Computer Vision Based Hotel Geolocation Using Color Histograms

Pam Ogbebor[1]

Abstract

Image geolocation is the act of identifying the location from which a photo was taken. This task is essential to human trafficking investigations because victims are often photographed in hotels and the resulting digital content is uploaded online. The ability to geolocate indoor scenes would help law enforcement find victims of sex trafficking and prosecute offenders. A computer vision algorithm would be an efficient and data-driven approach to recognizing hotel rooms. The algorithm should be able to assess the similarity between two rooms despite varying image quality, camera angles, lighting variations, occlusions, or photo croppings. Color is an intuitive cue when it comes to image differentiation. This report analyzes the accuracy of using different color spaces (RGB and HSV) and different similarity measurements (Euclidean, correlation, cosine, histogram intersection, chi-squared, and Bhattacharyya) to recognize hotel rooms. A combination of HSV-Bhattacharyya was shown to produce the most accurate retrieval results for hotel chain recognition and hotel branch recognition. However, a recognition cue other than color may be more promising for image geolocation.

1. Introduction

Over the last decade, digital human trafficking (HT) content has become more prominent online. This has paralleled the growth of escort and content-sharing platforms on the internet. Human trafficking is defined as the exploitation of individuals either through labor or sex [1]. In the case of sex trafficking, offenders use the internet to recruit, advertise, and sell sexual services. They often photograph victims in hotels, exchange these images with accomplices, or use them for escort advertisements. If an investigator recognizes the hotel room in an image, they can rescue victims quickly, build evidence against offenders, and detect HT hotspots. Historically, this process has been done manually [2]. To geolocate a photo, a law enforcement officer may browse through hundreds, if not thousands, of hotel room images to find a potential match. This quickly becomes impractical. Therefore, a computer vision approach would offer a more efficent solution.

1.1. Hotel Recognition

Hotel recognition is an indoor classification task that is considered more complex than other geolocation tasks [3]. In the case of outdoor geolocation, many outdoor spaces contain specific skylines, street signs, and architectural qualities that make them easier to classify. However, indoor geolocation has to work with a limited set of recognizable objects. Indoor scenes contain general characteristics that are more susceptible to change. Therefore, two rooms from completely different hotels may be virtually indistinguishable, while two rooms within the exact hotel may look nothing alike (e.g. different color schemes and different bed arrangements) [4]. The ideal computer vision model should detect the unique similarities between two hotel rooms but also differentiate them from other rooms. The model should also extract accurate details from low-quality, cropped, or visually obstructed images.

1.2. The Hotel-50K Dataset

The Hotel-50K dataset is a conglomeration of over 1,000,000 hotel room images derived from 50,000 different hotels and 92 major hotel chains [5]. Its existence has helped establish the hotel recognition problem in data science communities. Images in the dataset were sourced from both travel website scrapings (e.g. Expedia's API) and an app called TraffickCam. They are accompanied with their associated hotel name, hotel chain, and geographic location. For artificial intelligence (AI) models to correctly identify hotel rooms, they need to be familiar with images with low qualities, lighting variations, and uncommon camera perspectives. These characteristics are more representative of the photos derived from real-world HT investigations and are present in the images sourced from TraffickCam. TraffickCam is a mobile application that was proposed by researcher Abby Stylianou [6]. It crowdsources hotel room images from everyday travelers by allowing them to upload photos of their accommodation. These images are likely to be captured on cellphones, depict unusual orientations, and feature clutter in the hotel rooms. They are the most ideal for training human trafficking algorithms, but only about 28% of the hotels in the dataset have a TraffickCam representation [5]. Instead, the majority of the dataset images are derived from travel websites and depict a hotel's most impressive and professionally captured rooms. Despite the unrealistic quality, their inclusion in Hotel-50k is still extremely beneficial. They can offer a comprehensive representation of hotels in a given area, unlike TraffikCam's users who may upload images from the most touristy hotels.

