music recommendations of a semi-autonomous vehicle's recommendation. For broader societal implications, an interpretable and accurate model that can compete with state-of-the-art neural networks means deploying an interpretable model in production, is possible.

Explanations can be harmful for both correct and incorrect predictions. When human vision is impeded (*i.e.* the image is low-resolution or features dimly-lit scenes), the user may treat saliency as ground truth to provide evidence for holding individuals liable – for example, for a car accident. High accuracy, interpretable segmentation models can also mislead practitioners for incorrect justifications – both the user of the final product or the developer.

References

[1] K. Ahmed, M. Baig, and L. Torresani. Network of experts for large-scale image categorization. volume 9911, April 2016.

[2] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[4] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3450–3457. IEEE, 2012.

[5] A. Fisher, C. Rudin, and F. Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.

[6] K. Gong, X. Liang, D. Zhang, X. Shen, and L. Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 932–940, 2017.

[7] T. M. Hehn, J. F. Kooij, and F. A. Hamprecht. End-to-end learning of decision trees and forests. *International Journal of Computer Vision*, pages 1–15, 2019.

[8] B. W. Heumann. An object-based classification of mangroves using a hybrid decision tree—support vector machine approach. *Remote Sensing*, 3(11):2440–2460, 2011.

[9] L. Hoyer, M. Munoz, P. Katiyar, A. Khoreva, and V. Fischer. Grid saliency for context explanations of semantic segmentation. In *Advances in Neural Information Processing Systems*, pages 6459–6470, 2019.

[10] P. Kontschieder, M. Fiterau, A. Criminisi, and S. Rota Bulo. Deep neural decision forests. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[11] D. Laptev and J. M. Buhmann. Convolutional decision trees for feature learning and segmentation. In *German Conference on Pattern Recognition*, pages 95–106. Springer, 2014.

[12] S. Li and K.-T. Cheng. Visualizing the decision-making process in deep neural decision forest. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 114–117, 2019.

[13] T. Liu, M. Seyedhosseini, and T. Tasdizen. Image segmentation using hierarchical merge tree. *IEEE transactions on image processing*, 25(10):4596–4607, 2016.

[14] H. Lombaert, D. Zikic, A. Criminisi, and N. Ayache. Laplacian forests: semantic image segmentation by guided bagging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 496–504. Springer, 2014.

[15] W. Luo, Y. Li, R. Urtasun, and R. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pages 4898–4906, 2016.

[16] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014.

[17] S. K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of artificial intelligence research*, 2:1–32, 1994.

[18] V. N. Murthy, V. Singh, T. Chen, R. Manmatha, and D. Comaniciu. Deep decision network for multi-class image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[19] V. Petsiuk, A. Das, and K. Saenko. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[20] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

[21] D. L. Richmond, D. Kainmueller, M. Yang, E. W. Myers, and C. Rother. Mapping stacked decision forests to deep and sparse convolutional neural networks for semantic segmentation. *arXiv preprint arXiv:1507.07583*, 2015.

[22] S. Rota Bulo and P. Kontschieder. Neural decision forests for semantic image labelling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 81–88, 2014.

[23] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5506–5514, 2016.

[24] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 618–626, 2017.

[25] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *International Conference on Computer Vision (ICCV) 2017*, 2016.

[26] A. Sharma, O. Tuzel, and D. W. Jacobs. Deep hierarchical parsing for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 530–538, 2015.

[27] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, et al. Efficient human pose estimation from single depth images. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2821–2840, 2012.

[28] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[29] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. Striving for simplicity: The all convolutional net. *CoRR*, abs/1412.6806, 2014.

[30] C. Straehle, U. Koethe, and F. A. Hamprecht. Weakly supervised learning of image partitioning using decision trees with structured split criteria. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1849–1856, 2013.

[31] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. *International Conference on Machine Learning (ICML) 2017*, 2017.

