Extrapolating Overstory Tree Density from Drone Images through a Spatial Convolutional Network

Dhanuj Gandikota
The University of Michigan
dhanujg@umich.edu

Video Presentation Link:
https://www.youtube.com/playlist?list=PLdhO7e7rD_m5f_SMwbL1FmrLe-1mP79kV

Abstract (150 Words)

Computer vision implemented with drone robotics provides the opportunity to increase wildfire management efficiency. My project is focused on developing a network architecture which can identify density approximations in overstory tree growth. My results show that Spatial Convolutional Networks, more specifically the Perspective Crowd Counting Network (PCC-Net), holds potential in distinguishing tree height differentials and navigating noisy images given forest foliage. The optimal Mean Squared Error value obtained was **80.9,** demonstrating that the Model performs comparatively to published PCC-Net by Gao et al. 2019[1], given the lower epoch and dataset selection . The output of Density Maps provides a general understanding of image tree structure and is effective at phasing out foreign objects and low-level forest growth (seen as darker sun-hindered locations). Improvements on data processing and quantity, combined with network specialization and ecological based tuning would further improve the final network.

1. Introduction

While cyclical wildfires provide great ecological benefit, current temperate forests are experiencing wildfires at unprecedented rates in history with over 128,000+ acres burning this year to date [3]. Environmental changes and habitat fragmentation, often anthropogenic, have changed California's temperate forest structure to sustain higher densities of fire-intolerant trees, suppressed trees, and understory growth at more spatially uniform height classes [3]. This present structure provides large surface fuels and ladder fuels promoting widespread unhealthy wildfires [Figure 1]. The ecological practices of 'thinning', manual removal of plant life and prescribed burns, is utilized to modify vertical arrangement and densities of forest sections to replicate historical stand structures. Current methods of 'thinning' are effective; however, they are often labor and time intensive and thus impractical to implement over large areas.

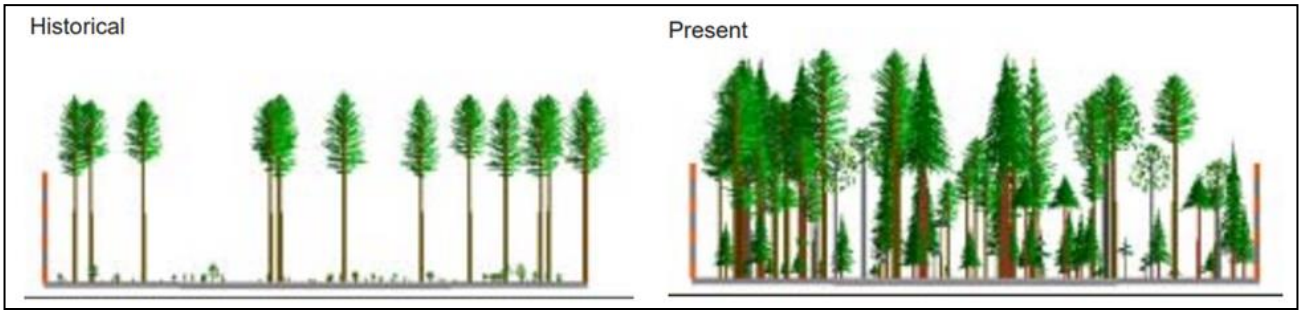

Figure 1 : Representation of historical growth in wildfire resistant forests compared to today. Healthy density levels of under growth and overgrowth tree heights prevents fires from spreading.

Computer vision implemented with drone robotics provides the unique opportunity to increase wildfire management efficiency. Automated search algorithms could identify density disparities in forest patches and provide management personally with efficient time and warming such that greater potential damage could be minimized. To search for such susceptible patches of growth, density identifying algorithms would need to be powerful enough to substantiate a difference between the very similar foliage and obtain clear differences in tree growth height. Convolutional Neural Networks offer the greatest optimization to provide a density metric that is also sensitive to the height of the tree growth. To optimize direction in this field of study, I will focus on developing promising accuracy from a network architecture which can classify density approximations in overstory tree growth (the highest level of tree growth).

1.1. Data

Concurrent with the utilization of drone technology for efficient analysis, the Data acquired for this project were aerial photos taken by drowns as shown in Figure [2].

