**CS677-1J1**

**Neural Backed Decision Trees**

****

**Proposal:**

Many computer vision applications such as medical imaging, autonomous driving, and other applications like Finance, Insurance, Defense demand accurate, justifiable, and explainable predictions. In certain cases where I need the network to justify its decision and explain how it arrived at its prediction it can be extremely difficult to know why the network arrived at its conclusion. Combining Neural Network with Decision Trees preserves high level interpretability while using neural networks for low-level decisions. The project has implemented Neural-Backed Decision Trees (NBDTs) to jointly improve accuracy and interpretability by replacing a neural network’s final linear layer with a differentiable sequence of decisions and a surrogate loss.

**Content:**

This report includes the following:

1. Link to Video Presentation
2. Link to Code Demo and Summary
3. Description of the Architecture of the DL model
4. Technical innovations introduced by the model and how it compares with the previous and related work
5. Novel Future Application of the model

**Dataset:**

I have used NBDT to run a classification task on the dataset CIFAR-10 Object Recognition in Images dataset. The dataset consists of 60,000 images of 10 different classes:

1. Plane
2. Car
3. Bird
4. Cat
5. Deer
6. Dog
7. Frog
8. Horse
9. Ship
10. Truck

Each image is either of a type of animal or vehicle belonging to the above classes.

Read more about the data here: <https://www.kaggle.com/competitions/cifar-10/data>.

### Exploring the Data:

First I have imported the data and examined its structure so that I am familiar with it. I have created batch sizes of 4 and transformed the images to normalize them all to the same shape. Next I have examined what an example batch will look like with the images and labels.

### Creating NBDT Model

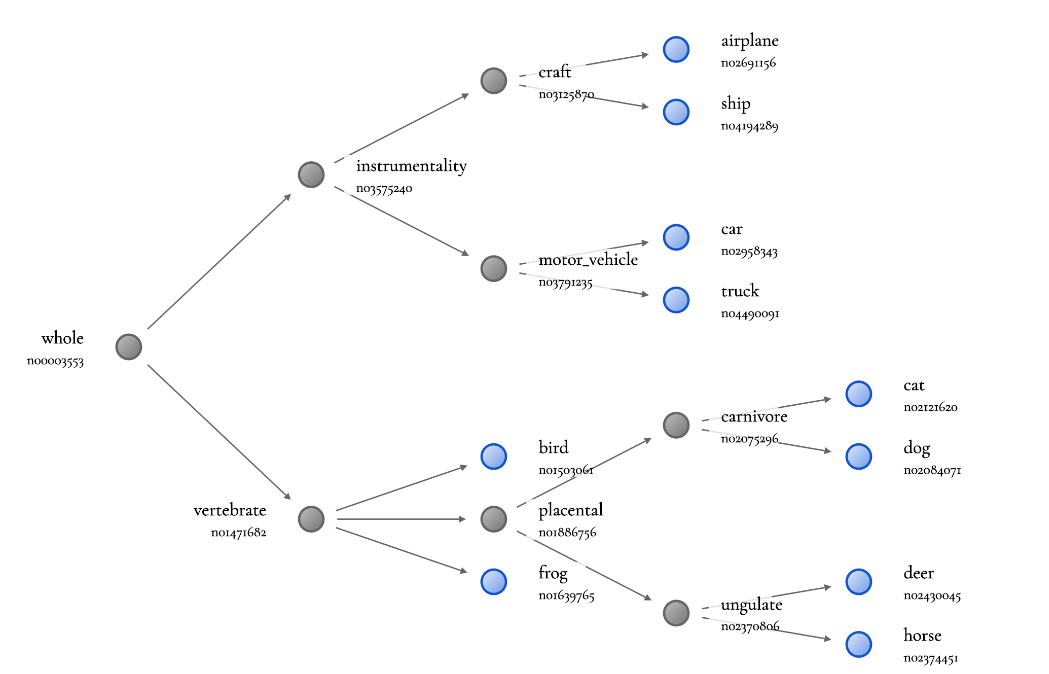
I have imported the libraries needed for NBDT and created model for the demo and testing. For convenience, I have used a pre-trained WideResNet model for computer vision from the pytorch torchvision.models library that is pretrained already.

### Testing Model

I have specified hierarchy='wordnet' to tell the NBDT library I want to build decision tree hierarchy and assign labels to intermediate nodes using WordNet.