[32] A. Wan, L. Dunlap, D. Ho, J. Yin, S. Lee, H. Jin, S. Petryk, S. A. Bargal, and J. E. Gonzalez. Nbdt: Neural-backed decision trees. *arXiv preprint arXiv:2004.00221*, 2020.

[33] L. Yaron and A. Loai. Decision trees based image segmentation using ensemble clustering. *International Journal of Scientific Research*, 8(10), 2019.

[34] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, pages 818–833. Springer, 2014.

[35] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision (ECCV)*, pages 543–559. Springer, 2016.

[36] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu. Interpreting cnns via decision trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6261–6270, 2019.

[37] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[38] Y. Zuo and T. Drummond. Fast residual forests: Rapid ensemble learning for semantic segmentation. In *Conference on Robot Learning*, pages 27–36, 2017.

A Figures

We include many more example figures, extensions of Fig. 6, Fig. 7, and Fig. 8. We include more examples of coarse visual decision rules (Fig. 9), fine-grained visual decision rules (Fig. 10), and the final visual decision rule figure for SegNBDT (Fig. 11).

B "Default" Predictions

The minimum required context maps exhibit a strange phenomenon: visually-similar pixels of different classes have vastly different context requirements. This is shown in Figure 12 (A). The top-left (1) is the original image, demonstrating the nearly-indistinguishable *Building*, *Road*, and *Sidewalk* pixels. The bottom-left (2) shows the labeled *Sidewalk* (pink), *Building* (gray), and *Road* (purple). The MRC map (3) shows the stark disparity in minimum required context for all three classes. As a result of this observation, we hypothesize that the neural network has preferences for a "default" prediction; in the absence of further evidence, the neural network will default to a certain class. Both aggregate MRC statistics and the SegNBDT hierarchy support this hypothesis.

1. **"Default" classes require less context on average**: Per Table 2, the class *Building* requires the least context to classify correctly on average, of all classes. This motivates the model's bias towards a *Building* "default" class when provided less context.
2. **"Default" classes occur at shallower depths in the hierarchy**: Per the induced hierarchy in Figure 13, the *Building* class is at a shallower depth than the *Sidewalk* class.
3. **"Default" classes are not the most common class**: Although the "default" class *Building* is preferred to *Road*, *Road* is actually the most common class in Cityscapes, making up 37.3% of pixels. Thus, "default" classes are not simply determined by class frequency.

The interpretability of the SegNBDT hierarchy allows us to reason about the "default" prediction phenomena. A vanilla neural network, by contrast, would not allow us to introspect the model's decision process.

C Pixel Shuffling for Object Removal

We show the ineffectiveness of zero masks by evaluating a fully-trained HRNetV2 on Cityscapes with a car accuracy of 53.1%. We replace all car pixels with zeros in the validation set, but the model still retains a car accuracy of 35.9%. Shuffling all car pixels, instead of replacing with zeros, further decimates car accuracy to 32.4%. Since pixel reshuffling results in lower car accuracy, we conclude shuffling is more effective than zero masks for object removal.

D Survey

The survey compares (1) a baseline Grad-CAM with (2) our SegNBDT with visual decision rules. The below describes our survey setup and how it was administered. For survey results, see the main manuscript Sec. 4.

Figures For each figure, we first sample a class uniformly at random. Second, we sample a random pixel from the Cityscapes validation set, where the model predicts that class. Third, we then generate two explanations for that pixel's prediction, using the two explainability techniques. Note the prediction may be correct or incorrect.

Grad-CAM inherently loses spatial information, as described in Sec. 3.3, but Grad-PAM, used to visualize our decision rules, does not. For comparable saliency maps, we compute Grad-PAM over all pixels in the cropped input.

