Computationally Determining Number of Available Parking Spots Using Statistical Inference

Jessie Glenn and Madison Kelsey, Advisor: Jeremy Becnel
Department of Computer Science, Stephen F. Austin State University

Abstract

The intent of this project is to develop program that provides the parking availability for a parking lot within the SFA university campus. The program captures a photo of a given parking lot as input, which is then converted to a bitmap. Statistical inference is applied to the pixel data for the image to determine the number of available parking spots in the lot.

This project serves as a backend component of a .NET MAUI application deployable on Android devices. The program was developed in Visual Studio.

Figure 1. Village Parking Lot

Benefit

On a campus full of commuting students and faculty, finding a parking spot is challenging. This project provides users with the ability to monitor how many parking spots are available in a user-selected parking lot without having to visit the lot.

The advantage of using statistical inference over standard machine learning techniques is that our method does not require time consuming training for each lot and expensive equipment.

Bitmap

A bitmap is a collection of pixels on a screen in the layout of a grid, with each pixel possessing its own color data. Each pixel contains a value for red, green, and blue. This value ranges from 0 – 255, 0 meaning there is no trace of that color within the pixel and 255 meaning the pixel contains the most amount of that color. For example, if a pixel consists of only pure blue, the red value relating to this pixel is 0, the green value is 0, and the blue value is 255. Each pixel can be accessed by an (x, y) coordinate within the bitmap corresponding to the row and column for the pixel.

Algorithm Determining The Amount of Available Spots

1. Every hour three 99% confidence intervals are determined for an empty parking spot using red, blue, and green values from the pixel data. A sample of pixels along a diagonal line is used. The sample size depends on the size of the diagonal which is different for each spot.

$$Confidence\ Interval = \bar{x} \pm z\frac{s}{\sqrt{n}}$$

   The collection of confidence intervals is used to indicate a range in which the average of each color's values will fall within to indicate an empty parking spot. The confidence interval represents the color of the grey empty spot.
2. To gather information on a single parking spot, the program selects pixels along a diagonal line within the boundaries of each spot in the given bitmap. The diagonal line that the pixels are selected from spans from either the bottom left corner of the parking spot to the top right, or the top left corner to the bottom right, depending on the angle of the bitmap's image. This is shown in **Figure 3**. The pixels along the diagonal are decomposed into their red, green, and blue numerical values
3. The average of the red, green, and blue numerical values along the diagonal is calculated.
4. Each average is then compared to its color's confidence interval from Step 1. If **any** of the three averages fall outside of their confidence interval's range, the program flags the spot as being occupied by a car (i.e. the parking lot grey was not detected). If **all** three averages fall within their confidence interval's range, then the spot is available.
5. The database information for the parking lot is updated.

Pixels

Figure 2 displays the pixels of an image that are not typically seen because they are so small. The image of the Village Parking Lot is made up of its own pixels in the same manner as this eagle eye.

