CPSC 6300 - Applied Data Science (Fall 2023)
Segmentation of image and video streams to detect unhealthy trees from drone footage

Abhiram Kasturi
Clemson University
kasturi@clemson.edu

ABSTRACT
This project focuses on the detection of unhealthy trees in drone footage through image and video segmentation. The dataset consists of keyframes extracted from drone videos and annotated for unhealthy trees based on verbal description. Two models were trained: Logistic Regression and Convolutional Neural Network(CNN). The models demonstrate their efficiency in predicting and segmenting, contributing to the automated detection of tree health issues in environmental applications.

1. INTRODUCTION
This project aims to detect unhealthy trees from drone footage and also seeks to answer the question, "How to find the best model for predicting unhealthy images?" The project presents two different models: Logistic Regression and Deep Learning.

Detecting and addressing tree health is crucial for effective environmental conservation, requiring an accurate method for early identification of potential issues. Existing approaches for assessing tree health may need more precision, motivating the project to overcome these limitations. Drawing inspiration from advancements in machine learning, particularly in the context of image and video segmentation, the goal is to develop a robust model capable of accurately predicting and segmenting unhealthy trees from drone footage. This project aims to provide a valuable tool for early detection, contributing to informed environmental management decisions.

The data in this project was gathered from the IBM Box website (https://ibm.box.com/s/dm5hmblioqtzyf0knw99hp90tg5g0svz), a repository containing a small dataset derived from drone video footage. The keyframes (images) were extracted for analysis, focusing on annotating trees as healthy or unhealthy based on verbal descriptions. The dataset consists of 73 images, with 45 representing healthy trees and 28 representing unhealthy trees. Notably, the dataset exhibits a balanced distribution between the two classes. Using verbal descriptions for annotation introduces a subjective element, relying on human interpretation for categorization.

2. EDA
In this project, each entry in the dataset corresponds to a specific image, with two columns: file_path denoting the file location of the image and label categorizing whether the image represents a healthy or unhealthy tree. These columns constitute the fundamental elements under examination. The unit of analysis revolves around individual images, allowing for a meticulous evaluation of model performance through metrics like Accuracy, Precision, and F1 Score. This focused approach aims to assess the models' proficiency in accurately classifying and segmenting tree health in drone footage.

2.1 Dataset
This dataset has a total of 73 observations, each corresponding to a specific image extracted from drone footage. These observations have been categorized into two distinct classes based on the health status of the trees:

- **Healthy Images:** Consisting of 45 observations, these images show instances where the trees are healthy.
- **Sick Images:** Consisting of 28 observations, these images show instances where the trees are unhealthy.

All the 73 images in the dataset are unique. It signifies that the dataset does not contain any duplicates or repeated entries; each observation is distinct. This directly makes the dataset more dependable and accurate, adding to the analysis.

2.2 Data Cleaning
In this analysis of a drone video dataset comprising 73 images, the data cleaning process was minimal, given the nature of the dataset consisting of images. No explicit data-cleaning steps were necessary as images were extracted as key frames from the drone footage. The dataset's focus on annotating healthy and unhealthy trees based on verbal descriptions suggests a subjective element in how the trees are classified.

3. VISUALIZATIONS
3.1 Class Distribution Analysis

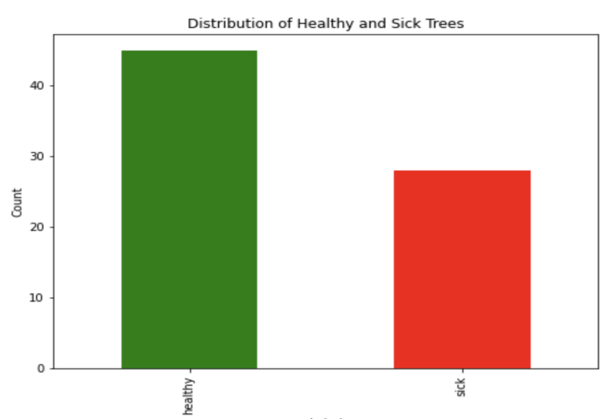

Figure 3: Distribution of healthy and sick trees.

The above bar plot visually depicts the count distribution of healthy and sick trees, using distinct bars for each category. This provides a rapid evaluation of class balance, which is crucial for machine learning model training. An uneven distribution may require considerations for data preprocessing strategies.