Administration As shown in the example (Fig. 14), each figure simply denotes one explanation as "Explanation A" and the other as "Explanation B". Every other figure switches the order of the explanations. We administer this survey to two groups of participants:

1. **Pre-qualified** participants have machine learning understanding. This survey, with only the first 100 figures, was administered to undergraduates in a machine learning course, using 10 Google Forms with 10 questions each. Each participant was presented with a random ordering of the 10 forms and was asked to complete as many as desirable.
2. **Non-pre-qualified** participants do not have machine learning understanding. This survey, with 1000 figures, was administered to mechanical turks.

Figure 9: More examples of Grad-PAM applied to all pixels in the image. Each column represents a different node in the SegNBDT hierarchy: A (Sidewalk vs. Verge), B (Person vs. Rider), C (Pavement vs. Not Pavement), D (Car vs. Not Car), E (Building vs. Vegetation), F (Traffic Light vs. Traffic Sign). In particular, note the following: In column A, saliency highlights objects generally close to the ground but offroad. Column B in turn highlights people fairly well, with small amounts of attention applied to co-occurring vehicles.

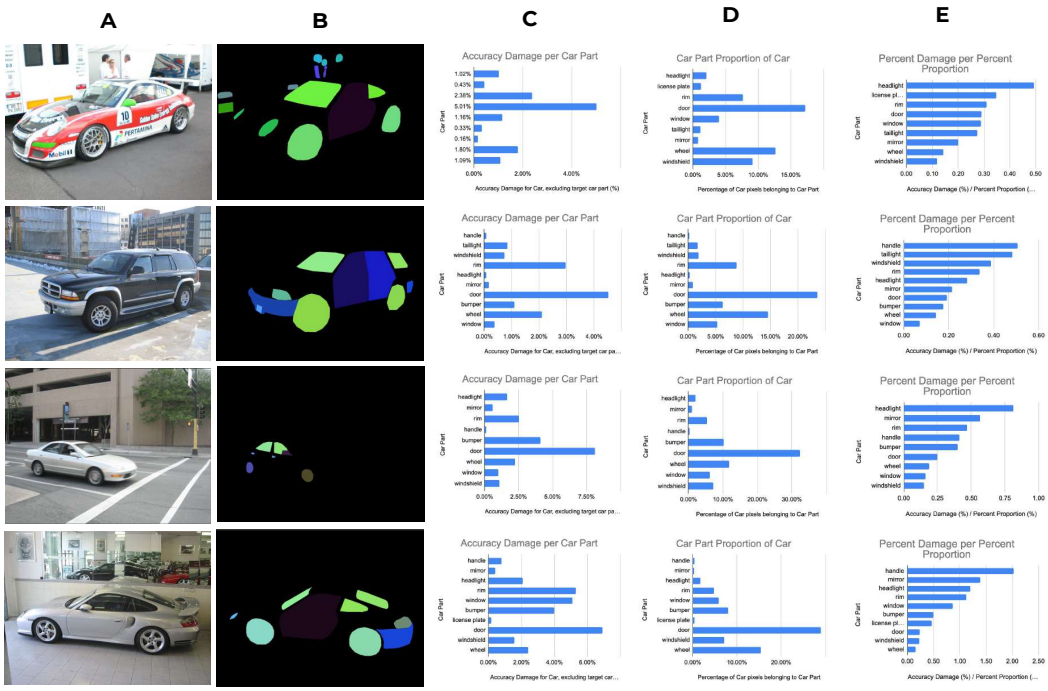

Figure 10: More examples of SIR applied to ADE20k fine-grained car segmentation. Per column E, headlights and handles are often the most discriminative, damaging accuracy the most per pixel. If we had only considered accuracy damage per object part (column A), we would have instead mistakenly identified car door as the most discriminative portion of a car. Normalizing by relative object part size in column B fixes this.