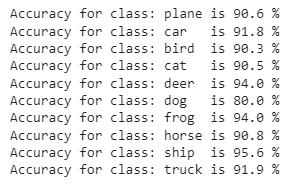
#### Generalizing for unseen classes

I have considered the case where I have introduced an image that is not part of the training classes but similar. I would expect the model to pick the class that is as visually similar as possible to the new unknown class. For example, if I introduce an image of a bear that is an unknown class to the model I should expect it to try and classify it as an animal and not vehicle and the closest looking animal in general to a bear. I have loaded 3 test images, 2 known and 1 unknown class, to test hypothesis. The Results support hypothesis and the bear image is classified as a dog which is the most similar visually to the image of the bear. The advantage of the NBDT is that for the unknown class I can accurately get a higher level generalization of what the image is of. For this example, the bear will be classified as an animal and a carnivore before being classified as a dog and I can use this information to get higher order classifications of unseen classes and see the models classification logic from top down. An example of the output decision tree for this case is something like the below:

****

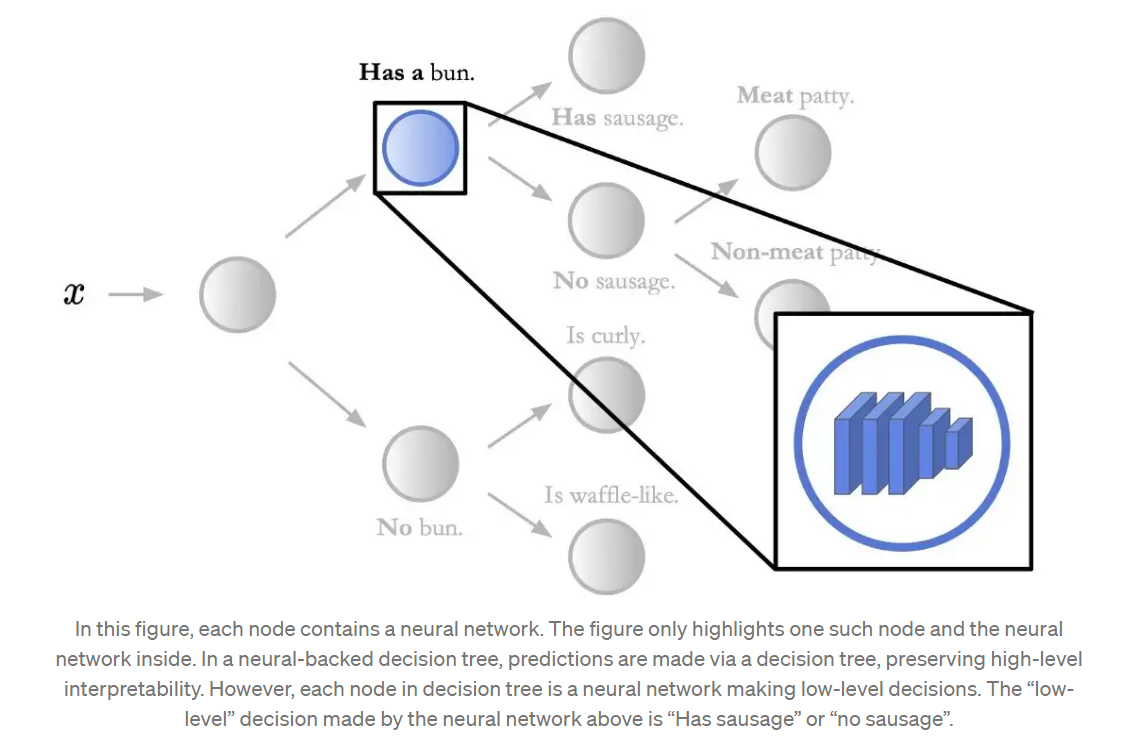
#### Total Classification Performance of Model

Lastly, I have tested to see the total classification accuracy of model to test and ensure that the NBDT layer did not result in any significant decrease of accuracy. Model has performed well with 90% accuracy so interpretability of last decision tree layer has not decreased accuracy. I can inspect for each class the accuracy to see if some more visually distinct classes perform better than others. The results confirm that more visually distinct classes such as ships have very high accuracy where other classes like dogs that have many different species that can look very different have the lowest accuracy as I expected.

