

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Leonardo Rodrigues Lima

VISÃO COMPUTACIONAL APLICADA PARA COMPARAÇÃO ANALÍTICA
ENTRE CORES DE DUAS IMAGENS DIGITAIS

Florianópolis

2021

Leonardo Rodrigues Lima

**VISÃO COMPUTACIONAL APLICADA PARA COMPARAÇÃO ANALÍTICA
ENTRE CORES DE DUAS IMAGENS DIGITAIS**

Trabalho de Conclusão do Curso de Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Joceli Mayer, Ph.D.

Florianópolis

2021

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

LIMA, LEONARDO

VISÃO COMPUTACIONAL APLICADA PARA COMPARAÇÃO ANALÍTICA
ENTRE CORES DE DUAS IMAGENS DIGITAIS / LEONARDO LIMA ;
orientador, Joceli Mayer, 2021.

123 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia Elétrica, Florianópolis, 2021.

Inclui referências.

1. Engenharia Elétrica. 2. Este trabalho apresenta o
projeto de uma aplicação capaz de automatizar o processo de
comparação entre cores de duas imagens digitais de forma
analítica. I. Mayer, Joceli . II. Universidade Federal de
Santa Catarina. Graduação em Engenharia Elétrica. III.
Título.

Leonardo Rodrigues Lima

VISÃO COMPUTACIONAL APLICADA PARA COMPARAÇÃO ANALÍTICA ENTRE
CORES DE DUAS IMAGENS DIGITAIS

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia Elétrica” e aceito, em sua forma final, pelo Curso de Graduação em Engenharia Elétrica.

Florianópolis, 03 de maio de 2021.



Documento assinado digitalmente
Jean Viane Leite
Data: 04/05/2021 21:08:49-0300
CPF: 003.474.909-80
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Jean Viane Leite, Dr.
Coordenador do Curso de Graduação em Engenharia Elétrica

Banca Examinadora:



Documento assinado digitalmente
Joceli Mayer
Data: 03/05/2021 17:03:29-0300
CPF: 546.833.519-00
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Joceli Mayer, Ph.D.
Orientador
Universidade Federal de Santa Catarina



Documento assinado digitalmente
Raimes Moraes
Data: 03/05/2021 20:10:02-0300
CPF: 054.997.488-10
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Raimes Moraes, PhD.
Universidade Federal de Santa Catarina



Documento assinado digitalmente
Bruno Catarino Bispo
Data: 03/05/2021 17:15:39-0300
CPF: 056.973.077-56
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Bruno Catarino Bispo, Dr.
Universidade Federal de Santa Catarina

AGRADECIMENTOS

A minha família, pelo carinho e incondicional apoio prestado durante todo o curso. A Universidade Federal de Santa Catarina, pelos cursos oferecidos com excelência na região de Florianópolis. A todos aqueles que contribuíram de alguma forma para que a conclusão deste trabalho fosse possível. Ao meu orientador, Prof. Joceli Mayer, pela sincera atenção dada ao meu trabalho e crescimento.

RESUMO

Este trabalho apresenta o projeto de uma aplicação capaz de automatizar o processo de comparação entre cores de duas imagens digitais de forma analítica, através de modelos de cores e sistema de visão computacional. Para isso discorre brevemente acerca da percepção humana das cores, para então detalhar sobre os modelos de cores estabelecidos desde a criação da CIE. Com o objetivo de automatizar o processo foi introduzido um conjunto de conceitos de visão computacional para a detecção das paletas de cores da carta padrão usada. Por fim, a aplicação foi testada em diferentes cenários com o objetivo de validar os métodos propostos.

Palavras-chave: Colorimetria. Visão computacional. Processamento digital de imagem.

ABSTRACT

This work proposes a method to automate the color quality comparison between two digital images. We employ computer vision and color models for this purpose. The human perception of colors is discussed in order to understand the color models established by CIE. A standard color standard is employed as color reference along with the proposed computer vision processes. The method is validated by applying it to various scenarios, contrasting quality from different cameras and illumination. Results illustrate the use of the method to achieve an analytical comparison of color quality and discrepancy between images acquired from different cameras and illumination.

Keywords: Colorimetry. Computer vision. Image processing.

LISTA DE FIGURAS

Figura 1 – Triângulo da visão.....	29
Figura 2 – Espectro eletromagnético.....	30
Figura 3 – Reflexão de um objeto verde	31
Figura 4 – Formação de imagem pelo olho humano.....	32
Figura 5 – Curvas de sensibilidade dos cones.....	33
Figura 6 – Sensibilidade da visão escotópica $V'(\lambda)$ e fotópica $V(\lambda)$	34
Figura 7 – CIE RGB.....	35
Figura 8 – Funções de correlação de cores CIEXYZ.....	36
Figura 9 – Diagrama de cromaticidade xy	38
Figura 10 – Representação cartesiana do espaço CIELAB.....	39
Figura 11 – Azul escuro e vermelho escuro	41
Figura 12 – Matiz diferente, porém iluminação semelhante.....	42
Figura 13 – Espaço de cor sRGB	46
Figura 14 – Etapas de um sistema de visão computacional	48
Figura 15 – Representação visual em escala de cinza de uma imagem	49
Figura 16 – Componentes de uma câmera digital	52
Figura 17 – Detalhe interno do conjunto de lentes.....	52
Figura 18 – Método de limiarização global.....	57
Figura 19 – Método de limiarização adaptativa	58
Figura 20 – Erosão: (a) Imagem original; (b) Imagem resultante.....	60
Figura 21 – Dilatação: (a) Imagem original; (b) Imagem resultante.....	61
Figura 22 – Carta de cores.....	63
Figura 23 – Conversão para uma escala de cinza.....	65
Figura 24 – Imagem original.....	65
Figura 25 – Imagem após a conversão	66
Figura 26 – Filtro gaussiano.....	66
Figura 27 – Após filtragem usando filtro gaussiano 5x5	67
Figura 28 – Limiarização adaptativa.....	67
Figura 29 – Erosão e dilatação	68
Figura 30 – Após erosão.....	68

Figura 31 – Após dilatação.....	69
Figura 32 – Transformada de Hough	70
Figura 33 – Após transformada de Hough	70
Figura 34 – Exemplo da função <i>approxPolyDp</i>	71
Figura 35 – Aproximação de contornos	71
Figura 36 – Após a aproximação de contornos com $\varepsilon = 1\%$	71
Figura 37 – Três regras para filtrar contornos indesejados	72
Figura 38 – Imagem após aplicar as três regras	72
Figura 39 – Centro dos contornos identificados.....	73
Figura 40 – Paleta ciano é a origem para percorrer a carta	74
Figura 41– Fluxograma para percorrer a carta de cores	75
Figura 42 – Todos os centros das paletas definidos	75
Figura 43 – Código da função <i>srgb_para_xyz</i>	76
Figura 44 – Código da função <i>srgb_para_xyz</i>	77
Figura 45 – Código da função <i>deltaE_2000</i>	78
Figura 46 – Processo completo	81
Figura 47 – Imagem original do caso 1, carta representa 30% da imagem.....	83
Figura 48 – Foram encontrados os 24 contornos no caso 1 no dispositivo X.....	83
Figura 49 – Foram encontrados os 24 contornos no caso 1 no dispositivo Y.....	83
Figura 50 – Foram encontrados os 24 contornos no caso 1 no dispositivo Z	84
Figura 51 – Foram encontrados os 24 contornos no caso 2 no dispositivo X.....	84
Figura 52 – Foram encontrados os 24 contornos no caso 2 no dispositivo Y.....	85
Figura 53 – Foram encontrados os 24 contornos no caso 2 no dispositivo Z	85
Figura 54 – Foram encontrados os 24 contornos no caso 3 no dispositivo X.....	86
Figura 55 – Foram encontrados os 24 contornos no caso 3 no dispositivo Y.....	86
Figura 56 – Foram encontrados os 24 contornos no caso 4 no dispositivo X.....	87
Figura 57 – Foram encontrados os 24 contornos no caso 4 no dispositivo Y.....	87
Figura 58 – Foram encontrados os 24 contornos no caso 4 no dispositivo Z	87
Figura 59 – $\Delta E = 1,6$ para a cor azul (#13)	89
Figura 60 – $\Delta E = 0,5$ para a cor azul (#13)	89
Figura 61 – $\Delta E = 2,0$ para a cor cinza (#22)	89
Figura 62 – $\Delta E = 0,5$ para a cor cinza (#22)	89
Figura 63 – $\Delta E = 1,8$ para a cor vermelha (#22)	89
Figura 64 – $\Delta E = 0,5$ para a cor vermelha (#15).....	89

Figura 65 – $\Delta E = 1,9$ para a cor verde (#14)	90
Figura 66 – $\Delta E = 0,3$ para a cor verde (#22)	90
Figura 67 – Imagem 1 do caso 5 usando o dispositivo do tipo X	91
Figura 68 – Imagem 2 do caso 5 usando o dispositivo do tipo X	91
Figura 69 – Imagem 1 do caso 8	93
Figura 70 – Imagem 2 do caso 8	93
Figura 71 – Imagem 1 do caso 9	96
Figura 72 – Imagem 2 do caso 9	96
Figura 73 – Histograma dos ΔE de todas as cores do dispositivo A no teste da mesma posição	100
Figura 74 – Histograma dos ΔE de todas as cores do dispositivo B no teste da mesma posição	100
Figura 75 – Histograma dos ΔE de todas as cores do dispositivo A em posições variadas	104
Figura 76 – Histograma dos ΔE de todas as cores do dispositivo B em posições variadas	104
Figura 77 – Ilustrativo do teste entre imagens de dispositivos diferentes	105

