Kynan Nedellec (260866794), Philippe Gauthier (260923926), Mingjun Tang (260793546)

MiniProject 4: Reproducibility in ML

Distilling a neural network into a soft decision tree by Frosst and Hinton

COMP 551

McGill University

December 13, 2021

Abstract

Convolutional neural networks are powerful tool for image classification but their blackbox design makes it difficult to understand and interpret their decisions. In this light, Frosst and Hinton investigated a new technique which consists in extracting the generalization ability of a neural network and instilling it into what they call a *soft decision tree* (SDT), thus exploiting the interpretability of decision trees. In this reproducibility challenge, we were able to 1. successfully replicate the results achieved by the authors, 2. obtain the same intermediary results from a thorough investigation of the original model's architecture, and 3. improve the model's performance through hyperparameter tuning and CNN architecture modifications. Finally, the soft decision tree model was tested on two other datasets: the MNIST fashion and a rock-paper-scissors image dataset.

Introduction

Due to their incredible performance, CNNs have recently become one of the most widely used algorithms for precise image classification. Through multiple layers, CNNs are able to find nebulous patterns in images to differentiate them from one another. Unfortunately, the architecture and complexity of these networks make it very difficult to understand the process behind the decisions that they make. This lack of interpretability which is often seen as unpredictability can generate mistrust between the algorithm and its user, as it has been noted in areas such as autonomous driving (Wang, 2021). On the other hand, decision trees are a lot easier to interpret and understand, but are less efficient at generalizing and classifying data than neural networks. Nicholas Frosst and Geoffrey Hinton offer a way to combine both the precision and generalization of a CNN and the interpretability and simplicity of a decision tree in their paper "Distilling a Neural Network Into a Soft Decision Tree".

They use a variant of decision trees called soft decision trees (SDT), which are trained to create optimal filters at every node to capture as many features as possible from pictures. SDTs perform "soft decisions": a sigmoid activation function at each step to calculate the most probable next filter in the tree. The authors use the generalization ability of a neural network and extract soft labels to train the soft decision tree, a technique known as distillation. With multiple techniques and improvements to the architecture combined, these trees are shown by Frosst and Hinton to be interpretable and faster than the classic CNN for the cost of a slight downgrade on performance on the MNIST digit dataset.

Methodology

We use the code from GitHub, which was not created by the author. All experiments are run using Google Colab with GPU acceleration. We chose to do

1. Reproducibility – to see if the model works consistently
2. Performance on other datasets – to investigate the generalization potential of the model
3. Ablation study – to investigate the effect of different choices of the model, such as the performance difference between distilled and non-distilled SDT
4. Improving the performance of the original model by tuning hyperparameters – to see if we can obtain a better accuracy on datasets that we test on, because the SDT does not perform well on the paper, rock, scissor dataset

Reproducibility

Using the Lukas Martak's model, we replicated the study and obtained a validation accuracy of 98% using the two layered convolutional neural network and a validation accuracy of 93% on the same soft decision tree with a tree depth of 8 and with 6 epochs, all while including the temperature coefficient, distillation and the regularization penalty as described in the paper. These results are as expected as the authors: in their original paper, described an accuracy of 99% on the CNN and 94% on the SDT.

Other datasets

Fashion-MNIST: The first dataset we tried with our SDT was the Keras Fashion dataset consisting of 60,000 28x28 gray images with the following 10 classes: t-shirt (0), trouser (1), pullover (2), dress (3), coat (4), sandal (5), shirt (6), sneaker (7), bag (8) and finally ankle boot (9). The model was trained with a batch size of 4, including 8 epochs and a tree depth of 5. The original convolutional neural network used to train the pictures had a validation accuracy of 88%, and the SDT had a validation accuracy of 78%. Appendix 2 shows the result of the SBDT for the fashion data set.

Rock-paper-scissors: We then fitted a CNN and an SDT to the Keras Rock-Paper-Scissors dataset. Given the fewer classes, 3 in total (0 representing rock, 1 paper and 2 scissors), and the small size of the dataset (2188 images) we had to do more modifications to the CNN architecture as well as more hyperparameter tuning to achieve 62.1% test set accuracy with the SDT of depth 3 (compared to 87.1% for the CNN). In particular, we had to reduce the number of convolutional layers to 2 instead of 6 and apply stronger dropout (0.5 instead of 0.2) to prevent large discrepancies between training and test set. This dataset presented an important difficulty associated with datasets with fewer classes and fewer distinctive features. Indeed, the tree presented in figure 1 is less instructive of the decision process than for MNIST or the fashion dataset. Further, usual data augmentation or transformation techniques such as rotation or edge extraction (see Appendix 8 & 11), while they enable higher accuracy, tend to break down the interpretability of the tree, defying the very purpose of SDTs.

