American University of Armenia
Zaven & Sonia Akian College of Science & Engineering

Capstone Project

Face Detection and Recognition Based Attendance System

Authors:
Syuzanna Harutyunyan, Eleni Khodabakhshi

Supervisor:
Suren Khachatryan

Spring, 2024

Acknowledgments

We would like to express our sincere gratitude to Professor Suren Khachatryan for his continuous, valuable support, and professional guidance throughout the entire research process. His consistent and professional feedback significantly contributed to the development and improvement of our Capstone project. His patience, belief in us, and assistance were integral to the development of our Capstone project and its final result.

Abstract

Face detection and recognition technologies have become important in modern applications, significantly in the automated attendance tracking. Our capstone project leverages these technologies to develop an Attendance System potentially applicable for diverse environments including workplaces, universities, and schools, with a particular focus on lecture rooms in universities. Our system is built around the ImageJ software and utilizes image processing capabilities. Through ImageJ and custom Java codes, we identified optimal HSV color space ranges for accurate facial pixel detection. The algorithm implemented in our system captures images, processes them to isolate facial features, and eliminates the background and non-facial pixels. Fitting ellipses are computed over detected faces, facilitating precise identification and attendance recording. In the end we discuss the further development of the system to enhance its accuracy and turn it into a consistent solution for automated attendance systems.

Keywords: *Face detection, Face recognition, ImageJ, Java, RGB, HSV, Attendance checking systems*

Contents

1. Introduction…………………………………………….… 6

2. Literature Review………………………………….. 2-12

2.1 Attendance Checking Systems

2.2 Skin-Color Classifiers

2.2.1 Comparison of Skin-Color Classifiers

2.2.2 Comparison of Color Space Representations

3. Problem Statement………………………………… 13

4. Method Description... 14-16

4.1 Face-specific Colors in RGB

4.2 Face-specific Colors in HSV

4.3 Datasets Used

4.3.1 FEI Face Database

4.3.2 American University of Armenia (AUA) Classroom Dataset

4.3.3. Web-Scraped Images

5. Experiments... 17-26

5.1 The classroom dataset

5.2 Experiments on facial dataset

5.2.1 Analyzing Hue, Saturation, and Brightness Behaviors for Person Detection

5.2.2 Breakdown of the "Face_Colors_Filter_HSV_Hist" Class

5.2.3 Breakdown of the "Face_Colors_Filter_HSV" Class

5.3 Post-Processing

6. Testing Results... 27-34

6.1 Image Grouping: Large vs. Small Classrooms

6.2 Background Contrast

6.3 Layout and Camera Direction

6.4 Frequency of Shots

6.5 Optimal Conditions for Face Detection in Classrooms

7. Conclusions and Future Work................................ 35

8. References.. 36-38

Appendix. ImageJ Plugins Using Java....................... 39-46

A.1 HueSpecific
A.2 Face_Colors_Filter_HSV
A.3 Face_Color_Sample_HSV
A.4 Face_Color_Filter_HSV_White
A.5 Face_Colors_Filter_HSV_Hist

1. Introduction

Attendance holds significant importance in different institutions and organizations, serving various purposes and being a key criterion for evaluating the performance of both students and employees. Each institution has its unique approach to tracking attendance. While some still use manual methods like paper or file-based systems, others have already adopted automated solutions such as biometric technologies and other new methods. However, in modern educational institutions, workplaces, and events, efficient attendance check remains critical. Traditional methods such as manual roll calls or biometric systems support attendance checking, but they are time-consuming, disruptive to the working and studying atmosphere, and can lead to inaccuracies. In addition, in a large classroom verifying students' attendance can be challenging. There can be some problems such as attendance sheets can be lost, or someone can pretend to be a student and so on.

Our research addresses these challenges by focusing on face detection and recognition within the context of attendance tracking. Face detection has been a subject of active studies for several decades due to its high practical value. Without face detection and recognition, an attendance system cannot function correctly. Our project aims to optimize this process by identifying the best color space region for face detection and recognition and will help to do attendance more accurately and easily.

The paper is divided into several chapters. It starts with the introduction and continues with the literature review part. After that, the problem statement of the Capstone project is defined, specifying the purpose of the paper and the questions it aims to answer by the end of the research. The Implementation part discusses the entire research and its steps, including the tools used, the code written, and the research problem as a whole. Following the Implementation chapter is the Testing chapter, which discusses the testing results and their challenges. Finally, the last chapter is the conclusion and future work.

2. Literature Review

2.1 Attendance Checking Systems

In the sphere of face recognition, detection, and attendance systems, multiple researchers, institutions, and organizations have done extensive studies. This research has explored diverse color spaces and implementation methodologies, each offering different approaches to the attendance system.

The papers that have been reviewed offer various approaches, problems, improvements, and implementation methods. However, we have created our own algorithm while trying to solve all the issues that appeared in others' work. Here are the examples of different projects about the Face recognition-based Attendance system and its algorithms.

According to the “Face Detection and Recognition Student Attendance System” project done at Kingston university of London [1], firstly the attendance system should be presented an image either via camera or from memory and it must detect the number of faces on it. After detecting faces, the system should crop them from the image and store them in memory for image recognition, which will occur in the next stage. Then the system needs to count the number of faces in the pictures. Then, it moves to the next stage which is recognition. This means it compares the faces it found with the ones it already knows. Special software is used for doing this process. This software makes sure everything runs smoothly and helps different parts of the system work together.

According to the “A model of Two-Factor Authentication using Facial Recognition in Automated Teller machines” Capstone project, done at University of Nairobi [2], offers other algorithms for the attendance system. The system consists of two parts. The first part, also known as the face detector, is a mobile component functioning as a camera. Its function is capturing student faces and saving them in a file through the utilization of computer vision face detection algorithms and techniques for face extraction. The second part, which is a desktop application, conducts face recognition on the stored images (faces) from the file. It records attendance of students and stores the outcomes in a database for future analysis.

According to the “Automated attendance system using face recognition through video surveillance” paper, done at Dept. of Computer Science & Engineering Maharaja Institute of Technology [3], the research is done using the video surveillance for the automated attendance system. Video surveillance is one of the best methods for the implementation of the automated attendance system. Through video surveillance, the system detects object movement, helps to do face detection and recognition processes within captured images. After that, the algorithm checks if the faces it found match with the students in a database. It only marks attendance for the students it recognizes.

As outlined in the "Face Recognition Student Attendance System" study conducted at Metropolia University of Applied Sciences [4], the project used LBPH in Django to construct a facial recognition system. This system was used for student attendance management. The main goal was to learn Django and web development and use that knowledge to create a working web app. Additionally, a lot of time was spent in researching face recognition technologies. Here, a "FacialRecognizer" class was made that works with OpenCV for recognizing faces. Then, OpenCV with Django was connected for helping to build the web app.

According to the "Study of Implementing Automated Attendance System Using Face Recognition Technique" paper [5], new method has been developed to conduct face recognition-based attendance, it uses the Personal Component Analysis (PCA) algorithm, The system will automatically keep track of students' attendance in the classroom. It will also make it easy for teachers to see students' information by keeping a log of when they come in and leave. The offered system has been implemented using three steps. The first step is to detect and extract face image and save the face information in an xml file for future references. The second step is to learn and train the face image and calculate the eigenvalue and eigenvector of that image. The final step is to recognize and match face images with existing face images information stored in xml file.

According to the "Face Detection and Recognition Student Attendance System" project conducted at Prince Mohammad Bin Fahd University [6], the project involves creating a device that can detect and recognize faces. The whole project is implemented using LabVIEW. Additionally, the project includes in it an Excel sheet which works with the program to manage student information, and a messaging device used for communication for sending messages to absent students or notifying parents. The project is done in LabVIEW and some smaller parts that work with it, like a list of student names in Excel. There is a need for a computer or laptop to put everything together.

