



# PKT: fast color-based spatial model for human skin detection

Grace L. Samson<sup>1,2</sup>  · Joan Lu<sup>1,2</sup> 

Received: 21 June 2019 / Revised: 3 February 2021 / Accepted: 14 April 2021 /  
Published online: 7 August 2021

© The Author(s) 2021

## Abstract

We present a new detection method for color-based object detection, which can improve the performance of learning procedures in terms of speed, accuracy, and efficiency, using spatial inference, and algorithm. We applied the model to human skin detection from an image; however, the method can also work for other machine learning tasks involving image pixels. We propose (1) an improved RGB/HSL human skin color threshold to tackle darker human skin color detection problem. (2), we also present a new rule-based fast algorithm (packed k-dimensional tree — PKT) that depends on an improved spatial structure for human skin/face detection from colored 2D images. We also implemented a novel packed quad-tree (PQT) to speed up the quad-tree performance in terms of indexing. We compared the proposed system to traditional pixel-by-pixel (PBP)/pixel-wise (PW) operation, and quadtree based procedures. The results show that our proposed spatial structure performs better (with a very low false hit rate, very high precision, and accuracy rate) than most state-of-the-art models.

**Keywords** Image processing · Skin detection · Information retrieval · Spatial data Modelling · Interpolation · Classification · Pattern recognition · Tree data structure · Computer vision

## 1 Introduction

Humanly related tasks like; pornographic image filtering, personal identity, hand detection and tracking, verifications video surveillance, face detection and tracking, image retrieval, human pose modelling, naked people detection, and facial expression analysis, depend largely on skin detection algorithms to perform optimally. Existing systems for human skin pattern

---

✉ Grace L. Samson  
zenhev@gmail.com

Joan Lu  
j.lu@hud.ac.uk

<sup>1</sup> Department of Computer Science, University of Huddersfield, Huddersfield, UK

<sup>2</sup> Department of Computer Science, University of Abuja, Abuja, Nigeria

classification/detection, suffer from certain major setbacks, including, individual pixel operation (otherwise known as pixel-by-pixel (PBP) or pixel-wise (PW) operation), high rate of false hit and poor performance especially in terms of predicting darker complexioned skin. These systems still pose challenging pattern recognition tasks for computer vision; thus, it has attracted a great deal of research in recent years [17, 28, 54]. Skin detection methods utilize color information from conventional color space. However, according to [8], there is a substantial disparity in the accuracy of classifying darker skin colors against their lighter counterparts, therefore, requiring urgent attention of commercial companies in building genuinely fair, transparent, and accountable skin analysis algorithms. Skin detection algorithms suggest the presence of human skin in a digital image. It is an important pre-processing step for techniques like face detection and semantic filtering of web content. According to [2], every color space contains an optimal skin detector scheme such that the performance of all the schemes is the same.

In [36], the basic steps in skin detection include representation of image pixels in color spaces, suitable distribution of skin and non-skin pixels, and skin colour modelling (which uses an underlying skin color distribution characteristic on a colour space to detect skin colour pixels quickly). However, human skin appearance in images is affected by various factors such as illumination, background, camera characteristics, and ethnicity, as such, skin detection using color information can be a challenging task [22, 28]. Numerous techniques exist in the literature for skin detection using color, nonetheless, due to real-world conditions such as illumination and viewing conditions, many of these studies are limited in performance. These techniques according to [28, 53] are prone to false skin detection in most cases, therefore, they are not able to cope with the variety of human skin colors across different ethnic groups. Thus, in this paper, we have proposed a fast algorithm based on an improved, combined (HSL and RGB) color model threshold value, for human skin detection from coloured 2D images using our new packed k-dimensional tree (PKT). The accepted skin colour threshold value was deduced from exhaustive experimentation for toning human skin color. The procedure involves the normalization of the RGB/HSL color channels of several randomly selected colored images. The final standardized RGB/HSL coordinate values, lead to the realization of the adopted skin color threshold. Additional comprehensive channel toning was equally adopted to facilitate enhancement on colour insensitivity due to luminance.

For the pixel-by-pixel (PBP) problem, common structures used for performance enhancement is the quad-tree. However, the idea of repeated deep quad-tree-like tedious partitioning seems to be cumbersome; and in some cases, the quad-trees have been proven to have a poor shape analysis and poor performance on pattern recognition due to their inability to compare two images with different translation or rotation efficiently. So, we present the PKT to overcome most of the challenges in these structures. The PKT algorithm starts by reducing the size of the image, thereby achieving only about 60% of the image size. The data reduction pre-processing technique only aims at increasing the speed of the application. The main purpose of the proposed model is to eradicate the common state-of-the-art PBP/PW approach of pixel classification, common in recognition procedures.

To the best of our knowledge, this is the first time a structure like PKT is developed. The structure shows high prospect, in terms of speed, low rate of false hits, reduced computational cost and complexity, high accuracy and precision rate. Going by to the performance comparison of existing models in [28], our experiments show that the proposed algorithm is characterized by a very high accuracy rate, precision, and efficiency (Table 4).

## 1.1 Description of color spaces (channels)

According to [2, 25, 38], the RGB (Red, Green, and Blue), HSV (Hue, Saturation, and Value), HSL (Hue, Saturation, Lightness) and YCbCr (Luminance, Chrominance) color models are some of the main parameters for identifying and recognizing a skin pixel. In [33], the HS color hexagon was described as what picture windows use in their color picker to display the brightest possible versions of all possible colors, based on their hue and saturation. This justifies our decision to adopt the color model as a choice tool for skin color detection. Additionally, the characterization of color range for skin detection is achieved by manipulating the H channel of the HS color model [13]. From the RGB coordinates of the image, the values for H, S, and L, are derived. The H channel of the HSL is applied to characterize the color range for skin detection. The S channel defines the saturation of the H pigment. The L channel normalizes the shade or saturation of both H and S.

We have used the PKT for the classification, prediction, and recognition of human skin pixel in an image, however, we have shown that the model is robust and versatile and can be useful in many other fields of machine learning procedures and pattern recognition including clustering (for instance, clustering skin pixels on the face as a blob in face recognition, cell/DNA clustering in biology for matching purpose, etc.), design of discovery systems (e.g. gene pattern discovery and identification in bioinformatics, data mining and knowledge discovery, etc.).

The major differences between the current study and existing models in terms of colour are

- a. The adaptability of the hue channel to different ethnic skin colour shades; achieved through significant range normalization between these color categories,
- b. The speed up (Table 5) of segmentation and classification procedures using a spatial model
- c. Most importantly, our model achieved a high precision and accuracy (Table 4).

## 2 Related work

Human skin related recognition and identification technologies according to [49] have proven to work less accurately on darker skin. One reason this may be so according to a study by [8], is that skin type classification systems are overwhelmingly designed to favour lighter-skinned subjects, with an error rate of up to 34.7%, leading to higher overall accuracy rates for identifying men than that for women. Their study established that darker-skinned females are the most misclassified group. Illumination, pose, noise and expressions, are the opposing factors faced during face capture and analysis. According to [53, 55], these factors greatly affect especially, the performance of facial recognition systems. Thus, among all biometric systems, according to the authors, facial recognition has shown the highest false acceptance and rejection rates. Several algorithms are proposed to solve this problem, notwithstanding, they only manage to attain slightly acceptable results characterized by high false positives [28, 54]. This means that human skin related recognition systems need to develop better skin detection algorithms to improve their performances. Besides, great attention has been paid to real-time human detection in applications such as vehicle autonomous driving, video surveillance and human activity understanding. However, despite a large body of work devoted to human detection in the last decades, it is still an open problem [48].

## 2.1 PBP (PW) operations

Most existing skin detection methods [4, 28, 51, 59], depend on building an n-dimensional histogram for pixel classification. In most cases, two histograms are constructed at the start with sample pixels, for training purposes. One histogram for skin-related pixels, the other for non-skin related pixels. Using these two histograms, a classification rule such as Bayes (or any other), is applied to each pixel of the test pixels or image to complete the detection process.

Many state-of-the-art techniques for human skin detection from images, depends wholly on PW operation. However, the efficacy of the PW classification is limited [23]. The main objective of PW skin color detection according to [57] is to build up a decision rule that classifies each pixel as skin or non-skin individually. [23, 31] claimed that the performance of skin detection algorithms has not been high in accuracy due to the high overlapped degree between “skin” and “non-skin” pixels. As a solution, they applied a Bayesian classifier and connected component algorithm to identify individual “true skin” pixels using the first posterior probability threshold. Though this method helps to improve skin classification performance, especially the false positive rate, it goes through the rigorous task of checking all pixels individually, thereby not efficient for speed.

Several state-of-the-art methods for skin detection use single color region approach according to [18]; in contrast to this, they applied genetic algorithms to determining optimal skin color regions from a selected color space, which considers skin color as a union of multiple smaller CbCr color regions rather than the aforementioned single color region counterpart. However, even though they applied an optimization on the CbCr color model, which they used in their work, the image pixels were as other systems tested individually. The work in [30], like our proposed system, starts by reducing the size of the given image and then applies the RGB and YCbCr colour models. However, it eventually ends up with processing individual candidate pixels that are in the range of skin color, for detecting human skin. Like other models presented so far [35] offered a similar method of individual pixel examination using only the HSV color model but applied two different types of noise filters NOGIE (Noise Object Global Image Enhancement) and NOWGIE (Noise Object with Global Image Enhancement) for an improved result.

In [9], an adaptive neuro-fuzzy inference system (ANFIS) for skin/non-skin pixels detection was proposed. [25] presented a new threshold based on a combination of RGB, HSV and YCbCr values for skin/non-skin pixels detection. A modified likelihood ratio, in addition to multi-scale, was used for classification in [40] for PBP skin pigment classification. Likewise, by establishing some correlation rules between the chrominance components PCr and PCb of a pixel P, [17] formulated two equations ( $PCr - PCb \geq IP$  and  $|PCb - PCbs| \leq JP$ ) that must be true before a single pixel (P) can be classified as skin or not. Note:  $IP = \max$  between (PCr, PCb) and  $JP = \max$ Distance between points ((PY, PCb), (PY, PCbs)), where PY = the pixel value of P on the Y components of the YCbCr space, and PCbs is an estimated value of PCb. [56] equally presented a system that uses a pixel-by-pixel operation for pixel classification.

In [23], a spatial based system for skin detection was presented. The system applies a discriminative feature space as a domain for spatial analysis of skin pixels, based on textural features extracted from skin colour probability maps. The texture is extracted in the form of seed, taking advantage of the fact that real skin areas have pixels with a high-skin probability threshold if an image is binarized. A distance function is applied for finding the shortest routes from large blobs of the seed to every pixel. Eventually, pixels that are not close to any of the seed blobs are rejected and then the skin regions are extracted. While this method seems

promising due to its texture-based idea, the method does not involve actual space partitioning, which is typical of hierarchical classification for efficient distance threshold queries, as such, query performance is impaired due to bulky distance calculation between seeds and individual pixels. Again, Pixel-wise (PW) classification was applied to hand detection procedure in [27], to find a human hand on pixel-level from a video. [28, 54] also applied the PW classification method for pixel matching purposes.

## 2.2 Tree-based solutions

Many tree-based systems have been proposed for the improvement of skin prediction procedures. However, as discovered in this study, these systems are still faced with a high rate of false hits. Therefore, the effectiveness of our model will contribute to improving time and computational complexity in learning systems. [14], in addition to Deep Neural Network and Naïve Bayesian models, presented a decision tree-based solution for skin detection that overcomes the challenge of color range thresholding. By calculating the probability of each pixel, their proposed equation is tested by PW technique before a pixel is classified using a skin/no-skin decision tree. Their method performs reasonably. However, it still portrays the shortcomings of most existing systems including inefficiency with time due to stages that are involved in processing individual pixels, and inaccuracy in certain skin type prediction, as stated by the author.

