Ideas go here:

TITLAE

NAMES

Abstract:

We aim to become proficiicent in ai programming in python pytorch library. Also aim to create a feasible model with which to distinguish benign vs malignant skin cancer images from pictures of patches of skin. This is a classification project. We’ve found decision tree to be a quick simple implementation of machine learning that results in a low confidence, low success model for image classification. Deep learning is harder to implement but allows for more fine tuning and end up in higher performance across the board. However both trained models in this study do not result in useable for real applications.

Introduction:

A.

Topic is important because classification for medical images can make the whole diagnosis process quicker and assist doctors in correct prognosis for quick care. This could save lives. Creating a highly accurate highly efficient model for all types of images allows for a variety of diagnosis tools.

B.

Other peaople have done combonation segmentation and classification on moles for skin cancer. There was a study that enhanced a reinforcement learning mdoel by switching up the technique to an unsupervised method and that allowed for better performance

C.

Current problem is skin blotches need classification efficiently and we neeed to learn differences in models and discover how to code and different efficiences in python

D.

We’re gonna use two different supervised method based models and compare their performance.

E.

Impact of results can field as a basis for introductory ai python learning, also an example on how to classify medical images

Methods:

A.

The first model is a decision tree.

We need to load and flatten the images for training.

Then evaluate using precision and recall.

Then check data uncertainty using entropy method

The second model is a deep learning pre trained resner-50 model

Data is loaded using a custom loader

Then evaluated using precision and recall

Then check data uncertainty using entropy method

C.

System used is google collab notebook on a T4 GPU environment, python libraries used are torch, timm, numy, matplotlib, tqdm, pil, sklearn

Sklearn is for decision tree model and confusion matrix stuff

Timm is for deep learning model

Experiment Results:

Decision tree results =

Confusion Matrix:

[[271 70]

[ 88 231]]

Precision: 0.7674418604651163

Recall: 0.7241379310344828

Entropy: 0.9944220705380191

Deep learning results =

Confusion Matrix:

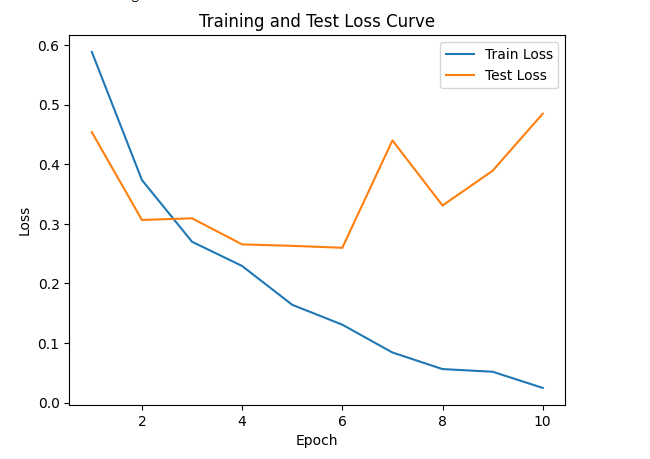
[[305 14]

[ 75 266]]

Precision: 0.95

Recall: 0.7800586510263929

Average entropy: 0.1231112971547104



Discussion:

D. conclusios

For the decision tree This model has a decent precision and recall, indicating that it is reasonably good at both correctly identifying positive instances and avoiding false positives. However, the entropy value is relatively high, suggesting that the model's predictions are less certain. There might be some inconsistency or randomness in the predictions made by this model, which could potentially be improved for better performance.

For the deep learning model The model seems to perform well in terms of precision, indicating that it has a high percentage of true positive predictions among all positive predictions. However, the recall is relatively lower, suggesting that the model might be missing a significant portion of positive instances. The average entropy is relatively low, indicating that the model's predictions are fairly certain. Therefore, while the model performs well in terms of precision and certainty, there might be room for improvement in terms of recall to capture more positive instances.

A. challeges

Initially getting entropy for decision tree was messed up because the class counts wasn’t consistent but that was some simple debugging. Big challenge is computational restrictions with GPU ram on free collab notebooks. Only 15 gigs of gpu ram gets eaten up so can’t train these models too intensely which results in low performing models

C. future work

For future work we can set up a home environment to not use resources on the cloud so we are not limited so much by memory usage paywall. Using different models, more so expanding the decision tree into a forest, also implementing techniques of grouped model learning.