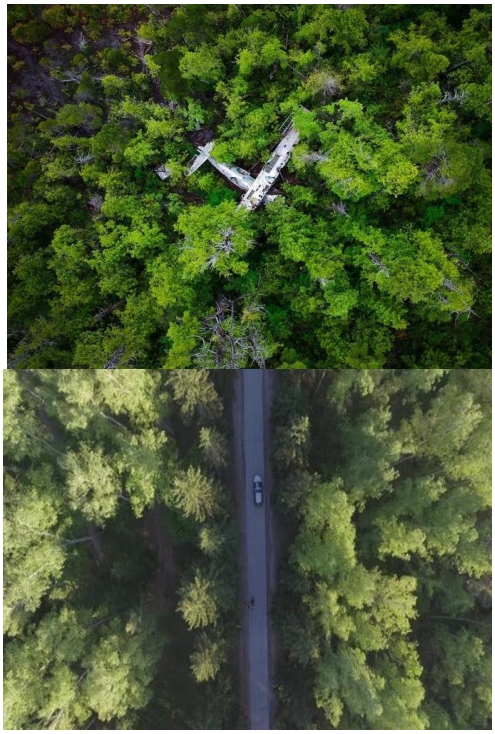

Figure 2

These images were collected from google images and were selected for their great ability to show varied tree types and different density distributions of overstory growth within the landscape. In addition, some included foreign objects such as the plane in Figure [2]. Unfortunately, as I learned in a disappointing way is that data collection is not only long but hard when maintaining integrity of specialized data. I was able to procure a set of 30 images of aerial drone forest footage.

1.2. Network Architecture

Initial research into this topic brought about the concept of object detection. . The R-CNN model family is a deep learning-based family well known for object localization and was described in the 2014 paper by Ross Girshick, et al. from UC Berkeley titled "Rich feature hierarchies for accurate object detection and semantic segmentation."[4] The network architecture of the R-CNN is shown in [Figure 3].

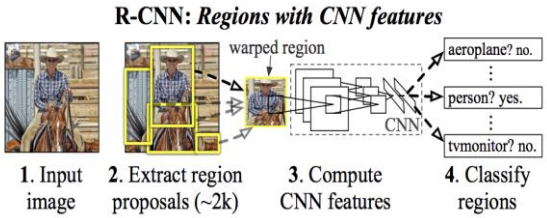

Figure 3

Research into the R-CNN model since 2014 has also seen greater work into object detection, especially with regards to obtaining density measurements from noisy and "crowded" images. Two current Convolutional Network Implementations on the forefront of density classification, more specifically in application to human crowd analysis, are Dilated Convolutional Neural Network and Spatial Convolutional Network systems. Research utilizing Dilated Convolutional Neural networks have provided an advantage in deconstructing highly congested data. Spatial Convolutional Networks, however, in recent research has shown promise in judging different camera perspectives. This judgement of dimension provides great benefit to judging overstory growth.

The inspiration for the network structure is from the 2019 paper from Junyu Gao of IEEE titled "PCC Net: Perspective Crowd Counting via Spatial Convolutional Network"[1]. This network looks to improve classification on regions with high clutter,

high appearance similarity and complex perspective changes[1]. Forest foliage provides a difficulty in all three of those goals. Furthermore, Gao highlights the design of the network architecture to aid situations in which "1) some background regions are similar to the congested region, which is usually prone to misestimation of the density; 2) the perspective change in crowd scenes cannot be effectively encoded, which causes the poor quality of a density map. "

The multi-task Perspective Crowd Counting Network (PCC Net) aims to highlight the change of perspective within images through the implementation of a perspective module, "DULR module"[1]. This module helps obtain and analyze the four directional ordinal data within each image. In addition to combat specialization of the classifier, a further stochastic approach is utilized to obtain "high" level features. The PCC Net showed very promising results with several crowd imaging datasets. The network outputs a 10-way classification score, with 1 being the highest density on the scale. In addition, the PCC-Net output a Density Map and Segmentation Map of each image.

