Introduction

Project Title and Members

Wood classification in computer vision: A Survey

Sarah Beaver, Thanh Le

https://github.com/haroldle/feature_engineering_class_project

Video links in case Canvas video does not have audio:

https://myunt-my.sharepoint.com/:v:/g/personal/thanhle5_my_unt_edu/Ed3TWUMitG5OgP1U1dZuPNAB1wmK-fNRJYe0P-TCpIkyGQ?e=wt0RXv

https://drive.google.com/file/d/1P8jrk7lrDkGskPwjj_FbqW7AfKiKbCXv/view?usp=sharing

Idea description

Benchmarking multiple state-of-the-art deep learning models (VGG16, ResNet, Vision Transformer) and traditional computer vision models to understand how each feature extraction technique affects the prediction. Traditional Computer vision techniques include LBP - Local Binary Pattern with variants with SVM - Support Vector Machine, and using feature fusion technique (fusing LBP with SIFT - Scale Invariant Feature Transform descriptor); inspired by [4]

Goals and Objectives:

Goal: This paper aims to answer two questions: Can deep learning models perform better than the proposed method in An automatic recognition system? How does each feature extraction technique affect the performance of the models?

Objectives: In answering those questions, these are the objectives:

1. Understanding the image dataset
2. Recreating the model from D. V. Souza *et al* [2]
3. Doing Transferred Learning on VGG16, Siamese Net

Background

Related work

LBP (multiple variants). LBP has been used intensively in wood classification [2, 5] and defection [7]. By fusing LBP variants, the machine learning model performances greatly increase [2].

Vanilla CNN. Vanilla CNN - (VGG16 and ResNet) - is used in wood classification [5]. These models have a potential in learning the feature patterns of the woods' images.

Your Model

We created Random Forest and Support Vector Machine models to classify the 46 classes of wood tissue. We looked at the accuracy of these models and the accuracy from the paper. These models were done twice, one time for non rotation invariant uniform images and second for the rotation invariant uniform images.

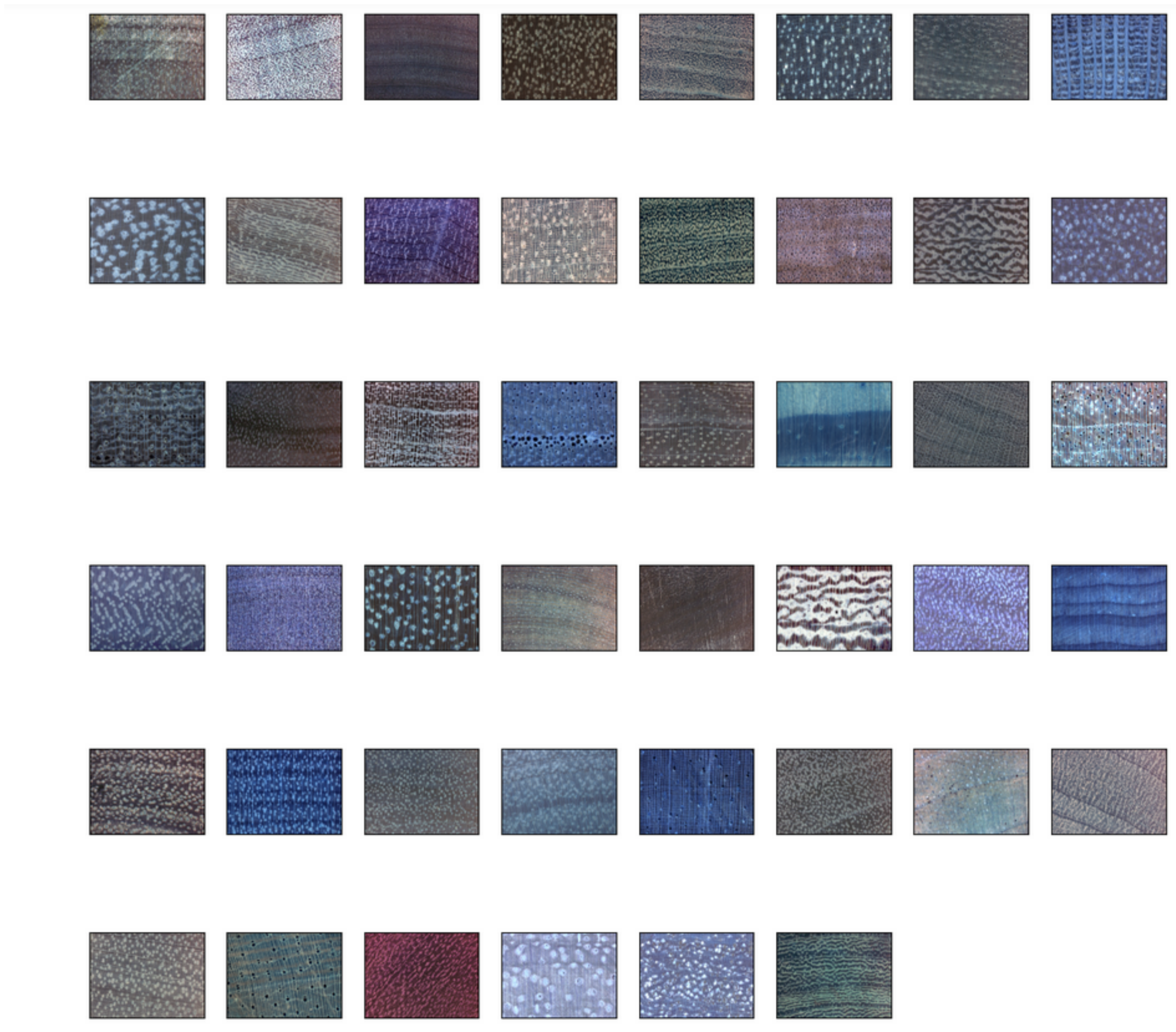
We also created VGGNet and used a pretrained version that is based on facebook. To make it work with our wood classification we did transfer learning on the final layers and trained those layers for 50 epochs. We had it flip on the images, shift on the images, rotation, and brightness range. It had a batch size of 64 and a learning rate of .001. This model was able to achieve an accuracy of 79.8%.