1.3. Context-Based Image Retrieval

The hotel recognition problem can be approached with an image retrieval algorithm. A context-based image retrieval model (CBIR) accepts query images as input and returns the most similar images from its database. Its performance can be evaluated by its Top-K accuracy which is the ratio between the correctly returned images and the total number of images returned. Properties that retrieval models commonly use to evaluate a photo's similarity include color, shape, texture, and object resemblance. Hotel-50K and its smaller dataset derivatives are promising contenders for the algorithm's database.

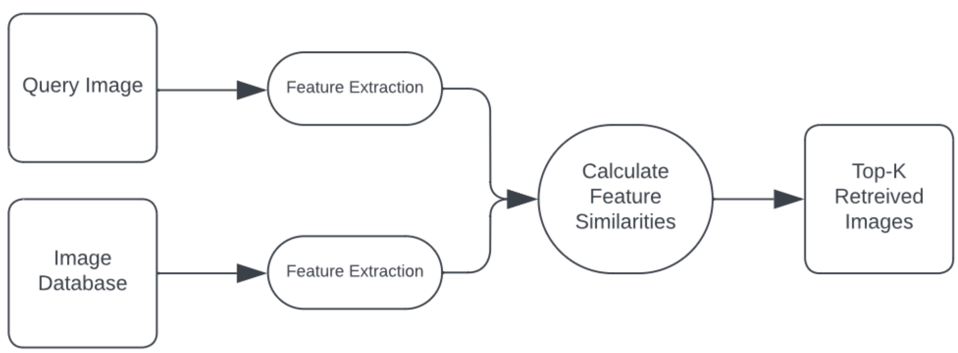

Figure 1: An example of a CBIR implementation

1.4. CBIR Drawbacks and Previous Approaches

Realistically, a database will not contain all possible images of human trafficking locations. For example, the majority of Hotel-50K's images are derived from U.S. and European hotels [5] which diminishes its global applicability. However, a dataset does not need to represent every hotel instance to be beneficial to law enforcement. In Feizi's paper, retrieval models still produced satisfactory results when thousands of hotel branches were removed from a dataset but the associated hotel chain was kept [7]. Stylianou's paper further demonstrated that hotel chain recognition was a significantly easier task to accomplish in comparison to branch recognition [5]. However, law enforcement can use the more accurate chain recognition to narrow down hotels in their areas of interest. If neither the chain nor the branch is present in the dataset, an idea would be to have a CBIR model return confidence ratings when images are below a certain similarity threshold. Therefore, low confidence ratings would notify law enforcement about the lack of possible matches. Stylianou proposed incorporating visual markers onto retrieved and query images to highlight features that the algorithm believed to be similar [8]. This provides insight into the algorithm's decision and would also prevent investigators from ignoring images that do not look overtly similar at first glance.

2. Color

2.1. Color Histograms

For two hotel chains, the most stark differentiation between their rooms is often the color palette. At the same time, the color palette can be the most similar feature between two rooms of the same chain. For this reason, color is an intuitive recognition cue for hotel rooms. A color histogram is a quantitative representation of the distribution of colors within an image. It graphs the number of pixels that contain a certain color value. They can be constructed to depict many color spaces like the popular RGB (Red, Green, Blue) and HSV (Hue, Saturation, Value). A color histogram using the RGB format creates three histograms that represent red, green, and blue. Their x-axis ranges from 0 to 255 (inclusive) and their y-axis ranges from 0 to the maximum number of times a color value occurs. The color histograms of two images can be compared with distance similarity equations. An image retrieval algorithm can then display the database images with the most similar color histogram as the query image. This recognition cue is robust against image reflections and rotations.