LISTA DE TABELAS

Tabela 1 – <i>White-point</i> dos iluminantes	40
Tabela 2 – Escala ΔE de percepção	41
Tabela 3 – Fatores de ponderação da fórmula CIE94	43
Tabela 4 – Filtro gaussiano 5x5	56
Tabela 5 – Paletas de cores.....	63
Tabela 6 – Caso 1 do teste de localização da carta	83
Tabela 7 – Caso 2 do teste de localização da carta	84
Tabela 8 – Caso 3 do teste de localização da carta	85
Tabela 9 – Caso 4 do teste de localização da carta	86
Tabela 10 – Caso 5 do teste usando amostras do dispositivo 1.....	92
Tabela 11 – Caso 8 do teste usando a câmera do teste 1.....	94
Tabela 12 – Resultado dos casos de comparação.....	95
Tabela 13 – Resultado (ΔE) do dispositivo A na mesma posição.....	98
Tabela 14 – Resultado (ΔE) do dispositivo B na mesma posição.....	99
Tabela 15 – Resultado (ΔE) do dispositivo A em posições variadas	102
Tabela 16 – Resultado (ΔE) do dispositivo B em posições variadas	103
Tabela 17 – Comparativo (ΔE) entre a amostra padrão com iluminância de 250 lux.	106
Tabela 18 – Comparativo (ΔE) entre a amostra padrão com iluminância de 200 lux	107
Tabela 19 – Resultado da subtração do ΔE em 250 e 200 lux	108
Tabela 20 – Resultado da subtração do ΔE em 250 e 100 lux	109
Tabela 21 – Resultado da subtração do ΔE em 250 e 50 lux	110
Tabela 22 – Resultado da razão entre a diferença de ΔE e a tolerância para o dispositivo 10 modelo A	111
Tabela 23– Resultado da razão entre a diferença de ΔE e a tolerância para o dispositivo 10 modelo B	112
Tabela 24 – Resultado da razão entre a diferença de ΔE usando D65 e D50 e a tolerância no modelo A disp. 10.....	113

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Motivação	25
1.2	Objetivo e proposta do trabalho.....	26
1.3	Organização do trabalho	27
2	PERCEPÇÃO DE CORES E MODELOS DE CORES.....	28
2.1	Introdução	28
2.2	Percepção humana	28
2.2.1	Luz visível	29
2.2.2	Objeto observado	30
2.2.3	Fisiologia do olho humano	31
2.3	Modelagem de cor	34
2.3.1	CIE RGB.....	34
2.3.2	CIE XYZ.....	36
2.3.3	Espaço CIELAB	38
2.3.4	CIE76.....	40
2.3.5	CIE94.....	41
2.3.6	CIE2000.....	43
2.3.7	sRGB	45
2.4	Conclusões.....	47
3	VISÃO COMPUTACIONAL E PROCESSAMENTO DE IMAGEM	48
3.1	Introdução	48
3.2	Imagem digital	49
3.2.1	Tipos de imagens digitais	50
3.3	Aquisição de imagem	51
3.3.1	Lentes.....	52
3.3.2	Sensores	53

3.3.3	Processador	53
3.4	Pré-processamento	54
3.4.1	Conceitos básicos.....	54
3.4.2	Filtro Gaussiano.....	55
3.5	Segmentação	56
3.5.1	Limiarização	56
3.5.2	Detecção de bordas	58
3.6	Operações morfológicas	59
3.6.1	Erosão	59
3.6.2	Dilatação	60
3.7	Conclusões.....	61
4	DESENVOLVIMENTO DE ALGORITMO DE VISÃO	
	COMPUTACIONAL PARA COLORIMETRIA.....	62
4.1	Introdução	62
4.2	Recursos.....	62
4.2.1	Carta de cores	62
4.3	Linguagem de programação Python	64
4.3.1	OpenCV	64
4.3.2	PyCharm	65
4.4	Código	65
4.4.1	Conversão em tons de cinza	65
4.4.2	Filtro Gaussiano.....	66
4.4.3	Limiarização adaptativa.....	67
4.4.4	Erosão e dilatação	67
4.4.5	Segmentação	69
4.4.6	Espaço CIELAB	76

4.4.7	CIE2000.....	77
4.5	Conclusões.....	79
5	SIMULAÇÕES E RESULTADOS.....	80
5.1	Introdução.....	80
5.2	Resultados das etapas.....	80
5.3	Localização da carta de cores.....	82
5.4	Comparação entre cores.....	88
5.4.1	Validação inicial.....	90
5.4.2	Comparação entre imagens do mesmo dispositivo.....	96
5.4.3	Comparação entre imagens de dispositivos diferentes.....	104
5.5	Conclusões.....	113
6	CONCLUSÃO.....	115
6.1	Considerações finais.....	115
6.2	Trabalhos futuros.....	115
	REFERÊNCIA.....	117
	Anexo A – FUNÇÕES IMPLEMENTADAS.....	120
	Anexo B – FUNÇÕES IMPLEMENTADAS.....	121

1 INTRODUÇÃO

Há mais sobre cores do que aparentemente nossos olhos veem. Desde que a Commission Internationale de l'Eclairage (CIE) estabeleceu a base para colorimetria moderna, os pesquisadores têm desenvolvido teorias e realizado experiências para encontrar um modelo unificado para explicar como o ser humano “vê” as cores.

A cor não é uma característica absoluta de um objeto, mas sim uma percepção humana que depende de aspectos fisiológicos e psicológicos, uma vez que “Os estímulos da cor registrados pela retina são provocados pela distribuição de energia e pelas propriedades espectrais da luz visível que passa através de um objeto ou que é refletida pelo mesmo.” (SAMPAIO; FRANÇA; BRAGA, 2007, p.543).

Em relação a colorimetria básica, pode-se dizer que ela fornece o embasamento fundamental das técnicas utilizadas para especificar os estímulos de cores ao sistema sensorial a um típico observador humano. Entretanto, com essas técnicas básicas é apenas possível especificar se dois estímulos são correspondentes ou não. Por outro lado, a colorimetria avançada tem como objetivo estender as técnicas para ser possível especificar a diferença entre dois estímulos através de modelos de aparência de cores. Esses modelos nos permitem descrever de forma matemática e objetiva os estímulos de cores em uma variedade de condições. Desse modo, são os modelos que permitem que a percepção de cores passe do âmbito subjetivo ao analítico matemático.

O atual trabalho apresenta o projeto do uso da colorimetria para determinar a diferença da percepção humana entre duas imagens digitais. Essa diferença será usada para automatizar a verificação da qualidade de imagem entre duas câmeras de segurança. Todavia, para que este projeto seja completo, é necessário o estudo de outras áreas que serão abordados nesse trabalho, tais como estrutura e anatomia do olho, fontes de luz, ótica, princípios de física, propriedades reflexivas e principalmente visão computacional, que será usada para automatizar o processo de identificação das amostras de cores.

1.1 MOTIVAÇÃO

O processo de desenvolvimento de produtos pode ser definido como um conjunto de atividades por meio das quais se busca, a partir das necessidades do mercado, restrições tecnológicas e considerando as estratégias da empresa, chegar as especificações de projeto de um produto e, de seu processo de produção, para que a manufatura seja capaz de produzi-lo

(ENESEP, 2008). Esse trabalho consiste em desenvolver uma aplicação capaz de garantir a principal especificação (imagem) de uma câmera digital durante o processo produtivo.

Câmeras digitais são dispositivos que têm como principal função gerar uma imagem digital, sendo que o processo de definição de cores padrão de uma câmera é estabelecido através de uma pesquisa de mercado durante a etapa de desenvolvimento do produto. Essa pesquisa define uma amostra padrão que deve ser seguida em todos os dispositivos produzidos até o fim do ciclo de vida do modelo.

Portanto, é esperado que amostras do mesmo modelo de câmera tenham o mesmo padrão de cores a fim de assegurar a qualidade ao cliente definida na etapa de desenvolvimento. Contudo no processo produtivo desses dispositivos, componentes críticos que podem impactar na cor da imagem gerada, como sensores e lentes, podem apresentar diferenças entre dispositivos. Essas diferenças são causadas por dois motivos: da própria variante do processo produtivo desses componentes ou do fato de terem sido adquiridos de fornecedores diferentes. O segundo motivo, apesar de não ser a prática ideal para itens críticos, é comum em função do risco de ter apenas um fornecedor para um determinado componente, uma vez que qualquer impacto em sua produção provocará um efeito cascata que irá afetar na produção do dispositivo.

Tendo isso em vista, por questão de padrão de qualidade do produto, é necessário que mesmo com itens diferentes, modelos idênticos tenham a mesma performance de cor. Sendo que, a cor ter a mesma performance representa ser imperceptível para um ser humano diferenciar as imagens entre dois dispositivos do mesmo modelo independentemente do lote de fabricação. Portanto há a necessidade da criação de um método que, garanta que durante todo o ciclo de vida produtivo do dispositivo, o padrão de cores definido durante o processo de desenvolvimento seja seguido.