According to the "Facial recognition-based attendance monitoring system for educational institutions" paper, done at University Tunku Abdul Rahman [7], The face recognition system in the project takes attendance by checking people's faces against a list of known faces. If there's a match, it marks them as present in a database. It's set up to run on a Raspberry Pi, with a web server for viewing attendance. The system has two main parts: hardware and software.

The "Student Attendance system using face recognition" project from the University for Business and Technology [8], introduces new ways to recognize faces for tracking student attendance. It uses computer vision to teach a system to recognize faces by taking 150 pictures of students with a laptop camera—50 for each student. Then, it trains the system to recognize these faces using two techniques: one for finding faces using Haar features and another for recognizing faces using Local Binary Patterns Histograms.

In the literature review part, the thesis talks about a method developed by Viola and Jones for spotting objects using rectangular features called Haar-like features, which was used in the algorithm for creating face recognition-based attendance systems. In the paper it is combined with a learning algorithm called AdaBoost and a series of strong classifiers. This method works on different devices, from powerful systems to mobile devices, and even in surveillance systems with many cameras.

Another method discussed is by Ahonen and team, who describe faces using Local Binary Patterns (LBP). They divide facial images into smaller parts and extract texture descriptors from each part. Then, they combine all these descriptors to make a complete description of the face. These two methods are the core methods which have been used in algorithms.

Moreover, in "Attendance System in Third -Level Irish Institutions and Colleges Using Face Recognition Approach" project [11], done at School of Computing National College of Ireland, a new face detection-based automated attendance system is proposed, which uses different machine learning techniques. By combining two techniques called CNN and HOG with another one called SVM, the system can find faces in images and recognize whose face it is. The research showed that this system is about 90% accurate when checked by looking at the results. It works better than older methods that used different techniques like Viola-Jones Face Detector, Eigen feature extractions, and principal component analysis.

According to “Face Recognition-based Lecture Attendance System” capstone, done at Kyoto University [12], again a new approach is discussed about the attendance system. In this time, the algorithm contains all the results of face recognition obtained by continuous observation. Continuous observation improves the performance for the estimation of the attendance. It continuously observes and records students' positions and face images during classroom lectures to estimate attendance and student positions. The paper is all about making the attendance system better. They want to improve how well it can recognize faces. They do this by giving more importance to certain seats and planning the camera positions better.

In conclusion, a superior attendance system needs to be implemented to be unique, precise, and inclusive of all the improvements mentioned in the papers.

2.2 Skin-Color Classifiers

2.2.1 Comparison of Skin-Color Classifiers

A good skin-color classifier needs to identify various skin types (like white, black, pink, yellow, brown, etc.) under different lighting conditions and backgrounds. Many techniques focus on only some skin types and lighting conditions. Performance is often measured in terms of true positive rate (correctly identifying skin) and false positive rate (incorrectly identifying non-skin), but evaluation across different datasets can be problematic.

For instance, the Compaq dataset consists of images from the web with diverse skin tones and lighting. Similarly, the ECU dataset includes images from digital cameras and the web, covering a wide range of skin types and lighting conditions. Some methods, like explicit thresholding, set fixed boundaries based on empirical data. While simple and fast, this method struggles with diverse lighting and background conditions.

In contrast, the histogram technique relies on the distribution of skin and non-skin colors. It can perform well but needs a large training dataset and has high storage requirements. Adjusting the number of histogram bins can improve performance but varies depending on the color space and dataset size.

Gaussian Mixture Models (GMMs) and Single Gaussian Models (SGMs) are also used. Based on the analysis done by Caetano et al on a dataset of 800 images containing people from a large spectrum of ethnic groups [13], the performance of SGM is similar to GMMs for low FPR (false positive rate), however, for medium and high TPR (true positive rate), the GMMs performs better. GMMs are more popular and perform better than SGMs due to their ability to generalize with less training data, but they are computationally expensive and slower during classification.

Phung et al performed an analysis for comparing the performance of Multilayer Perceptron (MLP) network and Bayesian, Gaussian, and Explicit threshold skin classifiers on ECU database [14]. The results of the analysis done suggest that MLP networks show similar performance to Bayesian techniques but with lower storage requirements. The choice between methods depends on factors like storage limitations and computational costs.

2.2.2 Comparison of Color Space Representations

Various color spaces are used for skin detection, each with its own advantages and limitations. A fundamental aspect of skin detection is the choice of color space representation, which significantly impacts the accuracy of the detection algorithm. Color spaces provide a mathematical model for representing colors, each with its unique way of organizing color information, and selecting the most suitable color space for skin classification is challenging.

Several comparison studies are done to evaluate the effectiveness of different color spaces for skin detection. Early research by Littman and Ritter compared the performance of neural approaches based on linear maps for skin color using RGB, YIQ, and YUV color spaces [15]. They found that performance was largely independent of color space, with neural methods performing better. Other studies, such as Zarit et al., expanded the comparison to include additional color spaces like CIE Lab, HSV, and Fleck HS [16]. These studies often used limited datasets for evaluation, making it challenging to draw definitive conclusions regarding the superiority of one color space over another. However, HS-based color spaces were often favored for their promising and better results. Terillion et al. evaluated nine chrominance spaces for skin segmentation using Single Gaussian Models (SGMs) and Gaussian Mixture Models (GMMs) [17], finding that normalized color spaces worked better with SGMs while un-normalized spaces performed comparably with GMMs. Albiol et al. provided a theoretical analysis demonstrating that for every color space, there exists an optimum skin detector with comparable performance, assuming there is an invertible transformation between the color spaces [18]. They compared skin detection performance using a Bayes classifier across three color spaces: RGB, YCbCr, and HSV. Their findings suggested similar performance across the color spaces evaluated.

To rank color spaces accurately, large datasets with diverse skin tones, lighting conditions, and backgrounds are essential. Shin et al. evaluated separability between skin and non-skin clusters across different color spaces, finding the highest separability in the RGB space [19]. The evaluation was done on nine different color spaces using metrics derived from scatter matrices and skin/non-skin histograms. They found that RGB color space exhibited the highest separability between skin and non-skin clusters, showing its effectiveness in skin detection tasks. Fu et al. compared four color spaces (RGB, HSV, YCbCr, and rg) using Gaussian Mixture Models (GMMs) for skin detection [20]. Their results indicated that HSV performed the best due to its ability to decorrelate chrominance and luminance information. However, they also highlighted the importance of considering the choice of skin modeling technique, which significantly influenced performance across different color spaces.

We can conclude by emphasizing the importance of considering various factors, such as dataset size, diversity, and choice of skin modeling technique, when evaluating color spaces for skin detection. The key criterion for skin classifier performance is the degree of overlap between skin and non-skin clusters in a color space. Non-parametric models like histogram-based Bayes classifiers are unaffected by color transformations, while parametric models like Gaussian modeling are influenced by the choice of color space.

RGB Color Space

The RGB (Red, Green, Blue) color space originated from applications like CRT displays, where colors are described as combinations of three primary-colored rays: red, green, and blue. It's one of the most commonly used color spaces for processing and storing digital image data due to its simplicity. However, despite its popularity, RGB has several limitations that make it less than ideal for certain color analysis and recognition tasks.

One significant drawback of RGB is the high correlation between its color channels. This correlation can make it challenging to separate color information accurately, especially in situations where distinct colors need to be differentiated precisely. Additionally, RGB suffers from perceptual non-uniformity, meaning that equal changes in RGB values may not result in equal perceptual changes in color. This inconsistency can lead to inaccuracies in color-based algorithms and analyses.

Another issue with RGB is the mixing of chrominance and luminance data within the same color space. Chrominance refers to the color information, while luminance represents the brightness or intensity. Mixing these two types of data in a single-color space can complicate color analysis and make it less straightforward to extract meaningful color features.

