Sky Detection and Segmentation using Computer Vision Algorithm

Vivien Mak Wing Yan
School of Science and Technology
Sunway University
Petaling Jaya, Malaysia
20012134@imail.sunway.edu.my

***Abstract* - Sky region detection in outdoor images is a significant challenge in computer vision applications to accurately identify the sky region. This paper proposes a novel algorithm for automated sky region detection and segmentation using Laplacian edge detection. The algorithm aims to differentiate between sky and non-sky regions, and daytime and nighttime images in order to generate a predicted sky mask, predicted sky image, and skyline. The proposed approach is evaluated against existing algorithms that utilize different strategies for sky region detection. The algorithm is tested on a diverse dataset and achieves an average accuracy of 94.62% across different datasets and lighting conditions. The proposed algorithm's execution time is also analyzed. While demonstrating high accuracy, the algorithm faces challenges with complex structures such as buildings. Future work could focus on addressing these limitations to further improve the algorithm's performance.**

Keywords—sky region detection, skyline detection, Laplacian edge detection, computer vision, Python, OpenCV

I. Introduction

A. Problem Statement

Teaching computers to identify the sky region in outdoor images is a challenging task. Recognizing the sky's region holds significant importance across various applications, including weather forecasting, solar exposure detection, ground robot navigation, and more. This recognition serves as the initial stage for various image processing and computer vision applications. The challenge lies in developing an algorithm that can effectively differentiate between sky region pixels and non-sky region pixels even when dealing with varying lighting conditions.

B. Aims and Objectives

The main goal of this project is to create a computer vision-based algorithm for detecting and segmenting the sky. This algorithm should automatically distinguish between pixels that belong to the sky region (represented by white, 1 or 255) and pixels that belong to non-sky regions (represented by black, 0). Additionally, the algorithm should be able to generate the skyline that corresponds to the segmented sky region.

The objectives aimed to be achieved in this project are:

- To develop an efficient computer vision-based algorithm for automated sky region detection in outdoor images.
- To identify sky and non-sky regions within the images.
- To identify images under different lighting conditions such as daytime and nighttime.

C. Difference Between Proposed Algorithm and Existing Algorithms

The main difference between the proposed algorithm and the existing algorithms is the approach to performing sky region detection and segmentation. Each algorithm employs a unique strategy to tackle the task of identifying sky regions. The border points algorithm [5], uses the concept of border points to identify the boundary between sky and non-sky regions in each column of an image. This algorithm is effective for recognizing sky regions with complex structures such as buildings and flags. Nonetheless, it encounters limitations when faced with images containing highly complex structures like ferris wheels, and flowers.

Conversely, the color classification algorithm [6] relies on color attributes for sky region segmentation specifically targeting bluish-colored pixels. It can handle images with snow, windows, and challenging illumination. Nevertheless, it encounters challenges when confronted with images containing greyish skies or sea regions. The hybrid probability model algorithm [7] combines boundary vector-based and pixel-based methods by using color, gradient, and vertical position of pixels. This method is efficient, accurate, and versatile, but it requires a longer execution time for non-VGA size images.

The vision-based skyline detection algorithm [8] analyzes image brightness differences to locate visible and invisible skylines which involves candidate point selection and filtering. This makes the algorithm to be able to detect skylines under different weather conditions such as cloudy or foggy. Meanwhile, the fusion of sky region segmentation algorithm [9] combines sky region segmentation with single image dehazing techniques using fusion methods. The algorithm improves image quality by reducing color distortion in foggy images.

The proposed algorithm utilizes Laplacian edge detection to detect and segment the sky regions. The algorithm can identify sky regions in both daytime and nighttime as well as determine the presence of sky in an image. However, the results may not be ideal for all images. In summary, the main difference between the proposed algorithm and the existing algorithms is the techniques used to identify sky regions and their strengths and limitations. The proposed algorithm may not improve the accuracy, efficiency, or handling in challenging scenarios compared to the existing algorithms, however, the proposed algorithm was still able to perform sky detection and segmentation in images that are not complex.