I adapted my network architecture from the PCC Net used in Gao et al. 2019 (PCC-Gao) with the exception of the addition level of segmentation that Gao utilized in the output given the reasoning that its implementation, while showing clarity, does not contribute computationally to the density classification and is utilized as a measure of cleaning up the 'Density Map' from disturbance.

2 Methods

2.1 Data Processing

In order to expand my dataset to encapsulate hypothetical drone images that would be obtained from the same forests, I implemented a function utilizing my experience in MATLAB to generate a copy of each image that was rotated 180 degrees and flipped along the horizontal axis to maintain largest variation from original. These images were added into the total dataset.

The data was processed and standardized to match the input utilized by PCC-Gao. Utilizing my experience within MATLAB, I wrote algorithms to initially read in the image data and utilized a piece of code[2] to allow myself to choose density points of which would be stored within a .mat object. Each [.mat] object consisted of a data structure of three dimensions being focal object x location x density. These density points are critical to generating the density map which will be utilized as the validation metric to train the classifier.

Figure 4: Each Crosshair is a point I specifically clicked

[Figure 4]demonstrates the points I was able to choose within each image as focal points (trees with a significant overstory growth). Overstory growth decision was made by myself as a human utilizing my 4 years of experience as an Ecology, Evolution and Biodiversity Major at the University of Michigan.

Then I converted the images to a single channel black and white and sized them all to the ideal network specification of 576 x 768 as per the original PCC-Net. The density points stored within the [.mat] objects were concatenated into a matrix within a csv file becoming the density map.

2.2 Mapping Network Architecture

The network architecture of my PCC-Net was heavily based upon the PCC-Gao, except for the segmentation layer[1]. Gao's network architecture is represented within [Figure 5]. Each convolutional layer in Figure[5] is represented through the cuboid shapes with the dimensionality represented above.

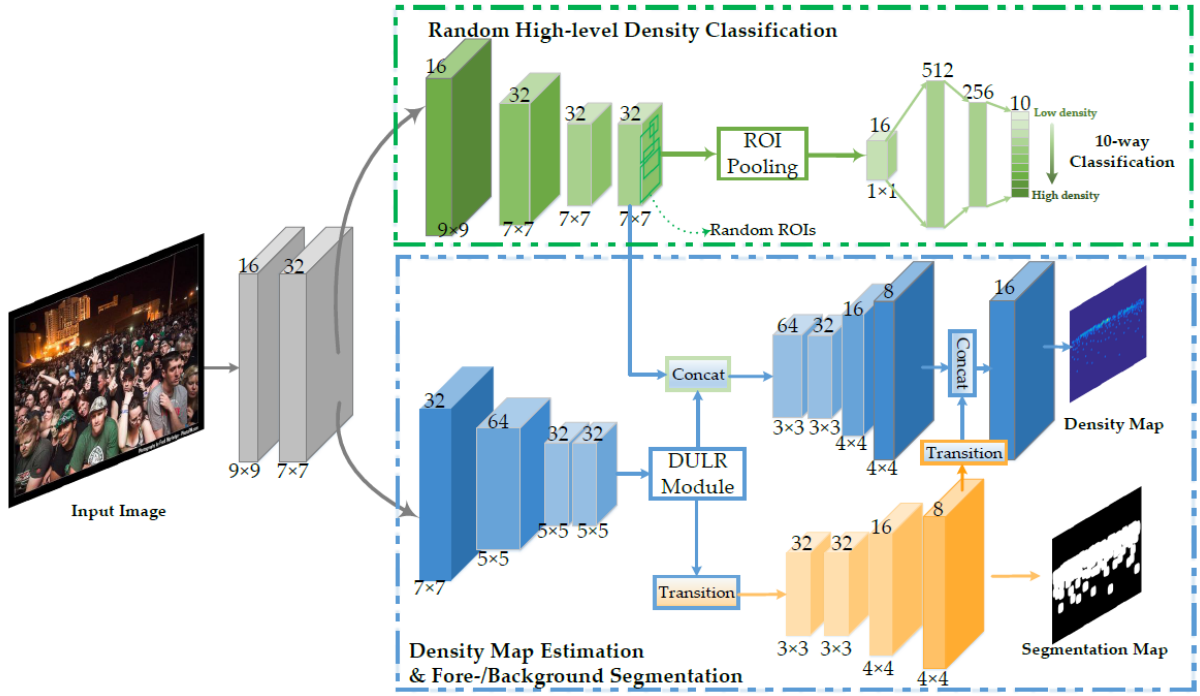

Figure 5: Implementation of PCC-Gao from Gao et al 2019

Within my implementation of the PCC-Net The 10-class density score was obtained through the following architecture.