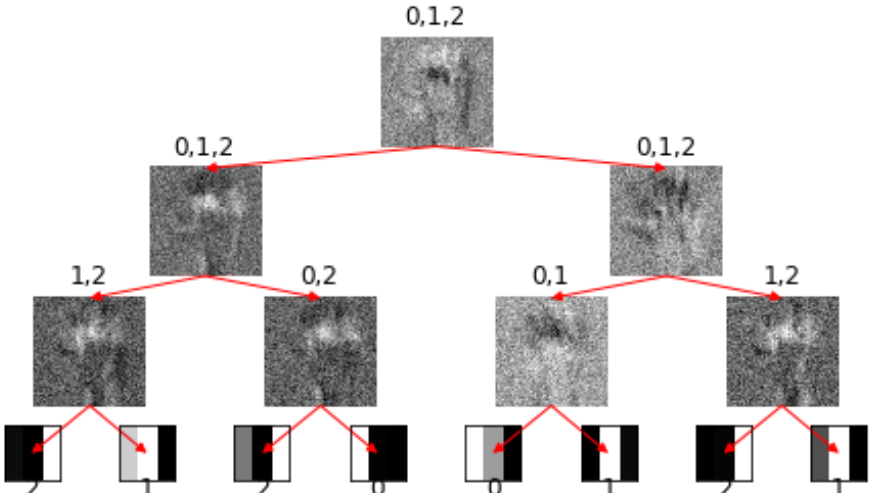


Figure 1: Visualization of a soft decision tree of depth 3 trained on the rock-Paper-Scissors dataset

Ablation study

All models for the ablation study were tested with a batch size of 4, trained over 4 epochs and with a tree depth of 4 on the MNIST digits dataset. The regular model had an accuracy of 86%.

Regularizers: The authors of the paper make use of a regularization penalty term, which they explain helped each node distribute classes with a probability around 50% instead of having one highly probable (and overfitted) path for each class. This encourages more equal and thus better use of decisions in the nodes. The penalty decays with depth, since lower levels of the tree need to make more explicit decisions on the class than higher levels which discriminate on more general features. Without this penalty term, we obtained a test accuracy of 76%. As expected, the penalty helped with generalization of the model, improving the accuracy.

Distillation: As we mention above, distillation from a neural network is a technique used "to extract the generalization ability of a neural network into a decision tree" (Hinton, Vinyals, Dean 2015). This is done by obtaining new y labels for the training from the CNN. Using hard labels (no distillation) we obtained a lower accuracy of 79%. These results showed the usefulness of the distillation technique.

Inverse temperature: Inverse temperature is a feature used to prevent very soft[1] decisions. This filter is applied in every node before the sigmoid activation function. Without the inverse temperature, we obtained an accuracy of 78%, with much more noise in the decision tree as shown by figure 4 in the appendix. With the softer decisions made by the SDT, the noise has less impact on the probability of a given path.

Improving performance

Hyperparameters and model parameters:

As stated in the paper and code, increasing the depth of the tree can improve the accuracy of soft decision trees, but the training time also increases exponentially with the depth of the tree. For example, increasing the depth to 8 of the fashion data set tree gave an accuracy of 80% instead of 78%. We note however that adding depth adds to the complexity and results in less interpretability.

Batch size has a significant impact on the accuracy of soft decision trees. Increasing the batch size of a soft decision tree is more likely to reduce the accuracy of the model (see Appendix 12), but it will also reduce the training time a lot.

Finally, we note that while the original CNN was successful for building the tree for the fashion dataset, the performance of the resulting tree for the other dataset was barely above random, albeit many tries at hyperparameter tuning. Further investigations into the necessary requirements for a CNN to produce performing trees should be done.

Data augmentation:

Since getting marginally higher accuracy proved quite difficult, we turned to data augmentation to improve performance. For the rock-paper-scissors dataset, apart from the rotations and edge detection discussed above, we tested image segmentation (see Appendix 8 & 11). We also tried to replicate the black-on-white format of MNIST digits in the hope of improving performance using edge detection coupled with image segmentation, but only with limited success. With a tree of depth 3 we achieved 44.35% (62.1% without data augmentation), and 63.2% for a tree of depth 5 (versus 72.1% originally).

Final words

For a more complicated dataset with more classes, the SDT requires a bigger depth which could hinder its interpretability. In the same fashion, when there are fewer classes and few discriminative features, there is an important tradeoff between interpretability and performance, with CNN's or even incomprehensible decision trees outperforming the best interpretable models. At what point is interpretability better than performance? It's certain that some tasks such as feature extraction, or tasks strongly involving human decision-makers could benefit from these interpretable DSTs. However, for other tasks where performance is the main priority, Hinton and Frosst's innovation doesn't yet solve the problem of interpretability, at least not without trading off a significant degree of performance.

[1] By very soft decision we mean a *decision* that is overly distributed over different possible paths (Frosst 2017).

