SKIN-PIXEL DETECTION FROM A GIVEN IMAGE

:==:

ML project report submitted by Group No : 3 for the degree of

B.Tech

in

Computer Science and Engineering

विश्वजीवनामृतं ज्ञानम्

Atal Bihari Vajpayee -

Indian Institute of Information Technology and Management, Gwalior

(An Institute of National Importance)

DOS: June 4, 2020

:==:

Members of Group No : 3

Akashdeep Goel(2017BCS-009)

Chirag Jindal(2017BCS-013)

Manvi Gupta(2017BCS-017)

Sajal Verma(2017BCS-025)

Saumya Gupta(2017BCS-028)

Machine Learning Instructor:

Dr. Sunil Kumar

Contents

List of Figures

1 Introduction

1.1 Abstract

Skin color is the most important and primary element for skin recognition in an image.But the only deciding factor cannot be skin color as skin tone varies according to different races.A different color space must be chosen such as HSV to convert the image so that it does not vary and is not sensitive to the changes in lightning.Then we can apply the algorithm and separate the skin parts in an image.

1.2 Motivation

There have been a lot of changes in the ways humans interact with computers. Modern Human-Computer applications include Speech synthesizers, Hand gesture recognition, Biometric identification etc. In all the above applications, skin color is often used as a cue for detecting, localizing and tracking targets containing skin, like faces and hands in an image. Skin detection is the process to find the pixels or regions containing skin in any video or image. This process is a preprocessing step to find regions that potentially have human faces and limbs in images. Skin detection applications are used for personality recognition, body-parts tracking, gesture analysis and adult content filtering etc.

1.3 Problem Statement

We are given a set of images in which we have to detect the area in which the skin is present by changing the image to different color spaces. We will then apply various types of algorithms on the image to find out the pixels in which skin is present and then select the algorithm which gives us the best results.

1.4 Literature Review

We studied some research papers for better understanding of the problem of skin detection. We found out some color spaces and algorithms which were applied to find out the desired results. Following are some of the research papers and their findings.

1.4.1 Kseniia Nikolskaia, Nadezhda Ezhova, Anton Sinkov, and Maksim Medvedev "Skin Detection Technique Based on HSV Color Model and SLIC Segmentation Method"

1. The paper uses techniques such as HSV color model and SLIC segmentation method.

2. The implemented skin detector converts the image into required color space such as HSV and then uses the image histogram to mark each pixel: whether it belongs to skin. Image pixels are grouped in super-pixels using the SLIC clustering method.

3. The main steps are:

(a) Download the input image.

(b) Convert image to HSV color space.

(c) Generate the image histogram.

(d) Apply classifier to determine the probability of a given pixel being skin colored.

(e) Divide the image into superpixels.

(f) Paint out superpixels where sum of probabilities less than the limit.

4. The Naive Bayes algorithm was chosen as a classifier because this algorithm is the most accurate and a small amount of data is required for it's learning.

5. SLIC algorithm was chosen for image segmentation. It had the fewest errors among those considered in, and also it is the fastest one. It is an adaptation of the k-means method for superpixels.

6. **Advantages:** High processing speed and invariance under rotation and lighting changes.

7. **Issue:** This algorithm is not absolutely accurate. Segments with a color similar to the skin color will be recognized as the skin."

1.4.2 E.Buza, A.Akagic and S.Omanovic, "Skin detection based on Image segmentation with histogram", 2017 10th International Conference on ELECO, Bursa, Turkey

1. **Steps:**

   (a) Removal of Background Pixel using thresholding.

   (b) Converion of RGB image to HSV and YCrCb.

   (c) Clustering is applied to extract the skin pixels based on the features of HSV and CrCb.

2. **HSV:** Hue, Saturation and Value is a color model that is often used in place of the RGB color model. In using this color model, a color is specified then white or black is added to easily make color adjustments.

3. **YCbCr:** Y is the luma component of the color. Represent the light intensity of the color. Cb and Cr is the blue component and red component related to the chroma component respectively.

4. Accuracy obtained on the Face and Family photos data set = 0.81

5. **Conclusion:** The method has low computational cost and is suitable for detecting skin color with reasonable accuracy and is suitable for image pre-processing.