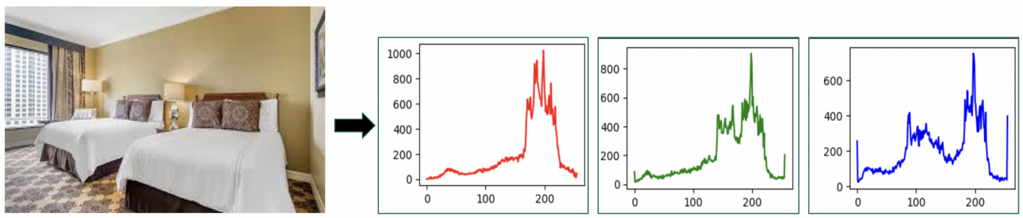

Figure 2: An RGB color histogram example

2.2. RGB and HSV

RGB (Red, Green, Blue) and HSV (Hue, Saturation, Value) are two popular color spaces used in computer vision applications. The RGB color space is an additive color model in which colors are created by combining red, green, and blue in various intensities, each ranging from 0 to 255. The more of each color is added, the closer the pixel approaches white. Its ability to represent over 16 million colors makes it the most popular choice for monitor displays. However, RGB is not considered intuitive because lighting variations can significantly affect its values. The HSV color space represents colors in terms of their hue, saturation, and value (brightness), which aligns more closely with human perception of colors and shades. Hue, which represents the type of color, ranges from 0 to 360 degrees. Saturation, which measures the color intensity, ranges from 0 to 100%. Value, the brightness of the color, also ranges from 0 to 100%. This can allow for accurate color detection under varying lighting conditions so it is widely used in computer vision.

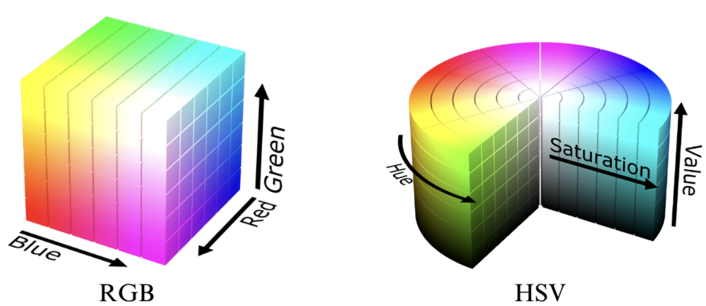

Figure 3: A visual representation of the RGB and HSV color space

2.3. Color Drawbacks

When used in isolation, color recognition has its flaws. For example, lightening or dimming an image reduces the accuracy of color-based retrieval algorithms [9]. This quickly becomes problematic because lighting variation is seen throughout images derived from real-world HT investigations. Oftentimes, HT photos also feature digital occlusions when victims are masked for legal and privacy reasons [5]. This further disrupts the results of a color histogram. Many hotel rooms reuse generic color schemes like whites, browns, and grays meaning that color may not be a distinct enough recognition cue. Additionally, color is not intrinsically related to an object's identity [10] so an algorithm that recognizes that a chair is white would theoretically outperform an algorithm that only recognizes that white is present. Photo enhancement or color correction could potentially mitigate some of these drawbacks.

3. Similarity Measurements

Histogram distance measurements are used to quantify the similarity of two color histograms. The distance measurements that this report focuses on include Euclidean distance, cosine similarity, histogram intersection, correlation coefficient, chi-squared distance, Earth Mover's Distance (Emd), and Bhattacharyya distance. These measurements were picked both for their availability in Python libraries (e.g. OpenCV and Scipy) and for their popularity in computer vision models. Analyzing these measurements side-by-side provides insight into the more accurate histogram similarity method for color-based image retrieval.

3.1. Euclidean

Euclidean distance measures the straight-line distance between two points. For histograms, it measures the distance in color space between each pair of bins, while disregarding their sizes. It is a popular metric because it is easy to compute. It is sensitive to large differences within a few bins and may ignore the smaller differences spread across multiple bins. If two histograms are similar they return a low Euclidean distance.

$$d(H_1, H_2) = \sqrt{\sum_{I=0}^{N-1} \left(H_1(I) - H_2(I)\right)^2}$$

Figure 4: Euclidean equation

3.2. Cosine Similarity

Cosine similarity measures the cosine of the angle between two non-zero vectors. It measures histogram similarity based on its orientation in a multi-dimensional space, rather than their absolute values. Its value ranges from -1 (dissimilar) to 1 (identical), and 0 indicates no similarity. It can identify images with similar color distributions regardless of the intensity or brightness levels. This can make it robust against varying lighting conditions because it prioritizes the relative distribution of colors and not their absolute quantities. However, this can make it ineffective in differentiating between histograms with similar shapes but different content.