The quad-tree structure in [40] differs greatly from what we have presented here, such that for each  $32 \times 32$  sized neighborhood pixels, if a boundary is detected, the  $32 \times 32$  neighborhood is further divided into four sub-neighborhoods, and the modified likelihood ratio test is performed on each of these sub-neighborhoods. The procedure is repeated recursively until a decision is reached, or the window becomes so small that a significant decision cannot be made. This method might be promising but the idea of repeated deep quadtree-like partitioning seems to be cumbersome. The method proposed in [1] applied the Bayesian Rough Decision Tree (BRDT) classifier to improve the accuracy of human skin detection. Quadtree classified vector quantization (QCVQ) method was used in [11]. This method firstly partitions a quadtree into its usual segmentation and then classified into smooth and high-detail blocks. The authors claim that the scheme yields better retrieval performance compared to the well-known vector quantization (VQ)-based image retrieval methods. However, even though quad-trees are very good on images with large areas of a single color, which eventually become compact, they have been proven to have a poor shape analysis and poor performance on pattern recognition due to their inability to compare two images with different translation or rotation efficiently, especially with an image that has different colors for every pixel. [16, 24, 50, 58] discussed other tree-based structures that are similar to decision trees or quad-trees.

## 2.3 Super-pixels

The Super-pixel (Sp) paradigm, introduced in [39] and presented in [19, 29, 32, 47], is a pre-processing activity that divides an image into adjacent regions/clusters of pixels. In most Sp based procedures, watershed image segmentation process is carried out on the image before creating Sp(s). Constructing Sp(s) requires the application of a pre-defined similarity measure that is based on perceptual features. The steps in creating Sp(s) using the common SLIC algorithm, include:

1. Create initial regions based on  $\phi$  (the parameter that determines the number of Sp(s))
2. Group the regions to cluster pixels using a similarity criterion.
3. Modify the connection between regions.

Thus, based on  $\phi$ , the number of pixels in a Sp would be:

$$Y = \sqrt{N/\phi}$$

Where  $N$  is the number of pixels in the input image,  $\phi$  is the estimated number of super-pixels (sub-regions) constructed, and  $Y$  is the number of pixels in each Sp(s). The above equation is different from the concept of PKT. After constructing the Sp(s), the sums of the probabilities of their (training set images) pixels are computed; these sums are compared to the probability map of an input image and the Sp(s) of the input image with lower sums are returned as the predicted cells/values. The procedure described above is totally different from the procedure of the PKT. Unlike the supervised/semi-supervised construction process (using an initial human marked image segmentation) of the Sp-based systems, PKT is fully automatic and unsupervised. It does not require hand annotated images, training sets, initial estimated regional clusters, or an input parameter (like  $\phi$ ). Rather than find  $Y$ , PKT computes  $\Delta$  as in the equation below (see ALGORITHM 1 for details).

$$\Delta = \lceil ((DL)/\mu)^c \rceil$$

In the above equation,  $\Delta$  is compared to  $\phi$ ,  $DL$  to  $N$ , and  $Y$  to  $\mu$ . However, while  $\phi$  is estimated in Sp(s),  $\Delta$  is automatically computed in PKT based on the number of dimensions of the image or the number of principal attributes of the data table (for non-spatial data). PKT is a *multidimensional* structure and can apply to higher dimensional space or high dimensional datasets. Moreover, the projected number of regions of Sp(s) based on  $\phi$ , can lead to over-segmentation [19], whereas the number of PKT sub-regions  $\Delta$  is optimally computed based on the fixed determinant value ( $\mu = 25$ ). In addition, setting an initial position for Sp regions using constant distance increments, presents the challenge of possibly placing these positions (centres) at an image border and thus, Sp might fail to obtain a good segmentation [29]. This is not the case with PKT. PKT cells (regions) are generated automatically and instantly through a recursive partitioning strategy. Furthermore, only 5 strategic pixels are selected from each PKT sub-region, as against computing the sum of the probabilities of pixels of Sp's sub-regions. Finally, the use of predefined determinants  $\mu$  to automatically detect the optimal number of PKT sub-regions, reduces the time taken to find the skin colour pigments, by eliminating the daunting search for  $\phi$ .

In fact, all the methods presented above are quite different from our proposed system. One thing that is common amongst some of the above methods and approaches is, PBP examination to develop a classification rule. Even with tree-based enhancement, there are still some required adjustments if performance is of interest. For example, the decision tree is only an analytical, decision or visualization support tool and might not be proficient for multidimensional or spatial analysis. The quad-tree as we have mentioned is not balanced and therefore not very efficient during computation. Moreover, an image that has different colours for every pixel, will involve very tedious partitioning, thereby losing effect. Additionally, the four children constraint, and constant partitioning in quad-trees limit the proper utilization of the leaf nodes of the quadtree. Besides, [51] reiterated that in neural network methods, the training

stage may take a long time if the number of training patterns is very large. In addition, even though we have stated that most spatial-based models for skin segmentation use the Bayesian classifier (BC), BC methods alone according to [28] are not efficient because they do not have the capability of detecting skin pixels without false alarms. Thus, to overcome the above limitations and challenges, including the complications associated with quadtree-like structures and to reduce the use of PBP/PW methodologies, we present the PKT and PQT.

### 3 Proposed system (PKT)

Our proposed tree structure is an integration of the methods described in [42, 43, 45, 46]. In these materials, various related spatial indexing and modelling mechanisms including a description of improvement strategies for spatial structures and techniques were presented. Nevertheless, this current work is an aggregate classic study that produces an efficient technique for spatial modelling, which we have applied in this paper to predicting human skin related pixels.

The basic concept/idea behind our technique is the fact that skin pixels are hardly isolated. That is, once a skin pigment is encountered at a certain position/location, there is a high probability that the neighbouring pixels are equally skin. As such, selecting a tiny fraction of the pixels, in that bounded area, will most probably guarantee the satisfaction of the skin/no-skin classification condition. Therefore, we build an effective  $k$ -dimensional tree structure, for partitioning and indexing the pixels in an input image into sub-groups. After the partitioning, we interpolate (by inverse distance weighting - IDW) through the leaf nodes using only a very few sample pixels ( $r$  out of  $\Omega_0$ , where  $\Omega_0$  is the total pixels in the leaf node) from the leaf node. Finally, only the leaf nodes where all  $r$  pixels satisfy the skin/no-skin criteria (i.e., Eq. 6 evaluates to T for those pixels) are returned.

Notice the high rate of false hit on the other methods with combined colours in Figs. 7b and 7c. Most parts of the skin area in some images were not captured. In some other images, non-skin areas were captured as skin. Our suggestion in Fig. 8 shows an improvement to these problems and this improvement contributes to the high precision and accuracy of our spatial model.

#### 3.1 Method description (spatial modelling)

The segmentation procedure in Section 3.4, for predicting skin and non-skin pixels in an input image based on the defined color threshold, is normally performed (in the most state-of-the-art systems) by testing individual pixels (PBP or PW operations). However, this process is very slow with a significant degree of false hits (Table 2), but it can speed up and perform better if enhanced by a  $k$ -dimensional data structure like PKT that is efficient for detecting patterns from colored  $k$ -dimensional images.

The model largely depends on an arbitrary value  $\mu$ , which determines its performance. With an exhaustive experiment,  $\mu=25$  was established as the most fitting value for any image type and size, although this depends on the underlying task. For dimension  $k=2$ , and an image of size = DL, given that  $\Delta = ((DL)/\mu)^{1/k}$ , the number of pixels to be processed reduces from DL to  $\delta$ , where  $\delta = r \times \Delta \times \Delta$ ,  $r = 5/\Omega_0$ ,  $\Delta =$  numbers of partitions in each dimension, and  $\Omega_0$  is the number of data elements

(pixels) on each leaf node ( $Ln$ ) of the proposed tree. This means that; for an image of size 3000 pixels, rather than processing the entire pixel elements, only 605-pixels are processed yet an efficient outcome is achieved.

Algorithm overview: General Pseudocode for the PKT algorithm

**The proposed algorithm:**

$k = 2$ , the number of dimensions

Input: The Image ( $I$ )

Output:  $\beta$  (array of skin pixels)

Methods:

1. Extract pixel locations ( $\partial_j$ ) from the image as in Fig. 1(a)
2. Store  $\partial_j$  in an Array ( $A$ ), as in Eq. 1, to obtain the vectorized form of ( $\partial_j$ ) as in Fig. 1(b)  
Where  $j = 1, 2, \dots, N \times N$ , for an  $N \times N$  image
3. Step 2 yields  $A'$  (an array of the xy pixel locations ( $\partial_j$ ) of  $I$ ) as in Eq. 2, this then give rise to the row/column vector  $A''$  (which stores only ( $\partial_j$ ) without xy tuple) as in Eq. 3.
4. Construct PKT (ALGORITHM 1) by Partitioning  $A''$  into  $\Delta$  sub-section/sub-regions (sorting in all dimensions).
5. After partitioning on the last dimension, every sub region, becomes a PKT leave node ( $Ln$ ), containing  $\Delta$  numbers of  $\partial_j$ , where each  $\partial_j$  stores the xy coordinates of the individual pixel.
6. Unwrap the tree (PKT) by recursively, extracting the  $Ln$  in a bottom-up approach.
7. For each  $Ln$  in STEP 6, compute R (ALGORITHM 2), an array containing r significant pixels for interpolating unknown skin pixels.
8. Use R in STEP 7 to determine  $Ln$  that qualifies as a skin cell by testing r elements against our defined skin color threshold in Table 1, store qualified  $Ln$  in  $\beta$  as in ALGORITHM 3.
9. Return  $\beta$  (A list of skin pixels).

### 3.1.1 Spatial analysis

**Elucidation** Given a set of P points/pixels in a 2-dimensional ( $\mathbb{R}^2$ ) space of an  $N \times N$  image (raster), each point  $p_j$ ,  $j = 1, 2, \dots, P$ ,  $p_j$  occupies a single location ( $\partial_j$ ) as shown in Fig. 1a — the grid of P pixels.

Thus, in  $k = 2$  dimension, we define a spatial operation for the image of the form:

$$A = f(\partial_1, \partial_2, \dots, \partial_{N \times N}) \quad \text{for } \partial_j \in \mathbb{R}^k, \quad j = 1, 2, \dots, N \times N \quad (1)$$

Eq. 1 depicts a region of local spatial features ( $A$ ) for the input image, that is to say,  $A$  is a function of a  $k$ -dimensional region of dispersed elements ( $\partial$ ). This means that  $\partial_j$  is the location of a group of features for the  $j$ th pixel. In this case,  $\partial_j$  is the X, Y position/location of pixel  $p_j$  ( $p_jx, p_jy$ ), on the  $N \times N$  image raster. Therefore,  $A$  can be represented as an  $N \times N$  matrix as in Eq. 2.

$$A' = [\partial_1(x_1, y_1), \partial_2(x_2, y_2) \dots \partial_{N \times N}(x_{N \times N}, y_{N \times N})] \quad (2)$$

Thus, in Eq. 3, we create a row/column vector representation of the image as in Fig. 1b, without explicitly reflecting the xy tuple.