Normalized RGB

To address some of the limitations of traditional RGB, researchers have explored normalized RGB representations.

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B}$$

Normalized RGB is obtained by applying a simple normalization procedure to the RGB values, ensuring that the sum of the three normalized components equals 1 ($r + g + b = 1$). This normalization effectively reduces the dimensionality of the color space by omitting the third component, which does not carry significant information. One notable property of normalized RGB is its invariance to changes in surface orientation relative to the light source, under certain assumptions. This property makes normalized RGB particularly useful for analyzing matte surfaces where ambient light can affect color perception [21]. Additionally, the transformation simplicity of normalized RGB has contributed to its popularity among researchers for various color-based applications [22], [16], [23], [24].

HSV

Hue-saturation based color spaces were developed to provide a way for users to specify color properties numerically. These color spaces describe color with intuitive values, based on the artist's idea of tint, saturation, and tone. Hue represents the dominant color (such as red, green, purple, and yellow), while saturation measures the colorfulness of an area in proportion to its brightness [25].

The separation of color properties, such as luminance and chrominance, in hue-saturation color spaces made them popular for skin color segmentation tasks. Researchers found these color spaces effective for discriminating between different colors, leading to their widespread use in skin color segmentation studies [16], [26], [27], [28], [29].

Some properties of hue were highlighted in the analysis done in multiple research projects. For instance, hue is invariant to highlights at white light sources and, for matte surfaces, to ambient light and surface orientation relative to the light source [21]. However, there are also criticisms of hue-saturation color spaces. Poynton (1995) points out undesirable features such as hue discontinuities and conflicts in computing "brightness" (i.e. lightness, value) with the properties of color vision [25].

An alternative approach to computing hue and saturation was introduced by Fleck et al. [30]. They proposed using log opponent values to compute hue and saturation, aiming to reduce the dependence of chrominance on the illumination level.

One challenge with hue-saturation color spaces is their polar coordinate system, resulting in the cyclic nature of the color space. This cyclic nature can be inconvenient for parametric skin color models that require tight clusters of skin colors for optimal performance. To address this challenge, Brown et al. proposed a different representation of hue-saturation using Cartesian coordinates [22]. This representation offers a more linear and structured approach to hue and saturation, potentially improving the performance of parametric skin color models.

Additionally, the conversion from RGB to the HSV color space is made easier with the formulas provided by Burger & Burge [31]. We start by determining the saturation of the RGB color components R, G, B within the range of [0, C_{max}], where C_{max} typically equals 255. This saturation is calculated as:

$S_{HSV} = \frac{C_{rng}}{C_{high}}, \; V_{HSV} = \frac{C_{high}}{C_{max}}$

Where:

$C_{rng} = C_{high} - C_{low}$,

$C_{low} = min\ (R, G, B),$

$C_{high} = (R, G, B)\ = R\ (for\ all\ cases\ considered\ for\ skin - based\ analysis)$

As a result of the formulas provided by Burger & Burge [31], we can compute the preliminary Hue as:

$H' = \frac{G-B}{R-B}$,

$H = \frac{1}{6}H'$

for the cases where $0 \leq H \leq 1$, and the approximate estimation of H ≈ 21 for the cases where $0 \leq H \leq 255$. Further explanation on the steps for achieving this will be discussed during this paper.

3. Problem Statement

The research and implementation of a face-recognition-based Attendance System in our project is defined using several steps. During our research, we have proposed the following system: We are going to conduct face recognition-based attendance checks in different small and large classrooms in universities. We assume that all lecture rooms have appropriate setups for taking several pictures during classes without disturbing the lecture process. Multiple pictures are taken from time to time during classes, and attendance checks will be performed based on these images. The classrooms have free entrance; no ID scanning machines, or registration procedures are available here. The purpose of our research is the implementation of an attendance system using different methods and tools. The core purpose of the research is finding the optimal way to do the attendance check in the classrooms regardless of conditions, the seat layouts, and other factors that can impact on the face-recognition-based attendance system. During the research, several questions are under study, including the optimal position of the camera for taking pictures, the optimal number of images needed for attendance checks, the frequency of taking pictures, the brightness of the room, and the seat layout of students. These questions will be answered later during the research.

4. Method Description

4.1 Face-specific Colors in RGB

Based on the analysis done in FLID Color Based Iterative Face Detection by S. Khachatryan [34], we aim to find a specific range for Hue.

First, we collect pixels from facial regions, focusing on collecting unique colors. Once we have collected the unique colors, we group them based on their Green (G) value. This will result in 256 groups.

Additionally, for each color compute $M = \frac{1}{2}(R + B)$, having that $G = \frac{1}{2}(R + B)$, $B < G < R$. In each group $Green = x$ we compute $Min\ \{M\}$, $Max\ \{M\}$, $Mean\{M\}$, $Var\{M\}$, then we draw $Min\{M\}$, $Mean\{M\} - Var\{M\}$, $Mean\{M\}$, $Mean\{M\} + Var\{M\}$, $Max\{M\}$.
The next step is to apply the quadratic regression to each of these 5 curves.

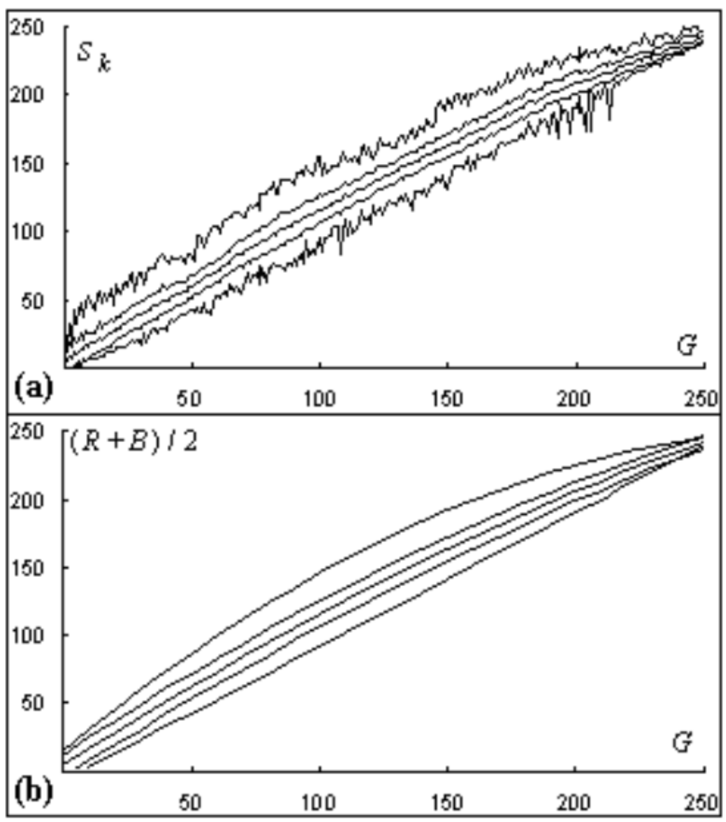


Figure 1 (a) $s_k(G)$ series, $0 \le k \le 4$. The curves are ordered from $k = 0$ at the bottom to $k = 4$ at the top; (b) the face loop in 2D space of $(G, (R + B) / 2)$

All face-related colors satisfy $s_0(G) \leq M \leq s_4(G)$, (1)

Table 1. Quadratic approximations to $s_k(G)$ series.