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H_1})(H_2(I) - \bar{H_2})}{\sqrt{\sum_I (H_1(I) - \bar{H_1})^2 \sum_I (H_2(I) - \bar{H_2})^2}}$$

Figure 5: Correlation equation

3.3. Histogram Intersection

Histogram intersection measures the overlap between two histograms by summing the minimum values of corresponding bins, so a higher value represents a greater similarity. It is useful in comparing histograms with similar distributions. The result is a value that ranges from 0 (no overlap) to the total sum of the histogram. It is robust against outliers but less sensitive to differences in overall shape.

$$d(H_1, H_2) = \sum_I \min(H_1(I), H_2(I))$$

Figure 6: Histogram intersection equation

3.4. Correlation

Correlation measures the degree to which two histograms vary and it ranges from -1 to 1. A high positive correlation shows high similarity, while a low or negative correlation shows dissimilarity. A value of 0 indicates no correlation. It is useful for identifying similar patterns between histograms when the overall distribution pattern is more important than the exact bin values.

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_I (H_1(I) - \bar{H}_1)^2 \sum_I (H_2(I) - \bar{H}_2)^2}}$$

Figure 7: Correlation equation

3.5. Chi-Squared

The Chi-Squared distance returns the sum of the squared differences for every pair of bins. Unlike Euclidean distance, it is sensitive to bin sizes because each bin is treated independently. This can lead to inaccurate results when one of the color histogram bins contains an unusual count. If the histograms have similar distributions, the result is a low Chi-squared distance.

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)}$$

Figure 8: Chi-squared equation

3.6. Bhattacharyya Distance

The Bhattacharyya distance calculates the amount of overlap between two distributions. It considers the distribution's mean, variance, and overall shape which makes it a more robust metric. It ranges from 0 (identical) to 1 (dissimilar), so similar histograms result in a low Bhattacharyya distance.

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_1 \bar{H}_2 N^2}} \sum_I \sqrt{H_1(I) \cdot H_2(I)}}$$

Figure 9: Bhattacharyya equation

4. Methods

4.1. Bedroom Classification

The database for my retrieval algorithm is derived from the "Hotel-ID to Combat Human Trafficking 2021" Kaggle dataset. The dataset contains more than 97,000 images from TraffikCam and is accompanied by a CSV with their associated hotel chain and hotel branch ID. The CSV allows me to verify the accuracy of the retrieval model. My objective is to analyze the importance of color as a hotel room recognition cue, therefore only images of bedrooms are needed in the database. However, the Kaggle dataset contains various photos of lamps, paintings, and chairs. I self-tagged around 2,000 images of “Bedrooms” and 3,000 images of “NotBedrooms” and utilized Google’s Trainable Machine to create a classification model for the two classes. I ran the model through the dataset and it classified 46,000+ images out of the 97,000 images as bedrooms. This became the final database for the image retrieval model.

Bedroom NotBedroom

Figure 10: An example of the ”Bedroom” and ”NotBedroom” class

4.2. Technologies

For the creation of my algorithm, I used the programming language of Python. I chose Python because of its established presence in computer vision communities. Within Python, I utilized the libraries of OS, OpenCV(cv2), Numpy, Pandas, Matplotlib, and Scipy. In the context of my program, the library of OS aided in file management and retrieval, OpenCV was crucial in reading images, extracting colors, and calculating histograms, Numpy managed the arrays for histogram data, Pandas read CSV files and created data frames, Matplotlib visualized the retrieved images, and Scipy.spatial provided additional measurement equations.

4.3. Implementation

The steps of the algorithm include image acquisition, histogram calculation, histogram comparison, and visualization. For image acquisition, images were read using OpenCV and converted from their default BGR color space to RGB or HSV. The histograms are calculated according to the user-specified color space. All histograms were given manageable bin sizes of (8, 8, 8).