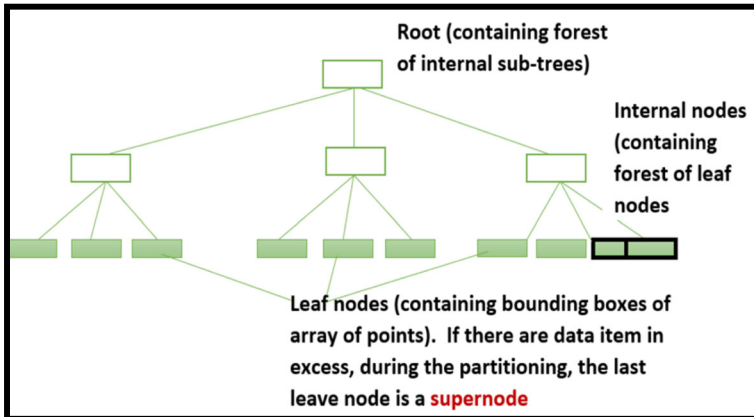


Fig. 2 Typical PKT in two (2) dimensions

in ALGORITHM 1. The partitioning is simply an array sub-division procedure for spatial data, where the dimension of the array is determined by the underlying space that is holding the image. For non-spatial datasets, the computation for  $\Delta$  in ALGORITHM 1 will vary, such that  $\mu$  might be the number of attributes/columns, or it might be the number of attributes divided by an undetermined arbitrary value (to avoid only one item per leaf node). Everything else remains the same. For now, we are concentrating on spatial datasets.

The tree does not go deeper than a maximum depth of two (2), for 2 dimensional, and depth of 3, for 3-dimensional spaces, respectively. This means that the partitioning will always end up with an axis-aligned bounding box of the leaf. Fig. 2 shows the root node as a forest of internal subtrees; each internal node is equally a forest of sub-nodes. These sub-nodes could be leaf nodes ( $L_n$ ) containing the image pixels if the partitioning procedure has reached the last dimension. Note: if the size of the pixel array ( $DL$ ) =  $P = N \times N$  is not even, the last leaf node will be extended to a super-node. The idea of a super-node does not affect the performance of the PKT, because in terms of pixel classification, the mid pixel in the bounding box, is assumed to carry the most classification weight (Eq. 10).

For static datasets, the tree employs a recursive top-down partitioning strategy, such that, partitioning only takes place if the number of elements in the dataset ( $DL$ )  $\geq$  a certain value ( $\Delta$ ). All sub nodes are stored in the root as internal node and then further partitioning of internal nodes occurs only if the number of elements in partition  $j$  ( $\Delta_{\text{subscript}>j}$ ) is greater than the value of  $\Delta$  (ALGORITHM 1).

For a dynamic dataset, the above procedure will start from the leaf, in the sense that a total of  $\Omega_0$  number of pixels are recursively stored on each leaf, and a total of  $\Delta$  leaf nodes are stored on the upper (internal) node until the root is reached.

The output of the tree after the partitioning is shown in Fig. 5 with different values of  $\mu$  (see Fig. 5e). The  $\mu$ , is the determinant of tree behavior and performance.  $\mu$

stands for the expected number of items in the leave node ( $Ln$ ), increasing the value means more items in the leave node, while decreasing the value means otherwise. Any of these actions will affect the tree significantly. After several repeated, exhaustive experimentation, we have chosen 25 as the most efficient value for  $\mu$  that behaves perfectly for all image sizes.

$$\Rightarrow \Omega_0 \cong \mu$$

---

ALGORITHM 1: Pseudocode of the partitioning algorithm (Constructing PKT)

---

```

Input:  $A''$  row/column vector (pixel array of length DL) derived in Eq. 3
1. Procedure Partition ( $A''$ )
2.  $e := 1/k1$ ;
3.  $Z := []$ 
4.  $\Delta := \lceil ((DL)/\mu)^e \rceil$ 
5.  $\mu$ : an arbitrary number (here, we choose 25)
6.  $\Delta$ : the value of  $\Delta$  determines the number of partitions on each dimension of tree
7.  $k1$ : is the number of dimensions (2 in this case)
8.  $k := 0$ 
9. While  $k \leq k1$ 
10. If  $DL \leq \Delta$ 
11. Stop /* terminate partitioning,  $A''$  is leave node*/
12. Else
13. a. Slice  $A''$  into  $\Delta$  sections
14. /* where  $\forall_k := DL/\Delta$ 
15.  $A''_{jk} :=$  new array of  $\forall_k$  pixel elements in partition  $j$  of the  $k$  dimension
16. for  $j = 1, 2, \dots, \Delta$ 
17. */
18. b. Add array  $A''_{jk}$  to  $Z$ 
19.  $Z$  /*the root node containing all nodes (partitions) when  $k := 0$ , dimension = 1*/
20. End If
21.  $Z \leftarrow Z_{jk}$ 
22. /*  $Z_k$  (is an internal node consisting of  $\Delta$  nodes.  $Z_k$  ( $k=2$ ) is PKT leaf nodes*/
23. /* Each leaf node is denoted as  $Z_{jk}$ */
24.  $DL \leftarrow \forall_k$ 
25.  $A'' \leftarrow A''_{jk}$ 
26.  $k \leftarrow k + 1$ 
27. End While
    
```

Output  $Z$ :  $k$ -dimensional forest of PKT nodes, where each dimension consists of  $\Delta$  partitions.

Note: This procedure forces the tree to stop at depth  $k=2$  (for 2 dimensions), thereby storing the data items on the leaf nodes of depth 2.

---

In ALGORITHM 1, we perform (by sorting  $A''$  on all the dimensions) the partitioning of  $A''$  (a row/column vector of the image pixels, with length DL, derived in Eq. 3), into predetermined sub-regions using the proposed partitioning strategy. After the partitioning, each array  $A''_{jk}$  containing a total of  $\forall_k$  elements, is further divided by  $\Delta$  until the last dimension is reached, and a certain condition is met. Ultimately, each cell in the grid (with  $\Omega_0$  elements, bounded by an axis aligned bounding box denoted as  $(Z_{jk=k1})$  forms the leaf node ( $Ln$ ) of the

PKT and a group of  $Ln$  forms an internal node  $Z_k$ .

---

ALGORITHM 1(a): Pseudocode Constructing PQT

---

Input:  $A''$  row/column vector (pixel array of length DL) derived in Eq. 3

1. Procedure Partition ( $A''$ )
2. Compute  $\Delta$  as in algorithm 1 /\*  $\Delta$  determines the number of children in a partition/leave node \*/
3. Let depth := 0
4. If  $DL \leq \Delta$ 
  - Stop /\* terminate partitioning\*/
5. Else
6.  $A \leftarrow$  new Bounding box ( $A''$ )
7.  $B \leftarrow$  centre point ( $A$ )
8.  $B_x \leftarrow$  x co-ordinate of  $B$
9.  $B_y \leftarrow$  y co-ordinate of  $B$
10.  $A1 \leftarrow$  array of  $A''_{jx} \leq B_x \ \&\& \ A''_{jy} < B_y$
11.  $A2 \leftarrow$  array of  $A''_{jx} \geq B_x \ \&\& \ A''_{jy} < B_y$
12.  $A3 \leftarrow$  array of  $A''_{jx} < B_x \ \&\& \ A''_{jy} > B_y$
13.  $A4 \leftarrow$  array of  $A''_{jx} > B_x \ \&\& \ A''_{jy} > B_y$
14.  $Z \leftarrow [A1, A2, A3, A4]$  /\*  $Z$  is the root node with four internal nodes\*/
15. End If
16. If  $\text{len}(Z) > 0$
17.  $A'' \leftarrow Z_j$
18.  $A''_j \leftarrow Z_{ji}, i=1, 2, \dots, \Delta, j=1, 2, \dots, 4$
19.  $DL \leftarrow \text{len}(\text{array } Z_j), j=1, 2, \dots, 4$
20.  $Z \leftarrow Z_{j\text{depth}+1}$  /\*  $Z_{j\text{depth}+1}$  is an internal node\*/
21. Repeat
22. Else
23. Return  $Z$
24. END If

Output  $Z$ : forest of  $PQT$  nodes, where each branch consists of  $\Delta$  partitions.

---

ALGORITHM 1. (a) describes the construction of the PQT (Fig. 14c), which is an improved quad-tree. Similar to the conventional quad tree, PQT continues partitioning until a certain condition is met. However, with PQT, the value of the partitioning parameter is pre-set by automatically computing  $\Delta$ .

After extracting the pixels of an input image into array  $A''$ , partitioning begins if the length of  $A''$  is greater than  $\Delta$ , this means that partitioning will terminate if:

$$A'' \div 4 \leq \Delta$$

The depth of the PQT is given as:

$$\text{dep} \cong \left\lceil \frac{\log_2(DL/\Delta)}{2} \right\rceil$$

and the number of children in each of the leave node of PQT is approximately:

$$\Omega_0 \cong^{DL} /_{4^{dep}}$$

The PQT is a highly balanced tree as all the leave nodes will always reside on the same level/depth.

Note, with the above equations, PQT will easily adapt to various tasks as the parameters that determines the structure can easily be adjusted to suit the task at hand. The depth of the tree and the number of children to reside in each leave node can be established a priori.

Next, we find R (ALGORITHM 2). R is an array of qualifying pixels selected by spatial interpolation from  $\Omega_0$  elements using the inverse distance weighting (IDW) function in Section 3.3. In this program, only the elements in R will be processed during the classification stage.

**ALGORITHM 2: Pseudocode for the finding R (array of significant pixels for skin prediction)**

Input:  $\mathbb{Z}$ : forest of nodes (from ALGORITHM 1)

1. Procedure find (R)
2.  $\Omega_0$  = total pixels in leaf node ( $L_n$ , bounded by  $\mathbb{Z}_{jk=k1}$ )
2. R = [] empty list to store significant pixels from each leaf nodes
3. For each  $\mathbb{Z}_{jk=k1}, j = 1, 2, \dots, \Delta * \Delta$ , (with length  $\Omega_0$ )
  - a. Sort elements in array  $\mathbb{Z}_{jk=k1}$
  - b. Using the IDW function described in section 3.3
    - While the length of R  $\leq 5$ 
      - Select  $r_{pixel} / * r_{pixel}$  is the 5 significant pixels out of  $\Omega_0$  pixel (Section 3.2.1) \*
      - Add each  $r_{pixel}$  to R

Output: R

An array with 5 elements signifying relevant pixels from each  $\mathbb{Z}_{jk=k1}$ , for the prediction procedure.

Note that the length of R is constant for any size of A''.

### 3.2.1 Selecting r pixels

r pixels are selected from each leaf node ( $\mathbb{Z}_{jk=k1}$ ) of  $\mathbb{C}$  to form a single R (ALGORITHM 2), such that they match the pixels depicted in Fig. 4c and 4d, where:

- $r_1$  pixel at (top left of the bounding box of  $\mathbb{Z}_{jk=k1}$ ).
- $r_2$  pixel at (top right of the bounding box of  $\mathbb{Z}_{jk=k1}$ ),
- $r_3$  pixel at (bottom left of bounding box  $\mathbb{Z}_{jk=k1}$ )
- $r_4$  pixel at (bottom right of the bounding box of  $\mathbb{Z}_{jk=k1}$ ),
- $r_5$  pixel at (middle of the bounding box of  $\mathbb{Z}_{jk=k1}$ )

After computing R, next we find the value of  $\delta$  (the total expected significant pixels in an image for efficient skin/no-skin classification).

It has been established that  $\Omega_0$  = total pixels in each leaf node ( $\mathbb{Z}_{jk=k1}$ ), r = size of array  $R \rightarrow 5$  (selected significant pixel in a cell), DL is the size of the input array of image pixels. Thus, in 2-dimension, the total pixels in the image to be processed should normally be:

$$DL = \Omega_0 \times \Delta \times \Delta \tag{4}$$

However, since we have selected only 5 pixels/elements out of  $\Omega_0$ , the new total pixels in the image to be processed would be

$$\delta = r \times \Delta \times \Delta \tag{5}$$

It is evident from Eqs. 4 and Eq. 5, that the total number of potential candidate pixels ( $\delta$ ) is lesser than DL.