	Trend line	**R^2**
s_0	$0.9848\,G - 6.7474$	99.63%
s_1	$-0.0009\,G^2 + 1.1917\,G - 4.0146$	99.98%
s_2	$-0.0011\,G^2 + 1.2262\,G + 4.0264$	99.96%
s_3	$-0.0013\,G^2 + 1.2608\,G + 12.067$	99.91%
s_4	$-0.0026\,G^2 + 1.5713\,G + 14.8$	98.86%

4.2 Face-specific Colors in HSV

Now using the RGB results we are going to connect it with HSV and find a range for hue: $G = \frac{1}{2}(R + B), B < G < R.$

1. $V = \frac{Max(R,G,B)}{255} = \frac{R}{255}$

2. $S = \frac{Max(R,G,B) - Min(R,G,B)}{Max(R,G,B)} = \frac{R-B}{R} = 1 - \frac{B}{R}$

3. $H' = B' - G' = \frac{R-B}{R-B} - \frac{R-G}{R-B} = \frac{G-B}{R-B}$

$G = \frac{1}{2}(R + B) \Longrightarrow H' \approx \frac{\frac{1}{2}(R+B)-B}{(R-B)} = \frac{1}{2}$

$$H = \frac{1}{6}H' \Longrightarrow H = \frac{1}{12}, if\ 0 \leq H \leq 1$$
$$if\ \ 0 \leq H \leq 255 \Longrightarrow H \approx \frac{255}{12} \approx 21$$
$$(1) \Longrightarrow 3 \leq H \leq 24$$

Finally, we can conclude that the desirable range for hue is [3, 24].

Given the evaluation of color spaces presented in the chapter, our project opts for the HSV color space over RGB for skin detection. The decision is motivated by HSV's ability to separate color information from brightness, making it more robust to changes in lighting conditions. This choice is crucial for achieving accurate skin detection across diverse environments, which is essential for applications like face detection. Additionally, dropping the luminance component in some methods may degrade performance, highlighting the importance of considering all components in color space representations.

4.3 Datasets Used

This section is dedicated to describing the datasets used in our analysis. We utilized three distinct sources to gather a diverse set of facial images for training and testing our face recognition system.

4.3.1 FEI Face Database

The FEI Face Database, hosted at https://fei.edu.br/~cet/facedatabase.html, served as a foundational dataset for our research. The FEI Face Database comprises a collection of facial images captured at the Artificial Intelligence Laboratory of FEI in São Bernardo do Campo, São Paulo, Brazil. Each of the 200 individuals in the database is represented by 14 images, resulting in a total of 2800 images, with a balanced gender distribution. These images were captured against a uniform white background, in an upright frontal position with profile rotation of up to about 180 degrees, with the original size of each image being 640x480 pixels. The individuals included in this dataset are aged between 19 and 40, showcasing diverse appearances, hairstyles, and accessories, featuring both neutral and smiling expressions.

4.3.2 American University of Armenia (AUA) Classroom Dataset

The AUA Classroom Dataset was sourced internally from the university's repository at https://drive.google.com/drive/folders/1N4Bod9GzOVAljoZhkjx2X8pz7ECuyah4?usp=share_link . This dataset was specifically created to include images of individuals seated in classrooms during different events held in various rooms across the university campus. The dataset offers a real-world representation of facial images captured in diverse environmental conditions, including variations in lighting, background, and pose.

Before conducting analyses, all datasets underwent thorough preprocessing procedures to standardize image formats, and compression of high-resolution photos, as large file sizes can significantly slow down image processing tasks. These steps not only speed up the process but also reduces computational resource demands, ensuring better results.

4.3.3. Web-Scraped Images

In addition to the previous datasets, we supplemented our analysis with a collection of facial images sourced from various websites. These pictures were taken from various sources on the internet, showing different types of people in different poses and backgrounds.

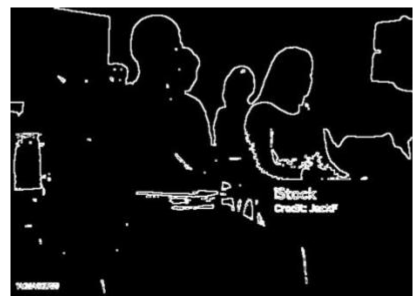
5. Experiments

5.1 The classroom dataset

We initiated our experiments by exploring edge detection techniques available in ImageJ. This involved converting colored images to black and white and working with grayscale representations to highlight image outlines. Converting images to black and white makes visualization of edges easier, simplifying subsequent processing steps by reducing data complexity and computations. Our initial attempts had promising results, laying the groundwork for further analysis.

Picture-1
Original Image

Picture-2
Image after converting to 8-bit picture and applying threshold



Picture-3
Image after applying "Find Edges" feature of ImageJ

Recognizing the limitations of traditional edge detection methods in addressing color and shape similarity issues, we concluded that edge detection alone could not help us and turned our attention towards exploring the feasibility of utilizing skin classifiers to distinguish skin tones from other colors in an image. Our objective was to leverage these classifiers to enhance the accuracy of person detection by effectively isolating individuals' faces from background elements.

In our search for finding a suitable skin classifier, we conducted research to identify existing algorithms and methodologies. This involved studying literature, academic papers, and open-source implementations to understand the underlying principles and performance metrics associated with various skin classification techniques. To evaluate the effectiveness of different skin classifiers, we established criteria based on factors such as accuracy, computational efficiency, lighting conditions, and adaptability to diverse skin tones. Despite our efforts with skin classifiers, we encountered persistent challenges in achieving consistent and satisfactory results. As a result, we explored alternative techniques and discovered the potential of color thresholding using the "Color Threshold" feature in ImageJ. Upon identifying a promising skin classifier, we integrated it into our workflow to enhance the accuracy of person detection. This involved preprocessing the images to isolate regions classified as skin tones, effectively reducing the impact of background color similarity on the overall detection performance. By leveraging the Hue, Saturation, and Brightness (HSB) color space and adjusting the corresponding thresholds, we observed significant improvements in isolating relevant objects within the images. This approach allowed us to selectively retain regions corresponding to people's faces while filtering out unwanted background elements.

Picture-4
Examples of isolating people's faces using "Color Threshold" method

5.2 Experiments on facial dataset

5.2.1 Analyzing Hue, Saturation, and Brightness Behaviors for Person Detection

With the goal of creating a method for spotting people in classrooms by studying facial colors, we closely examined how colors change in faces using three main features: Hue, Saturation, and Brightness. Our objective was to find patterns and correlations among these features to make our person detection method more accurate. The dataset used in this stage of analysis was the FEI Face Database, using only the frontal images of each individual with neutral or non-smiling expression.

To study the data and facilitate our analysis, we created a structured table to document the behaviors of Hue, Saturation, and Brightness across different face images using ImageJ. This helped us spot trends and connections between these features, which was crucial for creating a better way to formulate a detection strategy to find and isolate people.

#	Hue	Saturation	Brightness
1	25	3-90	25-80
2	23	3-55	60-100
3	24	3-100	51-105
4	22	3-95	0-70
5	24	3-90	0-90
6	24	3-70	0-100
7	23	3-85	0-100
8	23	3-80	0-80
9	33	3-95	0-80
10	24	3-105	0-85
11	24	3-70	90-110
12	28	3-120	30-80
13	18	3-70	30-80
14	24	3-90	40-100
15	23	3-75	80-115
16	24	3-120	0-60
17	23	3-95	55-90
18	23	3-90	0-90
19	24	3-95	0-90
20	23	3-115	0-85
21	23	3-85	85-115
22	25	3-100	0-80
23	23	3-80	40-85
24	25	3-100	0-90
25	26	3-100	0-80
26	25	3-115	0-90
27	23	3-70	50-100
28	21	3-80	25-130
29	25	3-95	60-105
30	25	3-95	0-110
31	24	3-75	0-100
32	21	3-110	0-110
33	23	3-95	0-85
34	23	3-70	60-100
35	25	3-80	30-100
36	25	3-90	50-100
37	22	3-70	50-125
38	19	3-100	55-110
39	24	3-100	30-105
40	27	3-125	0-85
41	21	30-100	30-100
42	25	3-120	0-100
43	25	3-75	90-120
44	24	3-100	0-90
45	24	3-90	55-90
46	21	3-80	50-100
47	21	3-90	0-110
48	21	3-80	0-90
49	23	3-70	0-90
50	19	3-70	60-110

Table-2
List of analysis of Hue, Saturation, and Brightness done on 50 pictures

Picture-5
Visualization of the results of the values obtained in Table-2

Through the examination of the dataset and analysis table, we made several key observations:

- **Consistent Hue:** Hue values exhibited a consistent range across various face colors, with an average interval identified for frontal face images.
- **Saturation and Brightness Changes:** Saturation and Brightness values showed some variations influenced by factors such as lighting conditions, skin tones, hair amount and color, etc.
- **Difficulty Setting Thresholds:** Establishing precise thresholds for Saturation and Brightness was challenging due to the diverse nature of classroom environments and individual characteristics.