HSV ranges in OpenCV are slightly modified with hue only ranging from 0-180 instead of 0-360. All histogram calculations were conducted with the cv2.calcHist function, but some parameters are changed depending on the color space chosen. The histograms were normalized and flattened to allow different sized images to be compared and to maintain their relative color value frequency. For histogram comparison, SciPy's Euclidean and Cosine functions, alongside OpenCV's Chi-Squared, Correlation, Histogram Intersection, and Bhattacharyya functions were used to calculate the similarity between color histograms. The calculations were stored in a Python dictionary and were sorted according to the returned distance metric. Correlation and Intersection return higher values when the histograms are similar, so the algorithm ranks Correlation and Intersection's similarity values in decreasing order. For visualization, a combination of OpenCV and Matplotlib was used to depict the Top-K images. Since the retrieval database contained over 46,000 images, it was time-efficient to save the histograms of all 46,000 images into a CSV and refer back to it when calculations were being conducted. This allowed me to extract RGB and HSV color histograms for each image only once. I randomized 1,000 images from the dataset to be used as query images and saved their accuracy rate for each combination of color space and distance measurement. The accuracy rate is defined as the likelihood that a chain or branch match is present in the Top-5 retrieved images. For my statistical analysis, I used the programming language of R to conduct ANOVA and Tukey HSD tests to evaluate the statistical significance between the groups' means.

Figure 11: An example output of the CBIR model

5. Discussion

5.1. Results

A combination of HSV and Bhattcharayya distance was shown to lead to the most accurate retrieval rate for both chain and branch identification. Without accounting for methods, HSV had a higher accuracy rate when compared to RGB. Without accounting for color space, the Bhattacharyya formula had a higher accuracy rate in comparison to the other distance measurements.

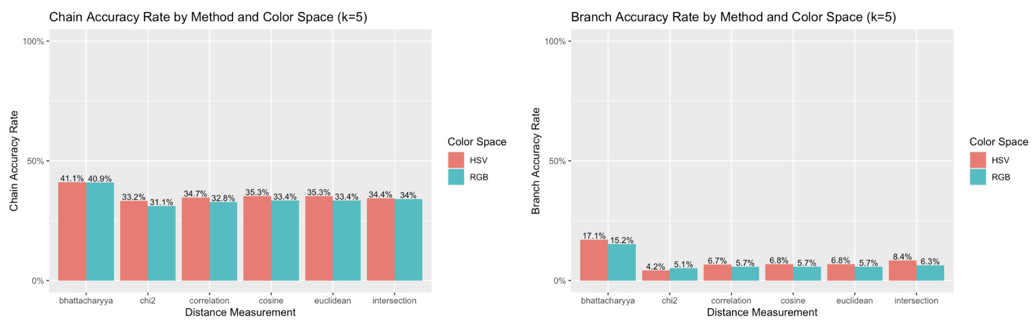

Figure 12: The likelihood that the retrieved images contains a match by color space and distance method. HSV-Bhattacharyya is the highest with 41.1% for chain accuracy.

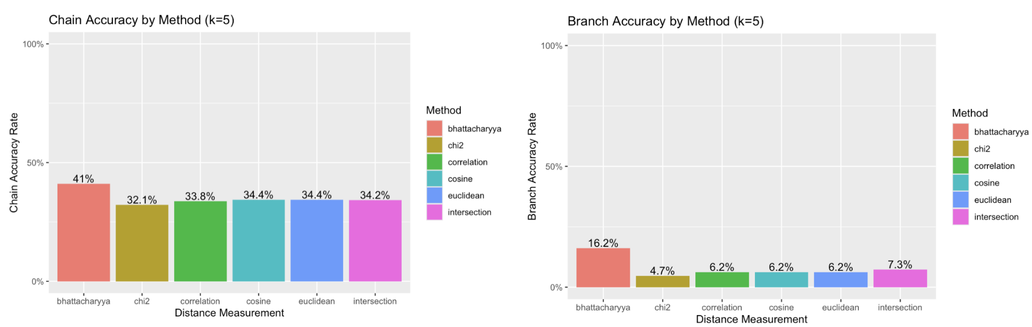

Figure 13: The likelihood that the retrieved images contains a match by distance method. Bhattacharyya is the highest with 41% for chain accuracy.

