Image Classification: Is There Parking?

Thomas (TJ) Gray, Maxwell McKee

Georgia Tech OMS

ISYE-6740: Computational Data Analytics

August 2nd, 2022

Contents

1. Problem Statement

This model attempts to answer whether there is parking or not on an average residential street using images taken of a street over the last 8 weeks. The images were taken at different times of day and day of the week, but from the same angle. Determining whether there is parking or not on a street is an important issue for commuters and business owners. Additionally, with the normalization and growth of recording technologies this model has the potential to be scaled to determine whether there is available parking across the country.

A common issue among those living in densely populated cities is the lack of available street parking. In the United States, over 75% drive as their primary means of transportation (1). Los Angeles, the second largest city in the United States, has 1.6 vehicles per household (2). Having a location to park one's car is a necessity in a car-dependent society. For the average person who is driving to and from work, knowing if they can park on a street at a given point of time would be a huge benefit in their life.

Beyond the average commuter, knowing whether an area has available car parking is valuable for property owners. Investors who are looking to purchase rental properties may want to know how much parking is available for units they are seeking to rent. The metric of available street parking could be used as an advertisement for potential renters. Parking disputes between landlords and tenants is often one of the biggest sources of aggravation between the two and is one of the primary factors that tenants may move out of a rental unit (3). This will continue to be an issue and an important factor as properties become more expensive and free parking in garages and in driveways becomes no longer feasible for many developers. Tenants will have to park on street.

A lack of parking issues can have large impacts on a local economy as well. This is especially true for small businesses. More and more people are shopping online and one of the driving reasons is the lack of available parking. It is of vital interest to various brick and mortar retail locations. There has even been analysis indicating that available parking is one of the main correlates with revenue a store is projected to generate (4).

The model that has been developed aims to start answering this important question on if there is parking by classifying sample images and determining whether there is available parking or not. Due to the limited number of images, this project looks to lay the foundation for creating a model to assess the availability of parking in a variety of places, and illuminate some of the challenges in doing so.

In a real-life scenario, the image capturing can be done through a variety of practical methods. There has been a rise in camera surveillance technology. For instance, Ring Doorbells sold 1.4 million units in 2020, and total sales for personal video surveillance were 7.9 million units in 2020 (5). Additionally, many municipalities already have street surveillance for safety reasons that can also be repurposed to capture information on whether there is parking or not.

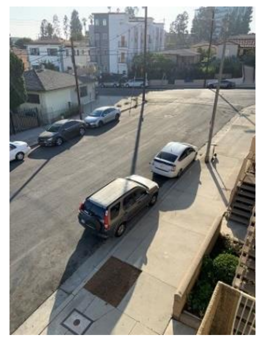
The mass adoption of image capturing technology makes this a solution that can be scaled to finally answer the question: Is there parking?

2. **Methodology**

2.1 Data Collection

We took 18 photos from the same angle of the same street at various times of various days. The first step was to read in the 18 images and confirm that they were all the same size (this consistency became important later on). One of the images was of size $(399{,}300{,}3)$ instead of $(400{,}300{,}3)$ so it was omitted.