For the Siamese model we created a pair of images. These pair images are where the images come from the same category (positive) or not the same category (negative). This is for the model to learn similarity between images and we trained the model to learn these similarities. For the architecture we used VGG19 to extract image features of the pair. Then we grab the feature vector and calculate the similarity distance between the pair of images. We used Euclidean distance for the similarity distance. We feed this to the neural net to decide if it is similar or not. The feature is the similarity distance and if the images are similar or not. This model got an accuracy of 42%.

Dataset

Dataset

The dataset is from D. V. Souza *et al* [2]. It contains 46 different wood species. For each species, there are roughly 17 images. A picture of all wood species can be seen on the next page.



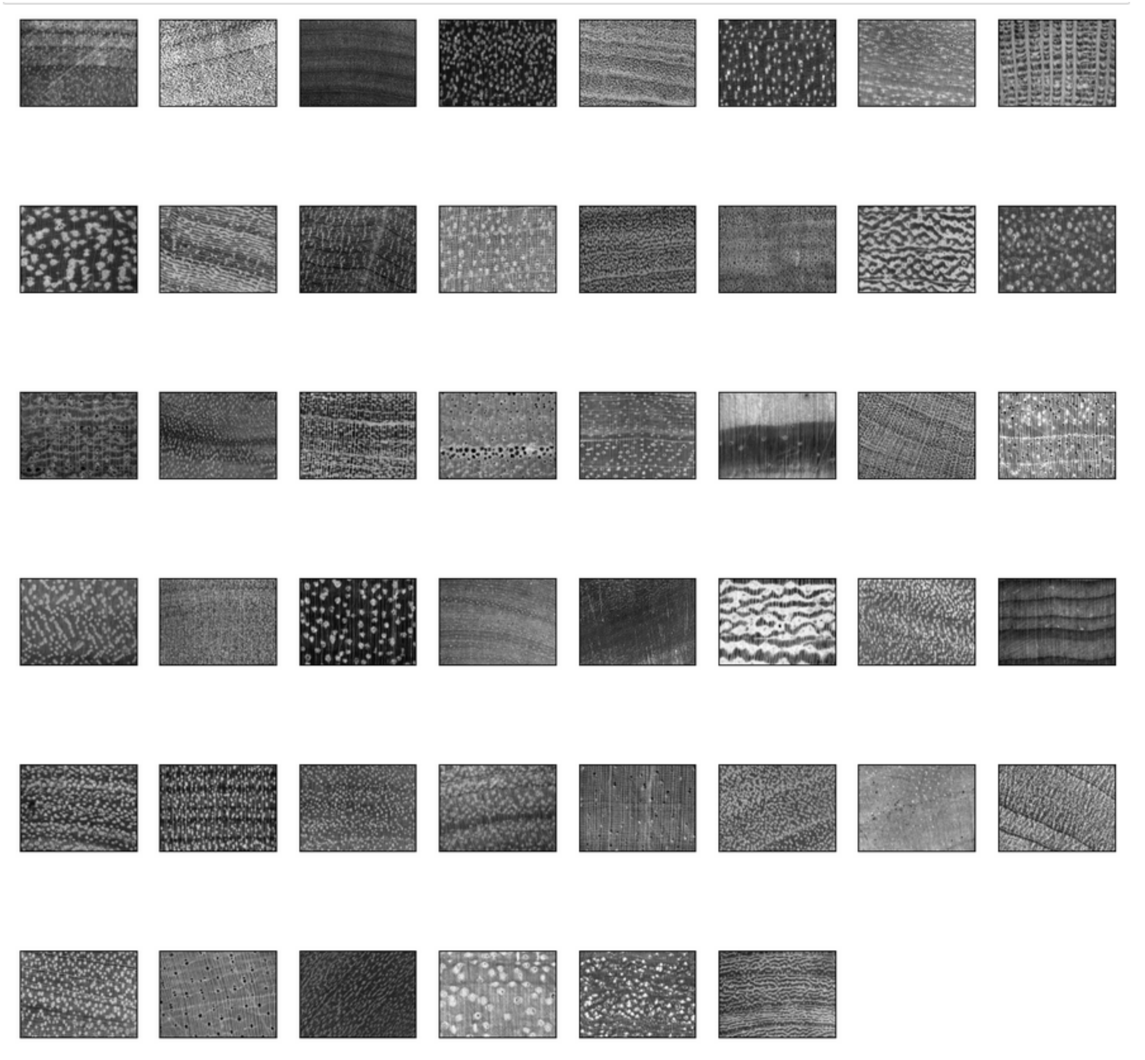
Detail design of Features

The dataset is from D. V. Souza *et al* [2]. It contains 46 different wood species. For each species, there are roughly 17 images.