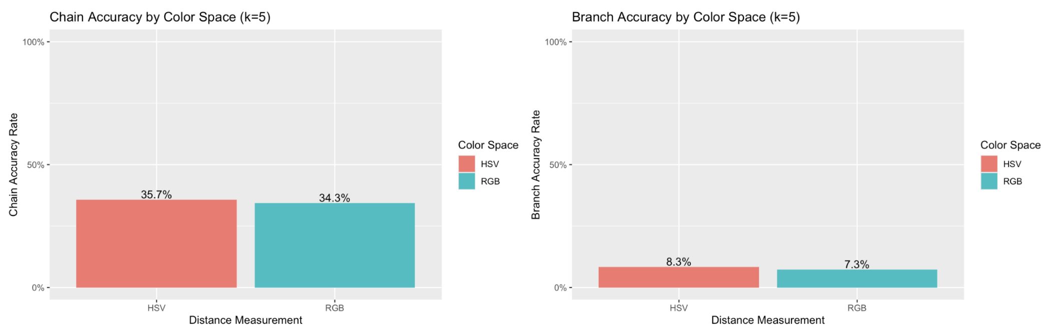

Figure 14: The likelihood that the retrieved images contains a match by color space. HSV is the highest with 35.7% for chain accuracy.

An ANOVA test showed that these differences between the distance method means are statistically significant for both chain and branch recogniton. The difference between color space means are only statistically significant in branch recognition. The interaction between color spaces and methods were not shown to be significant. However, their interaction becomes significant when the percentage of correctly matched images to the total number of retrieved images is the statistic being measured.

```
[1] "Two-Way Anova for Chain Accuracy:"
                      Df Sum Sq Mean Sq F value   Pr(>F)    
Color_Space            1    0.6  0.5880   2.593    0.107    
Method                 5    9.4  1.8865   8.319 7.34e-08 ***
Color_Space:Method     5    0.2  0.0368   0.162    0.976    
Residuals          11988 2718.6  0.2268                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
[1] "Two-Way Anova for Branch Accuracy:"
                      Df Sum Sq Mean Sq F value Pr(>F)    
Color_Space            1    0.3   0.331   4.688 0.0304 *  
Method                 5   17.4   3.488  49.445 <2e-16 ***
Color_Space:Method     5    0.3   0.056   0.799 0.5504    
Residuals          11988  845.8   0.071                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 15: An ANOVA test for the difference between the means.

A Tukey-HSD pairwise comparison of the means further confirmed that Bhattcharayya statistically resulted in more accurate retrievals than any other distance method. When the percentage of correctly matched images to the total number of retrieved images is measured, the combination of either RGB-Bhattcharayya or HSV-Bhattacharyya has a higher accuracy rate than any other color space/measurement combination for both chain and branch identification.

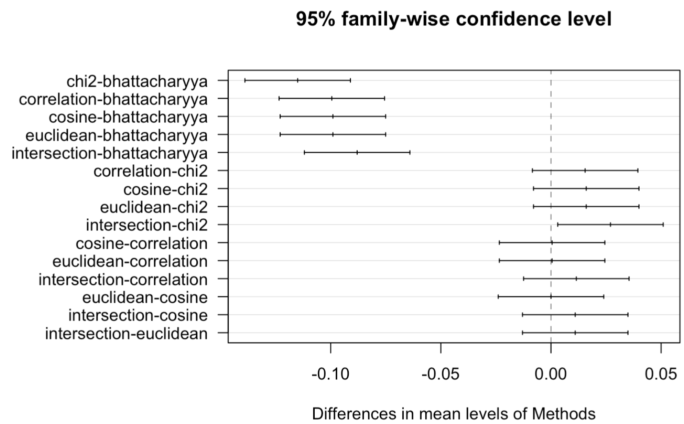

Figure 16: A TukeyHSD test for the difference between pairs of group means.

5.2. Reflection

Despite the HSV-Bhattacharyya retrieval producing notable results, there is still room for improvement. On its own, color may not be a suitable recognition cue because of the lack of detail it gives about a hotel room. More research can be conducted to include different color spaces (e.g. L*a*b*), different distance measurements (e.g. Earth Mover's Distance), or different histogram bin sizes. However, it may be more productive to consider a second recognition cue beside color.

5.3. The Potential of Object Detection

Oftentimes, hotel branches maintain the same furniture decorations across their suites but arrange them in different configurations. Hotel chains that share the same color palettes may differ when it comes to their actual furniture. An algorithm that prioritizes object similarity would perform well when dealing with repositioned furniture, a rotated image, an unusual camera perspective, or an occlusion. This would require the implementation of an object recognition ensemble within a retrieval model. The algorithm should be equipped with multiple classification systems that can extract and label furniture from query images even when they appear in uncontemporary shapes (e.g. lamps, desks, and chairs). The model will then compare the similarity of these objects with those available in the database and assess all object similarities to create a singular prediction. This has potential