Journal of Artificial Intelligence & Machine Learning (JAIML)
Volume 1, Issue 01, January-December 2021, pp. 1-6. Article ID: JAIML_01_01_001
Available online at https://iaeme.com/Home/issue/JAIML?Volume=1&Issue=1
Journal ID: 9128-7499, DOI: https://doi.org/10.17605/OSF.IO/X5EJH

© **IAEME** Publication

OPEN ACCESS

ROBUST IMAGE PROCESSING ALGORITHM TO COUNT SAREES/T-SHIRTS PLACED IN SHELVES OF A RETAIL SHOP

Lakshmi Namratha Vempaty
International Institute of Information Technology Bangalore,
Bangalore, India
https://orcid.org/0009-0005-8426-8577

ABSTRACT

This study presents a robust image processing algorithm for accurately counting sarees/T-shirts placed on retail shop shelves. The proposed algorithm employs k-means clustering for color quantization, a technique that reduces the number of colors in an image. Through iterative refinement of centroids, the algorithm effectively categorizes objects into distinct clusters, achieving accurate counts. This research offers a valuable approach to automate inventory management in retail settings, enhancing efficiency and precision.

Keywords: Image Processing Algorithm, Retail Shop Inventory, Sarees/T-Shirts Counting, Color Quantization, K-Means Clustering

Cite this Article: Lakshmi Namratha Vempaty, Robust Image Processing Algorithm to Count Sarees/T-Shirts Placed in Shelves of A Retail Shop, Journal of Artificial Intelligence & Machine Learning (JAIML), 1(1), 2021, pp. 1-6.
https://iaeme.com/Home/issue/JAIML?Volume=1&Issue=1

1. PROBLEM STATEMENT

Develop a robust Image Processing algorithm to count sarees/T-shirts placed in shelves of a retail shop.

2. PROJECT REQUIREMENTS

1. Count the number of sarees in a given image.

2. Count the number of sarees with in a specified bounding box.

3. PROCEDURE

The first step is to detect the shelves in the given image. So detect the shelves the process followed in color quantization followed by thresholding and contour drawing.

3.1. Gaussian filtering

The idea of Gaussian smoothing is to use this 2-D distribution as a 'pointspread' function, and this is achieved by convolution. Since the image is stored as a collection of discrete pixels we need to produce a discrete approximation to the Gaussian function before we can perform the convolution. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail.

3.2. Color Quantization

Color Quantization is the process of reducing number of colors in an image. Here I have used k-means clustering for color quantization. K means algorithm classifies a given data set through a certain number of clusters (k). The idea behind this algorithm is, it defines k centroids, one for each cluster. These centroids are placed far away from each other. Then each point in the data set is associated to the nearest centroid and a groupage is done. After this, k new centroids are calculated and are named as barycentres of the generated. This loop is repeated till the centroids don't move anymore. At each iteration, centroids change their location and at some point, they stop moving. This produces a separation of the objects into k groups which is called as k means clustering.

3.3. Thresholding

Thresholding: This refers to the point transformation in which the output gray level is either 0 or L depending on whether the input gray level is respectively below or above some constant T.It is used to distinguish between foreground and background. Thresholding can also be useful if the original image is some binary image corrupted with (non-binary) noise.

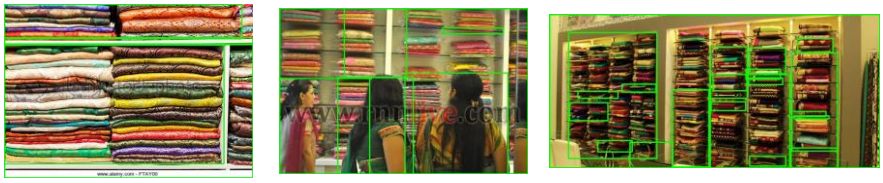
Adaptive Gaussian Thresholding: In thresholding a global value is used as threshold value. But it may not be good in all the conditions where image has different lighting conditions in different areas. In that case, we go for adaptive thresholding. In this, the algorithm threshold for a small regions of the image is calculated. So we get different thresholds for different regions of the same image and it gives us better results for images with varying illumination. In adaptive Gaussian thresholding the threshold value is the weighted sum of neighbourhood values where weights are a gaussian window.