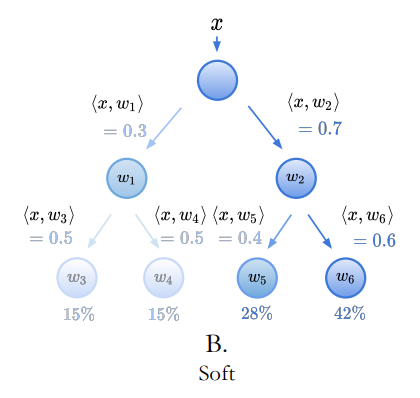


1. **Description of the Architecture of the DL model:**

The key insight of the model is to combine neural networks with decision trees, preserving high-level interpretability while using neural networks for low-level decisions. These models are called [Neural-Backed Decision Trees](http://nbdt.alvinwan.com/) (NBDTs). They can match neural network accuracy while preserving the interpretability of a decision tree.



NBDTs replace a network’s final linear layer with a decision tree. Unlike classical decision trees or many hierarchical classifiers, NBDTs use path probabilities for inference to tolerate highly-uncertain intermediate decisions, build a hierarchy from pretrained model weights to lessen overfitting, and train with a hierarchical loss to significantly better learn high-level decisions (e.g., Animal vs. Vehicle). NBDT first features each sample using the neural network backbone; the backbone consists of all neural network layers before the final linear layer. Second, the final fully-connected layer is run as an oblique decision tree. However, a classic decision tree cannot recover from a mistake early in the hierarchy. Thus, modified decision rules are presented, where each node simply returns probabilities, as normalized inner products, of each child. For each leaf, the probability of its path to the root is computed. Leaf with the highest probability is picked.



The modified decision rules are as follows:

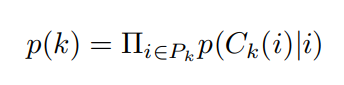
1. **Seed oblique decision rule weights with neural network weights**. An oblique decision tree supports only binary decisions, using a hyperplane for each decision. Instead, a weight vector ni is associated with each node. For leaf nodes, where i = k ∈ [1, K], each ni = wk is a row vector from the fully-connected layer’s weights W ∈ R^( D×K). For all inner nodes, where i ∈ [K + 1, N], find all leaves k ∈ L(i) in node i’s subtree and average their weights:



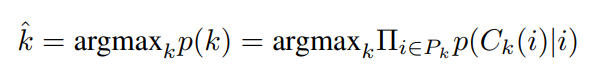
2. **Compute node probabilities**. Child probabilities are given by softmax inner products. For each sample x and node i, compute the probability of each child j ∈ C(i) using p(j|i) =



3. **Pick a leaf using path probabilities**. Consider a leaf, its class k and its path from the root Pk. The probability of each node i ∈ Pk traversing the next node in the path Ck(i) ∈ Pk ∩ C(i) is denoted p(Ck(i)|i). Then, the probability of leaf and its class k is