Note:  $\delta$  in Eq. 5 can vary greatly, depending on the size of  $A''$ . However,  $r$  and  $\Omega_0$  (typically  $\cong 24$  Eq. 8) are constant.  $r$  is not arbitrary, as we have chosen only 5 strategically positioned pixels from each cell/leaf node). Note also that the number of occurrences of  $\Delta$  in ALGORITHM 1, will directly depend on the number of dimensions of the space holding the input image or number of principal components, for non-spatial datasets. Therefore, for a  $k$  dimensional space, we would have

$$\Delta \times \Delta \times \dots \times \Delta_k \text{ as the total number of leaf nodes in PKT.}$$

Below, we show that Eq. 5 will greatly reduce the size of the computation in terms of predicting pixels that correspond to human skin. i.e., for our example image of size DL (length of  $A''$ ), where  $DL = 3000$  in 2-dimensions, we will have:

$$\begin{aligned} \Delta &= \left[ (3000/25)^{1/2} = 11 \right] \\ \Rightarrow \delta &= 5 \times 11 \times 11 = 605 \\ \therefore \delta (605) &< DL (3000) \end{aligned}$$

The above enhancement indicates that rather than matching all 3000 available pixels in  $A''$ , against the skin color threshold in Table 1 to find human skin, only compare 605 pixels are compared, reducing the number of computations to only 20% of the actual total.

### 3.3 Interpolation process

For pixel matching purposes, the skin segmentation program should normally test a total of DL elements/pixels (Eq. 4). However, this will make the segmentation/classification process highly inefficient and impractical. Therefore, we find  $R$  (from ALGORITHM 2) to reduce the computation from DL to  $\delta$  (Eq. 5). Hence, the interpolation procedure below is adopted to classify the pixels into skin or non-skin pixels, using only a total of  $\delta$  pixels.

Consider the  $P$  pixels and the locations in  $A''$  as a set of  $S$  spatial events  $\{p_1, p_2 \dots\}$ , as illustrated in Eq. 3. Let  $H$  be some color threshold as we defined in Table 1, for classifying  $p_j$ ,

**Table 1** Skin color threshold comparison

Others	Our model
HSV: $(0.0 \leq H \leq 50.0 \ \&\& \ 0.23 \leq S \leq 0.68)$ .	HSL: $H \geq 10.0 \ \&\& \ H \leq 30.0 \ \&\& \ S \geq 0.20 \ \&\& \ L1 \geq 0.10$
RGB: $(R > 95 \ \&\& \ G > 40 \ \&\& \ B > 20 \ \&\& \ R > G \ \&\& \ R > B \ \&\& \  R - G  > 15 \ \&\& \ A > 15)$ .	RGB: $(R > 50 \ \&\& \ G < 220 \ \&\& \ B! = 60 \ \&\& \ B < 200 \ \&\& \ R > G \ \&\& \ R > B \ \&\& \  (R - G)  > 20)$

as skin/no-skin i.e.:

$$H_b(p) \Rightarrow T \parallel H_b(p) \Rightarrow F, \forall p_j, j = 1, \dots, DL \tag{6}$$

If the function evaluates to T, it implies that  $p_j$  is skin and non-skin otherwise.

**Proposition** If  $H_b$  in Eq. 6 is accurate, only  $\delta$  pixels instead of DL is significant for the prediction in Eq. 10 and Eq. 11 to hold, with an accurate, precise, and fast result.

**Proof 1** Let the instances of S be each  $p_j$  located on a single location ( $\partial_j$ ) on the grid of P pixels. This means that for the array of pixels ( $A'$ ), since there is a discrete hybrid partition (i.e., PKT) of  $A'$  based on spatial proximity between locations,  $\partial_j, j = 1, 2, \dots, DL$ , a spatial inferential rule (as in Eq. 7) can be discovered faster and more efficiently.

Following Proof 1, we can now say that with  $\mu = 25, k = 2$ , the total  $p_j$  (for any subset  $p_j \subseteq P$ ), in each leaf node of PKT (ALGORITHM 1) is:

$$C \cong \lfloor DL / \Delta / \Delta \rfloor$$

$$\text{Where } \Delta \cong \left\lceil \left( \frac{DL}{\mu} \right)^{1/k} \right\rceil \text{ (ALGORITHM 1)} \tag{7}$$

Eq. 7 means that following the analysis in Proof 1, for any object or image, in k-dimensional space, there will always be only approximately C (computed in Eq. 8) data elements (pixels in the case of images) in the leaf node of PKT.

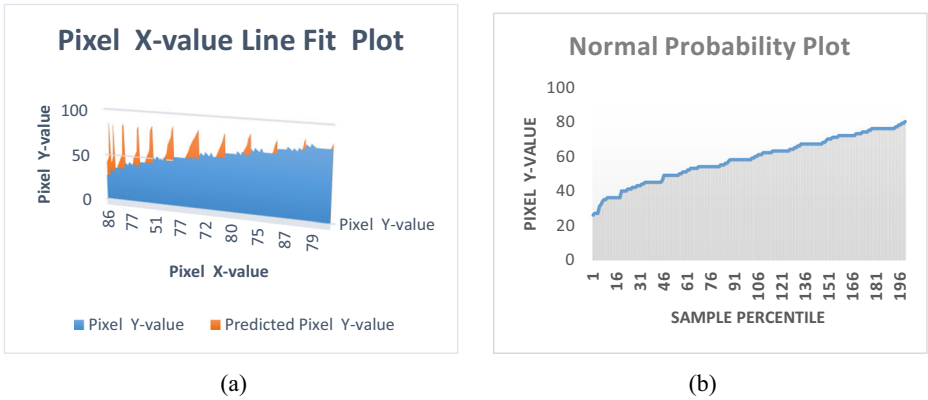
For example, let  $DL = 3000, k = 2$

$$\begin{aligned} &\Rightarrow \text{ceil} \left( \frac{3000}{25} \right)^{1/2} = 11 \\ &\Rightarrow \text{floor} (3000/11/11) \cong 24 \\ \square C = \Omega 0 \cong 24 &\text{ (number of pixels/points in leave node for all input sizes)} \end{aligned} \tag{8}$$

Eq. 8 is affected significantly by the value of  $\mu$ . Notice that if  $\mu$  is 100, Eq. 8 will become  $\cong 102$ , this means there are approximately 102 items in the leave node. Having such huge number of elements in the leave node might lead to an inefficient model due to oversized sub-region. Similarly, having less than 24 elements in the leave node might equally lead to inefficiency.

**Proof 2** Spatial autocorrelation measures the similarity between samples of a given population, as a function of spatial distance [6, 7, 44]. Figure 3 is a plot showing the relationship between sample skin pixels from the input image (Fig. 5). Fig. 3a shows how these pixels ( $p_j$ ), are highly correlated with a correlation coefficient  $r = 0.9181$ . The figure also shows that the pixels are densely clustered. Figure 3b shows the degree of normality in pixel distribution, within the sample image.

Thus, with the assumption that the skin pixels are highly correlated and are located within near-zero proximity with their neighbours, we define  $q$  –the measure of the spatial distance (SP) between locations  $\partial_j$ – as any spatial construct:



**Fig. 3** Sample skin pixels from the input image in Fig. 5 (a) shows relationships between image pixels, with a correlation coefficient  $r = 0.9181$  (b) The degree of normality in pixel distribution

$$q = \{q : q \in SP\}, \text{ Thus } q = \begin{cases} 0, & \text{if } SP'(p_j) \\ \geq 1, & \text{otherwise} \end{cases} \quad (9)$$

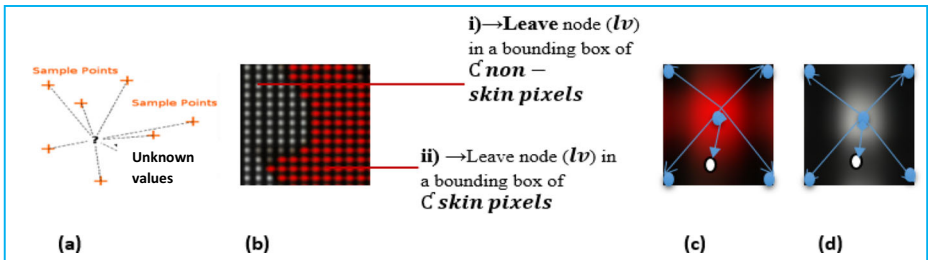
$SP'$  signifies a very close proximity (within a bounding box) between  $p_j$  in  $P$ .

$\square \forall p_j$ , in Proof 1, since spatial autocorrelation occurs due to correlation of a variable with itself through space [10], we assume  $q=0$  in Eq. 9. That is,  $\forall p_j : p_j \in P, \forall p_r : p_r \in P$ , observations made from  $p_r$  include information present in  $p_j$ . Therefore, it is rational that the sample size,  $r$ , be less than the total number of observations  $\Omega_0$  in each  $Ln$  bounding box/cell.

We, therefore, select only  $r$  sample pixels (described in ALGORITHM 2) on strategic locations from each  $Ln$  cell (Fig. 4), as the five significant sample points for interpolation and then move on to the interpolation procedure in Section 3.3.1.

### 3.3.1 Prediction by interpolation using IDW

In this section, we describe how we predicted 24 points (pixels) from only 5 points (pixels). Now, since all the  $\Omega_0$  elements in the  $Ln$  cell are highly correlated pixels, and since the cell is



**Fig. 4** Interpolation procedure (a) underlying idea of predicting new unknown cell values from known ones based on known sample points, (b) Expanded cells showing skin and non-skin pixel (in their bounding boxes) as classified by PKT cropped from Fig. 11f, (c) Expanded single cell from skin area from (b-ii), (d) Expanded single cell from the non-skin area from (b-i)



very small (Fig. 5e), there is a high probability following proof 2, that if the mid  $p_j$  in the cell ( $p_jM$ ) passes the threshold test in Eq. 10, then every other  $p_j$  in the cell will pass i.e.

$$Hb(p_jM) = T \Rightarrow Hb(p_j) \cong T, j = 1, 2, \dots, \Omega_0 \tag{10}$$

Interpolation as depicted in Fig. 4a, is a way of predicting values in a cell from a limited number of sample data points. The diagram in Fig. 4, depicts the prediction procedure, the white points in Figs. 4c and 4d are arbitrary unknown pixel values. The blue point at the middle  $p_jM$  is a known sample point, which carries the largest classification weight ( $w = d(p_r, p_j)^2$ ), where  $d$  = distance between known pixel  $p_r$  and unknown pixel  $p_j$  as in Eq. 12.

We already showed that  $q = 0$  in Eq. (12). That means the distance between  $p_jM$  and all  $p_j$  in  $R$  (ALGORITHM 2) = 0. Therefore, to find the value of  $p_j \notin R$ , for each  $Ln$  cell, we interpolate (Eq. 12) through locations  $\hat{p}_j$  to test the color channels of  $p_j$  against the threshold  $Hb$  based on the known points/pixels in  $R$  i.e.:

$$\Rightarrow Hb(p_j) \cong Hb(p_jM), j = 1, 2, \dots, \Omega_0 \tag{11}$$

So, let  $p_r$  be  $p_j \in R$ , and  $Hb(p_r)$  be their threshold value, let  $d$  be the distance between  $p_r$  and unknown  $p_j$  thus from Eq. 12, we find a discrete assignment of  $p_j$  in each cell using inverse distance weighting (IDW) with a power of 2:

$$\Rightarrow Hb(p_j) = \begin{cases} \frac{\sum_{i=1}^r \left( \frac{1}{d(p_r, p_j)^2} \right) Hb(p_r)}{\sum_{i=1}^r \left( \frac{1}{d(p_r, p_j)^2} \right)}, & \text{if } d(p_r, p_j) \neq 0 \text{ for all } j \\ Hb(p_r), & \text{if } d(p_r, p_j) = 0, \text{ for some } j, j = 1 \dots \Omega_0 \end{cases} \tag{12}$$

Following the evaluation in Eq. 12, the threshold values of all  $p_j$  in each cell are projected such that:

$$Hb(p_j) = Hb(p_r), \text{ if } d(p_r, p_j) = 0. \tag{13}$$

Thus, all cells that meet the criterion in Eq. 13 are returned as skin cells (ALGORITHM 3). Based on this proposition, the calculation in Eq. 4, reduces to the calculation in Eq. 5 warranting up to 80% less work and time.