4. RESULTS

The shelves and the number if sarees are detected from all of the given images. The results can be found below:

4.1. Detection of shelves:

In this section we can observe that when given an image as input the shelves where the sarees are placed are found below. Snapshots of the results can be found below:



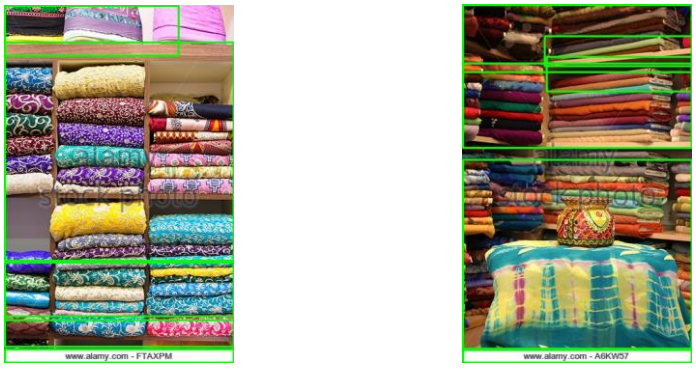
(a) Image-1 (b) Image-2 (c) Image-3

Figure 1: Detection of shelves in all of the images - 1,2 and 3

(a) Image-4 (b) Image-5 (c) Image-6

Figure 2: Detection of shelves in all of the images - 4,5 and 6

(a) Image-7 (b) Image-8 (c) Image-9



(d) Image-10 (e) Image-11

Figure 3: Detection of shelves in all of the images - 10 and 11

4.2. Counting the number of sarees

The count of sarees in a given image can be viewed below. For the given inputs the number of sarees are counted using hysteresis thresholding and contours are drawn around each detected saree.

Hysteresis Thresholding: When we know that pixel values above the threshold U would most likely be a part of an edge. You also know that values below a lower threshold L would have very extremely less probability of being a part of an edge. Then we choose the threshold lower limit to be L and upper limit to be U. This is known as hysteresis thresholding. This method has consistently been able to defeat absolute thresholding. Hence using this both strong and weak edges are preserved and noise and unnecessary edges are removed.

Contour Drawing: After setting up the threshold values using hysteresis thresholding, we will be left with those edges that are of the segmented image using k-means clustering and the unnecessary ones are removed. Now we need to draw rectangles around these objects of the image which are nothing but sarees. Then counting these rectangles would give the count of number of sarees. So after getting the count, I converted it to a string and kept it on the image. These results can be seen below:

(a) Image-1

(b) Image-2

(c) Image-3

(d) Image-4

(e) Image-5

(f) Image-6

(g) Image-7

(h) Image-2

(i) Image-3

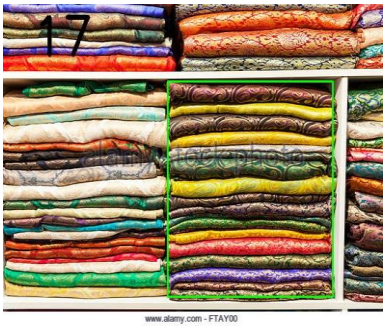
(j) Image-3

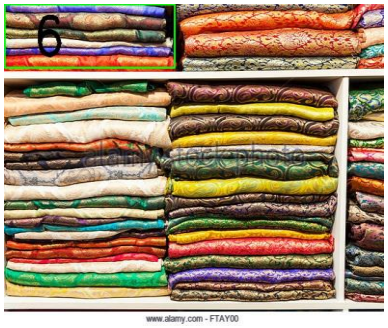
Figure 4: Number of sarees in all the Images