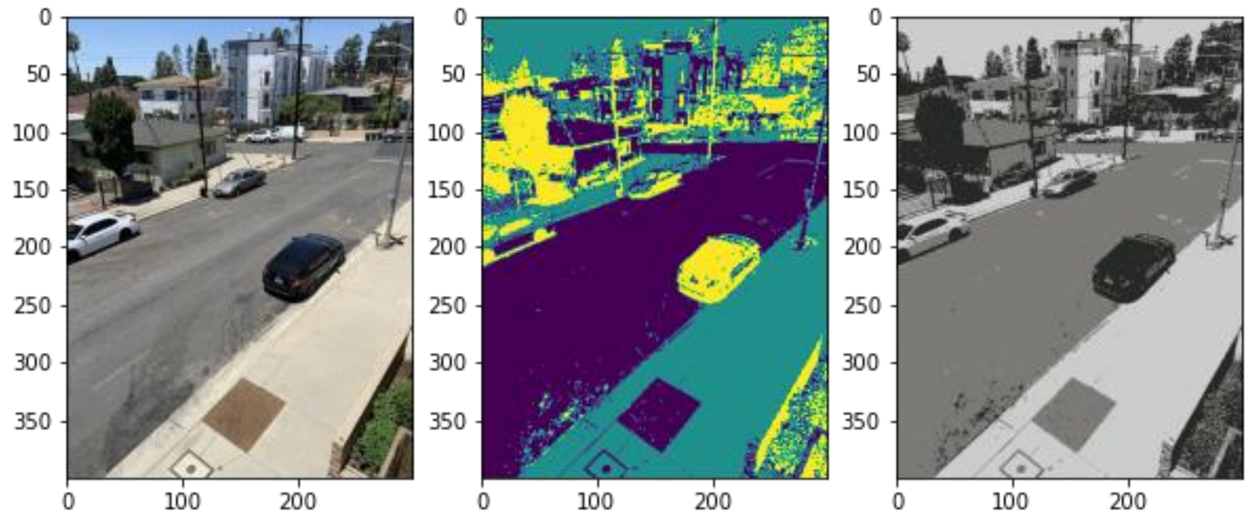
Next, we wanted to standardize the colors of the images as much as possible before compressing them to reduce variation across unimportant dimensions (ex. Brightness). The result was to have the RGB of each pixel have a mean of 0 and a standard deviation of 1, which was then rescaled to fall between $(0{,}1)$.

2.2 Image Compression with KMeans

After standardizing the images, we compressed them using KMeans clustering. Each image was compressed, or reduced, to $k = 2{,}3{,}4{,}8{,}16{,}32$ colors; this means we ran KMeans on each image individually. Each image is of size $(400{,}300{,}3)$ which was reshaped to $(120000{,}3)$ prior to running KMeans so that each pixel is treated as an observation with R, G, B as the features. The centroids were initialized spaced apart from each other to speed up convergence and to increase the likelihood of converging to global minimums instead of local minimums ("minimum" is referring to minimizing the within-cluster sum-of-squares criterion)

$$\sum_{i=0}^{n} \min_{\mu_j \in C} \left(\left| \left| x_i - \mu_j \right| \right|^2 \right)$$

This resulted in each pixel being assigned to a cluster $1, 2, \dots, k$ where each cluster label corresponds to a centroid being an average color for that cluster. The images below show a KMeans clustered image with $k = 3$. The first image is the original, the second is the image reconstructed using the labels as colors, and the third is the image using the centroids as colors.



2.3 Image Storage

The information needed to reconstruct each KMeans clustered image with k clusters is each pixel's label, the label's RGB centroid color, and the shape of the original image. This information was calculated for each image across $k = 2,3,4,8,16,32$ KMeans clusters and stored as a JSON file.

