CVPR
#*****

Transfer learning models for tree species detection in agroforestry

Anonymous CVPR submission

Paper ID *****

Abstract

Section 0: Abstract (1-2 paragraphs) The abstract is required. It should contain a 1-2 sentence summary of each major section of the paper.

1. Introduction

Section 1: Introduction (0.5-1 page) Explain the problem and why it is important. Discuss your motivation for pursuing this problem. Give some background if necessary. Clearly state what the input and output is. Be very explicit: "The input to our algorithm is a image, video, patient age, 3D video, etc.. We then use a SVM, CNN, GAN, etc. to output a predicted age, cancer type, restaurant, ramen, etc.." This is very important since different teams have different inputs/outputs spanning different application domains. Being explicit about this

Trees are a large carbon pool for our planet. They play a vital role in offseting anthropogenic carbon emissions and attenuating the effects of climate change. We can grow this pool through afforestation and deferred deforestation, oftentimes supported through carbon offset payments by sectors emitting carbon to the landowners growing these forests.

One high-potential opportunity for afforestation is through the transition of pastureland operations to silvopasture operations. Silvopasture is simply the agricultural practice of combining tree cropping with livestock management.

In order for a payment system like this to work, we need a highly accurate estimate of the amount of carbon stored in trees. Fortunately, decades of research has been invested into tree growth projections and today, allometric equations are readily available for many tree species, which can accurately estimate above and below-ground tree biomass as a function of tree height, tree diameter, species, and local climate.

This research develops a novel dataset of tree photographs and trains a deep learning model to identify tree species from images in the silvopasture setting. These tools will be combined with a broader suite of tools being developed to accurately estimate tree carbon from phone-based tree height and diameter measurements to enable carbon payments on agricultural operations transitioning to silvopasture. In the future, these phone-based measurements will be used to augment and improve the models developed in this paper.

Specifically, in this paper I compile a novel dataset of high-fidelity tree photographs, taken in profile perspective. The dataset contains classified images of the 7 most common tree species used in silvopasture in the southeastern US, where pilot projects for this silvopasture transition are underway. I then train a deep learning model to predict tree species from these profile-vew tree photographs.

2. Related work

(0.5-1 page): You should find existing papers, group them into categories based on their approaches, and talk about exemplary ones in each category: Discuss strengths and weaknesses. In your opinion, which approaches were clever/good? What is the state-of-the-art? Do most people perform the task by hand? You should aim to have at least 10 references in the related work. Include previous attempts by others at your problem, previous technical methods, or previous learning algorithms.

Image classification is a common problem solved using convolutional neural networks. [[DESCRIPTIONS OF IMAGENET, ALEXNET, VGGNET, GOOGLENET, AND RESNET]] [4–6].

[[PAPER]] Expanded the possibilities of using pre-trained models in the described frameworks for use in more targeted settings. [[SUMMARY OF APPROACH]]. [[SUMMARY OF FINDINGS]].

In Feng et al.'s work on long-tailed object detection, they explore training classification models on highly-similar objects, a similar challenge to that conducted in this paper [2]. [[SUMMARY OF APPROACH]]. [[SUMMARY OF FINDINGS]]

Carpentier et al. use a self-collected dataset to a very similar problem: tree detection through bark [1]. [[SUMMARY OF APPROACH]]. [[SUMMARY OF FINDINGS]].

Fricker et al. also attempts to classify trees, however

CVPR #*****

from an aerial perspective instead of the profile perspective taken in this paper [3]. [[SUMMARY OF APPROACH]]. [[SUMMARY OF FINDINGS]]

3. Methods

Methods (2 pages): Describe your learning algorithms, proposed algorithm(s), or theoretical proof(s). Make sure to include relevant mathematical notation, e.g. when you formulate your input(s), output(s) and the loss function(s). It is okay to use formulas from the lecture notes. For each algorithm, give a brief description (2-3 sentences) of how it works. Again, we are looking for your understanding of how these deep learning algorithms work. Although the teaching staff probably know the algorithms, future readers may not (reports will be posted on the class website). Additionally, if you are using a niche or cutting-edge algorithm (e.g. binary network, SURF features, or anything else not covered in the class), you may want to explain your algorithm using several paragraphs. Note: Theory/Algorithms projects may have an appendix showing extended proofs (see appendix description below). Assume the reader has completed CS231N. You don't need to explain filters and max-pooling, but if you use something like stochastic strides, you should explain that

4. Dataset and features

Dataset and Features (0.5-1 pages): Give details about your dataset: How many training/validation/test examples do you have? Is there any data preprocessing you did? What about normalization or data augmentation? What is the resolution of your images? Include a citation to where you got your dataset from. Depending on available space, show some examples from your dataset. Try to include examples of your data in the report (e.g. include an image, show a waveform, price graph, etc.).

This paper develops a novel dataset of high-fidelity tree photographs, taken in profile perspective, labeled with the tree species. The species included are the seven most common species of tree used in silvopasture in the southeastern US: Black Locust, Black Walnut, Honey Locust, Loblolly Pine, Northern Red Oak, Pecan, Chinese Chestnut.