Statements of contribution

Philippe Gauthier: fashion dataset, reproducibility, ablation study, introduction, abstract

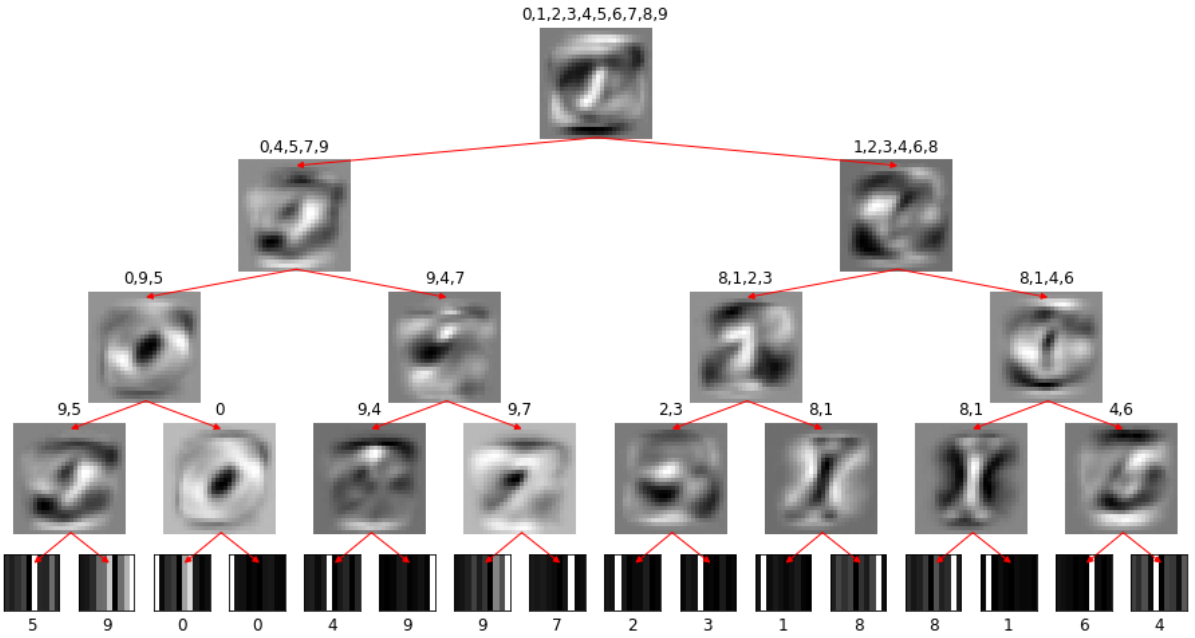
Mingjun Tang: rock paper scissor dataset, improving performance, merge of codes

Kynan Nedellec: rock paper scissor dataset, noise filtering, data augmentation

References

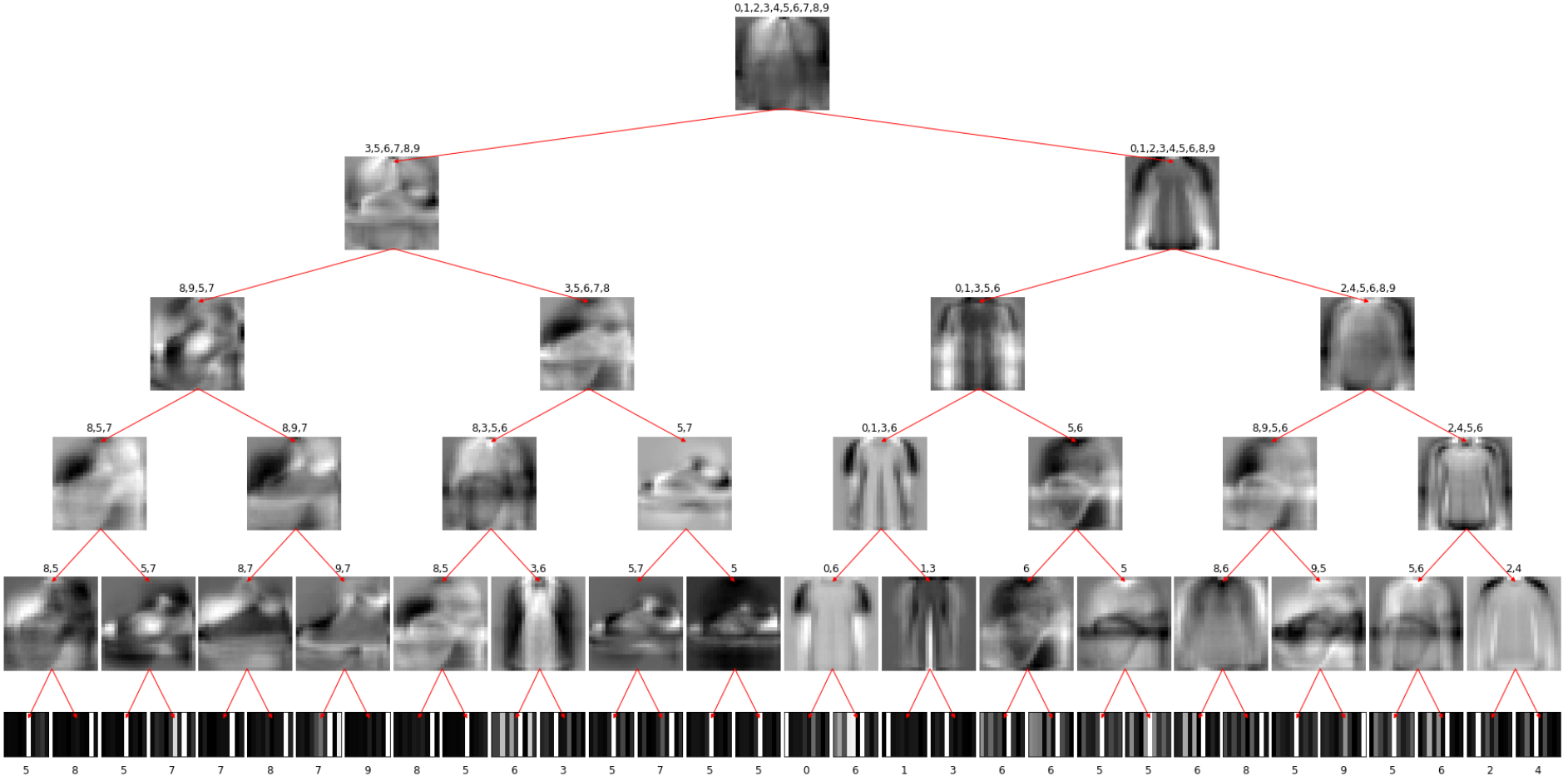
1) Distilling a Neural Network Into a Soft Decision Tree, Nicholas Frosst, Geoffrey Hinton, https://arxiv.org/pdf/1711.09784v1.pdf
2) Distilling the Knowledge in a Neural Network, https://arxiv.org/abs/1503.02531
3) Wang, Hengli et al. “Learning Interpretable End-to-End Vision-Based Motion Planning for Autonomous Driving with Optical Flow Distillation”. *arXiv [cs.CV]* 2021. Web.
4) https://github.com/lmartak/distill-nn-tree