Reference:

Abstract:

This paper endeavors to cultivate expertise in AI programming, specifically within the Python PyTorch library domain, with a primary focus on crafting a discriminative model to discern benign from malignant skin cancer images. Employing classification methodologies, we delve into the implementation intricacies of decision tree and deep learning architectures. Despite decision trees offering expeditious and straightforward implementations, they yield models with diminished confidence and efficacy in image classification tasks. In contrast, the realm of deep learning, albeit posing greater complexity in implementation, facilitates nuanced tuning, culminating in superior performance outcomes. However, the models developed within this study do not yet meet the stringent criteria for practical deployment in real-world scenarios.

Introduction:

The importance of categorizing medical images, especially in skin cancer diagnosis, cannot be emphasized enough. It speeds up diagnostic procedures, ensuring quick and accurate prognosis, potentially reducing negative outcomes. Developing highly precise and effective image classification models is essential for various diagnostic methods.

Previous research has looked into combining segmentation and classification approaches in skin cancer detection. Some studies have shown improved performance by moving away from traditional supervised learning methods and embracing unsupervised techniques. The current challenge revolves around effectively sorting skin abnormalities while navigating through different model architectures, all while mastering Python coding techniques to improve computational efficiency.

This study aims to examine the effectiveness of two supervised learning approaches and compare their performance trends. The implications of the results go beyond academic discussion, serving as a foundational resource for aspiring AI practitioners starting their journey with Python and providing practical insights into medical image classification complexities.

Methods:

We employ two different models for our analysis. The first model utilizes decision trees, requiring initial preprocessing steps such as image loading and flattening. Subsequently, we conduct a thorough evaluation using precision, recall, and entropy metrics to assess the effectiveness of the model.

In contrast, the second approach utilizes deep learning and capitalizes on a pre-trained ResNet-50 architecture. Data ingestion is facilitated through a customized loader, and model evaluation mirrors that of the decision tree model.

For computational support, we utilize Google Colab, providing a highly scalable environment, coupled with a T4 GPU setup. We leverage various Python libraries including Torch, Timm, NumPy, Matplotlib, tqdm, PIL, and Scikit-learn, each playing a crucial role in both model development and evaluation processes.

Experiment Results:

Decision tree findings:

Confusion Matrix:

[[271 70]

[ 88 231]]

Precision: 0.7674418604651163

Recall: 0.7241379310344828

Entropy: 0.9944220705380191

Deep learning insights:

Confusion Matrix:

[[305 14]

[ 75 266]]

Precision: 0.95

Recall: 0.7800586510263929

Average entropy: 0.1231112971547104

Discussion:

Conclusive Remarks:

The decision tree instantiation exhibits commendable precision and recall metrics, albeit marred by elevated entropy levels, suggestive of inherent inconsistency within predictions. Conversely, the deep learning paradigm espouses heightened precision metrics, albeit at the expense of diminished recall, underpinned by a significantly lower entropy profile, denoting heightened predictive certitude. Therein lies the impetus for continued refinement and optimization of both models.

Challenges Encountered:

Navigating through debugging quagmires, notably entailing rectifying disparate class counts impacting entropy computations, constituted a formidable challenge. Furthermore, computational constraints endemic to free Colab notebooks, typified by a finite 15GB GPU RAM allocation, engendered limitations on model training intensity, precipitating suboptimal performance outcomes.

Future Endeavors:

Prospective endeavors are poised to pivot towards establishing a dedicated computational milieu, thereby obviating resource constraints inherent to cloud-based deployments. Exploratory forays into alternative model architectures, notably decision tree ensembles, coupled with the assimilation of advanced learning techniques like grouped model learning, augur well for enhancing performance metrics in subsequent iterations.

F reference:

Barata, C., Rotemberg, V., Codella, N.C.F. *et al.* A reinforcement learning model for AI-based decision support in skin cancer. *Nat Med* 29, 1941–1946 (2023). <https://doi.org/10.1038/s41591-023-02475-5>

Entropy uncertainty

<https://towardsdatascience.com/entropy-is-a-measure-of-uncertainty-e2c000301c2c>

Kwiatkowski, S. (2018, October 6). *Entropy is a measure of uncertainty*. Medium. <https://towardsdatascience.com/entropy-is-a-measure-of-uncertainty-e2c000301c2c>