5. COUNT THE NUMBER OF SAREES IN THE WINDOW SPECIFIED BY USER:

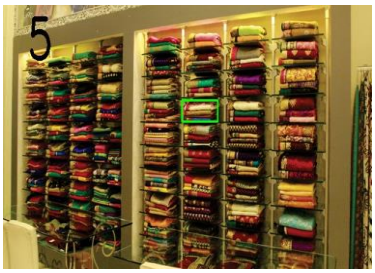
The user will be shown an image. Then using the mouse user needs to select the region of interest. Once the region of interest is selected then the user needs to press 'c' then it is finalized and the number of sarees in that region are displayed. If the user wants to un select the selection then 'r' needs to be clicked. Some samples are shown below:

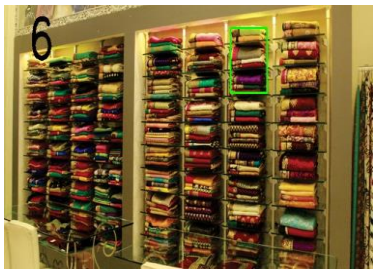
So when we select the region of interest using mouse interactively, that area is shown by drawing a contour around it and then counting the number of sarees. Then that number is put on top of the given image along with the region of interest.



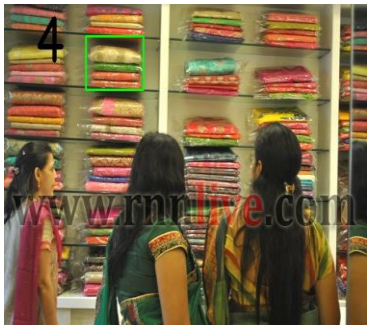


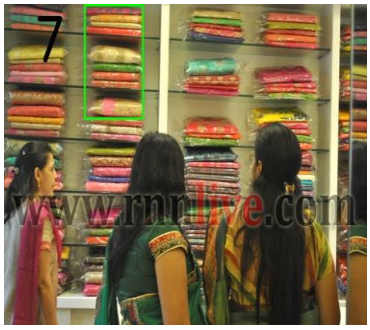
(a) Interactively selected ROI of Image-1 (b) Interactively selected ROI of Image-1



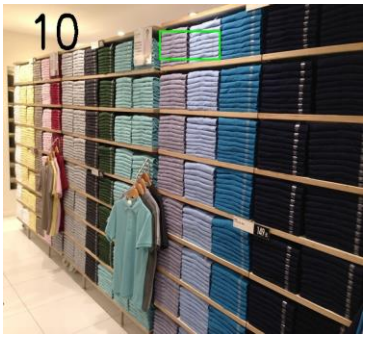


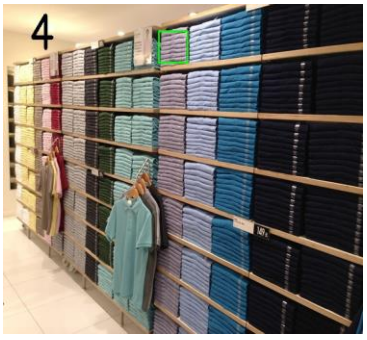
(c) Interactively selected ROI of Image-7 (d) Interactively selected ROI of Image-7





(e) Interactively selected ROI of Image-2 (f) Interactively selected ROI of Image-2





(g) Interactively selected ROI of Image-10(h) Interactively selected ROI of Image-10

Figure 5: Number of sarees in selected region of an image

Citation: Lakshmi Namratha Vempaty, Robust Image Processing Algorithm to Count Sarees/T-Shirts Placed in Shelves of A Retail Shop, Journal of Artificial Intelligence & Machine Learning (JAIML), 1(1), 2021, pp. 1-6.

Article Link:
https://iaeme.com/MasterAdmin/Journal_uploads/JAIML/VOLUME_1_ISSUE_1/JAIML_01_01_001.pdf

Abstract:
https://iaeme.com/Home/article_id/JAIML_01_01_001

Copyright: © 2021 Authors. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

This work is licensed under a **Creative Commons Attribution 4.0 International License (CC BY 4.0)**.

editor@iaeme.com