1.2 OBJETIVO E PROPOSTA DO TRABALHO

O objetivo desta pesquisa é desenvolver uma aplicação capaz de fazer a comparação de cores entre duas imagens digitais de dispositivos diferentes de forma analítica e automatizada. Essa comparação deve levar em conta a percepção humana e não apenas os dados digitais e deve ser feita analiticamente e não de forma subjetiva.

Essa aplicação pode ser usada para assegurar a qualidade dos dispositivos produzidos pelo processo de manufatura, fazendo a comparação entre a amostra padrão definida no desenvolvimento do produto e o dispositivo proveniente da produção em massa.

1.3 ORGANIZAÇÃO DO TRABALHO

O trabalho desenvolver-se-á em seis capítulos, como descrito a seguir.

O Capítulo um descreve a motivação do trabalho e introduz o tema que será discorrido.

O Capítulo dois descreve o processo de percepção de cores dos seres humanos com o objetivo de demonstrar a subjetividade da reação de um indivíduo aos estímulos às cores. Além disso, são apresentados os conceitos fundamentais de cores, que serão a base para todo o estudo exposto neste trabalho. Baseado nesse processo, são descritos modelos matemáticos de cores, entre eles o CIELAB, sRGB e CIE2000, que serão imprescindíveis em nossa aplicação.

O Capítulo três faz um breve apanhado das bases teóricas necessárias para desenvolver o sistema de visão de máquina que é capaz de reconhecer a carta de cores. Esse tema é abordado desde a aquisição até a segmentação da imagem.

O Capítulo quatro descreve a implementação dos conceitos tratados nos Capítulos dois e três para desenvolver uma aplicação que reconheça a carta de cores e que seja capaz de medir a diferença de cores entre duas imagens usando o modelo CIE2000.

O Capítulo cinco apresenta e discute as simulações e os resultados obtidos pela aplicação desenvolvida em diferentes cenários e dispositivos.

2 PERCEPÇÃO E MODELOS DE CORES

2.1 INTRODUÇÃO

A percepção que um indivíduo tem de uma determinada cor nunca é isolada sobre ela, pois é sempre em relação a um contexto. “Assim, a reação do indivíduo à cor é particular e subjetiva, dado que cada um tem vivências e culturas diferentes, além disso, biologicamente ninguém tem exatamente o mesmo número de células perceptivas no olho e muito menos as têm dispostas da mesma maneira” (DELTA, 2014, p.11).

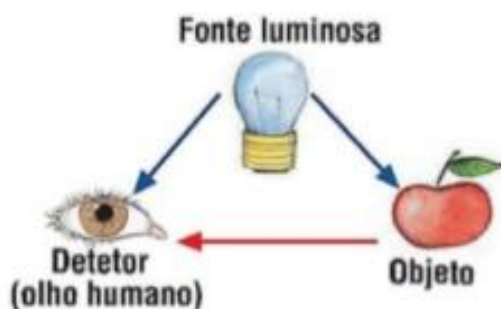
A fim de uma melhor compreensão da colorimetria, na seção 2.2 será exposto o processo de percepção das cores dos seres humanos, com o objetivo de descrever sua subjetividade e demonstrar a necessidade de um modelo analítico.

Na seção 2.3, serão apresentados os modelos analíticos e explicados os conceitos fundamentais das cores, que serão a base para a aplicação desenvolvida neste estudo. Também serão abordados, de forma evolutiva, os principais modelos de cores até chegarmos ao modelo CIE2000.

2.2 PERCEPÇÃO HUMANA

O processo de percepção visual é composto por três fatores: fonte, objeto e olho humano, formando o triângulo da visão (MEYER; ZOLLINGER, 1998), como pode ser notado pela Figura 1. O primeiro requisito é uma fonte de radiação visível eletromagnética que será modulada por um objeto devido as suas características físicas e químicas. Essa radiação então será detectada pelo olho humano e processada pelo sistema neural para produzir a percepção visual de cores.

Figura 1 – Triângulo da visão



Fonte: <https://scientificusblogpt.wordpress.com/>. Acesso em: 2 set. 2020.

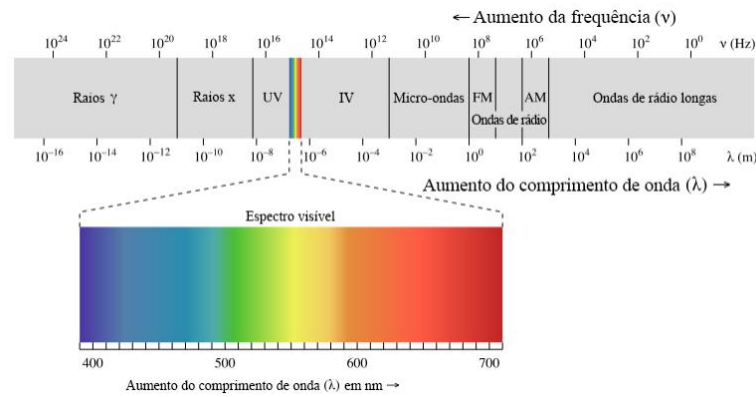
2.2.1 Luz visível

O primeiro componente do triângulo da visão é a fonte luminosa, responsável por fornecer a energia eletromagnética. O espectro de radiação eletromagnética, que é o conjunto de todas as possíveis frequências de ondas eletromagnéticas que existem, se estende desde as baixas frequências das ondas de rádio até as altas frequências correspondentes aos raios gama (OKUNO; YOSHIMURA, 2010).

Esse espectro eletromagnético é dividido em bandas, seguindo o tipo de utilização ou os efeitos causados, como exposto na Figura 2. O espectro de luz visível corresponde a banda visível pelo olho humano situada entre 400 THz e 790 THz, mas esse limite pode variar de pessoa para pessoa (KRONBAUER; SCHOR; CARVALHO, 2008). Em termos de comprimento de onda, tem-se por correspondência que o espectro visível é definido entre 480 nm e 760 nm.

Temos também uma divisão das cores que se dá em espectrais e visíveis. As cores espectrais são aquelas que correspondem a comprimentos de onda bem definidos do espectro visível e que possuem apenas uma correspondência direta. Ao passo que as cores visíveis, também denominadas não espectrais, são a combinação de mais de um comprimento de onda, como é o caso do magenta, composta por ondas da luz azul e da vermelha na mesma proporção.

Figura 2 – Espectro eletromagnético



Fonte: UC Davis ChemWiki. Acesso em: 3 set. 2020.

A CIE padronizou diversas distribuições espectrais de iluminantes e recomenda seu uso nas avaliações calorimétricas dos quais os principais iluminantes são o D65 e D50 (PEREIRA. 2015). O grupo de iluminantes D representa a luz nas variações do sol (época do ano, posição geográfica e horário), sendo o D50 e o D65 os mais adotados, com temperaturas de cor de 5003 K e 6504 K respectivamente (PEREIRA. 2015).

2.2.2 Objeto observado

O segundo item do triângulo da visão é o objeto observado, que não é capaz de emitir luz própria e apenas interage com a luz incidente proveniente de uma fonte luminosa.

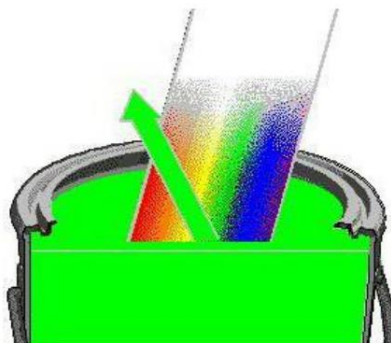
Seguindo a lei da conservação de energia, a luz incidente no objeto pode ser absorvida, refletida ou transmitida e sua soma deve ser igual à radiação incidida, conforme (2.1) em que $\Phi(\lambda)$ é a energia radiante incidente, $R(\lambda)$ a refletida, $T(\lambda)$ a transmitida e $A(\lambda)$ a absorvida pelo objeto.

$$\Phi(\lambda) = R(\lambda) + T(\lambda) + A(\lambda) \quad (2.1)$$

A luz refletida será a responsável pela interação com o olho humano, uma vez que uma pequena parcela é refletida já na superfície, ao incidir sobre esse tipo de objeto, devido ao índice de refração divergente entre o material do objeto e o ar. A maior parte da luz reflete difusamente (em todas as direções) ao penetrar na camada exterior do material, sendo responsável pela luz que chega até nossos olhos (FERNANDES, 2002).

A Figura 3 exemplifica um corpo que é visto na cor verde, posto que reflete difusamente a componente dessa cor da luz incidida e absorve o restante.

Figura 3 – Reflexão de um objeto verde



Fonte: www.educabrasil.org.br. Acesso em: 7 set. 2020.

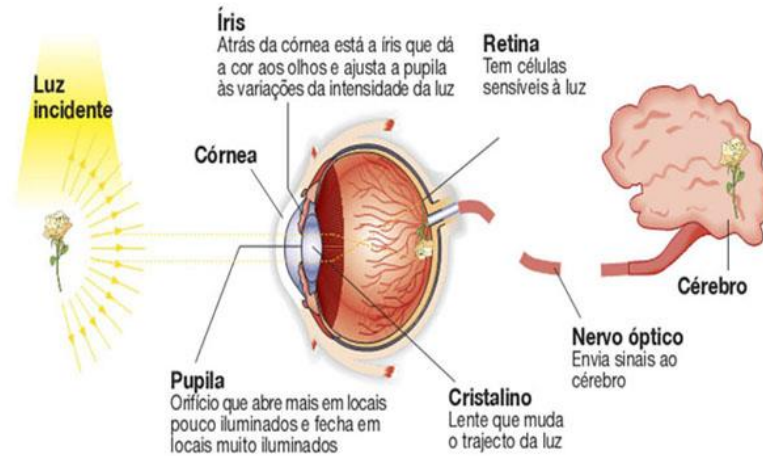
2.2.3 Fisiologia do olho humano

Tendo em vista que qualquer modelo de aparência de cores representa o sistema visual humano, é essencial entender o funcionamento desse sistema, que será descrito neste tópico.