1.4.3 ABM Rezbaul Islam, "Skin detection in image and video founded in clustering and region growing", August 2019, UNIVERSITY OF NORTH TEXAS

1. Clustering based skin detection method is proposed and validated.

2. The number of clusters defined dynamically. This is among the main contributions of this research.

3. Finding number of clusters dynamically is based on the number of objects present in the image.

4. In summary, the proposed algorithm shows promising results but there is room for improvements.

2

Implementation Methodology

2.1 Input Description

We have gathered different images for skin detection from various sources on Internet. Input images are in RGB format. We have selected images of people with different skin color, different gender and different races.

2.2 Mathematical Model

Elbow Method for optimal number of K in K-Means++

Finding an optimal number of clusters in an unsupervised learning algorithm, such as k-means, is an important task. The Elbow Method aids us find the ideal number of k clusters.

In this method the algorithm runs iteratively. The algorithm starts with a number of clusters initialized to one and after every iteration the number of clusters is incremented by 1 and for each value of clusters algorithm calculates the Sum of squared errors (SSE). Now Plot a graph (line chart) between SSE and Number of clusters. From the plot we select the last value of the cluster that had an abrupt decrease in the line chart. This value is the optimal number of clusters. This is called the elbow method because the line chart looks like an elbow.

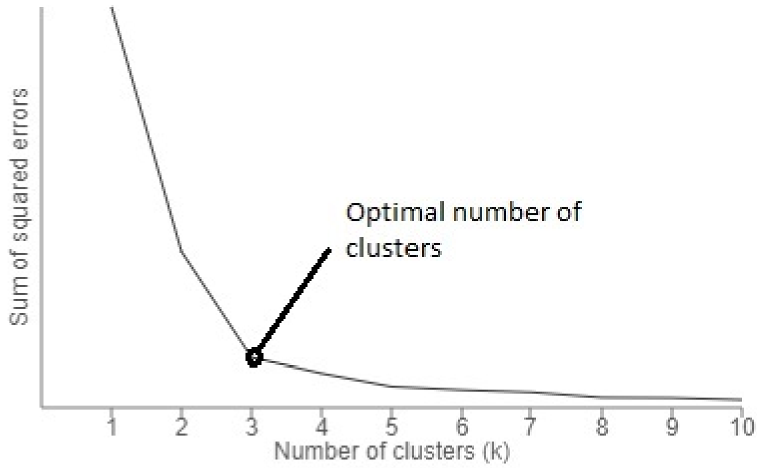


Figure 2.1: Plot of SSE vs Number of clusters

2.3 Machine Learning Model

K-Means++

K-means++ is an unsupervised technique, meaning that the target class is not provided in the set of data. K-means++ is used to partition the whole data into several categories (cluster) based on some patterns (Similarity). We could use techniques like euclidean distance or Manhattan distance to find the similarity between the points. The process of dividing the data-set is known as Clustering. The data points in the same category (cluster) are similar and dissimilar to the data point of other categories.

Steps of K-means++ algorithm:-

1. State the number of clusters K.

2. Initialize the centroids.

   (a) Select any data point randomly from the data.

   (b) From the selected data point, compute its distance from every other data point.

   (c) Now choose the next centroid such that its distance between the previously chosen centroid is maximum.

   (d) Repeat the above steps until we get a K number of centroids.