II. Literature Review

A. A Novel Sky Region Detection Algorithm Based on Border Points

The border points algorithm [5] uses the concept of border points to identify the boundary between the sky and the non-sky regions in each column of an image. It then calculates an energy function for both regions and adjusts the threshold to detect complicated sky areas. This algorithm is effective in recognizing sky regions with complex structures, such as buildings or flags, but may fail in highly complex images, such as those with ferris wheels or flowers.

B. Detection and Extraction of Sky Regions in Digital Images based on Color Classification

The color classification algorithm [6] segments the sky regions based on their color properties. It extracts pixels that have a bluish color range and converts them into a binary image. It then selects the largest white region as the sky region and generates a sky color reference. It finally extracts all regions that match the sky color reference and excludes those that do not. This algorithm can handle images with snow, windows, or challenging illumination, but may struggle with images that have a greyish sky or a sea region.

C. An Efficient Sky Detection Algorithm Based on Hybrid Probability Model

The hybrid probability model algorithm [7] combines the advantages of boundary vector-based and pixel-based algorithms. It first estimates the sky color by finding a rough sky-ground boundary using an objective function. It then constructs a hybrid probability model using the color, gradient, and vertical position of pixels to determine the probability that they belong to the sky. This algorithm is efficient, accurate, and versatile, but may have a longer execution time for non-VGA size images.

D. A Robust Vision-based Skyline Detection Algorithm under Different Weather Conditions

The vision-based algorithm [8] analyzes the difference in image brightness to locate both visible and invisible skylines. It first selects skyline candidate points by using a threshold parameter and an objective function. It then filters out false candidate points by calculating the brightness difference between image blocks. It finally connects the candidate points by comparing their similar characteristics. This algorithm can detect skylines in different weather conditions, such as cloudy or foggy days, but may face challenges with complex scenarios that involve multiple overlapping objects or structures.

E. Single Image Dehazing Based on Fusion of Sky Region Segmentation

The fusion of sky region segmentation algorithm [9] combines sky region segmentation with single image dehazing techniques. It uses a fusion method that combines the Otsu algorithm and Canny edge detection to split the foggy image into two regions: the sky region and the non-sky region. It then optimizes the transmittance of non-sky regions by using fast-guided filtering and enhances the estimation of atmospheric light value. It finally produces a defogged image by using a non-linear overlay image enhancement method. This algorithm can improve image quality and reduce color distortion in foggy images but may be sensitive to variations in image quality and lighting conditions.

III. Proposed Algorithm

The algorithm is designed to detect and segment the sky region and skyline within an image. The process yields three main outcomes: the predicted sky mask, the predicted sky image, and the skyline. The predicted sky mask is a binary image filled with white pixels (non-zero) that represent the sky region and black pixels (zero) representing the non-sky region. The predicted sky image is the segmented sky region on the input image based on the predicted sky mask. The skyline is a binary image that illustrates the boundary between sky and non-sky regions based on the predicted sky mask produced.

To assess the accuracy of the predicted sky mask, a confusion matrix is utilized. The goal of the system is to achieve a prediction accuracy of more than 80%. At the end of the program, it displays the performance metrics for the algorithm's performance and records the execution time.

In this section, we will discuss the system design of the proposed algorithm and the results will be discussed in the following section. Figure x displays the flow of the algorithm.

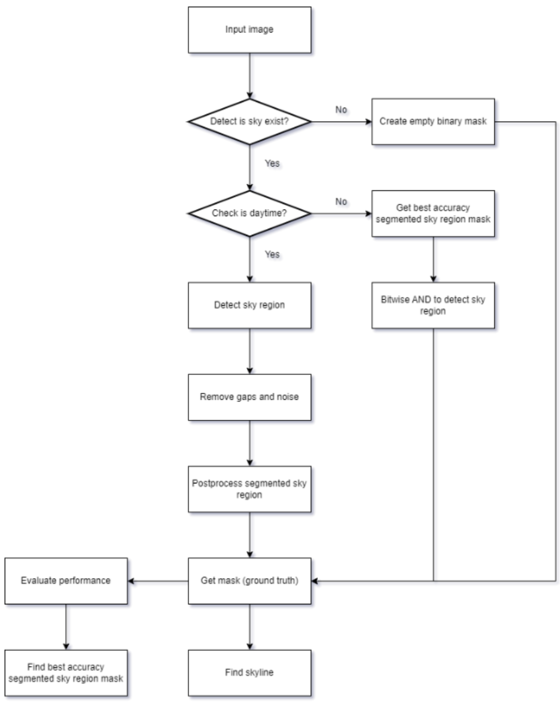

Fig. 1. Flowchart of the proposed algorithm.