3.2 Visualization of Sample Images

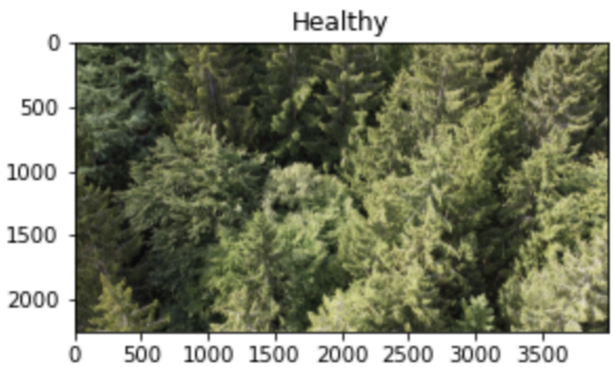

Figure 1: Healthy Tree

Figure 2: Unhealthy Tree

The above two images are two sample images selected from the dataset. The first image depicts a healthy tree, and the second image depicts an unhealthy tree.

3.3 Pixel Intensity Analysis

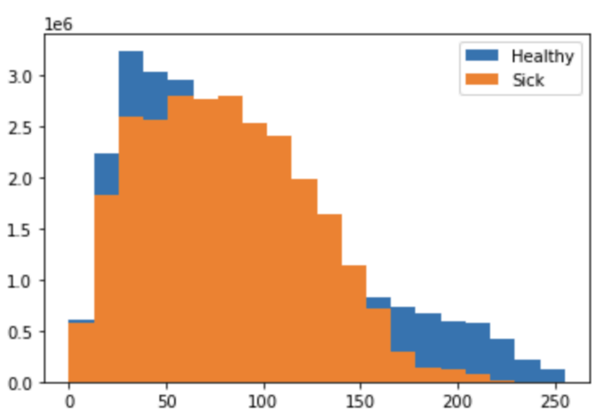

Figure 3: histogram of pixel intensities

The above histogram shows the distribution of pixel intensities for healthy and sick trees. Histograms are practical visual tools for understanding the frequency and variation of pixel values within an image. In this context, the histogram provides insights into the contrast or color variations between healthy and sick trees.

Analyzing the histogram allows us to see patterns and characteristics in the pixel intensity distribution, such as variations in peaks or spread, indicating differences in color or brightness levels between healthy and sick trees. Peaks within specific intensity ranges may correspond to common features associated with each category.

3.4 Edge Distribution Comparison

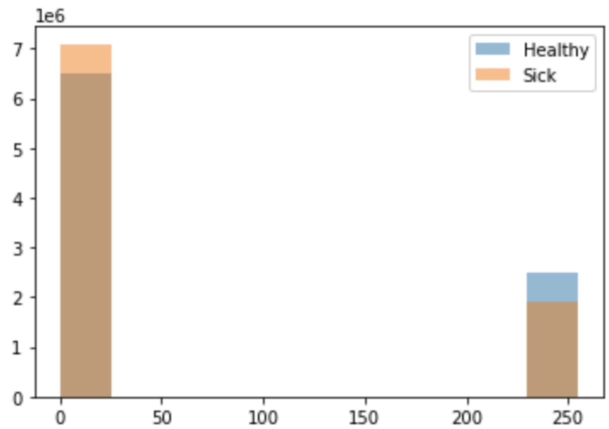

Figure 4: The graph compares edge distributions.

The provided graph compares edge distributions, reflecting the presence and arrangement of edges or boundaries between objects in an image. This graph gives valuable insights into similarities or differences in edge characteristics between the compared elements.

Examining the graph allows for the identification of patterns in edge distribution, signifying shared traits or distinctions in structural features.

4. SEGMENTED TREE REGION ANALYSIS

```
# Calculate size of segmented tree
print(f'Healthy tree size: {np.sum(healthy_seg)}')
print(f'Sick tree size: {np.sum(sick_seg)}')
```

```
Healthy tree size: 574614355418

Sick tree size: 592263490417
```

Figure 5: Size of segmented tree region

Computing the size of the segmented tree regions refers to the process of determining the spatial dimensions or extent of individual tree structures within an image or dataset that has undergone segmentation.

5. IMPLEMENTED MODELS

Two model choices were made for this project. One is the logistic regression model for the binary classification task of determining whether tree images are healthy or not, and the other model is Convolutional Neural Networks (CNN). Initially, the logistic regression provided computational efficiency, making it suitable for scenarios with linear relationships, but it could not capture intricate spatial patterns and complex relationships.