3. Now assign a data point to that centroid (cluster center) which is more closer than other centroids.

4. Repeat steps 2 and 3 until no data point changes its cluster.

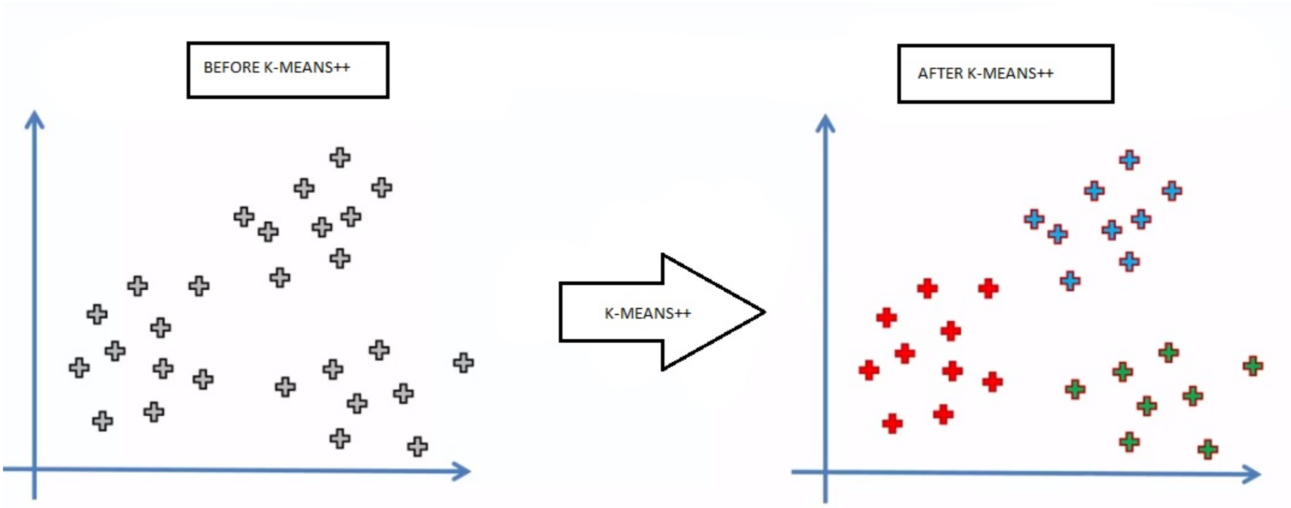


Figure 2.2: Clustering by K-means++

Image Segmentation

Image segmentation is a procedure of partitioning of an image into its constituent parts (also known as segments). The main motive of doing segmentation is for further analysis of each constituent or object in each constituent. There are two approaches of segmenting an image.

1. Discontinuity based.
2. Similarity based.

In discontinuity based approach, the image is segmented on the basis of sudden pixel value change. Point detection, line detection and edge detection are different types of discontinuity based approach. In Similarity based approach the image is divided on the basis of similar pixel values, i.e, we try to group those pixels which have similar pixel values.

2.4 Methodology

Input image is given to the algorithm for skin pixel detection. Input image is read in BGR format. Then BGR format is converted to RGB format and finally RGB format is converted to HSV format. Image is converted to HSV format because it is easier to distinguish colors in HSV format. Elbow method is used to find the optimal K number of clusters. Then K-means++ algorithm is applied to the input image for clustering. K-means algorithm groups the pixels into K clusters. Similar pixel values are grouped together.The cluster which belongs to the skin pixel region is separated and finally, the image is converted from HSV format to RGB format.