A. Create Result Directories and Loop Through Input Image

The program starts by removing existing result directories. Then, it creates new directories for storing predicted masks, predicted sky images, and skyline images for every image group. This structuring is done to manage the output images effectively. Once the result directories are set up, the program iterates through each image group within the ‘dataset’ folder along with the images in each group.

B. Check Sky Existence

In this step, the program determines whether there is a presence of sky in the currently processed image group. This is achieved by analyzing the individual images within the group and identifying the presence of the sky based on color. The input image is converted into the HSV (Hue, Saturation, and Value) color space to examine the blue, white, and gray tones in the image that are common in the sky.

Next, a logical OR operation is performed to calculate the percentage of potential sky pixels in the image by counting the number of non-zero (sky) pixels. A pre-defined threshold value of 0.4 determined through trial and error is applied to differentiate between sky and non-sky images. The threshold value is to be fine-tuned depending on the dataset images for more accurate sky existence detection. The sky percentage is then accumulated. An image is categorized as a sky image when the average sky percentage is greater than the predefined threshold.

C. Check Daytime or Nighttime

After confirming the presence of the sky in the image, the system proceeds to determine whether the current image was captured during the daytime or nighttime. This step is crucial as nighttime images might struggle to accurately detect the sky region due to poor lighting conditions. Consequently, this stage involves calculating the average pixel intensity of the image in grayscale.

A threshold value of 100 is defined to distinguish between images taken during the day and those taken at night. Images with an average intensity exceeding the threshold are classified as daytime images. Conversely, images with an average intensity below the threshold are categorized as nighttime images. This approach allows us to account for varying lighting conditions and enhance the accuracy of subsequent processing steps.

D. Process Daytime Images

If the images within the current group are identified as taken during the daytime, this stage involves further processing. The process includes detecting the sky region within the images by employing Laplacian edge detection to generate a binary gradient mask. Following this, a morphological erosion operation is applied to remove small areas of noise in the image. To enhance the accuracy of detecting the skyline, the function ‘remove_skyline_gaps’ is then used to refine the binary mask. This ensures a more precise focus on identifying the skyline.

Subsequently, the predicted sky mask is generated by setting the largest connected component pixels with non-zero values as white (255) in the refined binary mask to indicate the sky region. Additional morphological operations are applied on the predicted sky mask to fill gaps, remove noise, and expand the sky regions. Additionally, a sky-segmented image is produced using the predicted sky mask and bitwise AND operation. Then, post-processing is conducted on the predicted sky mask to fill up holes in the non-sky region. After successfully detecting the sky region within the image, both the predicted sky mask and the segmented sky image are stored within the designated result directories.

E. Process Nighttime Images

During the handling of nighttime images within the ongoing processing group, the task of identifying the sky region involves utilizing the most accurate predicted sky mask from the previously processed daytime images. As a prior step, the program determined the most optimal predicted sky mask derived from the daytime image processing once all the daytime images in the group had been processed.

For the nighttime images, a bitwise AND operation is performed with the most favorable sky mask associated with the group. This operation aims to detect the sky region effectively. Similar to the daytime process, the predicted sky image is generated by performing a bitwise AND operation between the input image and the predicted sky mask. Subsequently, both the predicted sky mask and the predicted sky image are stored within the assigned output directories.

F. Process Images With No Sky

If the program identifies an absence of sky in the current image group, it generates an empty binary mask filled with zeros that have the same dimensions as the input image to serve as the predicted sky mask. This approach is employed since a ‘no sky’ image does not have any sky region. Subsequently, a bitwise AND operation is executed to obtain the predicted sky image. Both the predicted sky mask and predicted sky image are then stored within the outcome directories.