1. *Initial Base Convolutional Layers*
2. *Initial Density Classification Layers*
    a. *Two 2D Convolutional Layers (Conv2D) followed by a 2D Max Pooling each (MaxPool2D)*
        i. *Rectified Linear Unit Correction (ReLu)*
    b. *Two more 2D Convolutional Layers*
3. *A middle ROI pooling layer*
4. *Single Conv2D*
5. *Linear Classifier with a hidden layer*
    a. *ReLu Correction*
6. *Output*

The Density Map output or my implementation followed this architecture,

1. *Initial Base Convolutional Layers*
2. *Convolutional Density Map Generation*
    a. *Two 2D Convolutional Layers (Conv2D) followed by a 2D Max Pooling each (MaxPool2D)*
        i. *Rectified Linear Unit Correction (ReLu)*
    b. *Two more 3D Convolutional Layers*
3. *Concatenation with Previous Network after node 2*
4. *DULR Module*
    a. *Down Up convolution (ConvDU)*
    b. *Left Right Convolution (ConvLR)*
5. *Second Stage Convolutional Density Map Generation*
    a. *Two layers of Conv2d Traspose*
        i. *PReLu Correction*
    b. *Two Conv2D layers*
6. *Output*

The DULR module structure and code was copied

within its entirety from Gao et al 2019 to maintain integrity of the PCC-Net and followed Figure [6]

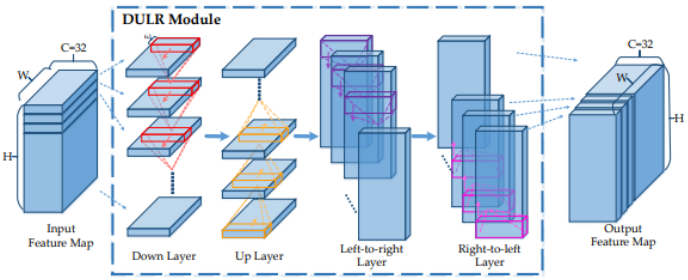

Figure 6: Image of Detailed DULR Module Architecture from Gao et al 2019

Both Python and MATLAB were utilized to complete this experiment. Within Python, the libraries utilized heavily for the PCC-Net generation was the PyTorch module. In addition, I utilized the code snippets from PCC-Gao with TorchVision and TensorBoard to generate plot analysis. The PCC-Net was trained and tested within the environment of Google Colabs utilizing the background Nvidia Tesla Graphics Processing Unit.

Implementation of the network structure took a fair amount of time and after much trial and error, the result relied heavily on structure of the PCC-Net code being similar in structure to Gao et al 2019. In order to remove the segmentation layer, I adjusted the forward pass of the final layers of the network structure to ignore an additional concatenation in [Figure 5] with the original segmentation layer dimensionality.

2.3 Network Training and Testing

The network was trained and tested utilizing a 1:4 random split in the dataset to divide the training and testing data. Training data was chosen as a random subset of 45 images with the Testing Dataset for final validation being the remaining 15. I implemented a Leave-One-Out training scheme in order to generate training validation data, in which a batch of 44 training cases was utilized to train the classifier and the remaining image was utilized to validate performance.

The network had tunable parameters within the architecture which were all originally set at the default values from PCC-Gao

- *Learning Rate = 1e -4*
- *Weight of Decay = 1*
- *Beta Weight = 1e-4*
- *Maximum Epochs : *40*
    - *Changed to conserve runtime, original experiment had epoch > 500*
- *Optimizer: Adam*

The error metric utilized for final output of the 10-class density score was the Mean Standard Error Metric (MSE) obtained from the Mean Aggregate Error (MAE) . The MSE was generated given the estimated density map given the original ground truth density map.

Network MSE results for varying the input parameters of Epoch Number and Learning Rate follows the Table below.