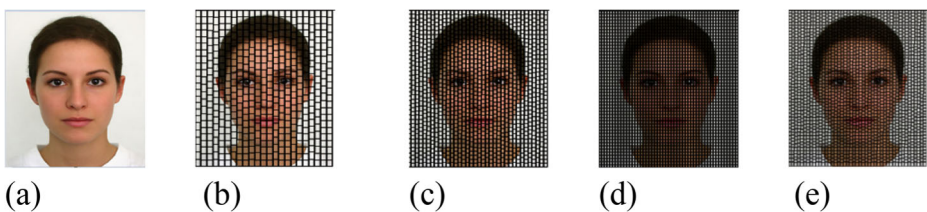


Fig. 5 Output of the PKT on different values of  $\mu$  (a) the original image (b, c, d, and e) outcome of the partition based on different values of  $\mu$ . (b)  $\Rightarrow \mu = 20$ . (c)  $\Rightarrow \mu = 40$ . (d)  $\Rightarrow \mu = 100$ . (e)  $\Rightarrow \mu = 25$

Note, combining the interpolation processes with a fast, spatial search structure like PKT, achieves an efficient  $\log N$  interpolation performance, which is highly suitable for large-scale problems.

It is evident from Eq. 13 that for an image or object of any size in 2 dimensions, only  $r = 5 \times \Delta \times \Delta$ , rather than  $\Omega_0 \times \Delta \times \Delta$  of the data elements/pixels, would be tested for matching purposes.

If we convert the calculation in Eq. 14 (Q), into a percentage using our example image of size 3000, that is, we find the ratio of the number of operations to the size of the input size and multiply by 100, then

$$\begin{aligned} Q &= \frac{\text{the number of operations}}{\text{Size of the input}} \\ \Rightarrow Q &= (605/3000) \times 100 \cong 20\%. \end{aligned} \quad (14)$$

Thus, we conclude that the percentage of pixels/points needed to detect human skin or match/classify patterns from any image is

$$\delta / DL \times 100 \cong 20\%$$

□ Only  $\cong 20\%$  of the pixels in an image is required to classify the pixels in an efficient skin detection/segmentation procedure

### 3.4 Finding the human skin pixel

We have described our interpolation procedure, using IDW in Section 3.3. Now, let us look at how the PKT carries out the classification/segmentation process (using our proposed HSL and RGB color models in Table 1), for identifying human skin presence in an image.

Initially, the program starts by extracting all the pixels from the image and then store them in a  $k-1$  dimensional array (as described in Section 3). Next, the array is partitioned to produce the leave node  $L_n$  following the procedure in ALGORITHM 1.

The process of mining patterns from images can generally be enhanced, by adopting a  $k$ -dimensional data structure like PKT, which is efficient for detecting patterns from coloured  $k$ -dimensional images. ALGORITHM 3, highlights the steps involved in this operation.

From each  $L_n$  (the tiny rectangles/cells in Fig. 5e), containing  $\Omega_0$  pixels, only  $r$  pixels elements belonging to  $R$ , are selected strategically. The RGB values of each  $r$  pixel ( $p_r$ ) in the array are converted to HSL values. All unknown  $p_r$  in each cell are compared recursively to match the skin/no-skins threshold in Table 1, eventual, only  $L_n$  where all  $p_r$  meet the colour matching criteria are returned.

---

### 3.4.1 ALGORITHM 3: Detecting skin pixels

---

Input:  $\mathbb{Z}$  from ALGORITHM 1

1.  $\Phi = (\mathbb{H}_{RGB})$  /\*accepted RGB color threshold\*/
2.  $\Phi_1 = (\mathbb{H}_{HSL})$  /\*accepted HSL color threshold\*/
3.  $\beta = []$  /\*an empty list to contain skin pixels\*/
4. Procedure Find the skin pigments ( $\mathbb{Z}$ )
5. While  $\mathbb{Z}_{jk=k_1}$ , /\* $\mathbb{Z}_{jk=k_1}$  is each leaf node from (ALGORITHM 2)\*/
  - a. Extract R from  $\mathbb{Z}_{jk=k_1}$
  - b. For each  $p_r \in R$  /\* $p_r$  are all the qualifying pixels in cell  $\mathbb{Z}_{jk=k_1}$ \*/
    - i. Extract RGB coordinates of  $p_r$  ( $p_{r RGB}$ )
    - ii. If  $(p_{r RGB}) \equiv \Phi$  //checks if pixel is identical to the defined RGB colour threshold
      - Compute the HSL values of  $p_r$  ( $p_{r HSL}$ )
      - If  $(p_{r HSL}) \equiv \Phi_1$  /\*checks if pixel is identical to the defined HSL colour\*/  
Add  $p_r$  to  $\beta_r$
      - Else Return
    - iii. Else get next  $p_r$
  - c. Add  $\beta_r$  to  $\beta$
6. End While

Output:  $\beta$  (2D array of skin pixels)

---

## 4 Results/discussion

### 4.1 Experiment

PKT experiment was implemented from scratch in JavaScript and ran on a machine with Intel® Core™ i5-5200U CPU @ 2.20GHz 2.20 GHz, 8 GB RAM, with Windows 10. All PKT modules/functions (ALGORITHM 1–3 etc.), sub-functions (e.g., bounding boxes, selecting  $r$ , etc.) and helper functions (e.g., color channel conversion and matching procedures, sort, sum, distance etc.) were custom-built de novo in JavaScript and visualised using Html tags ran on Python's HTTP local host server.

### 4.2 Datasets

We conducted the experiments using real-world datasets, synthetic datasets, and image segmentation datasets (all two-dimensional). The ColorFERET dataset [34] (Fig. 12b), Pratheepan human skin dataset [52] (Fig. 12a), and various other images of diverse complexion, pose, orientation, age, variation of illumination and sex, selected from the internet (Fig. 12c). The results of applying our algorithm to these datasets is shown on Figs. 12 and 13.

### 4.3 Elucidation

The proposed improved skin colour threshold in Table 1 and the multidimensional spatial structure (PKT) are applied in this work for effectively detecting human skin from an image. Below we have presented the results of the model. The results show that the structure is very versatile as it is promising, showing tendencies of greater prospects (Table 4).

By performing a few geometric operations on the reverse aspect of the model (Fig. 6), some facial features including the face, nose, eyes mouth and so on, can be detected. In Fig. 6, the non-skin areas have been marked with red points by PKT. Using some simple distance metrics, the head and neck could be extracted. Additionally, working out the position of facial features can help find the nose, mouth, and eyes, but this will be looked at in a later version.

### 4.4 Comparison between common color thresholds for human skin classification and ours

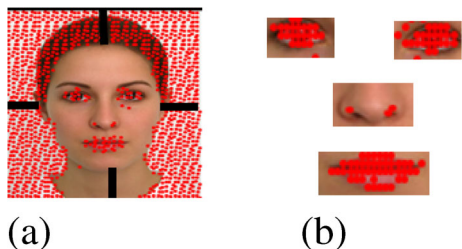
[12] Described several color thresholds for modeling skin colors. However, researchers including [3, 5, 21, 25, 37], adopted similar RGB/HS color models for human skin identification and possible recognition. These models fall within a given threshold for all skin color types (Table 1)

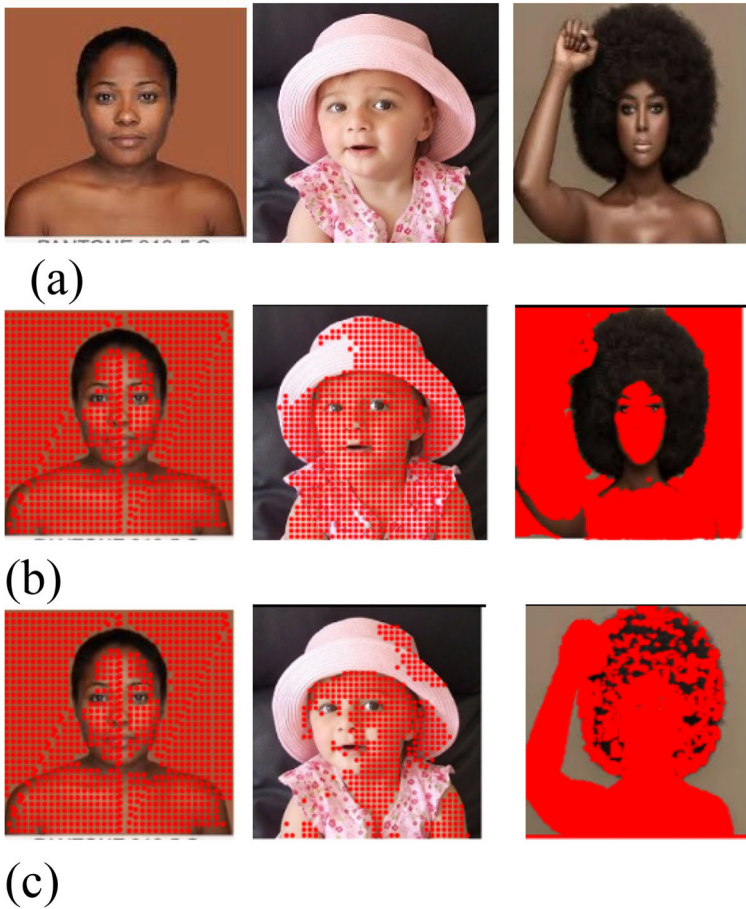
Our study is disputing the human skin color threshold premise and assumption made by authors [3, 5, 21, 25, 37], as it falls short of reality for certain human skin color codes. Three main colors (Red, Pink, Brown) pose the most problem in the human skin color threshold setting. Of course, this is because they are very close in shades to the red color underlying the human skin [20]. After an exhaustive toning on both the HSL and RGB color models, to enhance insensitivity to luminance, we came up with a more efficient threshold (Table 1) to tackle darker skin color problems mentioned earlier in Section 2 (Figs. 7 and 8).

It was noted, however, that these color models do not perform efficiently when applied discretely. That is, there is always a high degree of false hit. Hence, we tested a combination of both, and the results are found in Fig. 9 (for suggestions from others), and Fig. 10 (for our suggestion). The improved colour threshold we suggested in Table 1 is efficient especially as an improvement for darker skin prediction. Evidence of this is shown is equally shown in Figs. 9 and 10.

Figure 11 shows the various stages of the procedures of the PKT. The ground truth image Fig. 11a, original image Fig. 11b and the result of various stages of the skin detecting process using different values of  $\mu$ . Figure 11f is our final result (with  $\mu = 25$ ). The red points on the face are midpoints of the bounding boxes on each leaf node  $Z_{jk=k1}$  where Eq. 6 evaluates to T.