The feature extraction techniques that we are going to use are:

1. Local Binary Pattern (texture feature), [2, 5, 7]
2. Feature fusion (fusing multiple features for classifying) (LBP with multiple variation),; inspired [4]
3. CNN (convolutional neural network - vanilla VGG), [8]

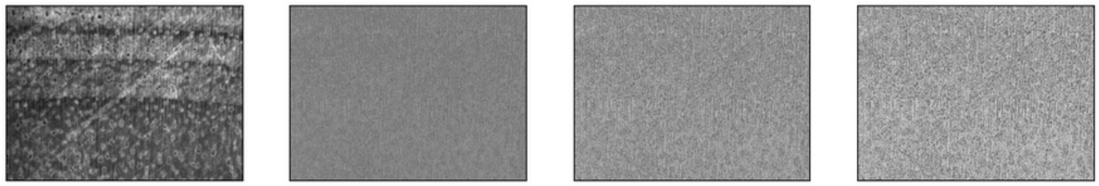
4. Siamese - calculating distance between feature vectors (two CNN models) [6]



The above image shows all the images as grayscale which is the first step for getting the features.

```
FROM LEFT TO RIGHT
ORIGINAL IMAGE => LBP 8 POINTS WITH RADIUS OF 1 => LBP 16 POINTS WITH RADIUS OF 2 => LBP 24 POINTS WITH RADIUS OF 3
```

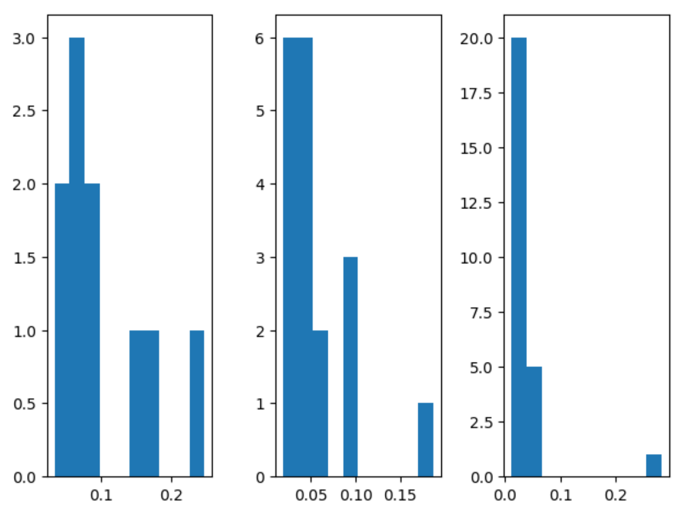
```
Out[5]: <matplotlib.image.AxesImage at 0x7f86ec4b7fa0>
```



The image above is an example of Local Binary Pattern(LBP) being performed on an image. In this case we are doing rotation invariant uniform images. We also performed for non rotation invariant uniform images but the result looks similar and one must look closely to see a difference.