Through our testing, we developed a simple ImageJ Plugin under the class named "HueSpecific". This plugin aims to target only specific hues within the RGB color space, for us to analyze the behavior of Hue, and perform hue-based isolation and filtering. In this stage of testing, we specified the Hue range as 0-25 based on the results of the obtained Table-2.

Picture-6
Results after applying only Hue filter

Understanding the relationship between hue, saturation, and brightness is crucial for accurate face detection. To explore this relationship, we used Microsoft Excel to analyze the behavior of Hue, Saturation, and Brightness with graphs and lines for each component resulting from the tests done on 25 photos from the FEI Face Database. This process involved systematically examining each photo within the established hue range and calculating key statistics such as maximum, minimum, mean, and variance for saturation, disregarding brightness to solely focus on understanding the distribution of saturation concerning hue. When testing this on the facial dataset, we specifically targeted the hue range of 0-25. The resulting analysis was then put in Microsoft Excel, for further analysis, which revealed a clear downward trend in the distribution of saturation across the hue range, as depicted in the following graph. These findings provided valuable insights into the relationship between hue and saturation.

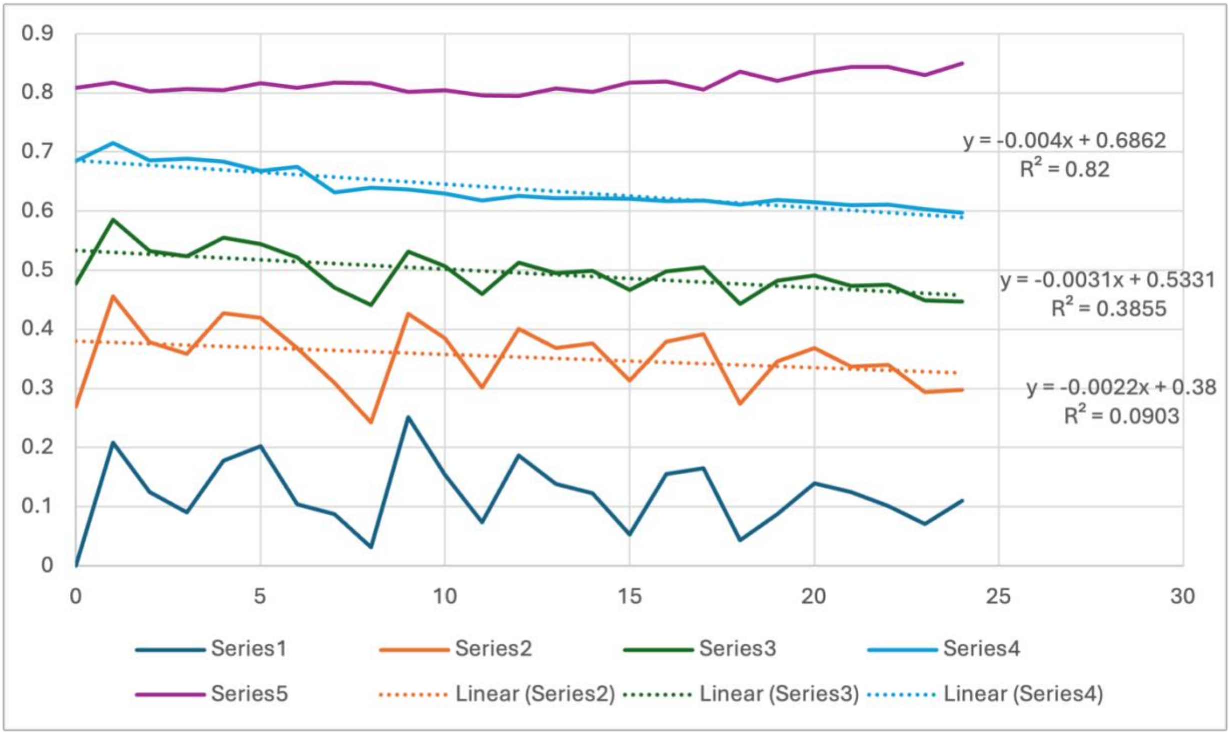


Chart-1
Analysis done on Excel on
maximum, minimum, mean, and variance values for saturation

To analyze our findings, we implemented a new code called the "Face_Colors_Filter_HSV," which included a saturation condition. When reanalyzing the face data using this code, it was clear that there was a connection between hue and saturation. However, we faced challenges as some face pixels couldn't be recognized, leading us to investigate how color and brightness are related. By recognizing the importance of understanding how colors and brightness affect each other, we focused our examination on the hue-brightness and saturation-brightness relationships and developed a new approach. We developed a new class named "Face_Color_Sample_HSV" for this. This approach involved calculating multiple possible relations for saturation and brightness, enabling a deeper understanding of their roles. Excel was again used to analyze the results, and to help us find patterns and relations for hue, saturation, and brightness, focusing mostly on finding a range for brightness values. We were able to conclude that the brightness values for skin colors are ranging between 0-0.8.