Nossa percepção visual é influenciada pela anatomia e estrutura do olho, portanto a seguir serão descritas as partes influenciadoras na percepção da imagem de forma sumarizada e na Figura 4 poderá ser observada cada uma das partes.

- Córnea: é um tecido transparente e avascular que está em contato com o ambiente externo. Por ser transparente, tem capacidade de realizar transmissão e refração da luz e, por ser resistente, representa uma barreira física entre as estruturas internas do olho e o meio ambiente (MAIA, 2020);
- Iris: é um disco que envolve a pupila e controla a entrada de luz no olho (MAIA, 2020);
- Retina: é o local no qual se formam as imagens, sendo a camada mais interna do olho, corresponde a um fino tecido nervoso que transmite as informações para o cérebro através do nervo óptico (MAIA, 2020). As células da retina sensíveis à luz são conhecidas como cones e bastonetes. Os cones são responsáveis pelas cores e pelos detalhes da imagem, e os bastonetes são encarregados da visão em ambientes escuros, ou seja, são sensíveis em baixa intensidade luminosa.

Figura 4 – Formação de imagem pelo olho humano



Fonte: <http://www.aulas-fisica-quimica.com/>. Acesso em: 7 set. 2020.

A luz, ao atravessar a córnea, penetra no olho humano passando pela íris e pelo cristalino, formando uma imagem do objeto observado sobre a retina (FERNANDES, 2002), que é a parte essencial para o estudo em questão, pois é responsável por transformar a imagem ótica em sinais elétricos para o cérebro (FAIRCHILD, 2005), e por isso será tratada detalhadamente.

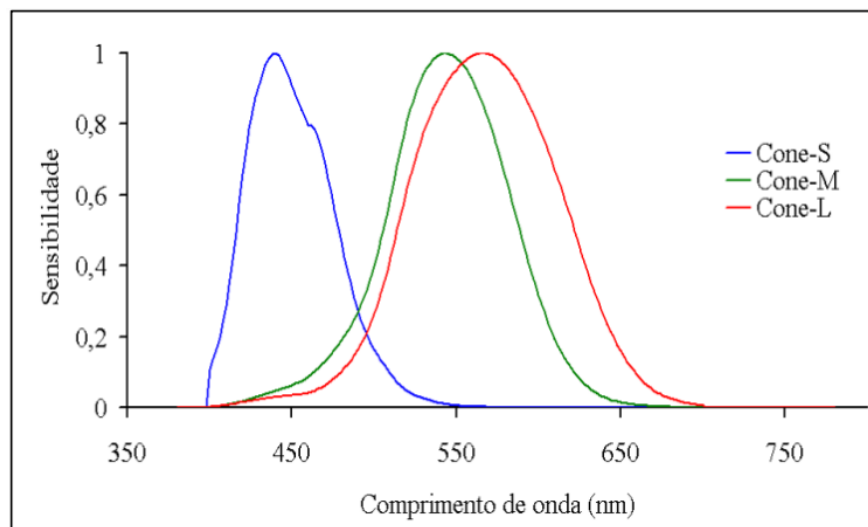
Na retina, são encontradas duas classes de fotorreceptores, os cones e os bastonetes (FAIRCHILD, 2005). E embora os dois estejam na retina, existe uma diferença entre eles que está na sensibilidade a excitação da luz, dado que os cones são sensíveis a alta luminosidade (maior que 100 cd/m^2), e os bastonetes a baixa luminosidade (menor que 1 cd/m^2). Em cenários com alta luminosidade os bastonetes estão saturados e apenas os cones estão operando, todavia em ambientes com iluminação intermediária ambos operam. Quando apenas os cones são utilizados temos a visão denominada de fotópica, já o termo visão mesópica refere-se ao uso das duas classes de fotorreceptores e visão escotópica ocorre quando somente os bastonetes estão ativados (FAIRCHILD, 2005).

Através do estímulo de fotorreceptores é possível enxergar as cores e vale evidenciar que há três tipos de cones e cada um contém um pigmento visual que corresponde a um comprimento de luz (HICKMAN; LARSON, 2012). Apenas uma parcela do espectro de luz pode excitar os cones, o que corresponde ao espectro de luz visível, que vai desde o vermelho até o violeta (LENT, 2010). O ser humano é capaz de diferenciar milhões de cores e isso só é possível através da capacidade de gerar respostas de forma simultânea a frequências relativas (GAZZANIGA, 2014).

Em relação aos cones podemos dizer que cada tipo de cone é sensível à luz em uma certa banda de comprimento de onda dentro do espectro visível. Os cones do tipo L (do inglês *Large*) são sensíveis à zona do vermelho, os do tipo M (do inglês *Medium*) à zona do verde e os do tipo S (do inglês *Short*) à zona do azul (GUALTIERI, 2004).

Na Figura 5, é possível notar a sensibilidade de cada tipo de cone ao comprimento de onda, há sobreposição das curvas de sensibilidade, ou seja, a reação de cada tipo de cone é diferente para uma mesma luz.

Figura 5 – Curvas de sensibilidade dos cones

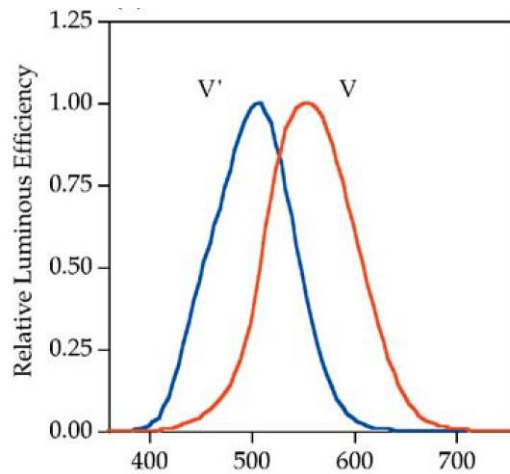


Fonte: (GUALTIERI, 2004)

A relação quantitativa dos cones do tipo L:M:S é aproximadamente 12:6:1 (FAIRCHILD, 2005). Atentando para essa proporção e a sensibilidade de onda de cada cone, tem-se que a sensibilidade máxima combinada está centrada em 550 nm, como mostrado pela curva $V(\lambda)$ na Figura 6. Diferentemente dos cones, os bastonetes são sensíveis a todos os comprimentos de onda do espectro visível, todavia não é capaz de distinguir diferentes comprimentos de ondas, ou seja, não é capaz de detectar cor.

A curva $V'(\lambda)$ da Figura 6 mostra a curva de sensibilidade de uma visão escotópica (definida exclusivamente pelos bastonetes), onde podemos observar a maior sensibilidade centrada no comprimento de onda de aproximadamente 500 nm.

Figura 6 – Sensibilidade da visão escotópica $V'(\lambda)$ e fotópica $V(\lambda)$



Fonte: (FAIRCHILD, 2005)

Por fim, as respostas dos fotorreceptores são transmitidas para o processamento no sistema nervoso que irá fazer a interpretação dos sinais, mas esse processamento não será contemplado no escopo deste trabalho.

2.3 MODELAGEM DE COR

É a diferença de sensibilidade de cada cone descrita anteriormente que permite interpretar diferentes comprimentos de ondas correspondendo a cores diferentes, a esta interpretação damos o nome de teoria dos três estímulos, conhecida desde o início do século XX. Porém eram necessários modelos matemáticos para a percepção das cores pelo olho humano, para correlacionar os três tipos de cones com as curvas espectrais dos objetos e luzes. A CIE (*Commission Internationale de l'Eclairage*), autoridade internacional que define padrões relacionados a luz, iluminação e cores, definiu alguns modelos que serão descritos nesta seção.

2.3.1 CIE RGB

Um dos primeiros modelos de cores propostos surgiu dos experimentos realizados por Wright e Guild e consistia em atingir a mesma percepção visual de uma determinada cor ajustando a intensidade das três cores primárias (LOPES, 2013). A datar desse experimento notou-se que havia cores impossíveis de serem reproduzidas a partir das cores primárias, nesses casos eram adicionadas às cores primárias na cor alvo para atingir o objetivo de alcançar a correspondência de cor.

Os testes realizados deram origem às funções de correlações de cores \bar{r} , \bar{g} e \bar{b} padronizadas pelo CIE, que são conhecidas como CIE RGB. As equações (2.2), (2.3) e (2.4) regem a normalização dos valores tricromáticos, sendo que a partir dessas funções os valores tricromáticos R, G e B podem ser calculados para uma determinada cor definida pela curva espectral $\phi(\lambda)$ (RIDOLFI, 2012).