I compile these photographs and labels from images scraped from the internet. I augment augment my dataset with reflections and random scaled croppings of each image. All original and tranformed images are then center-cropped and scaled for use in a deep learning model. Lastly, the dataset is filtered using a binary classifier trained to identify profile images of trees to ensure that the image dataset scraped from the internet has a high likelihood of only containing trees. I provide sample images from this dataset with their tree likelihood in Figure 1 and I provide full detail of how this novel dataset is constructed in the subsections below.

Figure 1. Four sample images with species labels from final dataset

4.1. Image scraping

I use three data sources to generate a dataset of tree images: the Harvard Arboretum Plant Image database [[CITE]], the Arbor Day Foundation website [[CITE]], and the results of Microsoft Bing image searches for each tree species of interest. I also evaluated BarkNet, a dataset of close-up photographs of tree bark supporting a tree-bark identification model, however, this dataset did not contain any of the species of interest in this project.

I developed website-specific python codes to scrape and download images of each tree species from each source. I downloaded all available images from the Harvard Arboretum and Arbor Day sources for each of the species of interest. And for the results of my Bing Image searches, I downloaded all of the first 750 images that were not protected from download access. I use Bing image search instead of Google image search because of recent changes in Google's website layout that make web scraping more difficult.

This results in a preliminary dataset of 7,707 images, detailed in Table 1.

Tree species (N)	Total	Bing	Arbor Day	Harvard
Black Locust	605	562	0	43
Black Walnut	651	563	4	84
Honey Locust	501	501	0	0
Loblolly Pine	495	488	7	0
Northern Red Oak	579	531	7	41
Pecan	611	591	6	14
Chinese Chestnut	662	467	3	192
Total	7707	6905	54	748

Table 1. Number of images scraped from each data source, by species

4.2. Dataset augmentation

I augment the dataset with a reflection of each image about its vertical axis. I do not include horizontal and diagonal reflections because all tree images classified with this model are expected to be up-right. I also augment the dataset with the original and mirror image of a random cropping of each image at 3 different relative sizes: 10%, 25%, and 50%.

In total, this augmentation increases my dataset by a factor of eight and results in a dataset of 32,832 images.

4.3. Image cropping, scaling, and

I center and scale all images to the same shape and size for use in a deep learning model. I first upscale all images so that the minimum dimension of each image (height or width) is 1024 pixels. I next center-crop each image to a 1024x1024 pixel square. And finally I downscale all images to 224x224 pixels and normalize them based on the mean and standard deviation of the datset used by Pytorch developers to pretrain their available models [1]. These transformations are implemented so that this image dataset follows the common dimensions and normalizations used by the pretrained models available for use through Pytorch [[CITE]].

4.4. Dataset filtering

Lastly, I filter my dataset to only include, with high likelihood, profile-perspective photographs of trees. I choose to filter my dataset because the method I use to construct my dataset introduces erroneous images with somewhat high likelihood. I confirmed this with manual inspection, observing errors like drawings of trees or images loosely related to trees (e.g., a table made from the wood of a black walnut tree). The former is more of a concern in the Harvard Arboretum and Arbor Day Foundation datasets, and both the former and latter are concerns in the results of Bing image searches. My dataset also includes random croppings of all images, which may not include any component of a tree and I choose to filter these croppings too.

There are no publicly available binary classification models for tree images, so I use transfer learning to leverage the pretrained weights of ResNet50 and construct a binary classifier [[CITE]] specific to this research. I do so by replacing the final fully connected layer of ResNet50 with the same fully connected layers and pooling described in the methods section above.

To train these final layers of the pretrained ResNET model, I create a separate dataset of tree and not-tree images, labeled as such, by scraping Bing for select search terms. I include images of ”tree photograph”, ”tree bark photograph”, and ”tree leaf photograph” in the tree class, and I include images of ”tree drawing”, ”table”, ”hand”, ”map”, and ”diagram” in the not-tree class; I determined this not-tree search terms with a manual review of a sample of observations of my original tree species dataset. I transform these images with the same scaling and normalizations as described above. With ten epochs of training, this binary classifier achieves 99.06% accuracy on the validation dataset.

I filter my tree species dataset by running this classifier on my images and only keeping those images whose prediction probability for the tree class exceeded a threshold of 85%. I include a histogram of tree class predictions for all images in my database in Figure 2. This filtering results in a final dataset of 24,944 images, 75.97% of my original augmented dataset.

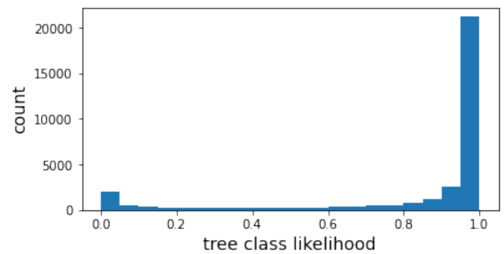

Figure 2. Histogram of tree class likelihood in tree species augmented dataset

Details of my final tree species dataset are presented in Table 2 by tree species and image source.