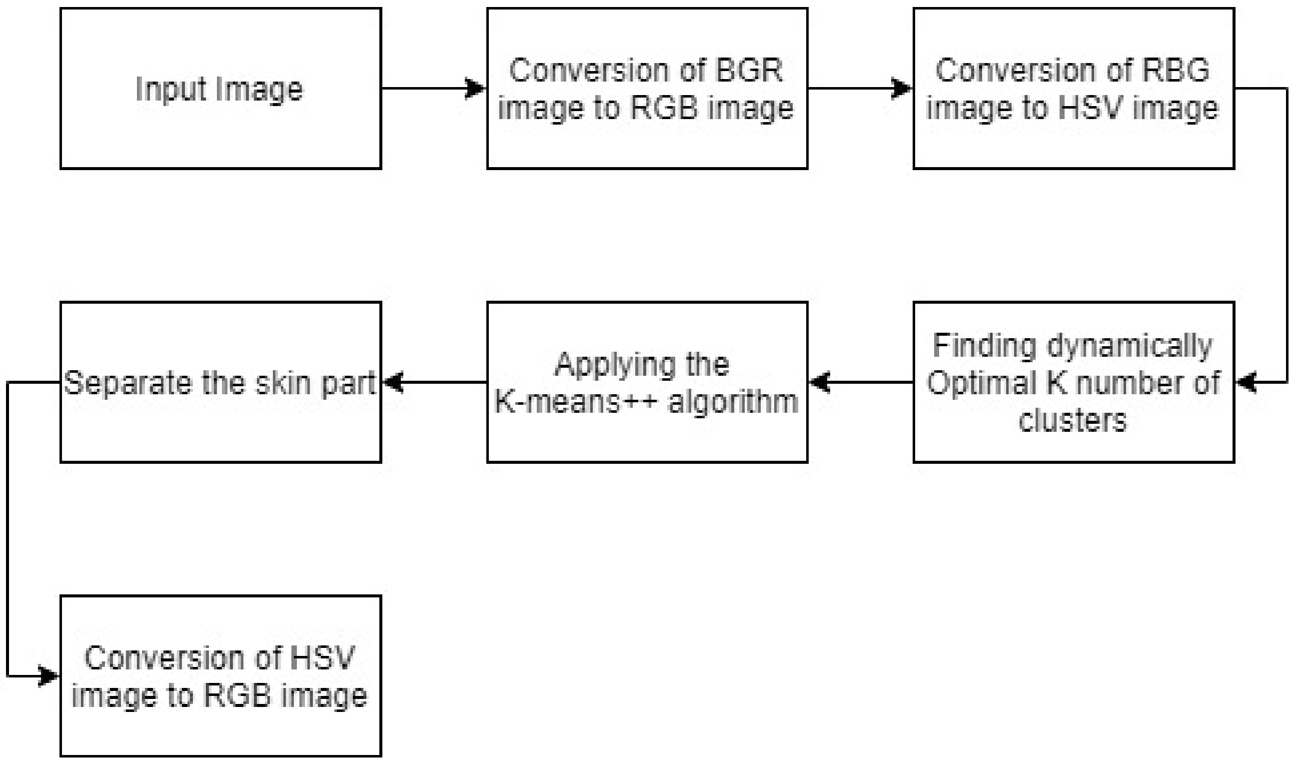


Figure 2.3: Flow Diagram

3

Tools Used

1. **Numpy:** Numpy is a python package used for scientific computing. It is used to process multidimensional arrays, and also used for various numerical operations.

2. **Pandas:** Pandas is a powerful python library which has Dataframe as its main data structure. It is used to manipulate data into rows and columns.

3. **Matplotlib.pylot:** It is a python library used for visualization. It is used for plotting graphs, charts etc.

4. **Scikit-learn:** It is a free python library. It consists of vast number of methods. It supports many machine learning algorithms like SVM, KNN, Decision Tree, K-Means Clustering, etc. It also supports numerical computations like calculating errors, accuracy, matrix, etc.

5. **OpenCV:** It is the Open Source Computer Vision library. It is a library containing programming functions that support real-time computer vision and image processing. It is written in C++. It is developed by Intel later supported by Willow Garage.

4

Code Description and Screenshots

1. The image for which skin pixel detection has to be done is extracted using OpenCV. The image is read in a BGR format, as per the characteristic of CV2.