**Fig. 6** a Is the reverse aspect of our test image; non-skin areas are shaded b cropped portion of the image isolating facial features





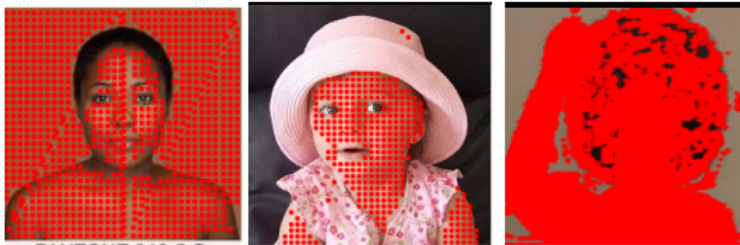
**Fig. 7** a Original image b and c shows common HSL, RGB color model as adopted in [3, 5, 21, 25, 37]. (b) RGB ( $R > 95$  and  $G > 40$  and  $B > 20$  and  $R > G$  and  $R > B$  and  $|R - G| > 15$  and  $A > 15$ ). (c) HS  $>> 0.0 <= H <= 50.0$  and  $0.23 <= S <= 0.68$

Figure 11g shows only the points representing the skin area. As we can see, the  $\mu$  affects the performance and behavior of the tree. In Fig. 11c, where  $\mu = 500$ , some parts of skin pigments were not detected, this will give rise to a high rate of false hits. Even though Fig. 11d—where  $\mu = 100$ —looks promising, there is still a tendency of some measure of false-negative hit. Figure 11e shows the outcome of Fig. 11d without displaying the tree. At  $\mu = 25$  in Fig. 11f, a perfect result was achieved. Figure 11h shows how the PKT was used to smooth out the result of the image in Fig. 10c.

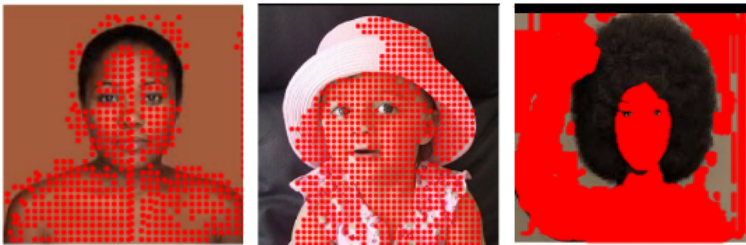
In Fig. 12, we show the result of applying the PKT algorithm to find human skin of varying types, complexion, illumination, shade, pose, position, etc. The images in Fig. 13 are skin pigments, detected using the same technique. However, the tree boundaries are not displayed.

#### 4.5 Evaluation

In order to evaluate the performance of the tree structure against the commonly used methods (PBP/ PW operations and quad-tree like structures) adopted by many authors, we have



(a)



(b)

**Fig. 8** (a) and (b) our suggested color models, (a) HSL ( $H > 10.0$  &&  $H < 30.0$  &&  $S > 0.20$  &&  $L1 > 0.10$ ), (b) RGB ( $R > 50$  &&  $G < 220$  &&  $B \neq 60$  &&  $B < 200$  &&  $R > G$  &&  $R > B$  &&  $|(R - G)| > 20$ )

compared the tree performance with these techniques. Figure 14, shows the performance of the various models. In Fig. 14a, we have the ground truth image, Fig. 14 (a-i) shows the skin area and Fig. 14 (a-ii) highlights non-skin areas as identified by the PW technique. No enhancement was applied, thus each pixel was checked individually based on our purported color threshold for skin pixel classification in Section 3.1.

In Fig. 14b, we show the same image with the skin area mapped out with the PKT. In Fig. 14 (b-i) the tree boundaries are not displayed and in Fig. 14 (b-ii), the reverse effect of the tree was depicted, showing non-skin areas as identified by the tree.

Figure 14c is the result of applying the PQT to the skin prediction procedure based on our improved color threshold in Section 3.1. Skin areas mapped out with the PQT is shown in Fig. 14 (c-i) without displaying the boundaries. In Fig. 14 (c-ii), the reverse effect of the PQT was depicted, showing non-skin areas as identified by the tree.



(a)

(b)

(c)

**Fig. 9** RGB + HSL shows the result of the combination of HSL, RGB color models (other methods)

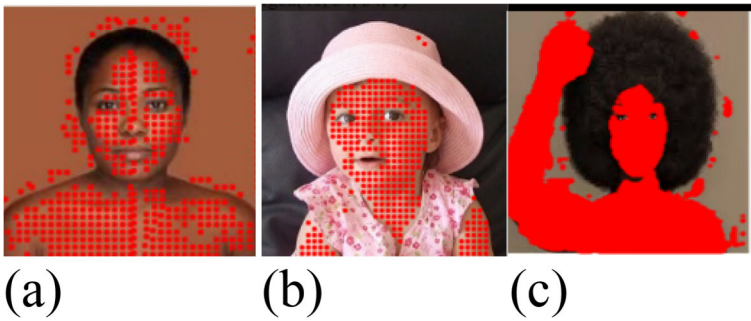


Fig. 10 RGB + HSL shows the result of the combination of HSL, RGB color models (our suggestion)

### 4.6 Performance graph

A plot of time performance for the various structures is shown in Fig. 15. In Fig. 15a, a comparison of time of construction between the PKT and the PQT is shown. Needless to say that the quad-tree deep quad partitioning strategies have a negative effect on the speed performance of the structure. Though the quad structure performs fairly in terms of classification (Fig. 14c), a little improvement might be necessary to speed up the system. As can be seen, the worst performance in terms of time consumption for pixel classification is the PBP technique, followed by the PQT method (Fig. 15b). This means that even though there is no structure to build in the PBP method, the method can not improve beyond a time complexity of  $O(n)$ . Note, the timing here includes the pixel extraction time, array manipulation, tree partitioning, and pixel classification. These comparisons are also shown in Table 5.

### 4.7 Precision, recall and accuracy

To evaluate the accuracy and precision of our proposed model, we have prepared Tables 2, 3 and 4 for different pixel sizes. The tables show the accuracy, recall and precision rate of these three methods in terms of pixel classification and skin segmentation. The accuracy, recall and precision calculation for proper evaluation of our skin/no-skin pixel classification model, were based on the formula in [15]. The higher precision and accuracy rate have been achieved by PKT because of the large IDW weight value attached to the midpoint pixel ( $p_jM$ ) of the leaf bounding boxes, such that  $H(p_jM)$  must evaluate to T (Eq. 10) for any other unknown pixel  $p_j$  to be a valid skin pixel. Even though the PQT method employs a similar restriction strategy, the four-child partitioning technique is bound to limit the restriction, by partitioning a cell with

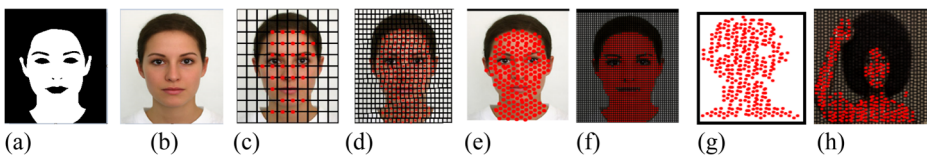
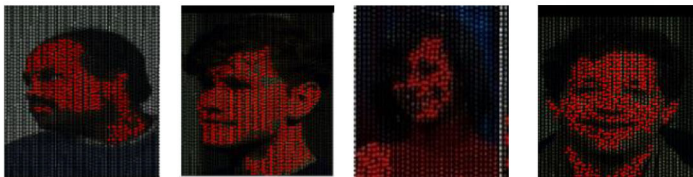


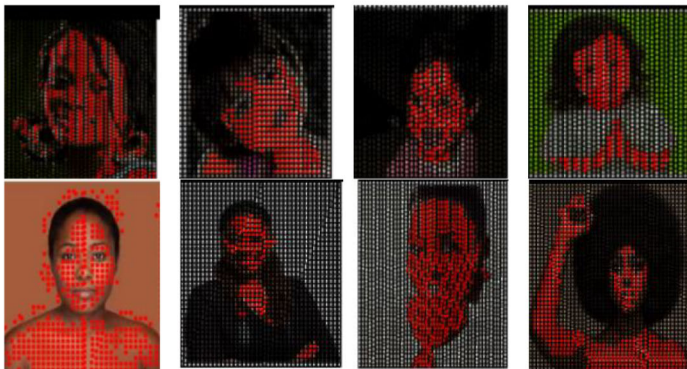
Fig. 11 The skin detection procedure completed with the proposed tree structure (a) ground truth image. (b) the original image. (c) Output showing skin area, when  $\mu = 500$ . (d) Output when  $\mu = 100$ . (e) Midpoints of leaf bounding boxes (that match our skin threshold value (ALGORITHM 3) shown as red points. (f) The expected and final result of the skin detection process. Extracted skin areas marked as red points. (g) Points representing skin area, extracted from an image using PKT (h) PKT smoothed image of in Fig. 10c



(a) Pratheepan images



(b) ColorFERET images



(c) Internet images

**Fig. 12** Other images, showing detected skin using our proposed model. **a** Pratheepan images, **b** ColorFERET images, **c** Internet images



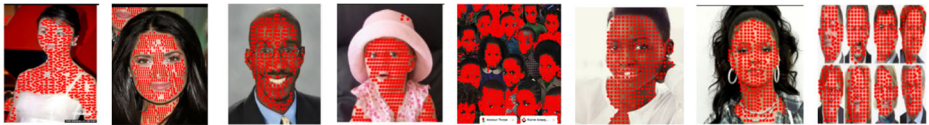
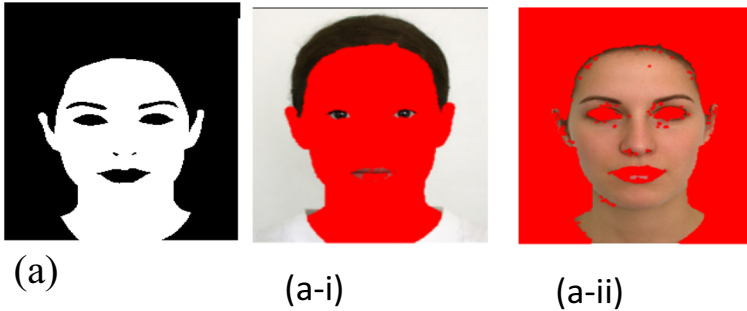


Fig. 13 More images tested (tree not displayed)

PBP

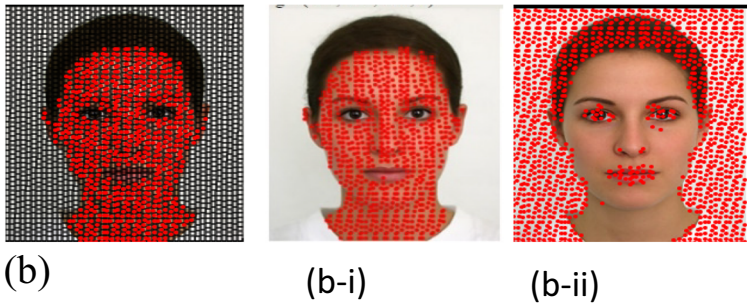


(a)

(a-i)

(a-ii)

PKT

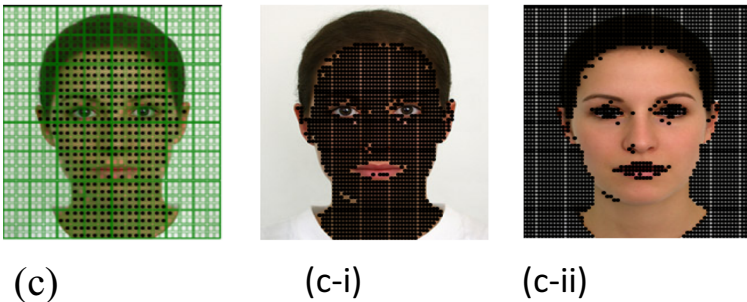


(b)

(b-i)

(b-ii)