Tree species (N)	Total	Bing	Arbor Day	Harvard
Black Locust	3587	3325	0	262
Black Walnut	3845	3327	23	495
Honey Locust	2925	2925	0	0
Loblolly Pine	3106	3059	47	0
Northern Red Oak	3535	3236	42	257
Pecan	3875	3754	42	79
Chinese Chestnut	4071	2905	17	149
Total	24944	22531	171	2242

Table 2. Final tree species image dataset, after augmentation and filtering, by species and source

5. Results and discussion

Section 5: Experiments/Results/Discussion (2-3 pages)

You should also briefly give details about what hyperparameters you chose (e.g. why did you use X learning rate for gradient descent, which optimizer did you pick, what was your mini-batch size and why) and how you chose them. Did you do cross-validation, and if so, how many folds? This should not take more than 1-2 paragraphs. If you want to list more details, please do so in the supplemental material.

Before you list your results, make sure to list and explain what your primary metrics are: accuracy, mAP, inception/mode scores, etc. Provide equations for the metrics if

[1]RGB means = [0.485, 0.456, 0.406]; RGB standard deviations = [0.229, 0.224, 0.225]

necessary.

For results, you want to have a mixture of tables and plots. Both quantitative and qualitative results are necessary. To reiterate, you must have both quantitative and qualitative results! This includes unsupervised learning (talk with the TAs on how to quantify unsupervised methods). Include visualizations of results, heatmaps, saliency maps, examples of failure cases and a discussion of why certain algorithms failed or succeeded. In addition, explain whether you think you have overfitted to your training set and what, if anything, you did to mitigate that. Make sure to discuss the figures/tables in your main text throughout this section. Your plots should include legends, axis labels, and have font sizes that are readable when printed.

Heres a list of qualitative quantitative methods for analysis that might be helpful in your project. None of these are necessary nor will be explicitly looked for by graders rather, we wanted to provide some (non-exhaustive) guidance on analysis methods:

Saliency maps

Class visualization

t-SNE

Confusion matrices

Common qualitative errors

GANs: compare the generated output to NN in training set (quantitative and qualitative)

GANs: image quality metrics like Inception and Mode scores

VAE: Reporting measures like Annealed Importance Sampling (AIS)

6. Conclusion and future work

Section 6: Conclusion/Future Work (1-3 paragraphs)

Summarize your report and reiterate key points. Which algorithms were the highest-performing? Why do you think that some algorithms worked better than others? For future work, if you had more time, more team members, or more compute, what would you explore?

7. Appendices

Section 7: Appendices

All the text in sections before this point must fit on eight (8) CVPR-style pages or less. Extra figures can go beyond the limit, but please do not put all your figures at the end just to fit the page limit. TAs will also be focusing their attention largely on the previous sections, so anything critical to your project should go in those sections, if possible. (If you are unsure of something, please feel free to make a private Ed post).

This section is optional. Include additional derivations of proofs which weren't core to the understanding of your proposed algorithm. Usually, you put equations or other details here when you don't want to disrupt the flow of the main paper.

8. Contributions and acknowledgements

Section 8: Contributions Acknowledgements (not part of page limit)

In this section, you must explicitly state what each person on your team did for the project. If you made use of public code (e.g. from GitHub), please provide a link to the original repo. Additionally, you must mention any non-CS231N collaborators and include a brief sentence on what they did for your project. See the AlphaGo paper's contributions statement for an example. If you're part of a research lab and made use of their job scheduling, containerization, or GPUs, briefly include a sentence description on this as well.

9. References

Section 9: References/Bibliography (No page limit)

This section should include citations for: (1) Any papers mentioned in the related work section. (2) Papers describing algorithms that you used which were not covered in class. (3) Code or libraries you downloaded and used. This includes libraries such as scikit-learn, TensorFlow, PyTorch, etc. For simplicity, please use the provided BibTeX file in the template (see our "Section 2" suggestions for how to easily get BibTeX citations from Google Scholar). Main body text, figures, and any discussions are strictly forbidden from this section. We are excluding the references section from the page limit to encourage students to perform a thorough literature review/related work section without being space-penalized if they include more references.

References

[1] M. Carpentier, P. Giguère, and J. Gaudreault. Tree species identification from bark images using convolutional neural networks, 2018. 1

[2] C. Feng, Y. Zhong, and W. Huang. Exploring classification equilibrium in long-tailed object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3417–3426, October 2021. 1

[3] G. A. Fricker, J. D. Ventura, J. A. Wolf, M. P. North, F. W. Davis, and J. Franklin. A convolutional neural network classifier identifies tree species in mixed-conifer forest from hyperspectral imagery. *Remote Sensing*, 11(19), 2019. 2

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015. 1

[5] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. 1

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions, 2014. 1