G. Evaluate Performance

After the images have been processed to identify the sky regions, this phase focuses on evaluating the performance of the predicted sky mask in comparison to a ground truth mask. This evaluation employs a confusion matrix that calculates four key metrics: true positives, true negatives, false positives, and false negatives [10]. These metrics collectively determine the accuracy of the predicted sky mask. The purpose of this evaluation is to assess the quality of the predictions.

TABLE I. CONFUSION MATRIX [10]

		Actual Value	
		Positive	**Negative**
Predicted Value	**Positive**	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

True positives are computed by summing the results of a logical AND operation between the predicted sky mask and the ground truth mask. This signifies the correctly identified sky regions. Conversely, true negatives are determined by summing the results of a similar operation performed on the complement of both the predicted sky mask and the ground truth mask. This indicates the accurate identification of non-sky regions.

False positives are calculated by summing the results of a logical AND operation between the predicted mask and the inverse of the ground truth mask. This indicates regions that have been mistakenly classified as the sky. Similarly, false negatives are derived by summing the outcomes of a logical AND operation between the inverse of the predicted sky mask and the ground truth mask. This indicates regions that have been inaccurately labeled as non-sky when it is part of the sky.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

The accuracy of pixel predictions is then determined using a formula presented above. The result is then multiplied by 100 to express it as a percentage. This process collectively offers a comprehensive assessment of the predicted sky mask's performance.

H. Find Skyline

This step involves detecting the skyline within the predicted sky region. It identifies and visualizes the skyline based on the predicted sky mask. Initially, it detects the contours of the predicted sky region by locating the boundaries of connected regions in the image. Subsequently, an empty image is created to serve as the canvas for drawing the skyline contours. The dimensions of this image are the same as the predicted sky mask. After that, the ground truth mask is applied to the empty skyline image using multiplication. This is performed to ensure that only the region of interest (RoI) will be considered for drawing the skyline. The final skyline image is then generated by outlining the boundary of the sky region in white color (255) with a thickness of 2 pixels. This visual representation provides a clear view of the skyline within the predicted sky region.

IV. Results and Discussions

A. Experimental Setup

The experiments for the proposed algorithm were conducted on a laptop with an AMD Ryzen 5 5500U processor, Integrated Graphics, and 16GB of RAM using Spyder IDE 5.1.5.

B. Dataset

The algorithm uses the SkyFinder dataset [4] which contains various types of images taken from different cameras at different locations. The proposed algorithm focuses on four camera numbers: 623, 684, 9730, and 10917. These cameras capture daytime, nighttime, sky, and non-sky images. In total, there are 13,257 images in the dataset, distributed as follows: 623 dataset (3,790 images), 684 dataset (4,094 images), 9730 dataset (1,805 images), and 10917 dataset (3,568 images). The SkyFinder dataset also provides a mask for each camera number dataset.

C. Qualitative Evaluation

The results undergo qualitative evaluation through careful observation. This observation involved comparing the segmented sky region outputs with the masks of each camera number dataset provided in the SkyFinder dataset. Fig.2 to Fig.8 illustrates the output of the segmented sky region for daytime and nighttime for each dataset with the best accuracy along with the provided mask (ground truth). Fig.9 to Fig.11 displays the output of the segmented sky region for each dataset that has a bad result.

Fig. 2. Best result of daytime sky detection of dataset 623.

Fig. 3. Best result of daytime sky detection of dataset 684.

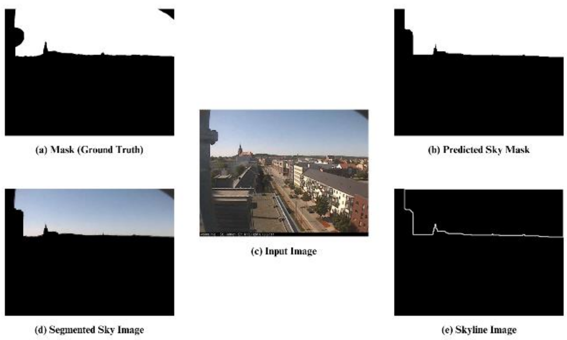

Fig. 4. Best result of daytime sky detection of dataset 9730.