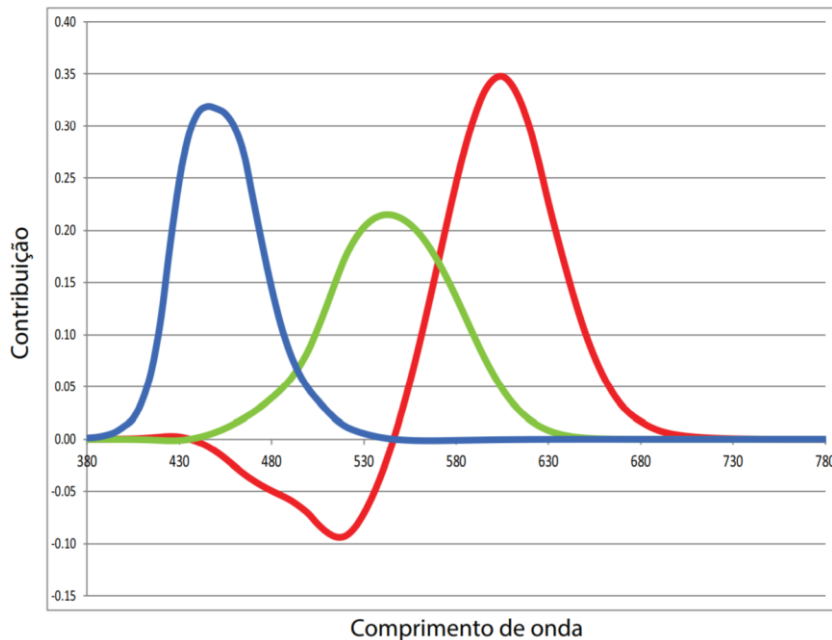
$$R = \int_{\lambda} \phi(\lambda) \bar{r} d\lambda = 1 \quad (2.2)$$

$$G = \int_{\lambda} \phi(\lambda) \bar{g} d\lambda = 1 \quad (2.3)$$

$$B = \int_{\lambda} \phi(\lambda) \bar{b} d\lambda = 1 \quad (2.4)$$

A Figura 7 mostra a contribuição das cores azul ($\lambda = 435,8$ nm), verde ($\lambda = 546,1$ nm) e vermelha ($\lambda = 700,0$ nm) para compor cada comprimento de onda. Vale ressaltar que a parte negativa corresponde aos casos em que é necessário fazer a adição da cor primária à cor alvo.

Figura 7 – CIE RGB



Fonte: (RIDOLFI, 2012)

2.3.2 CIE XYZ

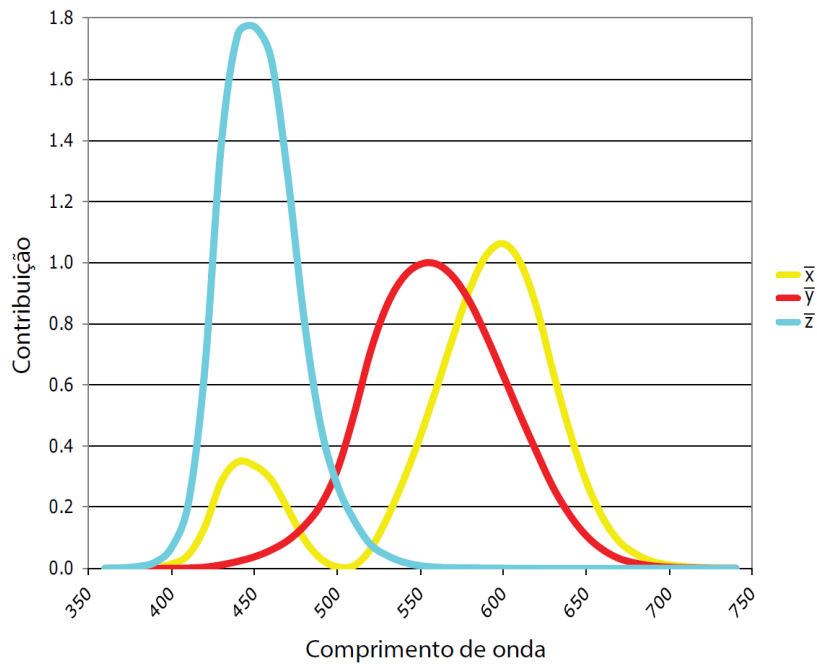
O conceito de “contribuição negativa”, que dificultava a assimilação do conjunto de funções, e o fato de não ser possível determinar a luminosidade da cor de forma direta no modelo CIERG B fez com que a CIE criasse em 1931 um novo modelo denominado CIE XYZ. A partir dele foi possível obter os valores tricromáticos X, Y e Z para uma determinada cor definida pela curva espectral $\phi(\lambda)$ consoante as equações (2.5), (2.6) e (2.7).

$$X = \int_{\lambda} \phi(\lambda) \bar{x} d\lambda = 1 \quad (2.5)$$

$$Y = \int_{\lambda} \phi(\lambda) \bar{y} d\lambda = 1 \quad (2.6)$$

$$Z = \int_{\lambda} \phi(\lambda) \bar{z} d\lambda = 1 \quad (2.7)$$

Figura 8 – Funções de correlação de cores CIE XYZ



Fonte: (RIDOLFI, 2012)

As funções de correlações de cores $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ e $\bar{z}(\lambda)$ são uma transformada linear aplicada em \bar{r} , \bar{g} e \bar{b} . Essa transformada, descrita em (2.8), além de eliminar os componentes negativos da função \bar{r} , torna \bar{x} uma medida precisa da luminância acromática da cor.

Consequentemente, todas as propriedades de cromaticidade da cor, ou seja, intensidade e tom, que não dependem da luminância estão representadas em \bar{y} e \bar{z} (RIDOLFI, 2012).

A Figura 8 mostra as funções de correlação de cores no modelo CIEXYZ análogo ao modelo CIERGB ilustrado na Figura 7, porém nota-se que não há componente negativa.

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \begin{bmatrix} 0,49 & 0,31 & 0,20 \\ 0,17697 & 0,81240 & 0,01063 \\ 0,00 & 0,01 & 0,99 \end{bmatrix} \begin{bmatrix} \bar{r} \\ \bar{g} \\ \bar{b} \end{bmatrix} \quad (2.8)$$

Com a finalidade de obter um plano em duas dimensões da representação da cor a partir dos valores X, Y e Z, foi desenvolvido um diagrama de cromaticidade. As coordenadas do diagrama são obtidas com base na normalização dos valores tricromáticos segundo as equações (2.9), (2.10) e (2.11) (FAIRCHILD, 2005).

$$x = \frac{X}{X + Y + Z} \quad (2.9)$$

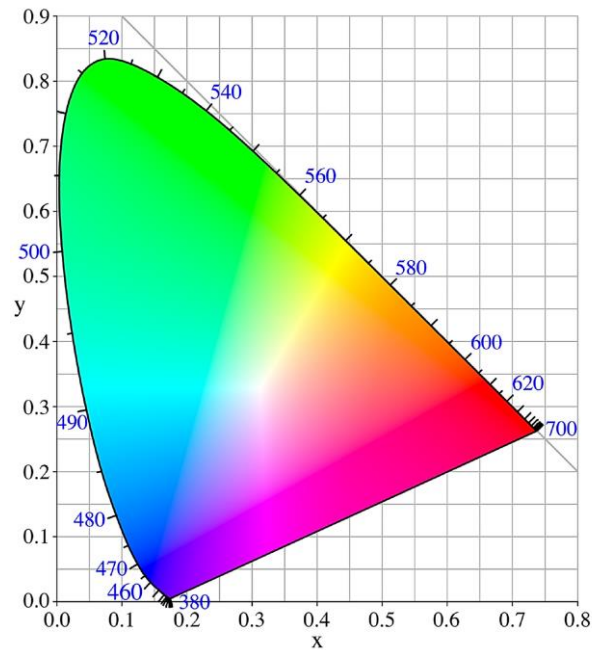
$$y = \frac{Y}{X + Y + Z} \quad (2.10)$$

$$z = \frac{Z}{X + Y + Z} \quad (2.11)$$

A equação (2.12) é uma consequência das equações (2.9), (2.10) e (2.11), evidenciando que x, y e z são linearmente dependentes. Assim, para especificarmos a cromaticidade são necessárias apenas duas das três variáveis, e que por convenção escolheu-se x e y (RIDOLFI, 2012). A Figura 9 reproduz o diagrama de cromaticidade xy descrito.

$$x + y + z = 1 \quad (2.12)$$

Figura 9 – Diagrama de cromaticidade xy



Fonte: (RIDOLFI, 2012)

2.3.3 Espaço CIELAB

Os valores tricromáticos X, Y e Z que definem um espaço psicofísico tridimensional descrito são de difícil interpretação visual, o que motivou o CIE a buscar um novo espaço de cores uniforme. Então em 1976, a CIE definiu o espaço psicométrico CIELAB, em que correlaciona a percepção da luminância com a variável Y.