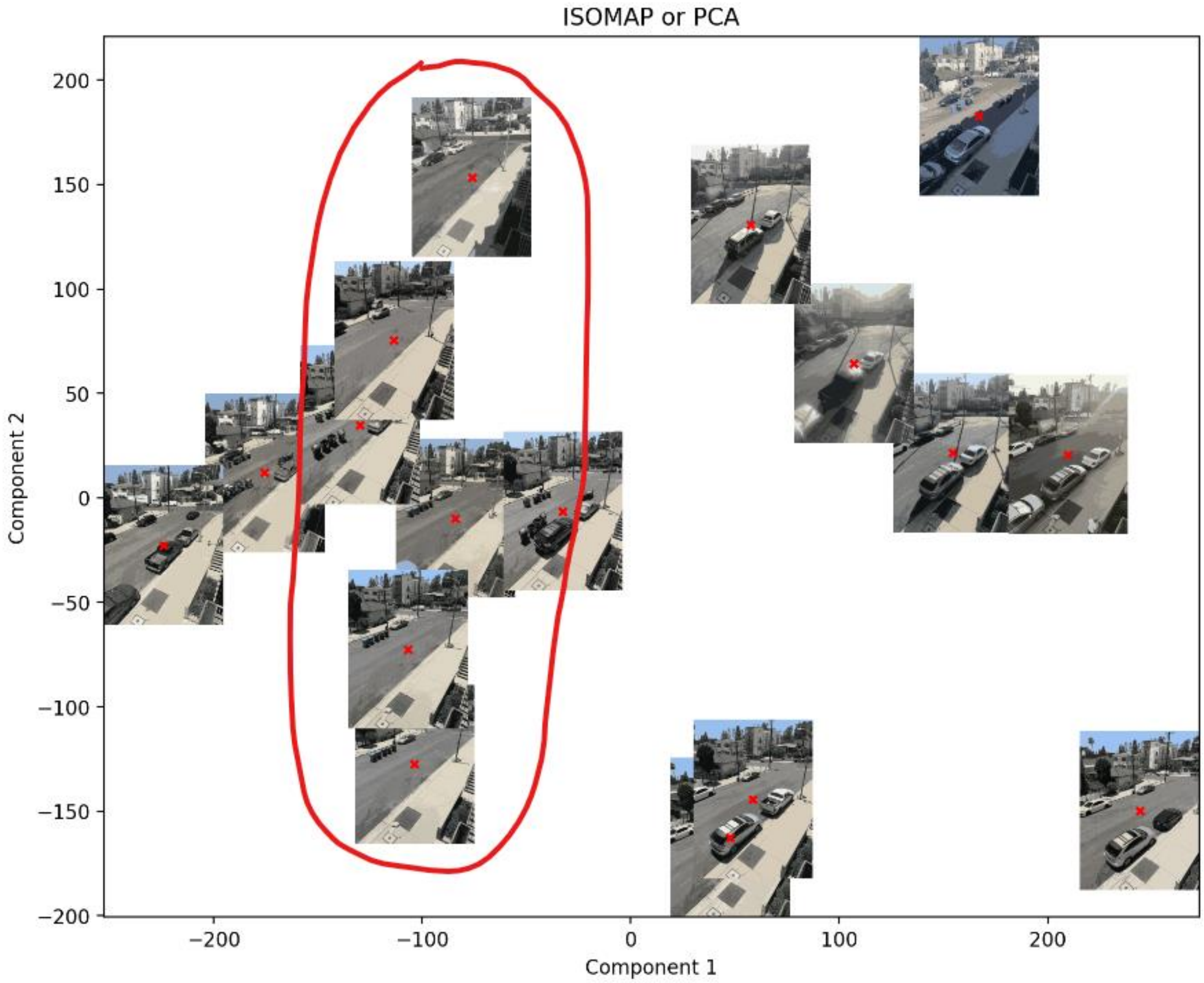
2.4 Dimensionality Reduction with ISOMAP and PCA

Our next step was to choose all the images compressed by k clusters and run ISOMAP and PCA dimensionality reduction techniques on the set of image arrays. Initially we tried this using the cluster labels as the pixels since those arrays are smaller and easier to work with, but we figured out that the label values were dependent on the initial cluster centers, so colors we inverted for different images.

PCA reduces data to n dimensions by reorienting the data such that $n = 1$ axis captures the greatest variance in the data, and so forth. The means of measuring this distance is a linear method - in this case it uses Euclidean distance. If our images dataset has linear relationship the first couple components of PCA should help identify patterns between the images.

If the data has a non-linear relationship, then we need to use a non-linear dimensionality reduction technique; hence, ISOMAP. ISOMAP can be thought of like PCA, except the distance between 2 points is not defined by Euclidean distance. A manifold is constructed amongst all the points where each point is connected to its k-Nearest Neighbors (in our case $k = 5$), and then the distance between points is the shortest path across the manifold. This measure of distance is often called "geodesic" distance and allows for a dimensionality reduction technique that considers the shape of the data.

We wrote a function that pulls in images that have been compressed by KMeans with k clusters and reduces the images to 2 selected components from either PCA, ISOMAP, or both (for ISOMAP the KNN for each point can also be specified). This function then graphs each image across the 2 selected dimensions and displays the according image. This is not a typical or necessarily "best practice" optimization strategy for finding the best components to determine image differentiation, but this manual method had it benefits too. Below shows an example of something we may have been trying to find with this display; the first 2 components of ISOMAP with $KNN = 5$ is shown.



As shown by what is circled in red, the first component of ISOMAP looks useful for identifying available parking even though the leftmost image looks out of place. This is an example of what we were looking for when plotting against these components.

2.5 GMM for Image Classification

The last step of the process was to use a Gaussian Mixture Model (GMM) to classify the images as having available parking or unavailable parking. The inputs to our GMM consisted of a combination of the PCA and ISOMAP dimensions as features. One of the primary reasons we chose to use GMM as our clustering method is because it allows for soft classification. For real-world application, while it may be useful to know if there is or is not parking, it is difficult in practice. We figured it may be more useful to have a likelihood of available parking. This helps potentially costly misclassifications, and it is just difficult to tell if there really is enough space to park a car.