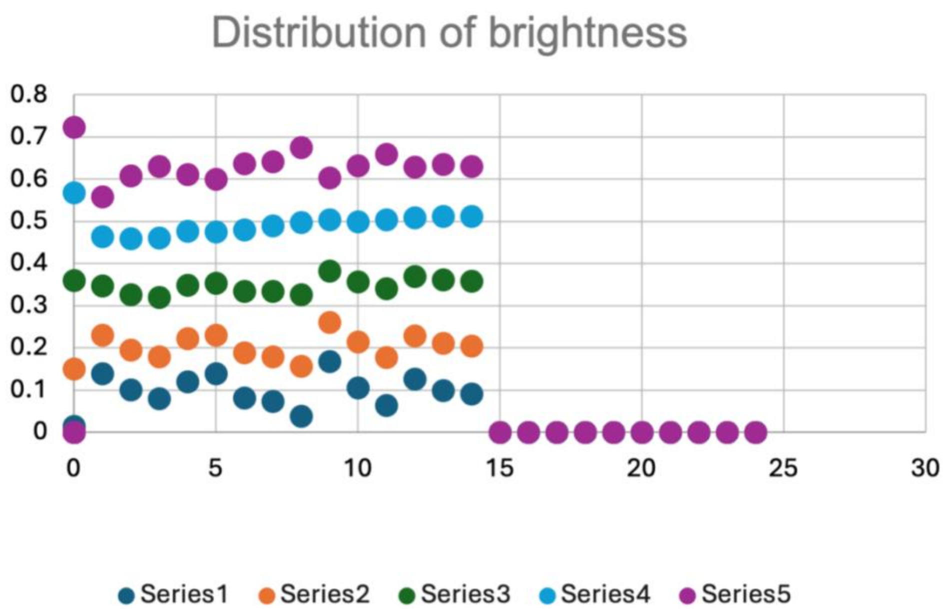


Chart-2
Visualization of brightness value distribution among 25 facial images

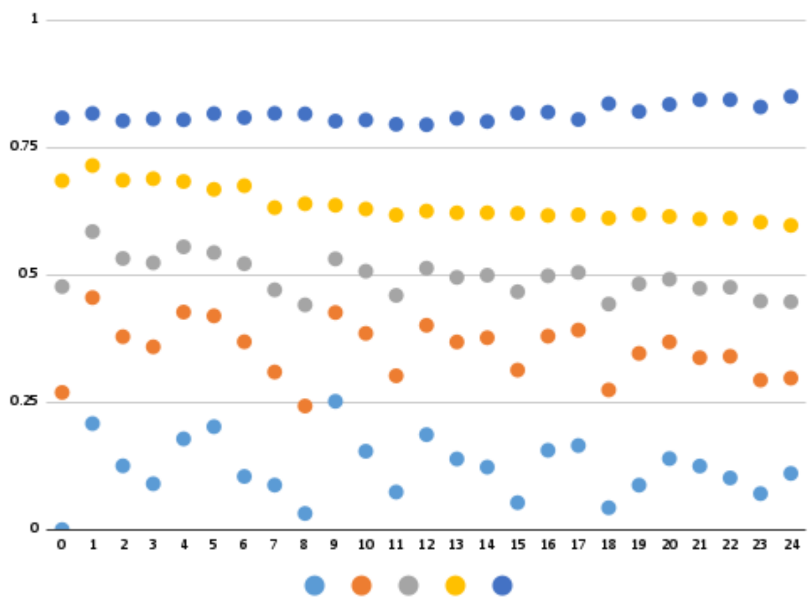


Chart-3
Visualization of brightness value distribution among 25 facial images

After some analysis, it was concluded that Hue alone cannot identify and isolate facial pixels only. In addition to this, combining saturation and brightness values could better represent skin tone intensity, and this idea became the foundation for our approach.

We developed the ImageJ Plugin "Face_Color_Filter_HSV_White" to experiment and analyze the behavior of all HSV color features together. In this code, we used a fixed hue range of 3-25, while saturation and brightness values were subject to change during our testing. We considered multiple ranges including 0.1-0.8, and 0.35-0.65 for saturation, and ranges 0.1-0.65 and 0.2-0.5 for brightness/value which worked best among others. All the pixels that were excluded from these ranges were colored white, isolating the pixels we were looking for.

Picture-7
Results after applying "Face_Color_Filter_HSV_White" on facial dataset with ranges: Hue: 3-25, Saturation: 0.1-0.8, Brightness: 0.1-0.65

Picture-13
Results after implementing "Face_Color_Filter_HSV_White" Plugin on facial dataset with ranges other than the ones obtained from analysis, which resulted in crucial facial pixels loss

5.2.2 Breakdown of the "Face_Colors_Filter_HSV_Hist" Class

We then developed the "Face_Colors_Filter_HSV_Hist" class to identify and filter skin tones within images. This class utilizes a histogram of hue-saturation values (HSV) to pinpoint the most common HSV values corresponding to human skin tones in classroom environments. This class is a tool we created to analyze skin tones in image by generating a histogram of saturation-value (SV) products for skin-tone colors identified based on their hue values. It processes RGB images and converts them to the HSV color model to isolate hues that are typical for human skin. The implementation was initially done by formulating the Hue, Saturation, and Brightness values separately, however, this resulted in a dead-end. This was a good reason for us to start finding a pattern and relation between two or all three values, and after some testing, we reached the conclusion of using the value of Saturation and Brightness combined as multiplication. The images used in testing this class are 200 pictures from the dataset of frontal face images by FEI Face Database.

We start by taking an image and go through each pixel. We then convert the pixel's color from red-green-blue (RGB) to hue-saturation-brightness (HSV). A conditional statement checks if the hue (normalized to 256 levels) is within the typical skin tone range (3 to 24). If it is, the saturation (S) and brightness/value (V) are multiplied and accumulated into a histogram array based on their product. The histogram “hist” stores the frequency of each SV product occurring within the skin tone hue range. This histogram is used to analyze the distribution of skin-tone intensity values across the image. We use math to find a smooth curve that fits the distribution of skin tones in the image. For this, we implemented the “quadReg” function which computes a quadratic regression on the cumulative histogram to find a parabolic fit to the data. This step is crucial as it helps in smoothing the histogram and identifying key points where the distribution changes significantly, which can be indicative of typical skin saturation and brightness levels to help identify significant points where skin tones change. This regression helps in understanding how densely the skin-toned pixels are distributed and identifies thresholds for filtering. Using the quadratic fit, the “roots” function calculates where the curve intercepts the histogram. These roots help determine significant boundaries in SV values, which can be used to isolate skin tones more accurately. Finally, the results from the quadratic regression and root finding are logged, which helps in debugging and using the results for further analysis and calculations.

After determining the relevant SV range for skin tones from the histogram, these values are used to set the thresholds in the “Face_Colors_Filter_HSV” class. We developed the “Face_Colors_Filter_HSV” class to highlight areas that match the skin tone HSV ranges obtained previously and exclude all the regions that don’t have components of skin tones.

5.2.3 Breakdown of the “Face_Colors_Filter_HSV” Class

This class uses the SV range determined by “Face_Colors_Filter_HSV_Hist” to isolate skin-tone regions in images. It filters the image based on these SV and hue thresholds to highlight potential face areas.

Similar to the “Face_Colors_Filter_HSV_Hist” class, this class operates on each pixel, converting RGB values to HSV. It then uses the thresholds for hue and the SV product determined from the histogram analysis to decide if a pixel should be considered as part of a skin tone region. For pixels within the skin tone hue and SV range, the pixel is set to a calculated value (potentially highlighting intensity based on SV), while all other pixels are set to white (background). This creates a contrast between skin-toned areas and the rest of the image, making the segmentation easier for our future steps. The application of the determined SV range is critical as it ensures that only those pixels that closely match the typical properties of skin in terms of saturation and brightness are considered, reducing false positives from objects with similar hues but different SV properties.

5.3 Post-Processing

Post HSV filtering, the image is converted to an 8-bit grayscale format. A binary threshold is then applied to segregate the skin-toned areas clearly from the background.
To enhance the binary image, several morphological operations are performed:

- **Dilation** increases the size of the object areas.
- **Erosion** removes noise around the object boundaries.
- **Median filtering** smooths the object surfaces.
- **Hole filling** ensures that the objects (faces) are solid, without gaps.

After isolating potential face regions, our next critical step involves analyzing these regions and fitting them with ellipses. This is achieved using ImageJ's "Analyze Particles" feature. The process of ellipse fitting is crucial for estimating the location and shape of each detected face within the classroom images. Here's a detailed explanation of how this process is integrated and executed in the workflow:

1. **Analyze Particles:**
    - ImageJ's "Analyze Particles" function is used to identify and measure properties of discrete objects (particles) in the binary image. The function calculates several properties for each particle, such as area, minimum, and maximum of all pixels.

2. **Fitting Ellipses:**
    - For each detected particle presumed to be a face, an ellipse is fitted. This fitting process involves calculating the best-fit shapes that are close to the shape of an ellipse. ImageJ computes the major and minor axes of the ellipse, which helps us in understanding the orientation and aspect ratio of each ellipse drawn. Additionally, the coordinates, size, and orientation of each ellipse are recorded. This data can be used for further statistical analysis including eliminating outliers for having a more accurate result.

3. **Output and Analysis:**
    - The final output includes an annotated image displaying all detected ellipses along with a detailed report containing the metrics of each detected particle. These metrics provide insights into the number of students, their approximate face sizes, and their distribution within the classroom.