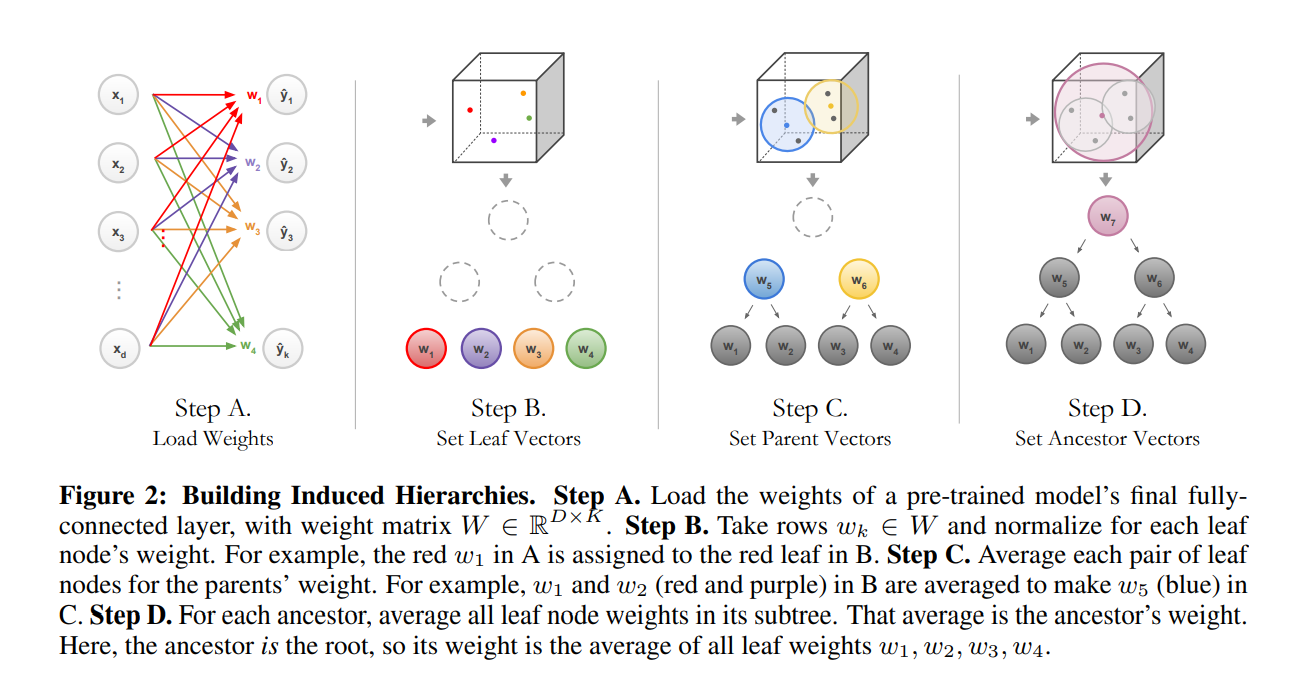
In soft inference, the final class prediction k^ is defined over these class probabilities,



This method has two benefits:

(a) Since the architecture is unchanged, the fully-connected layer can be run regularly or as decision rules.

(b) Unlike decision trees and other conditionally-executed models, this method can recover from a mistake early in the hierarchy with sufficient uncertainty in the incorrect path.



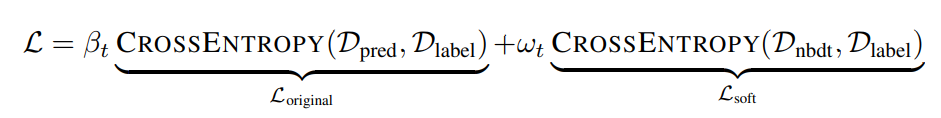
Existing decision-tree-based methods use hierarchies built with data-dependent heuristics like information gain or existing hierarchies like WordNet. However, the former overfits to the data, and the latter focuses on conceptual rather than visual similarity: For example, by virtue of being an animal, Bird is closer to Cat than to Plane, according to WordNet. However, the opposite is true for visual similarity: by virtue of being in the sky, Bird is more visually similar to Plane than to Cat. Thus, to prevent overfitting and reflect visual similarity, a hierarchy using model weights is built. The hierarchy requires pre-trained model weights. Row vectors wk : k ∈ [1, K] is taken, each representing a class, from the fully-connected layer weights W. Then, **hierarchical agglomerative clustering** is run on the normalized class representatives wk/||wk||2. Agglomerative clustering decides which nodes and groups of nodes are iteratively paired. Each leaf node’s weight is a row vector wk ∈ W and each inner node’s weight ni is the average of its leaf node’s weights. This hierarchy is the induced hierarchy.

WordNet is a hierarchy of nouns. To assign WordNet meaning to nodes, the earliest common ancestor is computed for all leaves in a subtree: For example, say Dog and Cat are two leaves that share a parent. To find WordNet meaning for the parent, find all ancestor concepts that Dog and Cat share, like Mammal, Animal, and Living Thing. The earliest shared ancestor is Mammal, so Mammal is assigned to the parent of Dog and Cat. This is repeated for all inner nodes. However, the WordNet corpus is lacking in concepts that are not themselves objects, like object attributes (e.g., Pencil and Wire are both cylindrical) and (b) abstract visual ideas like context (e.g., fish and boat are both aquatic). Many of these are littered across the induced hierarchies. Despite this limitation, WordNet is used to assign meaning to intermediate decision nodes, with more sophisticated methods left for future work.