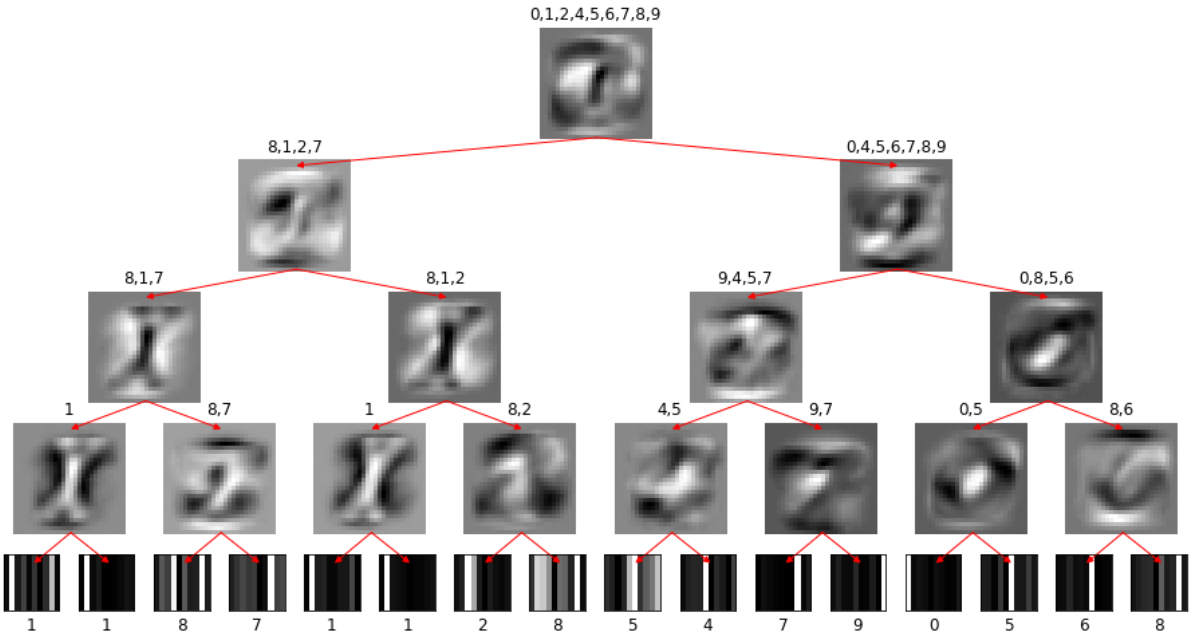
Appendix



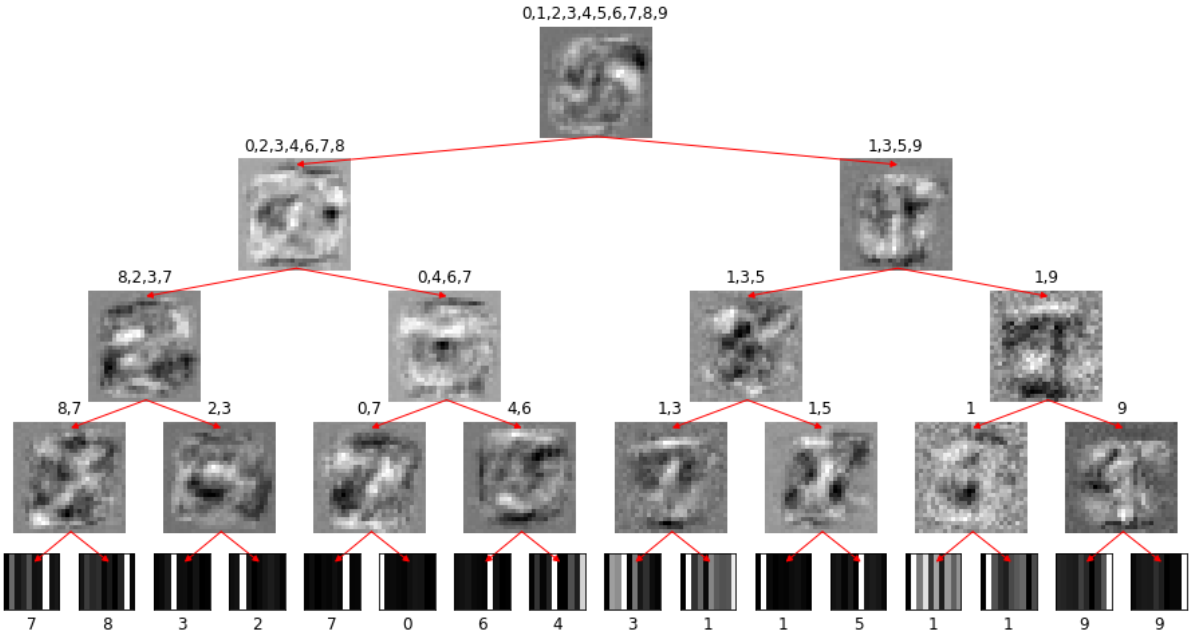
Appendix 1: (ablation study) regular model, batch size 4, 4 epochs, val accuracy 86%

Appendix 2: SDT for the MNIST fashion data set with a batch size of 4, 8 epochs and a depth of 5