The ellipse analysis process effectively summarizes the detected facial regions into geometric shapes that are easier to quantify and analyze. By converting detected skin-toned regions into ellipses, the system can offer a structured and standardized method of interpreting the data.

6. Testing Results

This chapter presents an in-depth analysis of the testing results obtained from our face detection system, primarily using the AUA Classroom Dataset. This dataset offered a real-world representation of classroom environments, providing diverse conditions for evaluating the effectiveness of our approach. We categorized the images into two groups based on the number of students: Large and Small. We also considered factors such as background contrast, layout and camera direction, and the frequency of shots to comprehensively assess the performance and challenges of our system.

6.1 Image Grouping: Large vs. Small Classrooms

The images were categorized into two groups based on the number of students present:

- **Large Classrooms:** These images included a higher number of students, which posed a challenge for the face detection system due to the increased number of faces and potential overlaps.

Picture-15
Tests done on large classroom with higher number of students faces which resulted in less accurate detection with multiple faces missing in detection

- **Small Classrooms:** These images had fewer students, providing clearer views of individual faces and less overlaps, which generally resulted in more accurate detection.


Picture-16
Tests done on small classroom with clearer views of students faces which resulted in a more accurate detection

In Large classrooms, our system sometimes struggled with accurately isolating individual faces, especially in crowded scenes where students were seated close to each other. Overlapping faces and varied lighting conditions further complicated the detection process. In contrast, small classrooms resulted in higher accuracy as the system could easily identify and isolate faces with fewer overlaps and distractions.

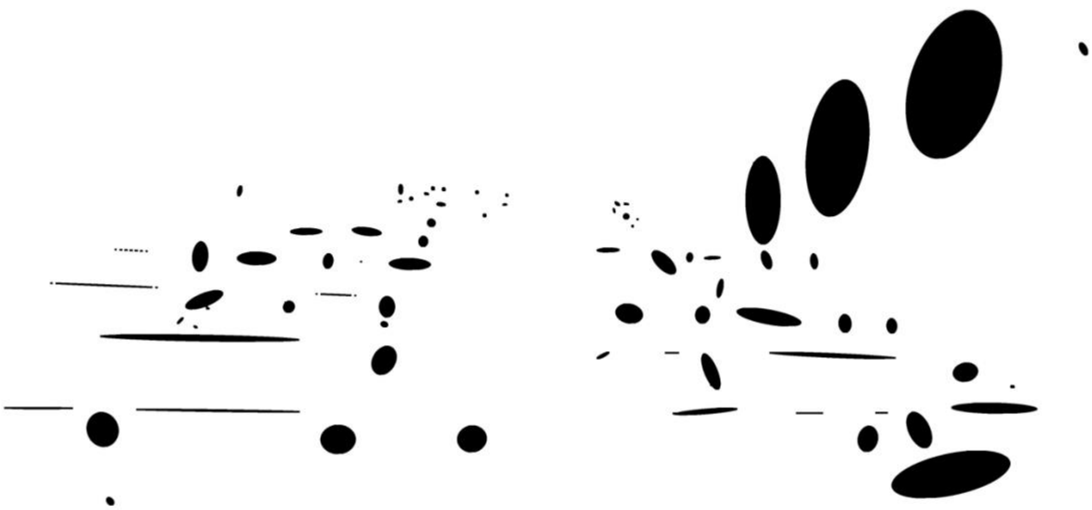
6.2 Background Contrast

The background of the images played a crucial role in the accuracy of our face detection algorithms. We classified backgrounds into two categories: contrasting and non-contrasting.

Non-Contrasting Backgrounds: Images where the background color and texture closely resembled the students' skin tones were particularly challenging. Classrooms with backgrounds that closely matched the skin tones of the students, such as beige or wooden walls, tables, curtains etc. led to numerous false positives and missed detections as the face detection algorithm struggled to differentiate between faces and background elements.

Picture-17
Original Image

Picture-18
Detection accuracy dropped with several false positives due to stairs and curtains colors being close to skin color and texture



Picture-19
Results after post-processing for removing the false positive from the detected areas.

Contrasting Backgrounds: These images had a clear distinction between the foreground (students) and the background. For instance, a dark classroom with well-lit faces, or a classroom with a plain white wall as the background provided a contrast to the skin tones, resulting in more accurate face detection.

Picture-20
Original Image

Picture-21
Same classroom as the Picture-17 with a different lightning, resulting in a higher accuracy with less false positives

6.3 Layout and Camera Direction

The layout of the classroom including the arrangement of desks and chairs, and the direction of the camera significantly influenced the detection results. We experimented with various camera angles and positions to understand their impact on face detection.

- **Frontal Layout**: Images taken from the front of the classroom, directly facing the students. This layout generally provided the best results as the faces were fully visible and oriented towards the camera.

Picture-22
A frontal shot of a classroom with a good layout which results in less potential overlaps due to the classroom being equipped with "Learning Stairs".
All the faces were correctly identified with some false positives

- **Side Layout**: Images captured from the side of the classroom, resulting in profiles or partial views of faces. This layout presented challenges in detecting faces, especially when students were looking away from the camera.

Picture-23
Example of overlapping faces which dropped the accuracy of detection as some faces are partially visible and difficult to detect

- **Back Layouts**: Images taken from back of the classroom, capturing only the back of students without their faces. These layouts provided the worst results since no face was visible and the face detection algorithm failed by mostly detecting objects other than students when looking for skin-colored objects.


Picture-24
A back shot capturing students from back, where the detection accuracy is low, depending on the visibility of faces.

6.4 Frequency of Shots

The frequency of image captures during the class session also affected the detection outcomes. We compared results from frequently taken shots versus one-time shots.

Frequently Taken Shots: Images captured continuously or at regular intervals. These shots helped in achieving better detection rates as multiple angles and moments were captured, providing a comprehensive dataset. Additionally, students turning their heads or covering their faces might be good reasons to aim for using this option.


Picture-25
A sequence of images taken frequently using different angles, which resulted in having multiple options to use if one or many of other shots missed some faces

One-Time Shots: Single images taken at a specific moment during the class. These images provided a snapshot of the classroom at one point in time, making the detection results highly dependent on that specific moment.

Picture-26
A single shot taken during the class. Some faces are missing due to a blockage at the time of capture.

6.5 Optimal Conditions for Face Detection in Classrooms

Based on our testing and experimentation with face detection techniques, we can conclude that the optimal camera setup for classrooms is a frontal angle. In addition to the camera setup, the classroom colors and objects present are also important. Ideally, classrooms should not contain objects with colors or textures similar to skin, as these can cause false positive detections. The best classrooms for minimizing detection errors due to overlaps are those equipped with "Learning Stairs." For instance, the Small Auditorium at AUA is a good example. However, this room has several skin-colored and textured objects, such as the stairs and curtains, leading to numerous false positives. Also, we opt for using frequently taken shots to avoid the risk of possible overlaps and blockages with a single shot option.

7. Conclusions and Future Work

This project aimed to improve face detection in classrooms using image processing, mainly through the HSV color space. HSV was chosen because it makes it easier to separate brightness from face-specific hues and handle different lighting conditions. By analyzing hue, saturation, and brightness components, we were able to accurately isolate skin tones. The project's findings showed the potential of using HSV color space and image processing techniques for face detection in dynamic classroom environments. Despite the challenges, the approach demonstrated significant promise, particularly in controlled lighting conditions and with clear background separation. By leveraging the AUA Classroom Dataset, we conducted multiple testing algorithms and analysis to validate our approach, revealing both successes and challenges. Although we encountered challenges such as background color confusion, variability in classroom conditions, and the similarity of non-human objects to skin tones, our method showed promising results. Additionally, we developed custom plugins for ImageJ to help separate and analyze faces, showing that our approach could work in real-world settings.