Even though standard cross entropy loss separates representatives for each leaf, it is not trained to separate representatives for each inner node. To amend this, a tree supervision loss is added, a cross entropy loss over the class distribution of path probabilities



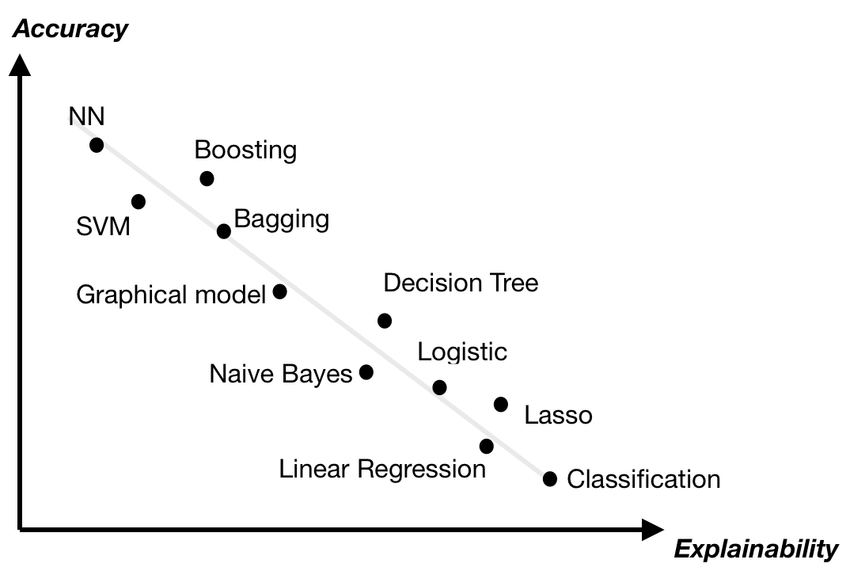
with time-varying weights ωt, βt where t is the epoch count:



1. **Technical innovations introduced by the model and how it compares with the previous and related work:**

**Why are Neural Backed Decision Trees Innovative:**

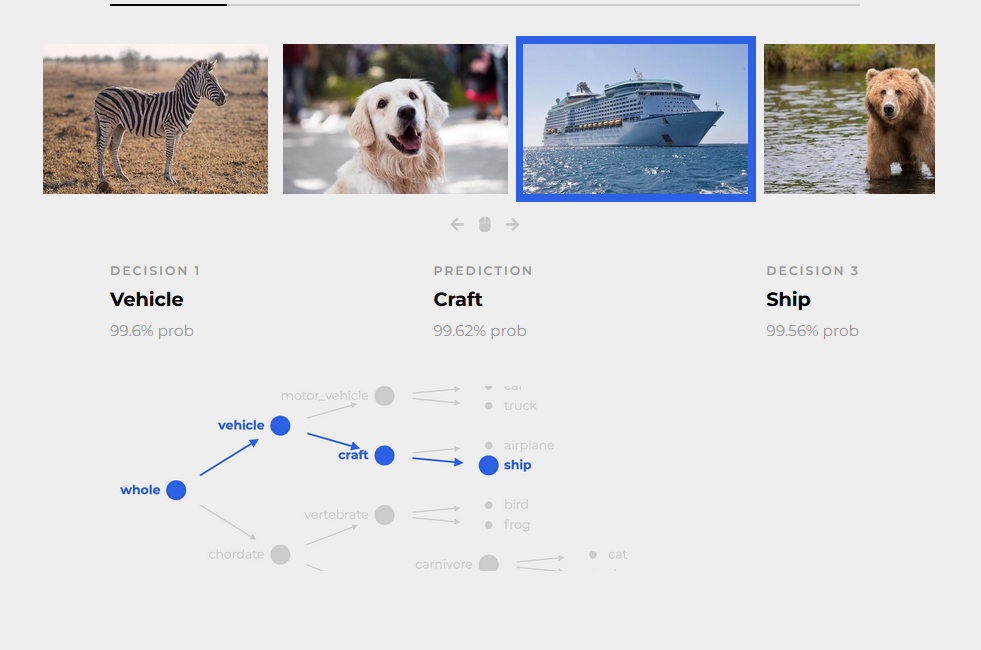
For machine learning to be widely adopted in fields such as medicine and finance, it needs to produce models that are both accurate and have easily interpreted and justifiable predictions. The current view in the field of machine learning is that I will always have to sacrifice accuracy for interpretability or vice versa. Consider the chart below visualizing this trend:



This illustrates nicely the current strong negative correlation between machine learning models accuracy and interpretability. NBDT are innovative as they jointly improve model accuracy and interpretability by replacing the neural network final layer with a decision tree defying the above trend in machine learning models. NBDT match or outperform modern neural networks on CIFAR, ImageNetm and better generalize to unseen classes by up to 16%. NBDT's ability to better generalize is the most innovative part of the improved performance because it comes down to how inferences are done differently in its last decision tree layer.