As equações que definem esse espaço são descritas a seguir pelas equações (2.13) à (2.16).

$$L = 116f\left(\frac{Y}{Y_n}\right) - 16 \quad (2.13)$$

$$a = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \quad (2.14)$$

$$b = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \quad (2.15)$$

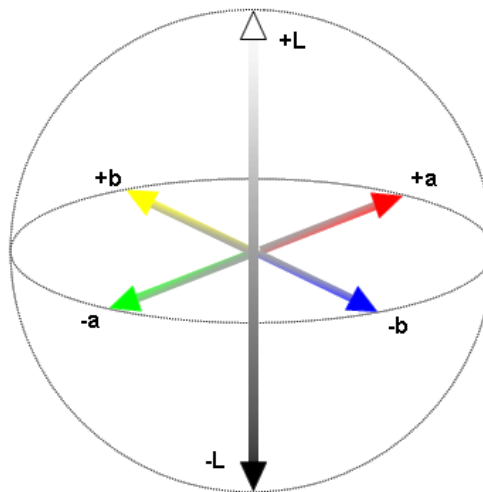
onde:

$$f(t) = \begin{cases} t^{\frac{1}{3}} & , \text{ se } t > \left(\frac{6}{29}\right)^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29} & , \text{ caso contrário} \end{cases} \quad (2.16)$$

Os valores tricromáticos L , a e b , que definem o sistema CIELAB dependem da luz que incide no objeto e no observador e são representados por X_n , Y_n e Z_n (RIDOLFI, 2012). Essas três coordenadas são denominadas *white-point* do iluminante, pois são os valores absolutos de luminância de um objeto branco em cd/m^2 (FAIRCHILD, 2005). Os valores *white-point* dos iluminantes são mostrados na Tabela 1.

Na coordenada L estão todos os dados de luminosidade, logo toda a informação relacionada a cromaticidade de cores está nas coordenadas a e b . Sendo que a coordenada a está relacionada a percepção verde-vermelho e a b , a percepção amarelo-azul. As três coordenadas formam uma representação cartesiana tridimensional de acordo com o que está ilustrado na Figura 10.

Figura 10 – Representação cartesiana do espaço CIELAB



Fonte: <https://afinkopolimeros.com.br/>. Acesso em: 15 out. 2020.

Tabela 1 – *White-point* dos iluminantes

Iluminante	Ângulo	Xn	Yn	Zn
A	2º	1,098	1	0,356
	10º	1,112	1	0,352
D50	2º	0,964	1	0,825
	10º	0,964	1	0,814
D55	2º	0,957	1	0,921
	10º	0,958	1	0,0921
D65	2º	0,951	1	1,089
	10º	0,948	1	1,073
D75	2º	0,949	1	1,226
	10º	0,944	1	1,206
E	2º	1	1	1
	10º	1	1	1

Fonte: (FAIRCHILD, 2005)

2.3.4 CIE76

Após a definição do espaço CIELAB, em 1976, a CIE estabeleceu como a diferença entre duas cores a distância Euclidiana entre dois pontos no espaço CIELAB. Essa equação ficou conhecida como CIE76 e é determinada em (2.17) para duas cores com coordenadas no espaço de cores CIELAB definidas em (2.18) e (2.19).

$$\Delta E_{76} = \sqrt{(L_2 - L_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2} \quad (2.17)$$

onde:

$$lab_1 = (L_1, a_1, b_1) \quad (2.18)$$

$$lab_2 = (L_2, a_2, b_2) \quad (2.19)$$

Esse modelo funciona muito bem para tons com a mesma luminosidade. Contudo devido a composição do espaço CIELAB para cores saturadas de matizes desiguais, teríamos um cálculo de diferença que sinaliza cores muito distintas na percepção humana.

Na Figura 11 temos a cor azul escuro à esquerda e o vermelho escuro à direita, em que o cálculo seria igual a dez ($\Delta E_{76} = 10$), o que significaria ser possível notar a diferença em um relance, conforme a Tabela 2, mas na verdade isso é possível apenas após uma análise próxima ao objeto.

Claramente a saturação é o maior problema do modelo CIE76, entretanto em função da simplicidade do cálculo ainda tem aplicação para situações nas quais não é necessária grande precisão e sim um processamento rápido.

Tabela 2 – Escala ΔE de percepção

ΔE	Percepção.
$\leq 1,0$	Não é perceptível pelo olho humano.
1~2	Perceptível através de uma análise minuciosa.
2~10	Perceptível em um relance.
11~49	Cor mais similar do que oposta.
> 49	Cor mais oposta que similar.
> 100	Cor oposta.

Fonte: (DELTA)

Figura 11 – Azul escuro e vermelho escuro



Fonte: www.zschuessler.github.io/. Acesso em: 16 out. 2020.

2.3.5 CIE94

Em 1994, a CIE definiu o novo modelo denominado CIE94, descrito pela equação (2.20), para sanar a limitação do modelo corrente para cores saturadas. Essa nova fórmula

atribui pesos para os valores de luminosidade, cromaticidade e matiz. Além disso foram adicionados à fórmula K_L , K_1 e K_2 fatores de ponderação que dependem da aplicação, segundo a Tabela 3 (RIDOLFI, 2012).

Apesar de resolver a limitação do modelo anterior, esse modelo apresenta incoerência ao comparar cores com grande diferença na matiz, porém com semelhança na luminosidade. A Figura 12 ilustra esse caso, onde através do modelo CIE94 encontramos $\Delta E_{94} = 128$, o que representa serem cores opostas, enquanto no modelo CIE2000 chegamos em $\Delta E_{00} = 49$, que faz uma melhor representação da disparidade entre as duas cores porém semelhança na luminosidade.

Figura 12 – Matiz diferente, porém iluminação semelhante



Fonte: www.zschuessler.github.io/. Acesso em: 16 out. 2020.

$$\Delta E_{94} = \sqrt{\left(\frac{\Delta L}{K_L}\right)^2 + \left(\frac{\Delta C_{ab}}{1 + K_1 C_1}\right)^2 + \left(\frac{\Delta H_{ab}}{1 + K_2 C_1}\right)^2} \quad (2.20)$$

onde:

$$\Delta L = L_2 - L_1 \quad (2.21)$$

$$C_i = \sqrt{a_i^2 + b_i^2} \quad i = 1,2 \quad (2.22)$$

$$\Delta C_{ab} = C_2 - C_1 \quad (2.23)$$

$$\Delta a = a_2 - a_1 \quad (2.24)$$

$$\Delta b = b_2 - b_1 \quad (2.25)$$

$$\Delta H_{ab} = \sqrt{\Delta a^2 + \Delta b^2 - \Delta C_{ab}^2} \quad (2.26)$$

Tabela 3 – Fatores de ponderação da fórmula CIE94

	Artes Gráficas	Têxteis
K_L	1	2
K_1	0,045	0,048
K_2	0,015	0,014

Fonte: (FAIRCHILD, 2005)

Os conjuntos de dados de diferenças perceptuais que deram origem à fórmula CIE94 contém apenas pequenas distinções de cores, logo a fórmula é válida se empregada em pequenas diferenças de cores (RIDOLFI, 2012). Considera-se uma estreita desigualdade de cores para aplicar esse modelo quando $0 < \Delta E_{76} > 5$ (L'ÉCLAIRAGE C. I., 1993).

Vale salientar que o modelo CIE94 é usado até os dias atuais em aplicações específicas, como no ramo têxtil e no da tintura.

2.3.6 CIE2000

A CIE corrigiu o problema da luminosidade através do modelo CIE2000, padronizado pelo órgão em 2002, que é um dos algoritmos mais usados até os dias atuais devido a sua precisão em diferenciar duas cores.

O modelo CIE2000 é descrito matematicamente para duas coordenadas de cores no espaço CIELAB $c_1 = (L_1, a_1, b_1)$ e $c_2 = (L_2, a_2, b_2)$ conforme as equações (2.27) à (2.45).

$$\Delta E_{00} = \sqrt{\left(\frac{\Delta L'}{S_L}\right)^2 + \left(\frac{\Delta C'}{S_C}\right)^2 + \left(\frac{\Delta H'}{S_H}\right)^2 + R_T \left(\frac{\Delta C'}{S_C}\right) \left(\frac{\Delta H'}{S_H}\right)} \quad (2.27)$$

onde:

$$\Delta L' = L_2 - L_1 \quad (2.28)$$

$$\bar{L} = L_1 + L_2 \quad (2.29)$$

$$\bar{C} = \frac{C_1 + C_2}{2} \quad (2.30)$$

$$C^*_{*i} = \sqrt{a_i^2 + b_i^2} \quad i = 1,2 \quad (2.31)$$

$$a'_i = a_i + \frac{a_i}{2} \left(1 - \sqrt{\frac{\bar{C}^7}{\bar{C}^7 + 25^7}} \right) \quad i = 1,2 \quad (2.32)$$

$$C'_i = \sqrt{a_i'^2 + b_i^2} \quad i = 1,2 \quad (2.33)$$

$$\Delta C' = C'_2 - C'_1 \quad i = 1,2 \quad (2.34)$$

$$C' = \frac{C'_1 + C'_2}{2} \quad (2.35)$$

$$h'_i = \tan^{-1} \left(\frac{b^*_{*i}}{a'_i} \right) \quad i = 1,2 \quad (2.36)$$

$$\Delta h' = \begin{cases} h'_2 - h'_1 & \|h'_1 - h'_2\| \leq \pi \\ h'_2 - h'_1 + 2\pi & \|h'_1 - h'_2\| \leq \pi \\ h'_2 - h'_1 - 2\pi & \|h'_1 - h'_2\| > \pi, h'_2 > h'_1 \end{cases} \quad (2.37)$$

$$\Delta H' = 2\sqrt{C'_1 C'_2} \sin \left(\frac{\Delta h'}{2} \right) \quad (2.38)$$

$$H' = \begin{cases} \frac{h'_2 - h'_1 + 2\pi}{2} & \|h'_1 - h'_2\| > \pi \\ \frac{h'_2 - h'_1}{2} & \|h'_1 - h'_2\| \leq \pi \end{cases} \quad (2.39)$$

$$T = 1 - 0,17 \cos \left(3\bar{H}' - \frac{\pi}{6} \right) + 0,24(2\bar{H}') \\ + 0,32 \cos \left(3\bar{H}' + \frac{\pi}{30} \right) \quad (2.40)$$

$$- 0,20 \cos \left(4\bar{H}' - 21 \frac{\pi}{60} \right)$$

$$S_L = 1 + \frac{0,015(\bar{L} - 50)^2}{\sqrt{20 + (\bar{L} - 50)^2}} \quad (2.41)$$

$$S_C = 1 + 0,045\bar{C}' \quad (2.42)$$

$$S_H = 1 + 0,015\bar{C}'T \quad (2.43)$$

$$\bar{H}'_{deg} = \frac{180\bar{H}'}{\pi} \quad (2.44)$$

$$R_T = \frac{2\bar{C}'^7}{\bar{C}'^7 + 25^7} \sin \left[\frac{\pi}{6} \exp \left(- \left[\frac{\bar{H}'_{deg} - 275}{25} \right]^2 \right) \right] \quad (2.45)$$