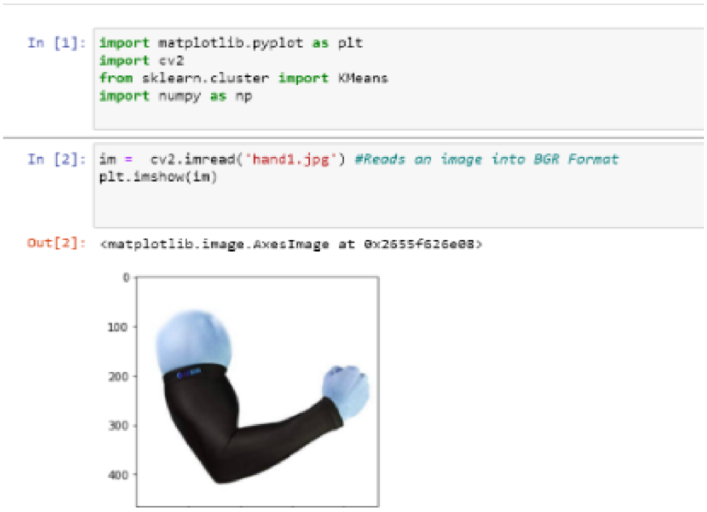


Figure 4.1: Read Image

2. The image is converted into RGB format and then to HSV format using the functionality of OpenCV.

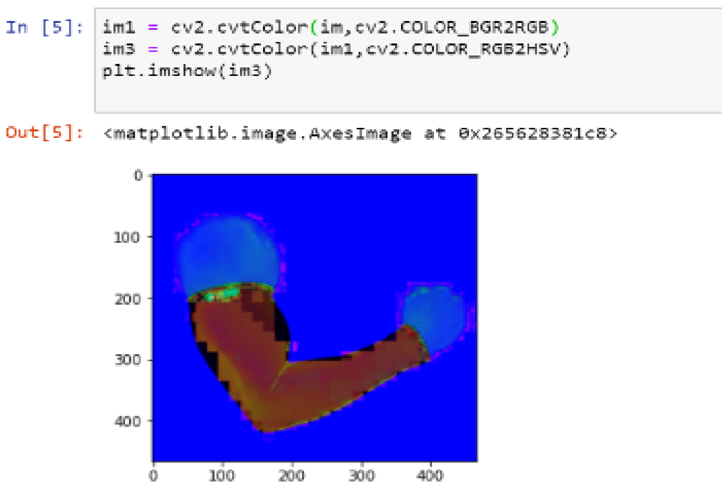


Figure 4.2: Color Inversion

3. The data structure of type dictionary labelled as SSE is developed which stores the number of structure as key and the loss as value. The graph is plotted for visualisation purpose. The elbow is visible in the graph which corresponds to the optimum number of clusters. Also the image is converted into a 2-D data structure from a 3-D data structure, which can be fed in for clustering.

```
In [8]: ## code to determine the no of clusters automatically

        all_pixels  = im3.reshape((-1,3))
        sse ={}

        for k in range (1,10):
            kmeans=KMeans(n_clusters=k,max_iter=1000).fit(all_pixels)
            sse[k]=kmeans.inertia_
        plt.figure()
        plt.plot(list(sse.keys()),list(sse.values()))
        plt.xlabel("number of clusters")
        plt.ylabel("sse")
        plt.show()
```

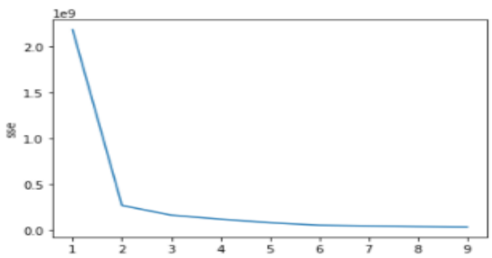


Figure 4.3: Elbow Method

4. The elbow in the graph or the number of optimum clusters is identified using the methodology that whenever the reduction in loss between successive number of clusters is less than 30%.

```
In [11]: ## code to determine the no of clusters automatically
t=sse[1]
no_of_clusters=11
for k,v in sse.items():
    if k==1:
        continue
    else:
        gain=t-v
        x=0.3*t
        t=v
        if gain<x:
            no_of_clusters=k-1
            break
print(no_of_clusters)

3
```

Figure 4.4: Finding K automatically

5. Clustering is implemented using KMeans++ using the number of clusters identified in earlier steps which is optimum for the given image. Also the dimensions of the image is stored in s1 and s2, which will be of use in future computations.

```
In [12]:
dominant_colors = no_of_clusters
print(all_pixels.shape)
km = KMeans(n_clusters=dominant_colors)
km.fit(all_pixels)

centers = km.cluster_centers_
centers = np.array(centers,dtype='uint8')
print(im3.shape)
s1=im3.shape[0]
print(s1)
s2=im3.shape[1]
print(s2)
print(centers.shape)
print(centers)

(217156, 3)
(466, 466, 3)
466
466
(3, 3)
[[  2   0 254]
 [101  46  35]
 [ 14  68 221]]
```

Figure 4.5: KMeans++ Algorithm

6. The pixels of all the corresponding cluster centres are plotted in the

HSV colour scheme.