**How are NBDT able to better generalize:**

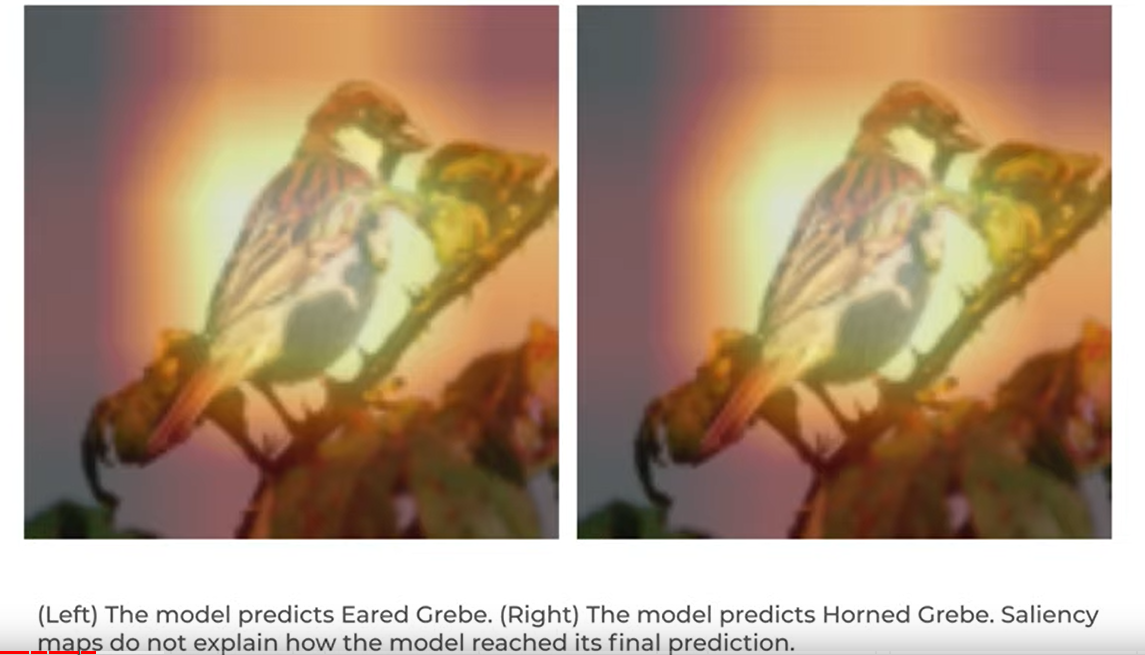
NBDT does not build its decision tree in a conventional way such as using data-dependent heuristics like information gain. It instead takes the pretrained weights from the last layer in the network and sets those to be the leaf nodes for each class. From there it uses hierarchical clustering by taking the average vector between the two neighboring leaf nodes and setting that as the parent and so on until I reach the root node. Using WordNet, I can then assign labels to intermediate nodes by finding the two nodes' nearest ancestor. As an example if I am tagging images as animals or vehicles I could then have an intermediate node with vehicle and a leaf node for the input image of a ship.



The path-probability cross entropy loss disproportionately up-weights decisions earlier in the hierarchy, encouraging accurate high-level decisions leading to the up to 16% better generalization for unseen classes.

**Why Decision Trees are more interpretable than saliency maps:**

Many computer vision applications (e.g. medical imaging and autonomous driving) require insight into the model’s decision process, complicating applications of deep learning which are traditionally black box. Recent efforts in explainable computer vision attempt to address this need and can be grouped into one of two categories: (1) saliency maps and (2) sequential decision processes. Saliency maps retroactively explain model predictions by identifying which pixels most affected the prediction. Focusing on the input this way fails to capture the model’s decision processes. Saliceny offers no insight for misclassification when the model is “looking” at the right object for the wrong reasons. Consider the below example where one is a saliency map of the right prediction and one is a map of the wrong prediction.