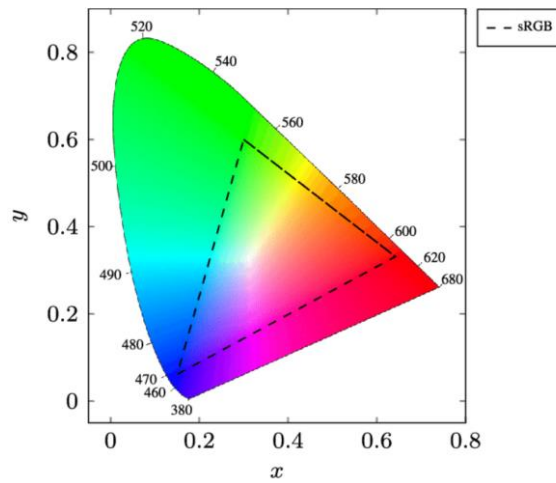
A CIE2000 aproxima-se muito do propósito de se obter uma distância entre cores equivalentes para todas as tonalidades e é considerada a fórmula mais avançada para aferição de diferenças de cores. Devido a isso, será o modelo utilizado neste trabalho para calcular a disparidade entre as cores de duas imagens digitais de forma analítica.

2.3.7 sRGB

O sRGB (*standard RGB*) é um espaço de cor derivado do espaço CIE RGB, criado em 1998 com o objetivo de padronizar cores para monitores de vídeos, ponderando-se a capacidade de exibição de cores (PASCALE, 2003). Esse espaço é utilizado em dispositivos de captura de vídeo até os dias atuais, incluindo as câmeras de segurança que serão alvo desse trabalho.

O espaço sRGB é um subconjunto do CIE RGB (PASCALE, 2003), conforme é mostrada na Figura 13. Consequentemente, também é possível fazer a conversão para o espaço CIE XYZ.

Figura 13 – Espaço de cor sRGB



Fonte: <https://www.avmakers.com.br/>. Acesso em: 2 nov. 2020

Como as imagens sRGB têm a função *gamma* aplicada à imagem, a primeira etapa consiste na conversão dos valores RGB para os espaços absolutos rgb, conforme (2.46), (2.47) e (2.48) (DALCIN, 2007).

$$r = \begin{cases} \frac{R}{12,92} & , \text{ se } R \leq 0,04045 \\ \left(\frac{R + 0,055}{1,055} \right)^{2,4} & , \text{ se } R > 0,04045 \end{cases} \quad (2.46)$$

$$g = \begin{cases} \frac{G}{12,92} & , \text{ se } G \leq 0,04045 \\ \left(\frac{G + 0,055}{1,055} \right)^{2,4} & , \text{ se } G > 0,04045 \end{cases} \quad (2.47)$$

$$b = \begin{cases} \frac{B}{12,92} & , \text{ se } B \leq 0,04045 \\ \left(\frac{B + 0,055}{1,055} \right)^{2,4} & , \text{ se } B > 0,04045 \end{cases} \quad (2.48)$$

A partir do espaço absoluto rgb, a conversão para o espaço XYZ é realizada através da conversão linear mostrada em (2.49) (DALCIN, 2007), análoga a conversão do sistema CIE RGB mostrada em (2.8).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0,412453 & 0,357580 & 0,180423 \\ 0,212671 & 0,715160 & 0,072169 \\ 0,019334 & 0,119193 & 0,950227 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (2.49)$$

2.4 CONCLUSÕES

Nesse Capítulo são inicialmente apresentados os fundamentos da percepção de cores de um ser humano e explicado cada componente do triângulo da visão. Baseado nesse triângulo, modelaram-se matematicamente as percepções de cores pela CIE. Esses modelos foram apresentados de forma evolutiva, sendo o espaço de cor CIELAB o destaque para nossa aplicação, tendo em vista que é através desse espaço que as distâncias de cores são calculadas pela fórmula CIE2000.

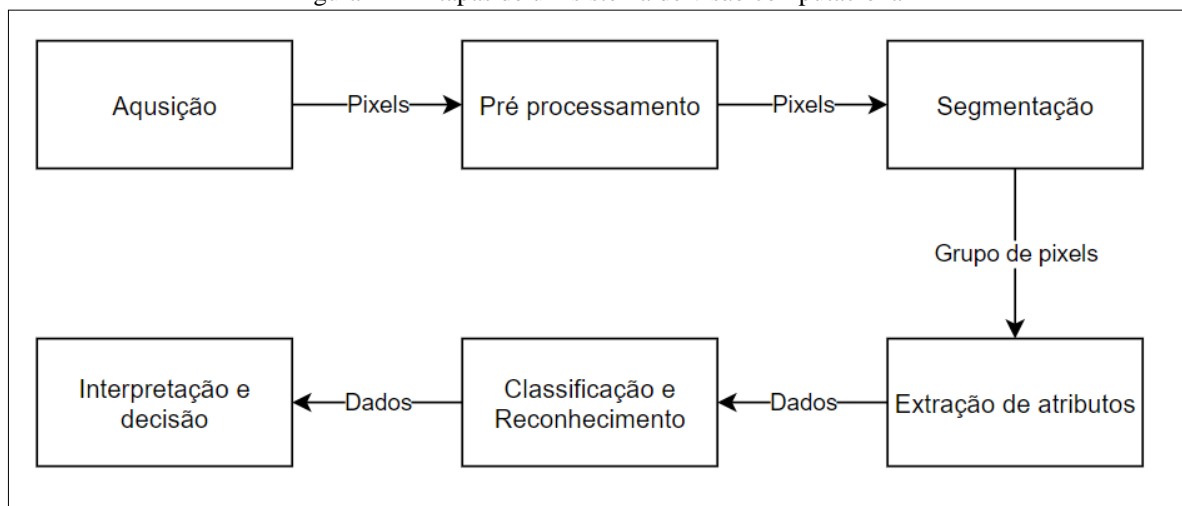
3 VISÃO COMPUTACIONAL E PROCESSAMENTO DE IMAGEM

3.1 INTRODUÇÃO

Com o objetivo de automatizar nossa aplicação, será necessário desenvolver um sistema de visão computacional, que envolve um conjunto de tarefas interconectadas, para o reconhecimento da carta de cores e aquisição da cor capturada.

Para este trabalho serão consideradas seis etapas, que estão descritas e ilustradas na Figura 14 a seguir:

Figura 14 – Etapas de um sistema de visão computacional



Fonte: autor

A etapa de aquisição de imagem consiste no processo de sua captura através de um dispositivo óptico; o pré-processamento envolve rotinas de restauração e realce para facilitar o processo seguinte. A segmentação é a etapa responsável por dividir a imagem em regiões de interesse. Posteriormente a região segmentada é utilizada para extrair atributos, que serão classificados na penúltima etapa. Por fim, é tomada uma decisão a partir da interpretação dos dados do processo anterior.

Este Capítulo descreve os aspectos teóricos de visão computacional e processamento de imagem para o reconhecimento da carta de cores.

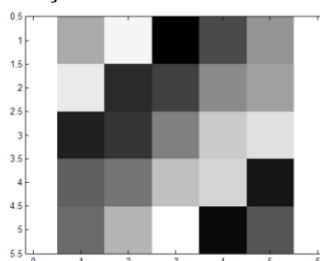
3.2 IMAGEM DIGITAL

A imagem digital é definida como uma função bidimensional, $f(x, y)$, em que x e y são coordenadas de um plano e a amplitude de f em qualquer par de coordenadas é chamada de intensidade da imagem nesse ponto (GONZALEZ; WOODS, 2008). Uma imagem é dita digital quando x e y e os valores de intensidade de f são quantidades finitas e discretas. Essa é considerada uma maneira conveniente da representação de uma imagem contínua (GONZALEZ; WOODS, 2008), que são as cenas do mundo real percebidas pelo olho humano (SONKA; HLAVAC; BOYLE, 2014).

Assim, a imagem digital é composta de um número finito de elementos, chamados de pixels, cada um com uma localização e um valor de intensidade específicos (GONZALEZ; WOODS, 2008). Sua representação é feita através de uma matriz de duas dimensões, contendo M linhas e N colunas, em que (x, y) são coordenadas discretas para células dessa matriz (FELISBERTO, 2015).

A Figura 15 é a representação visual de uma imagem digital 5 x 5 pixels na qual o nível de cinza de cada ponto é proporcional ao valor da intensidade de f nesse ponto.