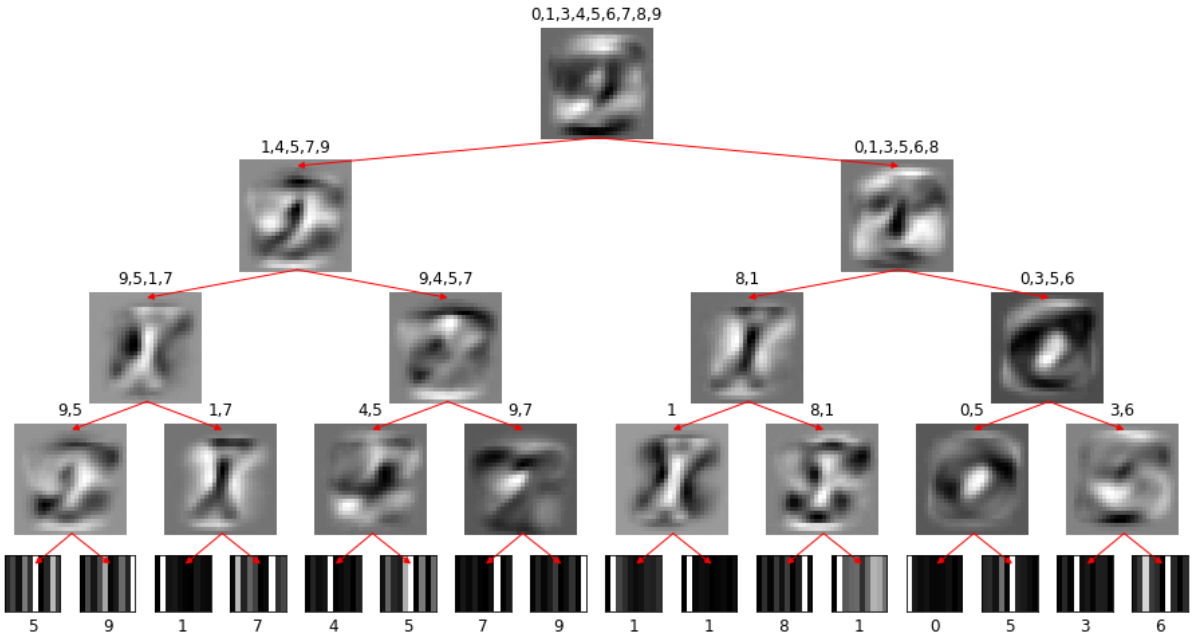




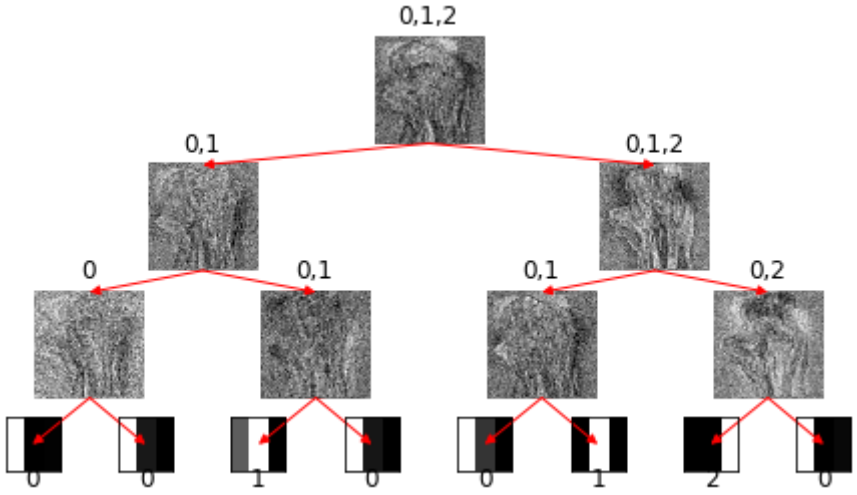
Appendix 3: (ablation study) no distillation val accuracy 79%



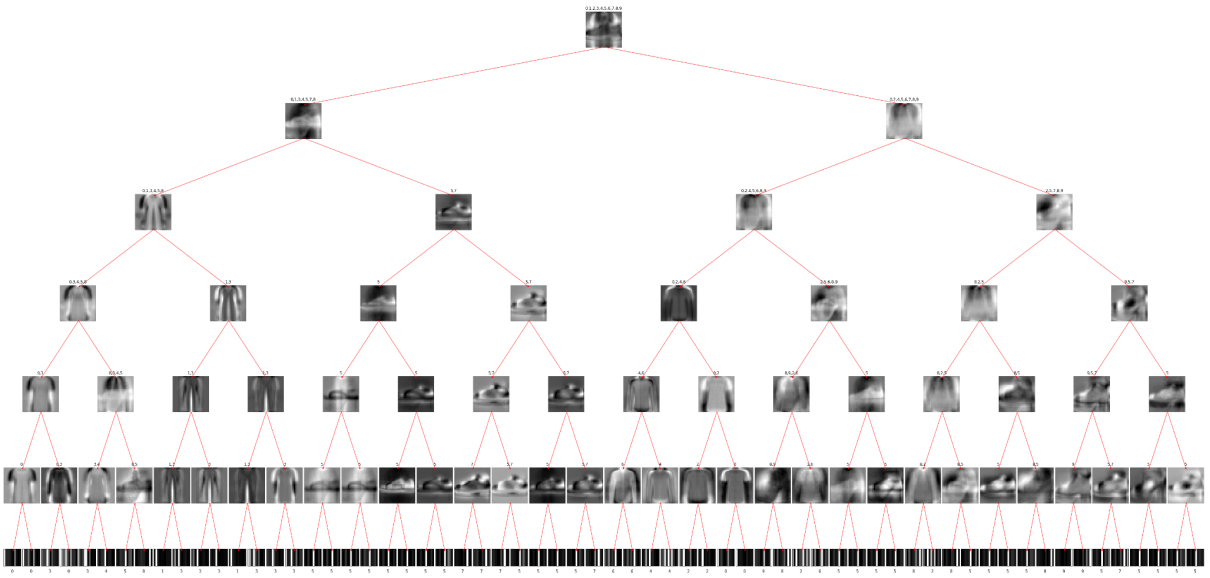
Appendix 4: (ablation study) no inverse temperature validation accuracy 78%