Epoch Number	*Learning Rate*	*Best MSE*
40	*1e-4*	*76.4*
30	**1e-4**	**80.9**
20	*1e-4*	*135.3*
25	*1e-2*	*146.7*

{*I do apologize as my computer is on the lower end of price point and computing efficiency.}

Minimizing the MSE resulted in an updated epoch number of 30 in order to maximize runtime efficiency for training/ testing without a large loss in MSE.

2.4 Network Output

The most notable network output are the generated density maps for each image within dataset. I utilized the code for the implementation of Tensor Board in order to obtain output visualization for both the density maps and graphs for the MSE over Epoch Number.

Terminal output was also modeled after the output of Gao et al 2019 in order to maintain full functionality with Tensor Board. This output contained Raw loss and MSE values generated after each epoch was run, with intermittent loss being displayed within each epoch. Additionally, Training time of the entire epoch is listed along with the best MSE and Raw Loss Value obtained thus far. The following selection represents an example output section from the 17th Epoch on my network's optimal run

```
PCC_NetTrees_Run42
    --------------------
    [mse 93.1], [val loss 0.00002432
0.00000240 0.0233 0.1959]
    --------------------
[best] [mse 84.0], [loss 0.00002476],
[epoch 14]
========================================
val time of one epoch: 52.93s
[ep 17][it 10][loss 0.00002688 0.00000157
0.0198 0.2333][1.15s]
        [cnt: gt: 12.1 pred: 76.101128]
[ep 17][it 20][loss 0.00001833 0.00000063
0.0166 0.1604][1.18s]
        [cnt: gt: 47.9 pred: 22.197775]
[ep 17][it 30][loss 0.00002465 0.00000183
0.0231 0.2051][1.14s]
…

train time of one epoch: 132.07s
========================================
========================================
```

3. Results

3.1 Numerical Result

The optimal MSE value obtained at the optimal parameters represented in section 2.4 was an MSE of **80.9** . Figure[7] shows the generation of MSE over each epoch of this experiment. Additionally, Figure [7] also shows the drop in training loss over each step taken within the classifier (epochs x iterations).

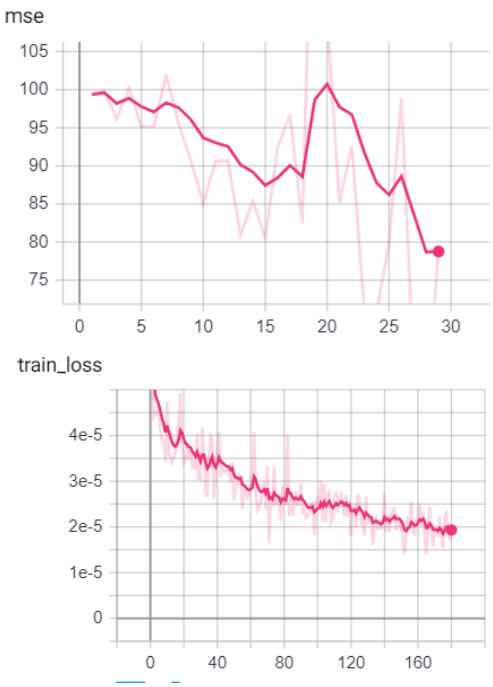

Figure 7: Graphs of MSE and raw Training Loss

3.2 Density Maps

{*I do greatly apologize, I tried everything I could think of to export the images from the TensorBoard implementation on Google Colabs, but these image qualities were the best I could obtain.}

Figure [8] shows three examples of the final output generated by the PCC-Net within TensorBoard. Each example is output given by the TensorBoard application and holds the original input image, the estimated density points of overgrowth trees and the generated density map for the image.