A human observer cannot determine why the right model made the wrong prediction because the highlighted pixels are still “Looking” at the correct object in the image. Alternatively, I can gain insight into the model’s decision process by breaking up predictions into a sequence of smaller semantically meaningful decisions as in decision trees. Research on human interpretability of decision trees vs saliency maps have shown the below:

* Out of 600 survey responses when given saliency maps and class probabilities, only 87 predictions were correctly identified as wrong. In comparison, when given the NBDT series of predicted classes and child probabilities 237 images were correctly identified as wrong. respondents can better recognize mistakes in NBDT explanations nearly 3 times better
* In the next study I offer the blurred image and two sets of predictions: (1) the original neural network’s predicted class and its saliency map, and (2) the NBDT predicted class and the sequence of decisions that led up to it (“Animal, Mammal, Cat”). For all examples, the two models predict different classes. In 30% of the examples, NBDT is right and the original model is wrong. In another 30%, the opposite is true. In the last 40%, both models are wrong. The image is extremely blurry, so the user must rely on the models to inform their prediction. When offered model predictions, in this survey, 255 of 600 responses are correct (42.5% accuracy), a 15.3 point improvement over no model guidance. I observe that humans trust NBDT-explained prediction more often than saliency-explained predictions. Out of 600 responses, 312 responses agreed with the NBDT’s prediction, 167 responses agreed with the base model’s prediction, and 119 responses disagreed with both model’s predictions. Note that a majority of user decisions (∼ 80%) agreed with either model prediction, even though neither model prediction was correct in 40% of examples, showing images were sufficiently blurred to force reliance on the models. Furthermore, 52% of responses agreed with NBDT (against saliency’s 28%), even though only 30% of NBDT predictions were correct, showing improvement in model trust.
* The explanation of an NBDT prediction is the visualization of the path traversed. I then compare these NBDT explanations to other explainability methods in human studies. Specifically, I ask participants to pick an expert to trust (Appendix, Figure 13), based on the expert’s explanation – a saliency map (ResNet18, GradCAM), a decision tree (NBDT), or neither. I only use samples where ResNet18 and NBDT predictions agree. Of 374 respondents that picked one method over the other, 65.9% prefer NBDT explanations; for misclassified samples, 73.5% prefer NBDT. This supports the previous survey’s results, showing humans trust NBDTs more than current saliency techniques when explicitly asked.

The above studies support that people trust decision tree models more than saliency maps and can more easily identify model errors using them.

1. **Novel Future Application of the model:**
2. Autonomous Vehicles - Did you know that 46,000 people died in car crashes in 2021 in the US? The promise of autonomous vehicles in the future run by AI systems have the potential to make roads exponentially safer for everyone with as much as 94% of accidents attributed to human error according to USDOT. That's 43,240 lives per year potentially saved by this technology. Not only that but it also has potential to improve traffic congestion and make transportation cheaper and more accessible, among other benefits as well. However, what happens if there is a scenario such as if a self-driving car finds itself in a position where an accident is inevitable, what measures should it take? Prioritize the protection of the driver and put pedestrians in grave danger? Avoid pedestrians while putting the passengers’ safety at risk? This is where explainable AI will come in and be crucial to its success. I will need AI systems that people can clearly inspect its reasoning on for such situations for legal and ethical concerns. Without it people may never have enough trust in the systems or regulation will outlaw them for fear of such situations taking place. Drivers and passengers will need to know ahead of time how AI systems will react in these situations to make an informed decision on whether to ride in them or not. Also, in case of an anomaly behavior of the system the automaker will need to be able to traceback and inspect where the model reasoning went wrong to correct it for future generations of the car.
3. Visual Explainability of deep medical models - Despite the progress of deep learning on medical imaging, there is still not a true understanding of what networks learn and of how decisions are reached. Deep learning has broad prospects in the medical field, as it is crucial to establish systems that assist diagnosis. Most methods, however, assist without knowing the decision basis. To increase their explainability, Neural Backed Decision Trees can be very helpful. For medical imaging, it is required to know both which area is a suspicious lesion and how the model makes decisions. False Negatives can be very detrimental. Suppose a malignant case is diagnosed as a benign one which can prove fatal in short or long term. NBDTs can help to classify diseases and visualize the decision path, thus offering interpretability which is crucial for clinical practices.