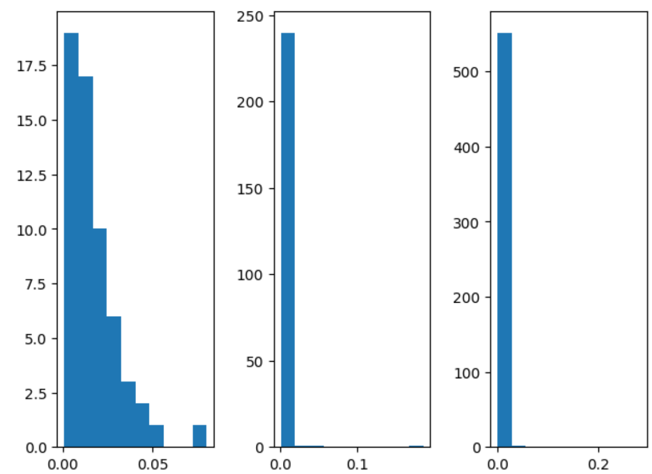
Below we can see a histogram of LBP rotation invariant with the left histogram being for 8 points with a radius of 1, the middle is the LBP rotation invariant for 16 and a radius 2, and the right histogram is LBP rotation invariant of 24 and a radius of 3.

```
In [7]: fig, ax = plt.subplots(1, 3)
        fig.tight_layout()
        ax[0].hist(extract(an_image, 8, 1.0, 'uniform'))
        ax[1].hist(extract(an_image, 16, 2.0, 'uniform'))
        ax[2].hist(extract(an_image, 24, 3.0, 'uniform'))
        plt.show()
```



Below we have histograms of LBP non-rotation invariant with the same setup as the above.

```
In [10]: fig, ax = plt.subplots(1, 3)
         fig.tight_layout()
         ax[0].hist(extract(an_image, 8, 1.0, 'nri_uniform'))
         ax[1].hist(extract(an_image, 16, 2.0, 'nri_uniform'))
         ax[2].hist(extract(an_image, 24, 3.0, 'nri_uniform'))
         plt.show()
```



Analysis of Data & Implementation

The local binary pattern (LBP) requires a grayscale image. This process is done by comparing the center pixel to the surrounding neighbor pixels. For each surrounding pixel we compare it to the center and if the center is larger than the neighbor then the neighbor gets assigned 1 and in the reverse gets assigned a 0. Then after that is done for each neighbor then it is flattened out and numbered by the index. This gets raised to power of 2 which then gives the decimal number which is the LBP value. This value is then placed in the output array. This technique extracts local features in the image and since we are doing tissue we care more about local features than about global features. We care about local features as with global features all the tissues will look the same. We perform basic fusion for the radiuses of 1,2, and 3 of 8,16, and 24 points. We perform it twice, once for rotation invariant uniform and non rotation invariant uniform.

We created Random Forest and Support Vector Machine models to classify the 46 classes of wood tissue. We looked at the accuracy of these models and the accuracy from the paper. These models were done twice, one time for non rotation invariant uniform images and second for the rotation invariant uniform images.

We also created VGGNet and used a pretrained version that is based on facebook. To make it work with our wood classification we did transfer learning on the final layers and trained those layers for 50 epochs. We had it flip on the images, shift on the images, rotation, and brightness range. It had a batch size of 64 and a learning rate of .001. This model was able to achieve an accuracy of 79.8%.

For the Siamese model we created a pair of images. These pair images are where the images come from the same category (positive) or not the same category (negative). This is for the model to learn similarity between images and we trained the model to learn these similarities. For the architecture we used VGG19 to extract image features of the pair. Then we grab the feature vector and calculate the similarity distance between the pair of images. We used Euclidean distance for the similarity distance. We feed this to the neural net to decide if it is similar or not. The feature is the similarity distance and if the images are similar or not. This model got an accuracy of 42%.

Results

For each result we are showing the accuracy, macro average, and weighted average of the overall model for precision, recall, f1-score, and support. We have it for each wood species as well but are only showing the overall models here as there are 46 classes and would be really long to see here. The full results can be seen in https://github.com/haroldle/feature_engineering_class_project/blob/main/Classification%20report.ipynb.