After conducting EDA and observing the nature of the images, it became evident that the dataset contained subtle shades of visual features crucial for accurate classification. In this aspect, CNN, well suited for non-linear data, was implemented to be more precise in classification since it is well done for non-linear data where convolutional and pooling layers capture spatial relationships in the image data.

```
class CNNModel(nn.Module):
    def __init__(self):
        super(CNNModel, self).__init__()

        self.conv1 = nn.Conv2d(3, 16, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(16, 32, 5)
        self.fc1 = nn.Linear(32 * 53 * 53, 120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 2)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = x.view(-1, 32 * 53 * 53)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

Figure 6: CNN Implementation

The selection of the CNN model for this classification task became more evident after achieving a Normalized Mutual Information (NMI) score of 1.00. This crucial metric justifies the model choice by evaluating it with the ground truth. The NMI score generally ranges between [0,1], where 1 indicates a perfect correlation, and 0 indicates no mutual information. Here, an NMI score of 1.00 indicates that the chosen model effectively captures the underlying structure of the data.

```
nmi = normalized_mutual_info_score(all_labels, all_preds)
print(f'Normalized Mutual Information: {nmi:.4f}')
```

```
Normalized Mutual Information: 1.0000
```

Figure 7: NMI score

6. MODEL METRICS

The test accuracy obtained in the evaluation of the logistic regression model was 86.67%, and the test error rate was 13.33%. This indicates that the model misclassifies a few tree images incorrectly, and there is still room for improvement in test accuracy. The main reason could be the model's capability to classify only the linear data, and to address this issue, a Convolutional Neural Network (CNN) was implemented.

```
# Calculate accuracy
accuracy = accuracy_score(all_labels, all_preds)
print(f'Test Accuracy: {accuracy * 100:.2f}%')
```

```
Test Accuracy: 86.67%
```

Figure 8: Logistic Regression Accuracy

```
# Calculate error rate
error_rate = 1.0 - accuracy

print(f'Test Error Rate: {error_rate:.2%}')
```

```
Test Error Rate: 13.33%
```

Figure 9: Logistic Regression Test Error Rate

By implementing the CNN model, the test accuracy obtained was 100%, which assessed the model's performance. This indicates that the model performs exceptionally well in classifying the images without misclassifications compared to the logistic regression model. Complementing this, a test error rate of 0% was obtained, which was way less compared to the test error rate of a logistic regression model. Hence, based on the accuracy results, the CNN model was concluded to work well for the tree image segmentation task compared to the logistic regression model.

```
    print('Test Accuracy: {} %'.format(100 * correct / total))
    print('Test Error Rate: {} %'.format(100 * (1 - correct/total)))
```

```
Test Accuracy: 100.0 %
Test Error Rate: 0.0 %
```

Figure 10: CNN Test Accuracy and Error Rate

The reports derived from the confusion matrix F1 score are analyzed to assess the model's fit to the data.

The logistic regression performs moderately in fitting the data, as the figures below depict. The figure below shows that the f1 score for the logistic regression model is below 90%, which indicates an unbalanced performance between precision and recall. However, the confusion matrix shows that the model incorrectly identified two of the ten healthy trees as sick, possibly needing help differentiating between the two categories.

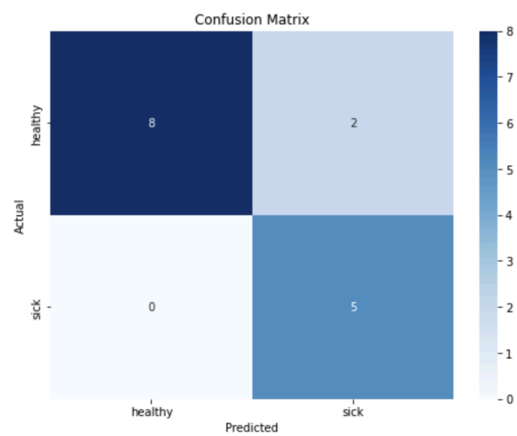

Figure 11: Logistic Regression Confusion Matrix