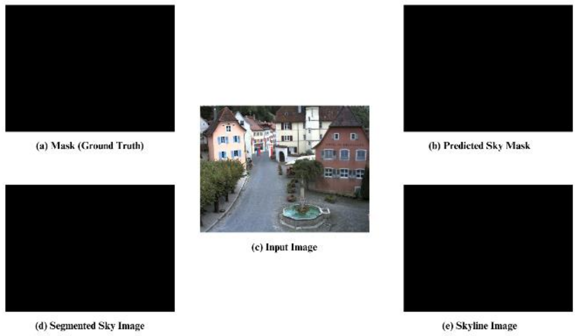

Fig. 5. Best result of daytime sky detection of dataset 10917.

Fig. 6. Best result of nighttime sky detection of dataset 623.

Fig. 7. Best result of nighttime sky detection of dataset 684.

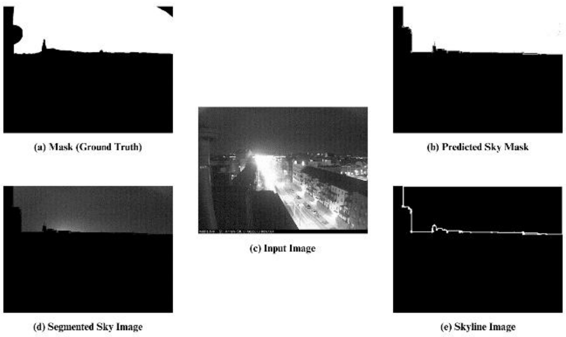

Fig. 8. Best result of nighttime sky detection of dataset 9730.

Based on Fig.2 to Fig. 8 above, we can observe that the output obtained is similar to the provided ground truth mask. The detected sky region is not able to handle cloudy images. It will consider the cloud as the non-sky region. Hence, the proposed algorithm is able to detect the sky region and skyline of it accurately during daytime and nighttime.

Fig. 9. Bad result of daytime sky detection of dataset 623.

Fig. 10. Bad result of daytime sky detection of dataset 684.

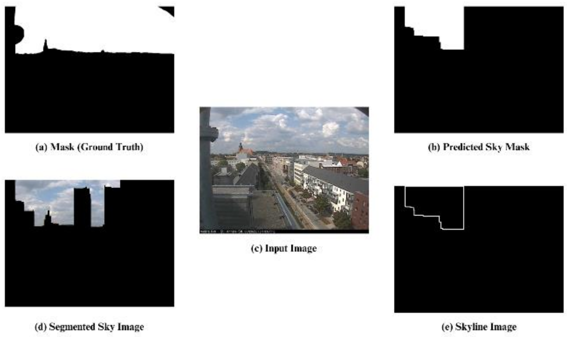

Fig. 11. Bad result of daytime sky detection of dataset 9730.

Based on Fig.9 to Fig. 11 above, we can observe that the output obtained is not favorable. The algorithm cannot handle cloudy images accurately and effectively. Hence, future enhancements need to be done to overcome the problem and achieve higher detection accuracy.

D. Quantitative Evaluation

The algorithm was also evaluated quantitatively using a confusion matrix to determine pixel accuracy. The confusion matrix helped assess how well the sky region predictions match with the ground truth mask. It identified True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) classifications for each pixel. TP indicates the number of correctly identified foreground pixels; TN represents the number of correctly identified background pixels; FP indicates the number of pixels incorrectly classified as foreground; and FN represents the number of pixels incorrectly classified as background.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

The formula above was then used to compute the pixel accuracy act as a quantitative measure of the algorithm's accuracy in image segmentation. The higher the accuracy percentage, the more accurate identification of sky regions by the algorithm.