to differentiate between the several hotels with standard white, brown, and gray color palettes. Bhavanasi and Stylianou's paper showed an object-centric model can consistently deliver more accurate results than a full-image model [10]. Their model's prediction also improved as the number of visible objects within the image grew. A potential flaw may be the limited amount of furniture present in some HT images. In this scenario, a secondary and more general cue (e.g. color) could supplement images with a few objects. This method of stacking multiple models upon one another has already shown promise. The winner of the Kaggle competition "Hotel-ID to Combat Human Trafficking 2022", David Austin, claimed to use a five-model ensemble [11].

6. Conclusion

The hotel geolocation problem may require a combination of multiple implementation methods. Color is an intuitive recognition cue and a combination of the HSV color space and the Bhattacharyya distance resulted in notable image retrievals. However, to increase the accuracy rate of the retrieved images, color should not be used in isolation. Real-life human trafficking images contain lighting variation and feature occlusions, which can make color detection unreliable. Hotel rooms often reuse generic color palettes like whites, browns, and grays, making color a less distinct feature. Implementing a second recognition cue like object detection may offer more accurate results.

References

[1] G. L. Dimas, R. A. Konrad, K. Lee Maass, A. C. Trapp, Operations research and analytics to combat human trafficking: A systematic review of academic literature, PloS one 17 (8) (2022) e0273708.

[2] A. Stylianou, A. Norling-Ruggles, R. Souvenir, R. Pless, Indexing open imagery to create tools to fight sex trafficking, in: 2015 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), IEEE, 2015, pp. 1–6.

[3] O. Bamigbade, J. Sheppard, M. Scanlon, Computer vision for multimedia geolocation in human trafficking investigation: A systematic literature review, arXiv preprint arXiv:2402.15448 (2024).

[4] R. Kamath, G. Rolwes, S. Black, A. Stylianou, The 2021 hotel-id to combat human trafficking competition dataset, arXiv preprint arXiv:2106.05746 (2021).

[5] A. Stylianou, H. Xuan, M. Shende, J. Brandt, R. Souvenir, R. Pless, Hotels-50k: A global hotel recognition dataset, CoRR abs/1901.11397 (2019). arXiv:1901.11397.
URL `http://arxiv.org/abs/1901.11397`

[6] A. Stylianou, J. Schreier, R. Souvenir, R. Pless, Traffickcam: Crowdsourced and computer vision based approaches to fighting sex trafficking, in: 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), IEEE, 2017, pp. 1–8.

[7] A. Feizi, A. Casanova, A. Romero-Soriano, R. Rabbany, Revisiting hotels-50k and hotel-id (2022). arXiv:2207.10200.
URL `https://arxiv.org/abs/2207.10200`

[8] A. Stylianou, R. Souvenir, R. Pless, Traffickcam: Explainable image matching for sex trafficking investigations, arXiv preprint arXiv:1910.03455 (2019).

[9] R. Chakravarti, X. Meng, A study of color histogram based image retrieval, in: 2009 sixth international conference on information technology: New generations, IEEE, 2009, pp. 1323–1328.

[10] S. S. Bhavanasi, A. Stylianou, Hotel recognition using object ensembles, in: 2023 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), IEEE, 2023, pp. 1–8.

[11] D. Austin, Hotel-id to combat human trafficking 2022 - fgvc9, https://www.kaggle.com/competitions/hotel-id-to-combat-human-trafficking-2022-fgvc9/discussion/328281, accessed: 2024-06-13.