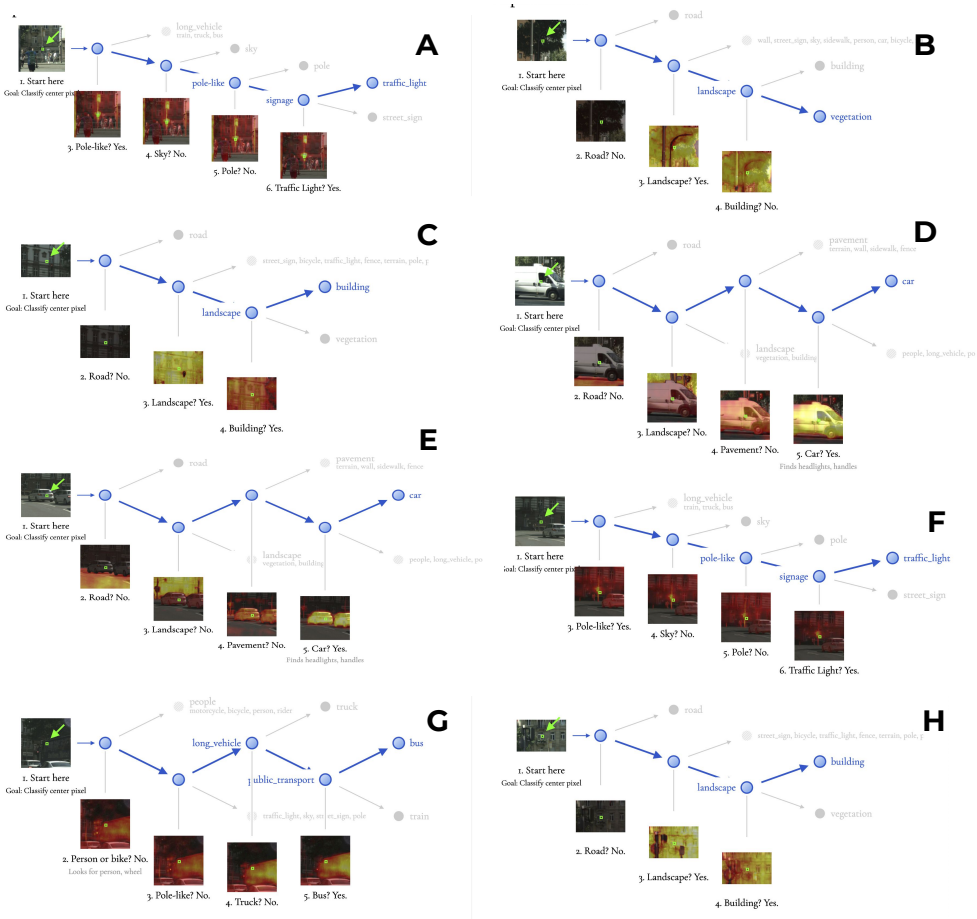

Figure 11: More examples of our final visual decision rule figures, produced for SegNBDT.