Future work will focus on improving these techniques to solve the problems we found, such as enhancing algorithms to better differentiate between skin tones and similar-colored objects, making the system adapt to different classroom conditions, and using machine learning models to increase detection accuracy and help us in the post-processing steps. As an example of this, we can point to Piture-19 in which the faces are isolated from false positive results in Piture-18, giving us chance to use further techniques such as machine-learning to have a result in future. Collecting and using more diverse datasets and real-time processing including face detection in both captured and live video will also be crucial steps forward.

In conclusion, this project provides a foundation for better face detection in educational settings. By overcoming the challenges and building on our successes, the methods developed here can help improve monitoring and analysis in classrooms, leading to better educational experiences and outcomes.

8.References

[1] Jam, J. (2018). Face Detection and Recognition Student Attendance System. *ResearchGate*. https://www.researchgate.net/publication/326986115_Face_Detection_and_Recognition_Student_Attendance_System. Retrieved from https://www.researchgate.net/publication/326986115_Face_Detection_and_Recognition_Student_Attendance_System

[2] K. David Biaru, A Model of Two-Factor Authentication using Facial Recognition in Automated Teller machines. Retrieved from http://erepository.uonbi.ac.ke/bitstream/handle/11295/76598/Kagiri_Enhancing%20community%20based%20health%20information%20system%20CBHIS%20reporting?sequence=4&isAllowed=y

[3] Rekha, A. L., & Chethan, H. K. (n.d.). *Automated attendance system using face recognition through video surveillance*. Ijtre.com. Retrieved from https://www.ijtre.com/images/scripts/2014011113.pdf

[4] Chatterjee, S., Jana, A., Ganguly, A., & Ghosh, A. (2018). Automated attendance system using face recognition technique. *International Journal of Engineering and Applied Sciences*, *5*(7). https://doi.org/10.31873/ijeas.5.7.18

[5] N. Kar, M. Kanti Debbarma, A. Saha, and D. Rudra Pal. Study of Implementing Automated Attendance SystemUsing Face Recognition Technique, International Journal of Computer and Communication Engineering, Vol. 1, No. 2, July 2012. Retrieved from https://www.researchgate.net/publication/274630584_Study_of_Implementing_Automated_Attendance_System_Using_Face_Recognition_Technique

[6] R. Abuazh, Y. Abdullah, A. Alluwaimi, Y. Aldakhail. Face Detection and Recognition Student Attendance System, Spring 2019-2020. Retrieved from https://www.pmu.edu.sa/attachments/academics/pdf/udp/coe/dept/ee/face_detection_system_report.pdf

[7] T. Shu Jing. Facial Recognition-Base Attendance Monitoring System for Educational Institution. January 2018. Retrieved from http://eprints.utar.edu.my/2861/1/CT-2018-1503979-2.pdf

[8] Sylaj, A. (2019). *Student attendance system using face recognition*. University for Business and Technology in Kosovo. https://knowledgecenter.ubt-uni.net/etd/1621/

[9] Viola, P., & Jones, M. J. (2004). Robust Real-Time Face Detection. International Journal of Computer Vision 57(2), 137–154.

[10] Pietikäinen, M. (2010). Scholarpedia. Retrieved from **http://www.scholarpedia.org/article/Local_Binary_Patterns**

[11] Rangarajan, R. (n.d.). Attendance System in Third -Level. Ncirl.Ie. Retrieved from https://norma.ncirl.ie/6641/1/rajalakshmirangarajan.pdf

[12] Kawaguchi, Y., Shoji, T., Weijane, L. I. N., Kakusho, K., & Minoh, M. (n.d.). *Face recognition-based lecture attendance system*. Psu.edu. Retrieved from **https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=4b6811cd2a7a6924fed4967c2b755c0942ca5351**

[13] T.S. Caetano, S.D. Olabarriaga, D.A.C. Barone, Performance evaluation of single and multiple-Gaussian models for skin-color modeling, SIBGRAPI02, 2002.

[14] S.L. Phung, A. Bouzerdoum, D. Chai, Skin segmentation using color pixel classification: analysis and comparison, IEEE Trans. Pattern Anal. Mach. Intell. 27 (1) (2005).

[15] E. Littmann, H. Ritter, Adaptive color segmentation: A comparison of neural and statistical methods, IEEE Trans. Neural Networks 8 (1) (1997) 175–185.

[16] B.D. Zarit, J.B. Super, F.K.H. Quek, Comparison of five color models in skin pixel classification, ICCV99, 1999.

[17] J.-C. Terillon, M.N. Shirazi, H. Fukamachi, S. Akamatsu, Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images, AFGR00, 2000, pp. 54–61.

[18] A. Albiol, L. Torres, E.J. Delp, Optimum color spaces for skin detection, ICIP01, 2001.

[19] M.C. Shin, K.I. Chang, L.V. Tsap, Does colorspace transformation make any difference on skin detection? IEEE Workshop on Applications of Computer Vision, Orlando, FL, December 2002, pp. 275–279.

[20] Z. Fu, J. Yang, W. Hu, T. Tan, Mixture clustering using multidimensional histograms for skin detection, ICPR04, 2004, pp. 549–552.

[21] SKARBEK, W., AND KOSCHAN, A. 1994. Colour image segmentation – a survey –. Tech. rep., Institute for Technical Informatics, Technical University of Berlin, October.

[22] BROWN, D., CRAW, I., AND LEWTHWAITE, J. 2001. A som based approach to skin detection with application in real time systems. In *Proc. of the British Machine Vision Conference, 2001.*

[23] SORIANO, M., HUOVINEN, S., MARTINKAUPPI, B., AND LAAKSONEN, M. 2000. Skin detection in video under changing illumination conditions. In *Proc. 15th International Conference on Pattern Recognition*, vol. 1, 839–842.

[24] OLIVER, N., PENTLAND, A., AND BERARD, F. 1997. Lafter: Lips and face real time tracker. In *Proc. Computer Vision and Pattern Recognition*, 123–129.

[25] POYNTON, C. A. 1995. Frequently asked questions about colour. In *ftp://www.inforamp.net/pub/users/poynton/doc/colour/ColorFAQ.ps.gz*.

[26] MCKENNA, S., GONG, S., AND RAJA, Y. 1998. Modelling facial colour and identity with gaussian mixtures. *Pattern Recognition 31*, 12, 1883– 1892.

[27] SIGAL, L., SCLAROFF, S., AND ATHITSOS, V. 2000. Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, 152–159

[28] BIRCHFIELD, S. 1998. Elliptical head tracking using intensity gradients and color histograms. In *Proceedings of CVPR '98*, 232–237.

[29] JORDAO, L., PERRONE, M., COSTEIRA, J., AND SANTOS-VICTOR, J. 1999. Active face and feature tracking. In *Proceedings of the 10th International Conference on Image Analysis and Processing*, 572–577

[30] FLECK, M., FORSYTH, D. A., AND BREGLER, C. 2002. Finding nacked people. In *Proc. of the ECCV*, vol. 2, 592–602.

[31] Burger, W., & Burge, M. J. (n.d.). *Digital Image Processing: An Algorithmic Introduction Using Java*. Springer Science & Business Media, 308-309

[32] V. Vezhnevets, V. Sazonov, A. Andreeva, A Survey on Pixel-Based Skin Color Detection Techniques, Proc. of the 13th GraphiCon, Moscow, 2003, pp.85-92. Retrieved from https://www.graphicon.ru/html/2003/Proceedings/Technical/paper509.pdf

[33] P. Kakumanu, S. Makrogiannis, N. Bourbakis, A survey of skin-color modeling and detection methods

[34] S. Khachatryan. FLID Color Based Iterative Face Detection, JHCR Journal of Hybrid Computing Research, pp 1-7, 2010