Similar to other clustering methods, for GMM you specify the number classifications you want. Although, instead of having clusters be something like points, they are gaussian densities. GMM starts by initializing gaussian densities with weights, distribution centers, and density shapes. These are the 3 parameters in the model. These parameters are used to then update the posterior distribution. This process of calculating the posterior distribution and then updating parameters is called the "E&M", and is repeated until convergence. As a result, the posterior distribution tells us the fraction of each point that belongs to each gaussian density.

3. Evaluation and Results

3.1 Results

The final goal of this GMM model was to rank all 17 of our images from most likely to have available parking to least likely to have available parking. Below shows that ranking of these images, along with their hard classification. The parameters used for this model are: images compressed with $k = 8$ clusters, and GMM run on the first three components of ISOMAP and the first component of PCA. ISOMAP used $KNN = 5$ for construction of the manifold. GMM used $k = 2$ gaussian densities.

The images are ranked by GMM's output of probability of belonging to the "parking available" category – 1 being the most likely and 17 being the least likely. We have circled the instances where we think the model is clearly displaying an incorrect result.

3.2 Evaluation and Pitfalls

Our method of GMM model selection involved playing with a few moving parts and was a fairly manual process. The image ranking from the GMM model was used to see if it looked like the resulting GMM was correctly classifying images as well as ranking them appropriately. The 2D image plot by PCA or ISOMAP reduced components was used to pick out the features that would be inputted into GMM. We toyed with different combinations of k for KMeans image compression, different values of k for ISOMAP's KKN manifold construction, and different dimensionality reduced components feeding into GMM get choose a final model.

Given the size of the data we had to model with, we would say our model works pretty well on the set we trained it on. Images 5 and 11 are clearly misclassified. Arguments could be made either way about how images 9 and 10 should be classified. Image 15 we would also say is misclassified, but the strong shadows in that image are difficult to account for with this small of a dataset. Besides those mentions, generally images without many cars on the street are classified as having parking, while images with many cars present are classified otherwise.

Maybe the most obvious issue with the final model is that we evaluated its effectiveness on the same data that we trained the model. With only 17 images to work with we felt that we needed to use all the images to model with. One idea that we could have implemented to cope with the limited data would have been to build GMM with the same components, but only use 16 images to train. Then we could predict the probability that the 17th image belongs to the "available parking" classification. This could be repeated for each image and then the images could be ranked from "most available parking" to "least available parking". Unfortunately, we did not have time to do that for this project.

For the future, the first thing to change would be to include a lot more data. The data's relationship would be much easier to decipher, variance in the data would be less affected by each individual image, and a better model could be picked due to being able to evaluate its performance on a different set than the model was trained on.

Optimizing the parameters is something that we also could have done for this project, but would have been much more effective if there was a lot of images to work with. We could have used something like SKLearn's GridSearchCV to input the parameters that we described above, pick parameters that optimized a score metric for GMM, and used those to create our final model. The score metric we would use would depend on whether we wanted the model to be supervised or not. If we labelled each image as having available parking or not, we could look at something like the misclassification rate for model selection. If we chose to not label the data then we could use something like the Silhouette Coefficient (which measures how well assigned each point is to its gaussian density).

In the end, this project became more of a “proof of concept” for creating a model that can classify parking availability given street view images. We think we have shown that this set of modelling techniques can be used to make, or at least help make, this evaluation! We are really curious to know how effective of a model we could make if we used more images, perhaps a different image angle, and implemented our suggestions for the future described above.

References

1. https://www.statista.com/chart/18208/means-of-transportation-used-by-us-commuters/
2. https://www.moneygeek.com/insurance/auto/car-ownership-statistics/
3. https://rentprep.com/property-management/landlords-guide-parking-rental-properties/
4. https://www.forbes.com/sites/cameronkeng/2018/08/30/parking-will-determine-your-maximum-revenue-possible-in-brick-mortar-businesses/?sh=53cd59c11e5b
5. https://www.businesswire.com/news/home/20210512005336/en/Strategy-Analytics-Amazons-Ring-Remained-atop-the-Video-Doorbell-Market-in-2020