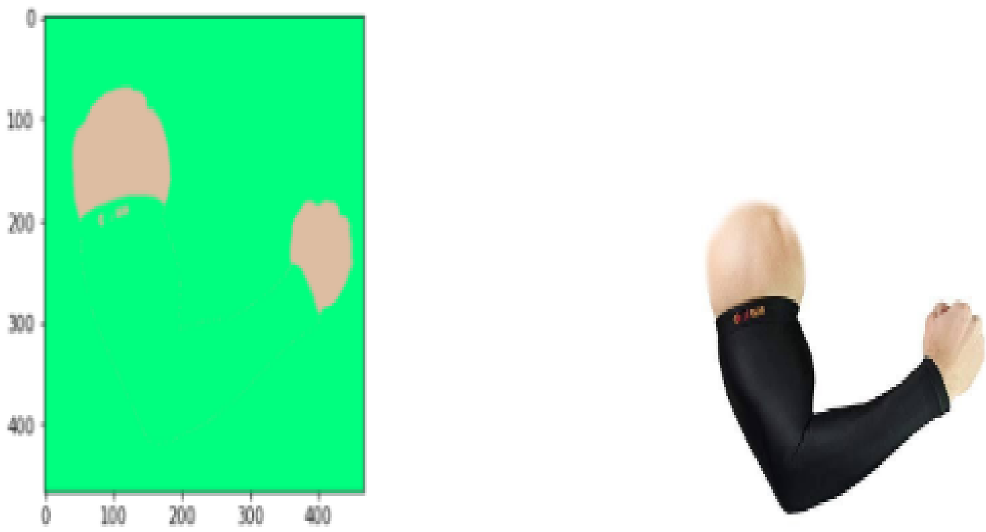
```
In [13]: i = 1
         #plt.figure(0,figsize=(8,2))


         colors = []

         for each_col in centers:
             plt.subplot(1,4,i)
             plt.axis("off")
             i+=1

             colors.append(each_col)

             #Color Swatch
             a = np.zeros((100,100,3),dtype='uint8')
             a[:,:,:] = each_col
             plt.imshow(a)

         plt.show()
```

Figure 4.6: HSV Plot

7. Now as the clustering is applied in the previous steps, the pixel values to the corresponding clusters are plotted in order to determine which pixel belongs to the skin part.

```
In [14]: for i in range (no_of_clusters):

             new_img = np.zeros((s1*s2,3),dtype='uint8')

             print(new_img.shape)
             for ix in range(new_img.shape[0]):
                 new_img[ix] = 255

         # for specific color extraction
             for ix in range(new_img.shape[0]):
                 if km.labels_[ix] == i:
                     new_img[ix] = colors[km.labels_[ix]]

             new_img = new_img.reshape((im.shape))
             im3 = cv2.cvtColor(new_img,cv2.COLOR_HSV2RGB)
             #im3 = cv2.cvtColor(new_img,cv2.COLOR_BGR2RGB)
             plt.imshow(im3)
             plt.show()

         (217156, 3)
```

Figure 4.7: Extracting Skin Part



Note: Green colour represents the background

Figure 4.8: Final Result

5 Results

6

Conclusion and Future Scope

6.1 Conclusion

1. We have used an unsupervised machine learning algorithm, K-means clustering with image processing techniques.

2. The method has low computational cost and is suitable for detecting skin color with reasonable accuracy and is suitable for image pre-processing.

3. Unsupervised methods are free from a training phase and work by automatically selecting the key skin features.

4. Supervised methods (including neural networks) require quality training data to build a model that can perform well on testing data. Yet, availability of all types of skin data for training is unrealistic.

5. The model is capable of intelligently determining the optimum number of clusters to be formed for segmenting the image.

6. The extraction of skin pixels using HSV colour scheme is better than RGB colour scheme.

6.2 Future Scope

1. Skin pixel detection can be taken to a level where it will be helpful in advanced biometric technologies.

2. These techniques can be used for robots or unmanned vehicles in future to detect the presence of human, to track them.

3. Skin pixel detection in future can be useful in medical aspects as it can be used to study different types of skins, skin tones and hence their related skin diseases.

7 References

7.1 References

1. W. R. Tan, C. S. Chan, P. Yogarajah and J. Condell, "A fusion approach for efficient human skin detection", IEEE Transactions on Industrial Informatics, vol. 8, no. 1, pp. 138-147, 2012.

2. Dr.R. Pon Periyasamy, V.Gayathiri, Melanoma Detection through K-Means Segmentation and Feature Extraction, vol. 4, Issue 5 May 2017.

3. ABM Rezbaul Islam, "Skin detection in image and videofounded in clustering and region growing", August 2019.

4. E.Buza, A.Akagic and S.Omanovic, "Skin detection basedon Image segmentation with histogram", 2017.

5. seniia Nikolskaia, Nadezhda Ezhova, Anton Sinkov, andMaksim Medvedev "Skin Detection Technique Based onHSV Color Model and SLIC Segmentation Method"