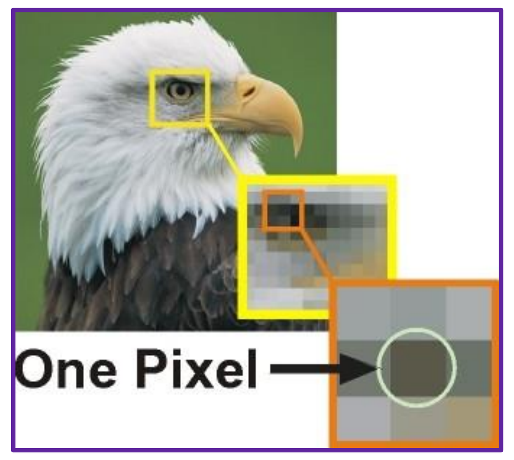

Figure 2. Eagle Eye Scaled Larger To Show Pixels

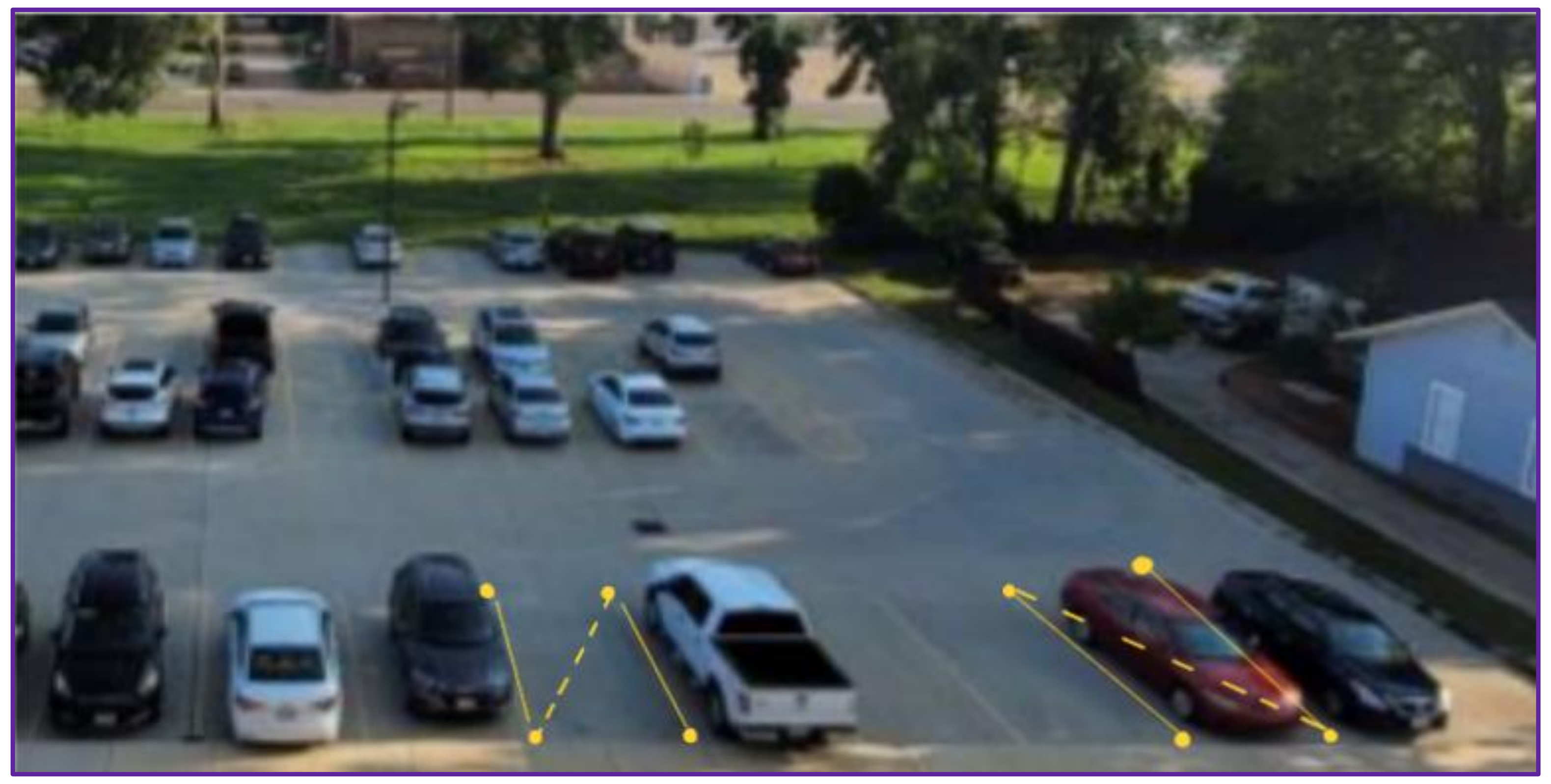

Figure 3. Village Parking Lot Displaying the "Diagonal" Algorithm

Results

We have tested 36 parking spots and the algorithm has correctly identified 28 spots. Those incorrectly identified were a result of shadow configuration. Our next goal is to broaden our algorithm to handle parking spots regardless of the amount of shade.

Future Work

A future improvement of this algorithm is to replace the still image of the parking lot with a video steam. This would allow for continual updates on the status of parking availability and increase accuracy of the number of available parking spots.

Another possibility for future work includes devising an algorithm that automatically determines the boundaries of a parking spot. This would allow for easy integration of additional parking lots.

Contact

Jessie Glenn
Department of Computer Science
P.O. Box 13063, SFA Station
Nacogdoches, Texas 75962
glennjf@jacks.sfasu.edu

Madison Kelsey
Department of Computer Science
P.O. Box 13063, SFA Station
Nacogdoches, Texas 75962
kelseymd@jacks.sfasu.edu

References

1. https://learn.microsoft.com/en-us/dotnet/api/system.drawing.bitmap?view=dotnet-plat-ext-6.0