```
Classification Report:
              precision    recall  f1-score   support

     healthy       1.00      0.80      0.89        10
        sick       0.71      1.00      0.83         5

    accuracy                           0.87        15
   macro avg       0.86      0.90      0.86        15
weighted avg       0.90      0.87      0.87        15
```

Figure 12: Logistic Regression Classification Report

Implementing the CNN model significantly improved the fitting of the data, as the figures below depict. Test accuracy, test error rate, F1 score, confusion matrix, and other metrics show that the model does an outstanding job classifying the images into healthy and sick trees with very few misclassifications.

At 100%, the F1 score—which considers precision and recall, is comparably high. This score demonstrates a robust model performance by indicating a balanced performance between recall (the number of relevant items selected) and accuracy (the number of relevant items assigned). Also, the confusion matrix demonstrates the model's performance in detail, showcasing that out of the ten instances of healthy trees, all trees were correctly classified, and none were misclassified. Also, all 5 cases of unhealthy trees were correctly identified.

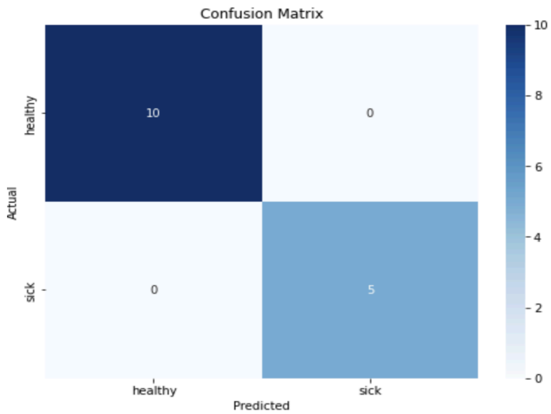

Figure 13: CNN Confusion Matrix

```
Classification Report:
              precision    recall  f1-score   support

     healthy       1.00      1.00      1.00        10
        sick       1.00      1.00      1.00         5

    accuracy                           1.00        15
   macro avg       1.00      1.00      1.00        15
weighted avg       1.00      1.00      1.00        15
```

Figure 14: CNN Classification Report

7. BEST FITTING MODEL

Based on the results of F1 scores (precision, recall) and the confusion matrix for both models, the performance of CNN is way better than that of the logistic regression model. An F1 score of 100% indicates a perfect balance of precision and recall, indicating a good classification of images. This is supported by the confusion matrix results, which show that the model perfectly identifies the healthy and sick trees. Therefore, based on these results, the Convolutional Neural Network model fits the data better than the logistic regression model.

8. RESULTS

```
Actual value: healthy
Predicted value:  healthy
```

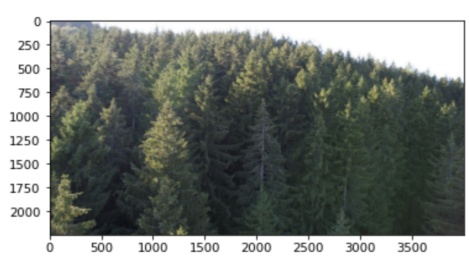

Figure 15: Result 1

```
Actual value: healthy
Predicted value:  sick
```

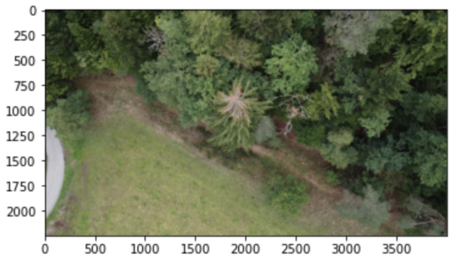

Figure 16: Result 2

```
Actual value: healthy
Predicted value:  healthy
```

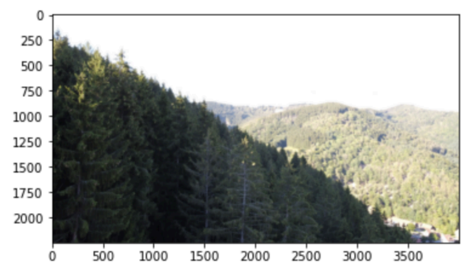

Figure 17: Result 3

9. CONCLUSION

This project demonstrates the segmentation of image and video streams to detect unhealthy trees from drone footage by employing logistic regression and convolutional neural network algorithms. The main motive of the model was to classify the input tree images into healthy or unhealthy. The developed model achieved an accuracy of 93.3% in the classification task.

Domain experts, such as forestry and agriculture departments, could significantly derive meaningful insights through this project. They can detect the issues beforehand, which might allow for timely interventions to prevent further deterioration. The outcomes of this project can automate the health assessment of trees, and experts could also use this technology to streamline and expedite tree health assessments across larger areas, saving time and effort in manual inspections.

Although the model classifies the images accurately, this project could be improved by enhancing the model's robustness and adaptability. This can be done by applying ensemble techniques combining CNN and RNN for sequential data analysis. Also, the model can be integrated with multimodal data, such as thermal imaging alongside visual images, which could provide complementary information about tree health.