TABLE II. AVERAGE ACCURACY OF EACH DATASET

Dataset	Number of Images	Average Accuracy (%)
623	3,790	100.00
684	4,094	96.22
9730	1,805	94.38
10917	3,568	95.57
Total	**13,257**	**96.54**

Table II presents the confusion matrix for the algorithm's evaluation. The proposed algorithm achieved an impressive overall accuracy of 96.54% when tested on a total of 13,257 images [4]. This shows that the algorithm is able to detect the sky region effectively since the average detection accuracy for each dataset is more than 90%.

E. Execution Time

Table III displays the program execution time to perform sky detection and segmentation for all the datasets (13,257 images).

TABLE III. EXECUTION TIME

	Execution Time (s)
Total (13, 257 images)	2463.23
Average per image	0.19

From the results obtained, the total execution time of the algorithm to process 13,257 images is 2463.23 seconds which is equivalent to 41.05 minutes. The total average execution time per image is 0.19 seconds. It is clear to see that a large number of datasets will require more time to process an image to detect and segment the sky region.

V. CONCLUSION

In conclusion, the proposed algorithm that employed Laplacian edge detection demonstrates effectiveness in handling both daytime and nighttime images. The sky segmentation algorithm achieves an accuracy rate of 96.54% which is very likely to the ground truth mask. Nevertheless, there are some limitations in the algorithm where it cannot handle images that are cloudy. The daytime outputs of cloudy images are not desirable which can be improved in future work.

ACKNOWLEDGMENT

This work is for the assignment of CSC3014 Computer Vision.

REFERENCES

[1] B. Zhao, X. Li, X. Lu, and Z. Wang, "A CNN-RNN Architecture for Multi-Label Weather Recognition", Journal of Neurocomputing, pp. 47-57, 2018.

[2] N. Laungrunthip, A. McKinnon, C. Churcher, and K. Unsworth, "Edge-based Detection of Sky Regions in Images for Solar Exposure Prediction", in Proceeding of 23rd International Conference on Image and Vision Computing, 2008, DOI: 10.1109/IVCNZ.2008.4762101.

[3] Y. Shen, and Q. Wang, "Sky Region Detection in a Single Image for Autonomous Ground Robot Navigation", International Journal of Advanced Robotic Systems, 2003, DOI: 10.5772/56884.

[4] R. P. Mihail, S. Workman, Z. Bessinger, and N. Jacobs, "Sky Segmentation in the Wild: An Empirical Study", IEEE Winter Conference on Applications of Computer Vision (WACV), 2016, pp. 1-6, DOI: 10.1109/WACV.2016.7477637.

[5] Z. Zhao, Q. Wu, H. Sun, X. Jin, Q. Tian, and X. Sun, "A Novel Sky Region Detection Algorithm Based On Border Points," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, no. 3, pp. 281–290, Mar. 2015, doi: https://doi.org/10.14257/ijsip.2015.8.3.26.

[6] A. Abdelrahman, "Detection and Extraction of Sky Regions in Digital Images based on Color Classification," Jun. 2013. Available: http://www.diva-portal.org/smash/get/diva2:628931/FULLTEXT01.pdf

[7] C. -W. Wang, J. -J. Ding and P. -J. Chen, "An efficient sky detection algorithm based on hybrid probability model," 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Hong Kong, China, 2015, pp. 919-922, doi: 10.1109/APSIPA.2015.7415405.

[8] Y. -J. Liu, C. -C. Chiu and J. -H. Yang, "A Robust Vision-Based Skyline Detection Algorithm Under Different Weather Conditions," in IEEE Access, vol. 5, pp. 22992-23009, 2017, doi: 10.1109/ACCESS.2017.2728826.

[9] S. Li and Q. Zhou, "Single image dehazing based on fusion of sky region segmentation," *Journal of Physics: Conference Series*, vol. 1971, no. 1, p. 012093, Jul. 2021, doi: https://doi.org/10.1088/1742-6596/1971/1/012093.

[10] I. Düntsch and G. Gediga, "Confusion Matrices and Rough Set Data Analysis," *Journal of Physics: Conference Series*, vol. 1229, no. 1, p. 012055, May 2019, doi: https://doi.org/10.1088/1742-6596/1229/1/012055.