Below we have a screenshot of the accuracy of Random Forest for non rotation invariant uniform images.

accuracy	0.897260	0.897260	0.897260	0.89726
macro avg	0.902413	0.887320	0.883882	438.00000
weighted avg	0.915166	0.897260	0.896519	438.00000

Below we have a screenshot of the accuracy of Random Forest for uniform images.

accuracy	0.840183	0.840183	0.840183	0.840183
macro avg	0.838656	0.833783	0.821604	438.000000
weighted avg	0.855396	0.840183	0.834999	438.000000

Below we have a screenshot of the accuracy of Support Vector Machine for non rotation invariant uniform images.

accuracy	0.888128	0.888128	0.888128	0.888128
macro avg	0.887396	0.886959	0.880041	438.000000
weighted avg	0.903899	0.888128	0.890186	438.000000

Below we have a screenshot of the accuracy of Support Vector Machine for uniform images.

accuracy	0.956621	0.956621	0.956621	0.956621
macro avg	0.948903	0.953359	0.946784	438.000000
weighted avg	0.962870	0.956621	0.956783	438.000000

We can see from these results that the SVM model for uniform images performs the images. We could not reach the level of accuracy for the models D. V. Souza *et al* [2] which had an accuracy of 97 percent.

The accuracy of the VGGnet model was 79.8%. The accuracy of the Siamese net is 42%.

Project Management

Based on the objectives we have completed 1 and 2. Sarah did the model training and classification report on SVM and Random Forest for 100%. Le did the data understanding and LBP feature extraction for 100%. Sarah has done 100% of the document while Le did 100% of the video and source code description.

For increment 2, Sarah did and completed 100% of the VGGNet model while Le did 100% of the Siamese model. Sarah fixed and wrote 100% for increment two. For the video we each discussed our part (30% for Sarah and 70% for Le). Le handled the github.

Reference

[1] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* vol. 60, *(2),* pp. 91-110, 2004. Available: https://libproxy.library.unt.edu/login?url=https://www-proquest-com.libproxy.libr

ary.unt.edu/scholarly-journals/distinctive-image-features-scale-invariant/docview/1113598206/se-2. DOI: https://doi-org.libproxy.library.unt.edu/10.1023/B:VISI.0000029664.99615.94.

[2] D. V. Souza *et al*, "An automatic recognition system of Brazilian flora species based on textural features of macroscopic images of wood," *Wood Sci. Technol.,* vol. 54, *(4),* pp. 1065-1090, 2020. Available: https://libproxy.library.unt.edu/login?url=https://www-proquest-com.libproxy.library.unt.edu/scholarly-journals/automatic-recognition-system-brazilian-flora/docview/2422917952/se-2. DOI: https://doi-org.libproxy.library.unt.edu/10.1007/s00226-020-01196-z.

[3] Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint* arXiv:2010.11929, 2020. https://arxiv.org/abs/2010.11929

[4] Feng Yang, Zheng Ma, and Mei Xie. 2021. Image Classification with Superpixels and Feature Fusion Method.Journal of Electronic Science and Technology 19, 1(2021), 100096. https://doi.org/10.1016/j.jnlest.2021.100096 Special Section on InSilico Research on Microbiology and Public Health. https://www.sciencedirect.com/science/article/pii/S1674862X21000239

[5] Hwang SW, Sugiyama J. Computer vision-based wood identification and its expansion and contribution potentials in wood science: A review. Plant Methods. 2021 Apr 28;17(1):47. doi: 10.1186/s13007-021-00746-1. PMID: 33910606; PMCID: PMC8082842. https://plantmethods.biomedcentral.com/articles/10.1186/s13007-021-00746-1

[6] Koch, G., Zemel, Richard, and Salakhutdinov, R., 2015. Siamese Neural Networks for One-shot Image Recognition. https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf

[7] R. N. N. Rahiddin *et al*, "Classification of wood defect images using local binary pattern variants," *International Journal of Advances in Intelligent Informatics,* vol. 6, *(1),* pp. 36-45, 2020. Available: https://libproxy.library.unt.edu/login?url=https://www-proquest-com.libproxy.library.unt.edu/scholarly-journals/classification-wood-defect-images-using-local/docview/2426493638/se-2. DOI: https://doi-org.libproxy.library.unt.edu/10.26555/ijain.v6il.392.

[8] Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv. 20141409.1556. https://arxiv.org/abs/1409.1556