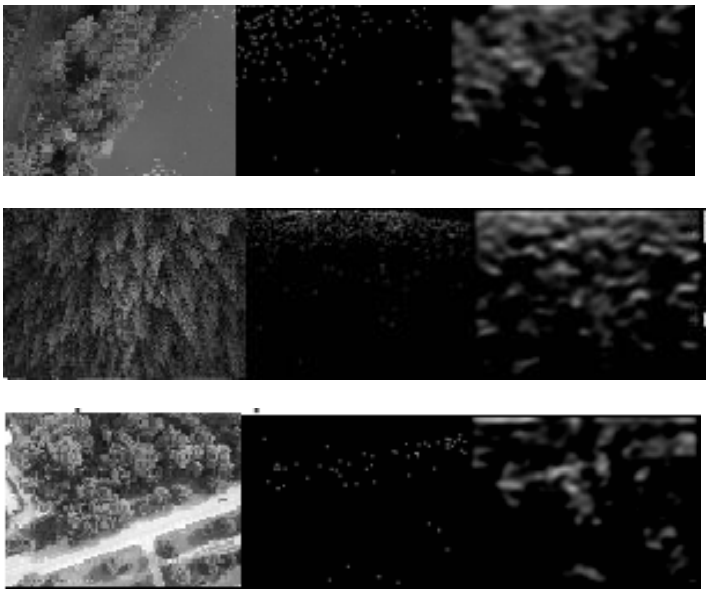

Figure 8: Three outputs of Input Image > Algorithm Focal Points > Algorithm Density Map

4 Discussion and Analysis

4.1 Comparative Results to Original Research

In order to compare the performance of my PCC-Net, I downloaded and computed the original code from PCC-Gao at the same epoch number of 30[1]. PCC-Gao utilized the well-known shanghai dataset of crowd pictures and had a data set of over 200 photos, however, chose sampling batches that were smaller than my own.

The difference between the PCC-Gao model and my own PCC-Net is Gao's selection to run epoch numbers of over 500 given the vast selection variety chosen from low batch numbers with high data pool.

As a result, given an epoch number of 30, PCC-Gao output an MSE of **124.3** for the similar epoch number, indicating results that are promising for my MSE score given my much lower epoch number and varied dataset.

4.2 Overall Results and Figures

Overall, the results given are far from conclusive on the decision to utilize a PCC-Net for overgrowth tree density estimation, however they provide a solid basis for potential in density estimation given a perspective factored method. The Comparative MSE to the original Experiment demonstrates that the Model stays within bounds of error for a density classifier such that greater dataset variety and epoch number could yield much more accurate results.

Visually, the output from the density maps provided strong evidence for the selection of tree growth from aerial drone footage. The Density Maps provide a general basis of the overall structure of the image and are effective at phasing out foreign objects (roads) and low-level forest growth (seen as darker regions of images given lower sunlight level). However as seen in Figure[8], the density maps do often phase out many degrees of growth in the process and/or perform a lighting bias and select highly contrasted objects.

4.3 Future Work and Implementation

Spending the time to truly understand the mechanisms behind this project made me also realize how technical the implementation of a complex neural network can be in an experimental status. I have numerous suggestions that would greatly increase the efficacy of this model and prevent many of the experimental biases that I may have exhibit.

Primarily would be the inclusion of much more varied and location standardized dataset. The ideal dataset would consist of not only a very large quantity of measurements, but also measurements that are taken from the same forest area in order to allow the algorithms to recognize the more nuanced characteristics of tree growth. In addition, the selection of density points for overstory growth should be performed by a much more experienced individual than myself. Final data processing could involve more contrast/ brightness manipulations to select against understory growth and foreign objects.

The network itself should not only be run on a greater number of epochs numbers but should also be analyzed with a detailed ecological perspective to justify the inclusion and exclusion of the convolutional layers. The PCC-net. While effective in density measurements, could be specialized to classify forest growth given general consistencies seen in biome growth.

References

[1] Gao, J., Wang, Q., & Li, X. (2019). PCC Net: Perspective Crowd Counting via Spatial Convolutional Network. *IEEE Transactions on Circuits and Systems for Video Technology*, 1–1. doi: 10.1109/tcsvt.2019.2919139

[2] princenarula222. (2018, July 19). princenarula222/Crowd_Annotation. Retrieved from https://github.com/princenarula222/Crowd_Annotation

[3] Overview, A. (2003). *Influence of Forest Structure on Wildfire Behavior and the Severity of Its Effects*. Retrieved from https://www.fs.fed.us/projects/hfi/2003/november/documents/forest-structure-wildfire.pdf

[4] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*. doi: 10.1109/cvpr.2014.81