PQT

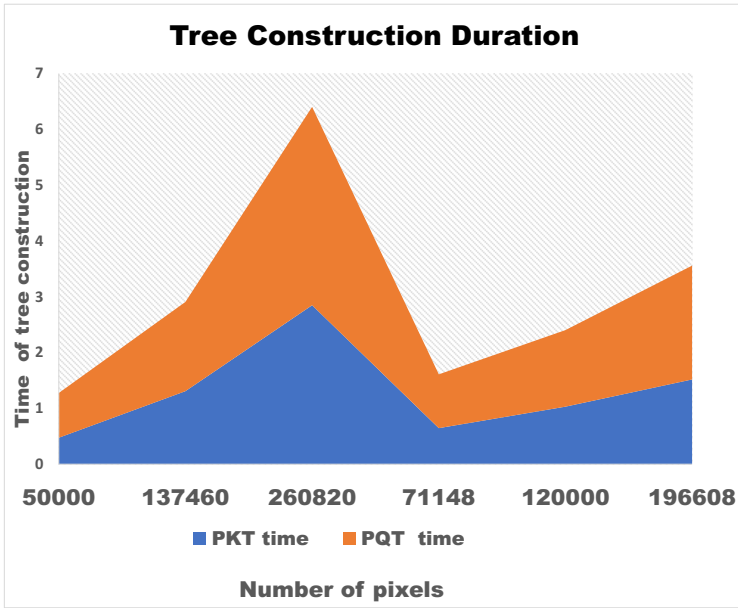


(c)

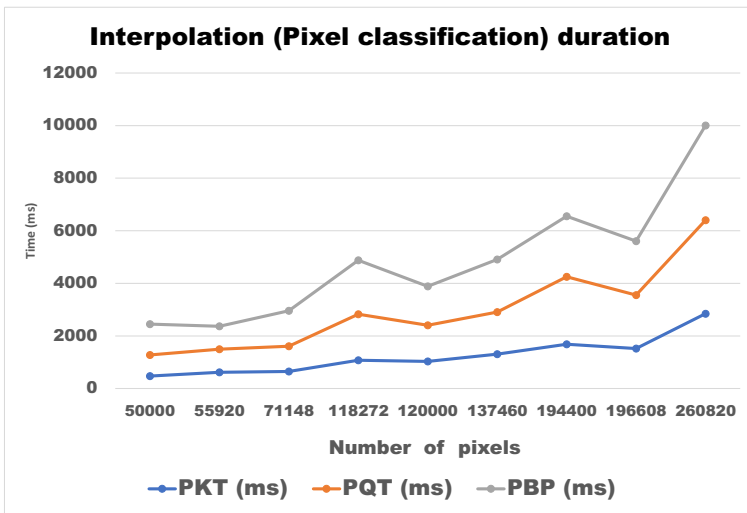
(c-i)

(c-ii)

Fig. 14 Comparison of performance between the three methods of study, using an image from Pratheepan dataset (a) the ground truth image, (a-i) PBP predicted skin pixels (in red), (a-ii) reverse process of PBP, showing non-skin pixels in red, (b) result and performance measure of PKT(proposed model) skin identification procedure, (b-i) extracted skin area in red (red points represents the midpoint of the leaf node ( $Ln$ ) in the region), (b-ii) reverse process showing non-skin area. (c) result and performance measure of the PQT skin identification procedure, (c-i) extracted skin area in black points, (c-ii) reverse process showing the non-skin area



(a)



(b)

**Fig. 15** Time of construction and classification (a) comparison of times elapsed between proposed system PKT and other (b) comparison between the size of data and the duration of pixel classification between the two methods of study

very close neighbors into two different cells of dissimilar subsets. PBP based systems do not have such constraint as such, the system selects every area of the image where there seems to be a trace of the defined colour threshold ( $H$ ).

**Table 2** Measurement of Accuracy and Precision of the proposed structure for the PBP method

Total Number of pixels	Actual Skin pixels (GT)	Skin pixels found (PBP)	Actual Non-Skin pixels (GT)	Non-Skin pixels found (PBP)	TP	FP	TN	FN	Precision (%)	Accuracy (%)	Recall (%)
50,000	18,926	20,510	31,074	29,490	18,926	1584	29,490	0	92.37	96.83	100
137,460	14,482	15,866	122,978	121,594	14,482	1384	121,594	0	91.28	98.99	100
71,148	19,666	24,239	51,482	46,909	19,666	4573	46,909	0	81.13	93.57	100
120,000	42,249	43,408	77,751	76,592	42,249	1159	76,592	0	97.32	99.03	100
196,608	55,836	68,818	140,772	127,790	55,836	12,982	127,790	0	81.13	93.40	100
218,700	32,935	32,834	185,765	185,866	32,834	0	185,866	0	100	100	100
43,200	14,865	16,016	28,335	27,184	14,865	1151	27,184	0	92.81	97.33	100
248,832	29,218	27,312	219,614	221,520	27,312	0	219,614	1906	100	99.23	93.47
260,820	18,418	17,229	242,402	243,591	17,229	0	242,402	1189	100	99.54	93.54
194,400	44,880	46,838	149,520	147,562	44,880	1958	147,562	0	95.82	98.99	100
159,506	9107	7357	150,399	152,149	7357	0	150,399	1750	100	98.90	80.78

**Table 3** Measurement of Accuracy and Precision of the proposed structure for PQT method

Total Number of pixels	Actual Skin pixels (GT)	Skin pixels found (PQT)	Actual Non-Skin pixels (GT)	Non-Skin pixels found (PQT)	TP	FP	TN	FN	Precision (%)	Accuracy (%)	Recall (%)
50,000	18,926	18,000	31,074	32,000	18,000	0	31,074	926	100	98.01	95.10
137,460	14,482	7368	122,978	130,092	7368	0	122,978	7114	100	94.82	50.87
71,148	19,666	13,584	51,482	57,564	13,584	0	51,482	6082	100	91.45	69.07
120,000	42,249	28,440	77,751	91,560	28,440	0	77,751	13,809	100	88.49	67.31
196,608	55,836	55,104	140,772	141,504	55,104	0	140,772	732	100	99.62	98.68
218,700	32,935	29,616	185,765	189,084	29,616	0	185,765	3319	100	98.84	89.92
43,200	14,865	13,230	28,335	29,970	13,230	0	28,335	1635	100	96.21	89.00
248,832	29,218	23,220	219,614	225,612	23,220	0	219,614	5998	100	97.54	79.47
260,820	18,418	15,066	242,402	245,754	15,066	0	242,402	3352	100	98.71	81.80
194,400	44,880	40,200	149,520	154,200	40,200	0	149,520	4680	100	97.59	89.57
159,506	9107	6580	150,399	152,926	6580	0	150,399	2662	100	98.15	72.25

**Table 4** Measurement of Accuracy and Precision of the proposed structure for PKT method

Total Number of pixels	Actual Skin pixels (GT)	Skin pixels found (PKT)	Actual Non-Skin pixels (GT)	Non-Skin pixels found (PKT)	TP	FP	TN	FN	Precision (%)	Accuracy (%)	Recall (%)
50,000	18,926	18,720	31,074	31,280	18,720	0	31,074	206	100	99.59	98.91
137,460	14,482	11,920	122,978	125,540	11,920	0	122,978	2562	100	98.14	82.30
71,148	19,666	20,040	51,264	51,108	19,666	374	51,108	0	98.19	99.47	100
120,000	42,249	34,518	77,751	85,482	34,518	0	77,751	7731	100	94.0	81.70
196,608	55,836	60,060	140,772	136,548	55,836	4224	136,548	0	92.97	97.85	100
218,700	32,935	32,664	185,765	186,036	32,664	0	185,765	271	100	99.87	99.17
43,200	14,865	15,860	28,335	27,340	14,865	995	27,340	0	93.72	97.69	100
248,832	29,218	27,125	219,614	221,707	27,125	0	219,614	2093	100	99.15	92.83
260,820	18,418	17,424	242,402	243,396	17,424	0	242,402	994	100	99.61	94.60
194,400	44,880	47,923	149,520	146,477	44,880	3043	146,477	0	93.65	98.43	100
159,506	9107	6950	150,399	152,556	6950	0	150,399	2157	100	98.64	75.31

**Table 5** Run-time of the different number of pixels

Number of pixels	PKT time (s)	PQT time (s)	PBP time (s)
50,000	0.473	0.803	1.173
137,460	1.309	1.597	2.000
260,820	2.847	3.555	3.602
71,148	0.645	0.965	1.346
120,000	1.030	1.372	1.484
196,608	1.519	2.037	2.050
55,920	0.616	0.880	0.870
118,272	1.073	1.755	2.048
194,400	1.686	2.566	2.300

## 5 Conclusion and future works

PBP operations for human skin detection or skin pixel classification is sometimes characterized by a high rate of false hits, and increased time consumption. In this paper, we have presented an improved color threshold-based algorithm for recognizing and classifying human skin pixels in an image using the combination of RGB-HSL color models. To speed up the process and improve performance, we have proposed and implemented a k-dimensional structure for the classification procedure. Our proposed model shows very high promising results in terms of precision, recall and accuracy as compared to most state-of-the-art systems.

Images from different sources were tested, and the model scaled high. From the results presented in Tables 2, 3 and 4 (the overall performance of the algorithm), it can be seen that the proposed model provides a very significant reduction in false detection rates as compared to the PBP testing mechanism and quad-tree like techniques applied in many systems. Quad-trees have been in use for speeding up of the detection process however, we proposed and implemented PQT, an improved quadtree structure to which we compared our main model PKT (Table 5).

Although there is a significant improvement as compared to the PBP techniques in terms of speed and accuracy, the quad-tree structure showed certain drawbacks in terms of speed of construction and speed of classification of pixels, which can be attributed to the structures' partitioning strategy. We can boldly say that the proposed approach yields better detection performance compared to that of the state-of-the-art PBP and the quad-tree based techniques with a significant reduction in time and computational cost.

We have equally shown that with little geometry, the algorithm can detect a face, hand, and other features and gestures. For future work, an improved PKT is currently being investigated, which performs a second level filtering of the PKT cells (sub-regions) to produce a (*skin*) patch rather than skin pixels. This second PKT variants (which will include an edge detection procedure), will apply more smoothly to generalised object detection, segmentation, and recognition. We shall equally investigate parallelizing the structure to further improve its speed and efficiency in terms of general pattern mining. We are also investigating the implementation of PKT and PQT for higher dimensional spaces, and for clustering non-spatial datasets.

Finally, we claim that using these structures (PKT, PQT), only  $\cong 20\%$  of the pixels in an image are required to classify the pixels in a skin detection procedure.