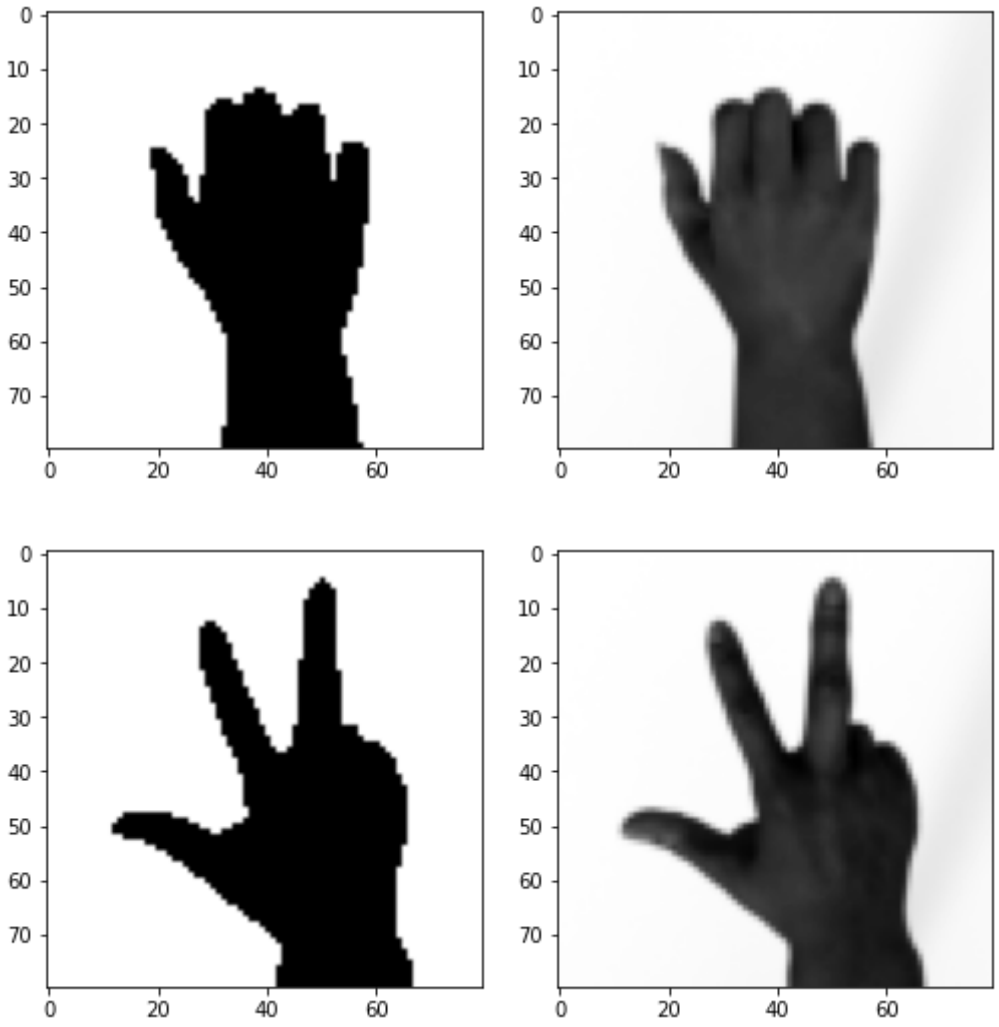
Appendix 5: (ablation study) no regularization penalty val accuracy 76%



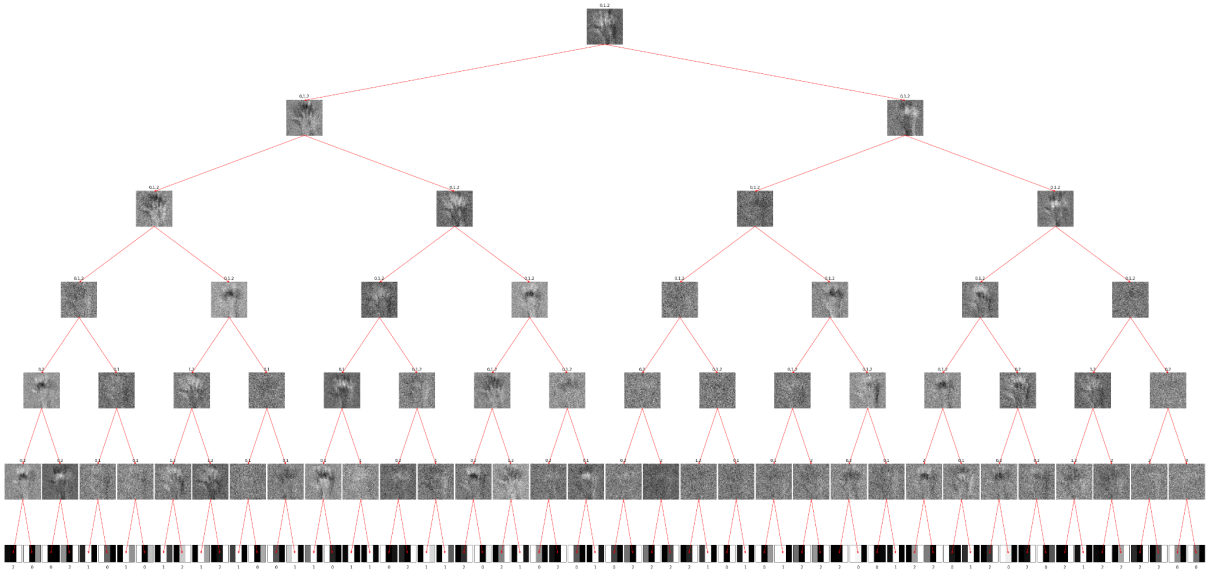
Appendix 6: Loss of interpretability of Rock-Paper-Scissor decision tree when augmenting images with Canny edge detection