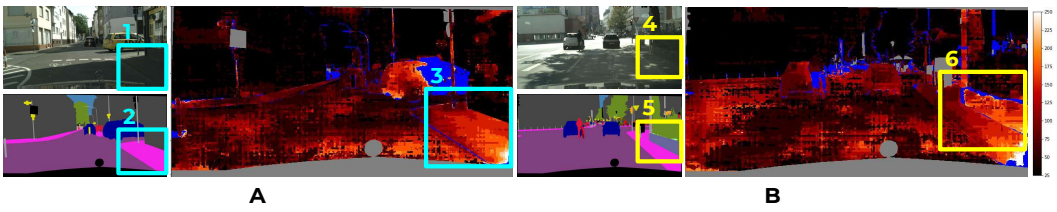

Figure 12: **"Default" Predictions**: Similar pixels from different classes require vastly different amounts of context to classify. (A) With a blue box, the top-left image (1) identifies a region of the image with visually-indistinguishable *Building*, *Sidewalk*, and *Road* pixels – all of which are a dark gray. The bottom-left (2) ground truth labels correspond to the top-left image, including *Road* (purple), *Sidewalk* (pink), and *Building* (gray). These visually-similar pixels in (1) of different classes (2) require vastly different amounts of context to classify correctly (3). Dark values indicate a small amount of context was required to correctly classify the pixel; light values indicate a large amount of context was required. Thus, the network correctly classifies *Building* even with small amounts of context. On the other hand, neighboring *Sidewalk* requires much more context to correctly classify. As a result, we refer to *Building* as the "default" class in this image. We see the same phenomenon occur with a new image (4), its corresponding ground truth (5), where *Vegetation* is green), and the minimum required context (6). Notice that here the *Building* class is missing and that the "default" class is the class *Vegetation*, which is as shallow as *Building* in the hierarchy.

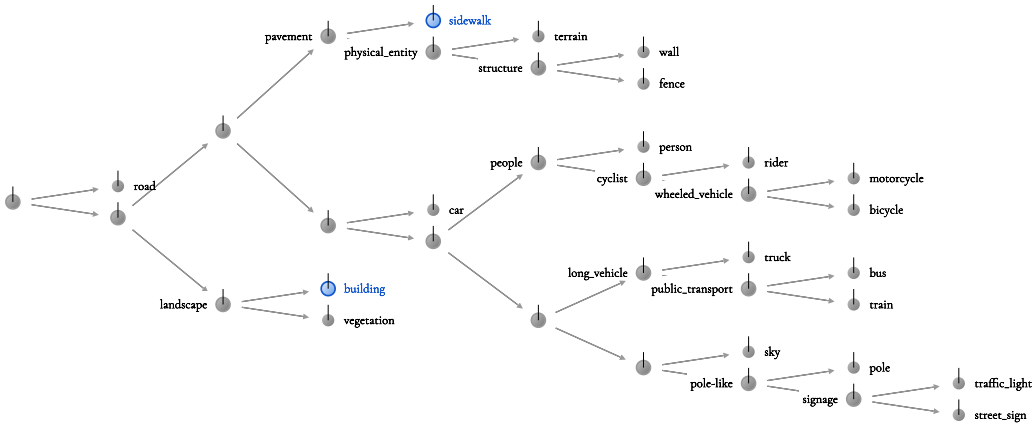

Figure 13: The induced hierarchy for SegNBDT, using a HRNetV2-W48 backbone, on Cityscapes. Note that the “Building" leaf is at a shallower depth than “Sidewalk".

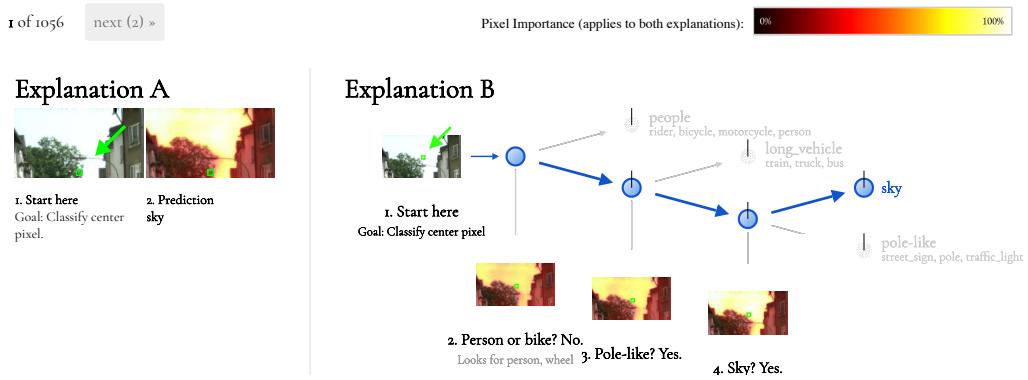

Figure 14: An example survey figure comparing the baseline Grad-CAM (“Explanation A") and SegNBDT with visual decision rules (“Explanation B")