**Acknowledgments** This work was supported by the University of Abuja, Nigeria and Tertiary Education Trust Fund (TETF)/Academic Staff Training and Development (AST&D), Nigeria.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Abbas A R, Farooq A O (2018) Human skin colour detection using bayesian rough decision tree. In: Almamory S, Alwan J, Hussein A (eds) *New trends in information and communications technology applications*. NTICT 2018. Communications in computer and information science, vol 938. Springer, Cham. [https://doi.org/10.1007/978-3-030-01653-1\\_15](https://doi.org/10.1007/978-3-030-01653-1_15)
2. Albiol A, Torres L, Delp E J (2001) Optimum color spaces for skin detection. In *proceedings 2001 780international conference on image processing*. IEEE, (cat. No. 01CH37205), (1):122–124. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.6402&rep=rep1&type=pdf>
3. Ali A, El-Hafeez T, Mohany Y (2019) A robust and efficient system to detect human faces based on facial features. *Asian Journal of Research in Computer Science* 2(4):1–12 <https://doi.org/10.9734/ajrcos/2018/v2i430080>
4. Ban Y, Kim S K, Kim S, Toh K A, Lee S (2014) Face detection based on skin color likelihood. *Pattern Recogn* 47(4):1573–1585. <https://doi.org/10.1016/j.patcog.2013.11.005>
5. Baskan S, Bulut MM, Atalay V (2002) Projection based method for segmentation of human face and its evaluation. *Pattern Recogn Lett* 23(14):1623–1629
6. Bentley JL (1975) Multidimensional binary search trees used for associative searching. *Commun ACM* 18(9):509
7. Berchtold S, Keim D A, Kriegel H P (2002) The X-tree: An index structure for high-dimensional data. Edited by Kevin Jeffay, Hongjiang Zhang, in the morgan kaufmann series in multimedia information and systems, readings in multimedia computing and networking, morgan kaufmann, pp 451–462. <https://doi.org/10.1016/B978-155860651-7/50124-8>
8. Buolamwini J, Gebru T (2018) Gender shades: intersectional accuracy disparities in commercial gender classification. *Conference on fairness, accountability and transparency*, p 77–91. <http://proceedings.mlr.press/v81/buolamwini18a.html>
9. Bush I J, Abiyev R, Ma'aitah M K S, Altuparmak H (2018) Integrated artificial intelligence algorithm for skin detection. *ITM Web of conferences*, EDP Sciences, (16): p. 02004.
10. Chen J, Chen Y, Yu J, and Yang Z (2011) Comparisons with spatial autocorrelation and spatial association rule mining. *Proceedings 2011 IEEE international conference on spatial data mining and geographical knowledge services*, pp 32–37. <https://doi.org/10.1109/ICSDM.2011.5969000>
11. Chen HH, Ding JJ, Sheu HT (2014) Image retrieval based on quadtree classified vector quantization. *Multimed Tools Appl* 72(2):1961–1984
12. Chen W, Wang K, Jiang H, Li M (2016) Skin color modelling for face detection and segmentation: a review and a new approach. *Multimed Tools Appl* 75(2):839–862
13. Conci OVA (2009) A skin detection using HSV color space, Pedrini, & J. Marques de Carvalho, *Workshops of Sibgrapi* p 1–2
14. Dastane T, Rao V, Shenoy K, Vyavaharkar D (2018) An effective pixel-wise approach for skin color segmentation-using pixel Neighbourhood technique. *Int J Recent Innov Trends Comput Commun* 6(3):182–186
15. Developers (2019) Classification: accuracy. *Machine Learning Crash Course*. <https://developers.google.com/machine-learning/crash-course/classification/accuracy>
16. Elfkihi S, DaoudiM, Aboutajdine D A (2006) A tree distribution for skin detection. *The second international symposium on communications, control and signal processing (ISCCSP'06)*. <https://www.eurasip.org/Proceedings/Ext/ISCCSP2006/defevent/papers/cr1259.pdf>
17. Faria R A D, Hirata Jr R (2018) Combined correlation rules to detect skin based on dynamic color clustering. In *proceedings of the 13th international joint conference on computer vision, imaging and*

- computer graphics theory and applications (VISIGRAPP 2018) - Volume 5: VISAPP, pp 309–316. <https://doi.org/10.5220/0006618003090316>
18. Hassan E, Hilal A R, Basir O, (2017) Using ga to optimize the explicitly defined skin regions for human skin color detection, 30th IEEE Canadian conference on electrical and computer engineering, (CCECE 2017), p. 1–4. <https://doi.org/10.1109/CCECE.2017.7946699>, <https://ieeexplore.ieee.org/abstract/document/7946699>
  19. Hua R, Wang Y (2017) Skin color detection based on super pixel. In proceedings of the 3rd IEEE international conference on computer and communications (ICCC), Chengdu, 2017, pp 1756–1760. <https://doi.org/10.1109/CompComm.2017.8322841>
  20. Jablonski NG (2006) Skin: a natural history. University of California Press, Berkeley
  21. Jati H, Dominic D D (2008) Human skin detection using defined skin region. In 2008 international symposium on information technology. IEEE, (1) p. 1–4)
  22. Kakumanu P, Makrogiannis S, Bourbakis N (2007) A survey of skin-color modeling and detection methods. *Pattern Recogn* 40(3):1106–1122
  23. Kawulok M, Kawulok J, Nalepa J (2014) Spatial-based skin detection using discriminative skin-presence features. *Pattern Recogn Lett* 41:3–13. <https://doi.org/10.1016/j.patrec.2013.08.028>
  24. Khan R, Hanbury A, Stoeftinger J (2010) Skin detection: a random forest approach. 2010 IEEE Int Conf Image Process 4613–4616
  25. Kolkur S, Kalbande D, Shimpi P, Bapat C, Jatakia J 2017 Human skin detection using RGB, HSV and YCbCr color models., (arXiv preprint arXiv:1708.02694)
  26. Legendre P (1993) Spatial autocorrelation: trouble or new paradigm? *Ecology* 74:1659–1673. <https://doi.org/10.2307/1939924>
  27. Khan R, Hanbury A, Stoeftinger J (2010) Skin detection: A random forest approach. 2010 IEEE international conference on image processing, p 4613–4616. <https://ieeexplore.ieee.org/document/5651638>
  28. Mahmoodi M R, Sayedi S M, Karimi F (2017) Color-based skin segmentation in videos using a multi-step spatial method. *Multimed Tools Appl* 76, 9785–9801. <https://doi.org/10.1007/s11042-016-3579-8>
  29. Mark S, Alberto S (2020) Region-based analysis, in feature extraction and image processing for computer vision (fourth edition). Academic press. Pp 399–432. ISBN 9780128149768. <https://doi.org/10.1016/B978-0-12-814976-8.00008-7>
  30. Mortazavi T M, Ebadati E O M (2019) An improved human skin detection and localization by using machine-learning techniques in RGB and YCbCr color spaces, *PeerJ reprints*
  31. Nguyen-Trang T (2018) A new efficient approach to detect skin in color image using Bayesian classifier and connected component algorithm, *Mathematical Problems in Engineering*
  32. Nikolskaia K, Ezhova N, Sinkov A, Medvedev M (2018) Skin detection technique based on HSV color model and SLIC segmentation method\* in proceedings of the 4th Ural workshop on parallel, distributed, and cloud computing for young scientists, Ural-PDC (pp. 123–135)
  33. Nishad PM (2013) Various color spaces and color space conversion. *J Global Res Comput Sci* 4(1):44–48
  34. National institute of standards and Technology (2011) Accessed [Online] Available at: <https://www.nist.gov/itl/iad/imagegroup/colorferet-database>. Accessed 23 March 2019
  35. Omer M A, Junaid J M, Bilal A H, Adnan M K 2018 Implementation of NOGIE and NOWGIE for human skin detection, *Int J Adv Comput Sci Appl (IJACSA)*, 9(7)
  36. Patil P M, Patil Y M (2012) Robust skin color detection and tracking algorithm, *Int J Eng Res Technol* 1 (8)
  37. Peer P, Solina F (1999) An automatic human face detection method. In *Computer vision - CVWW'99: proceedings of the computer vision winter workshop, Rastenburg, Austria, 8-10 February 1999*. - Str. 122–130. <https://plus.si.cobiss.net/opac7/bib/ferlj/1456724#full>
  38. Phung S L, Bouzerdoum A, Chai D (2002) A novel skin color model in ycbcr color space and its application to human face detection, *IEEE international conference on image processing (ICIP 2002)*, (1) p.289–292
  39. Ren X, Malik J (2003) Learning a classification model for segmentation proceedings of the 9th IEEE international conference on computer vision. IEEE Computer Society, Washington DC, pp 10–17. <https://doi.org/10.1109/ICCV.2003.1238308>, <https://ieeexplore.ieee.org/document/1238308>
  40. Roheda S (2017) A multi-scale approach to skin pixel detection. *Electron Imaging* 4:18–23
  41. Rossi J P, Queneherv P (1998) Relating species density to environmental variables in presence of spatial autocorrelation: a study case on soil nematodes distribution. *Ecography*, (21) p. 117–123
  42. Samson GL, Lu J (2016) PaX-DbSCAN: a proposed algorithm for improved clustering. In: Pańkowska MR (ed) *Studia Ekonomiczne. Zeszyty Naukowe*, (269524th ed.). Wydawnictwo Uniwersytetu Ekonomicznego w Katowicach, Katowice Retrieved from [www.sbc.org.pl/Content/269524](http://www.sbc.org.pl/Content/269524)
  43. Samson G L, Lu J (2018) Spatial clustering in large databases using packed X-tree. *Egypt Comput Sci J* 42(2):68–79. [http://ecsjournal.org/Archive/Volume42\\_Issue2.aspx](http://ecsjournal.org/Archive/Volume42_Issue2.aspx)
  44. Samson GL, Lu J, Showole AA (2014) “Mining Complex Spatial Patterns: Issues and Techniques”, *Journal of Information & Knowledge Management (JIKM)*, World Scientific Publishing Co. Pte. Ltd., 13(02): 1–20. <https://doi.org/10.1142/S0219649214500191>



45. Samson GL, Lu J, Usman MM, Xu Q (2017) Spatial databases: an overview. In: Lu J, Xu Q (eds) Ontologies and big data considerations for effective intelligence. IGI Global, Hershey, PA, pp 111–149
46. Samson GL, Usman MM, Showole AA, Lu J, Jazzaa H (2018) Large spatial database indexing with aX-tree. *Int J Sci Res Comput Sci Eng Inf Technol (JSRCSEIT)* 3(3):759–773
47. Saxen F, Al-Hamadi A (2014). Superpixels for skin segmentation. In 20. Workshop Farbbildverarbeitung, Wuppertal (20), pp. 153-159. <https://doi.org/10.13140/2.1.3293.3124>
48. Shen J, Zuo X, Li J, Yang W, Ling H (2017) A novel pixel neighborhood differential statistic feature for pedestrian and face detection. *Pattern Recogn* 1(63):127–138
49. Simonite T (2018) Photo algorithms ID white men fine—black women, Not So Much, Wired. (<https://www.wired.com/story/photo-lgorithms-id-white-men-fine-black-women-not-so-much>)
50. Smit AJ, Smit JM, Botterblom GJ, Mulder DJ (2013) Skin autofluorescence based decision tree in detection of impaired glucose tolerance and diabetes. *PLoS One* 8(6):e65592
51. Sun H-M (2010) Skin detection for single images using dynamic skin color modelling. *Pattern Recogn* 43(4):1413–1420
52. Tan W R, Chan C S, Pratheepan Y, Condell J (2012) A fusion approach for efficient human skin detection. *IEEE trans Ind Inf* 8(1):138–147 (T-II 2012). <https://doi.org/10.1109/TII.2011.2172451>, <https://ieeexplore.ieee.org/document/6051482>
53. Tan WR, Chan CS, Yogarajah P, Condell J (2012) A fusion approach for efficient human skin detection. *IEEE Trans Ind Inf* 8(1):138–147
54. Tavallali P, Yazdi M, Khosravi MR (2019) Robust cascaded skin detector based on AdaBoost. *Multimed Tools Appl* 78(2):2599–2620
55. Thakkar D, (2018) Top five biometrics: face, Fingerprint, Iris, Palm and Voice, Bayometric. (<https://www.bayometric.com/biometrics-face-finger-iris-palm-voice>)
56. Vezhnevets V, Sazonov V, Andreeva A (2003) A survey on pixel-based skin color detection techniques, *Proc. Graphicon* 3:85–92. <https://www.semanticscholar.org/paper/A-Survey-on-Pixel-Based-Skin-Color-Detection-Vezhnevets-Sazonov/bc1b5ff4fdb70c10a9aa0e9b8f6b260b2e1f4fed>
57. Xu T, Wang Y, Zhang Z (2013) Pixel-wise skin color detection based on flexible neural tree. *IET Image Process* 7(8):751–761
58. Zhang J, Wang H, Davoine F, Pan C (2012) Skin detection via linear regression tree. 21st IEEE international conference on pattern recognition. (ICPR2012), p. 1711–1714
59. Zortea M, Flores E, Scharcanski J (2017) A simple weighted thresholding method for the segmentation of pigmented skin lesions in macroscopic images. *Pattern Recogn* 64:92–104

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.