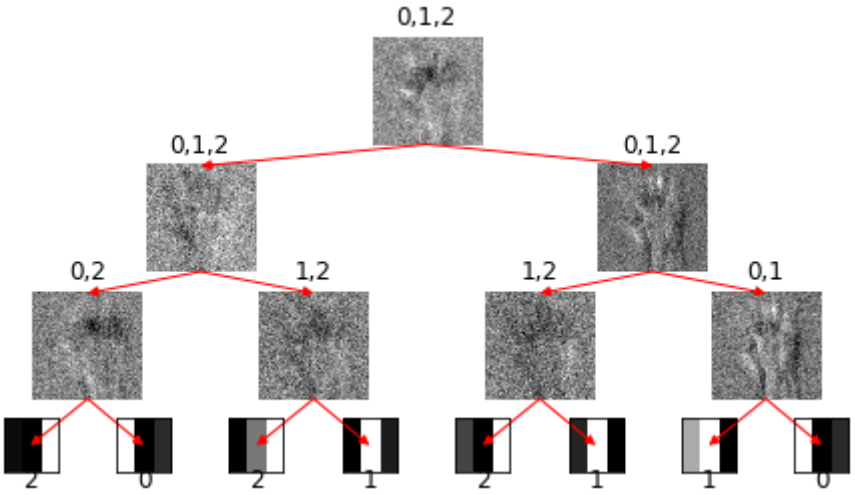
Appendix 7: SDT for mnist fashion data set batch size of 10, tree depth of 6, 10 epochs, 0.77 val accuracy



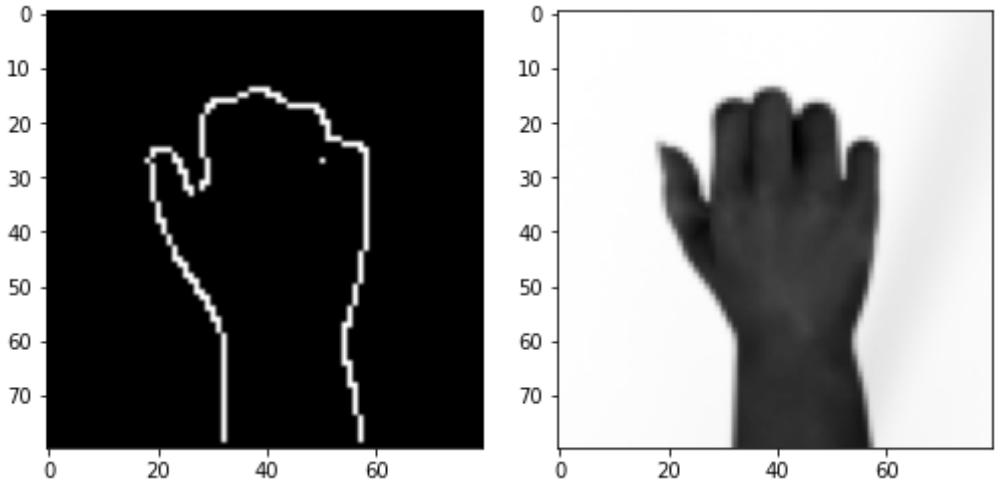
Appendix 8: Sobel edge detection and watershed segmentation



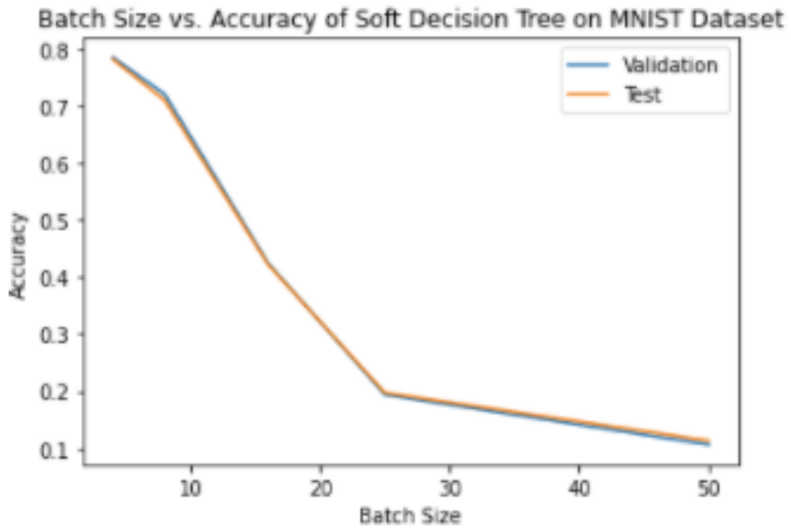
Appendix 9: Rock paper scissor decision tree of depth 6 achieving 71.2% accuracy



Appendix 10: Rock paper scissor decision tree with image segmentation achieving 44.35% test accuracy



Appendix 11: Canny edge detection for data augmentation



Appendix 12: Batch size vs accuracy of SDT on the original model