Figura 15 – Representação visual em escala de cinza de uma imagem



Fonte: (FELISBERTO, 2015)

Outra representação de imagem é realizada através de matrizes numéricas (GONZALEZ; WOODS, 2008), que é comumente utilizada para exprimir quantitativamente imagens digitais para processamento, em que cada posição da matriz representa a localização de um pixel, e o valor numérico a sua intensidade (FELISBERTO, 2015). A matriz mostrada pela equação (3.1) ilustra esse conceito, constituindo a imagem da Figura 15 por meio de uma matriz de números.

$$f(x, y) = \begin{bmatrix} 170 & 245 & 0 & 74 & 148 \\ 235 & 42 & 64 & 138 & 160 \\ 32 & 54 & 128 & 150 & 225 \\ 96 & 118 & 192 & 213 & 21 \\ 106 & 181 & 255 & 10 & 85 \end{bmatrix} \quad (3.1)$$

A definição matemática genérica para uma imagem digital de tamanho $M \times N$ é descrita pela equação (3.2), na qual cada elemento é um pixel (GONZALEZ; WOODS, 2008).

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N-1) \\ f(1,0) & f(1,1) & \dots & f(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1, N-1) \end{bmatrix} \quad (3.2)$$

3.2.1 Tipos de imagens digitais

A partir da intensidade dos pixels a imagem digital é classificada e essa classificação é em três tipos: binária, em escala de cinza e colorida (SONKA; HLAVAC; BOYLE, 2014), que serão abordados neste tópico.

3.2.1.1 Imagem binária

Uma imagem binária é uma imagem digital na qual há apenas dois valores possíveis de intensidade para cada pixel (GONZALEZ; WOODS, 2008). Sendo “0” e “1” os valores mais comuns para essa classificação, em que zero representa a cor preta e um a cor branca.

3.2.1.2 Imagem em escala de cinza

Uma imagem em escala de cinza é uma imagem digital na qual o valor de cada pixel representa apenas a informação de intensidade (GONZALEZ; WOODS, 2008). Sendo que a variação é caracterizada oscilando entre o preto, como a menor intensidade, e o branco, como maior intensidade, passando pelo cinza (NOGUEIRA, 2016).

A representação digital de uma imagem em escala de cinza é feita por 8 bits por pixel, ou seja, são 256 intensidades diferentes possíveis. Sendo de 0 até 255 o intervalo de valores que cada pixel pode assumir, como pode ser notado na matriz (3.1).

A transformação de uma imagem em escala de cinza para uma imagem binária é realizada através da operação denominada *thresholding* (SOLOMON; BRECKON, 2010), que será detalhada na seção 3.5.

3.2.1.3 Imagem colorida

Uma imagem colorida é uma imagem digital na qual cada pixel é representado por um sistema de cores (GONZALEZ; WOODS, 2008). Podemos dizer que o sistema sRGB é o mais popular.

Dessa forma, entendemos que a imagem é representada pela composição das três cores para cada pixel e a representação de cada pixel é realizada por 8 bits em cada canal, ou seja, são necessários 24 bits para representar um pixel. Portanto são possíveis 256^3 combinações, que totalizam 16.777.216 cores diferentes.

A transformação de uma imagem colorida para uma imagem em escala de cinza é realizada através da média ponderada das três componentes. Nesse caso, os pesos utilizados são os mesmos da sensibilidade visual de cada cone do olho humano.

A equação (3.3) descreve essa transformação.

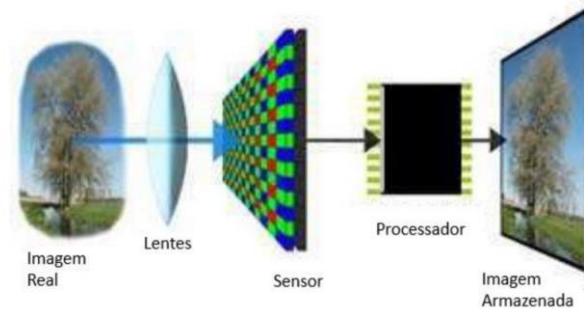
$$I_{cinza} = 0,299 I_{vermelho} + 0,587 I_{verde} + 0,114 I_{azul} \quad (3.3)$$

3.3 AQUISIÇÃO DE IMAGEM

A aquisição de imagem é a primeira etapa de um sistema de processamento de imagem, uma vez que sem uma imagem digital nenhum processamento pode ser realizado (GONZALEZ; WOODS, 2008). A aquisição consiste em capturar imagens analógicas (cenas reais) através de um dispositivo óptico (câmera), que faz a conversão para uma representação digital, sendo essencialmente necessária a utilização dos componentes ilustrados na Figura 16. Quanto a essa primeira etapa, ela é responsável pela diferença de qualidade nas imagens entre dispositivos, que será a razão da proposta da aplicação apresentada neste trabalho.

Constituídas fisicamente por lentes que atuam de modo simultâneo com componentes eletrônicos, as câmeras digitais necessitam de perfeita integração e controle de qualidade de seus componentes físicos e eletrônicos para seu correto funcionamento e para que tenham uma performance padrão entre dispositivos idênticos (SIMÃO, 2020).

Figura 16 – Componentes de uma câmera digital



Fonte: (SIMÃO, 2020)

Na realização deste estudo foram utilizadas câmeras de segurança, cuja característica construtiva segue o modelo apresentado e que serão detalhadas a seguir, com exceção da etapa de armazenamento, pois não tem consequências em nossa aplicação.

3.3.1 Lentes

As lentes são o primeiro elemento construtivo a ter contato com a luz refletida ou emitida pelo cenário a ser capturado (SIMÃO, 2020), conforme ilustrado na Figura 16. Em câmeras mais simples existe apenas uma lente acoplada, porém há câmeras compostas por conjuntos de lentes ajustadas por mecanismos, como mostrado na Figura 17.

Por conta de imperfeições durante o processo de fabricação e também da natureza construtiva das lentes, esse elemento está sujeito a pequenas diferenças que podem impactar na imagem capturada.

Figura 17 – Detalhe interno do conjunto de lentes



Fonte: (SIMÃO, 2020)

3.3.2 Sensores

Para a conversão da imagem real para um dado digital é necessário um dispositivo físico sensível a uma faixa de espectro eletromagnético e capaz de produzir um sinal elétrico proporcional ao nível de energia recebido (GONZALEZ; WOODS, 2008). Em câmeras digitais uma matriz de sensores é responsável por essa conversão.

Esses sensores podem ser entendidos como uma matriz $M \times N$, na qual cada elemento é um micro sensor de luz que recebe um sinal luminoso que será convertido em sinal elétrico (SIMÃO, 2020). Sendo que cada elemento da matriz contém um valor digital proporcional à intensidade luminosa sofrida naquela posição.

Atualmente existem dois tipos de sensores: CCD (*charged coupled device*) e o CMOS (*complementary metal oxide semiconductor*). A principal diferença entre eles é o modo como a energia luminosa é transmitida ao substrato, que é a camada sensível à luminosidade, ou melhor, o lugar em que a energia luminosa é convertida em energia elétrica (BARCELLOS, 2011).

Os sensores CCD são fabricados com a tecnologia MOS (*Metal-Oxide Semiconductor*), caracterizada pelo fato de cada pixel ser um circuito eletrônico. Esse circuito pode ser considerado um capacitor MOS, que é responsável por converter os fótons em cargas elétricas que serão lidas e processadas posteriormente (SOUZA; CARDOZA, 2012). Já os sensores CMOS têm em cada pixel um circuito constituído de fotodiodos, capacitores e transistores interligados, formando assim o pixel individual (SOUZA; CARDOZA, 2012).

Podemos dizer que atualmente os sensores CMOS são os mais empregados em razão de serem mais eficientes, apresentarem menos ruído na imagem, menor nível de aquecimento e utilizarem menos componentes e elementos eletrônicos na sua fabricação em relação ao CCD.

3.3.3 Processador

Tendo em vista que os computadores trabalham com valores discretos é necessário que o processador faça a discretização do sinal na saída dos sensores. Para a discretização e conseqüentemente para a conversão de uma imagem digital os processos de amostragem e quantização são fundamentais.

Levando em conta a definição de imagem digital descrita na seção 3.2, entendemos que imagem é um sinal contínuo de cor em duas dimensões, no qual o domínio é o plano e o contradomínio é o espaço de cor.

Na amostragem, obtém-se amostras de $f(x, y)$ nas direções x e y (domínio) em instantes de tempo discretos, tomados em um intervalo de tempo uniforme (FELISBERTO, 2015). Em vista disso é gerado uma matriz $N \times M$ de amostras, digitalizando o plano em que a imagem está inserida (GONZALEZ; WOODS, 2008). Essa amostragem é seguida da digitalização da amplitude $f(x, y)$ (contradomínio) em L níveis inteiros de intensidade, denominada quantização (GONZALEZ; WOODS, 2008).

3.4 PRÉ-PROCESSAMENTO

Esta seção apresenta fundamentos e técnicas de processamento de imagem utilizadas para que o pré-processamento possa reconhecer uma carta de cor e extrair cada paleta.

3.4.1 Conceitos básicos

Levando em consideração p um pixel localizado na coordenada (x, y) , temos que a vizinhança ou janela $N(x, y)$ é o conjunto de pixels que cercam p , incluindo o próprio p (GONZALEZ; WOODS, 2008).

Para este trabalho só serão utilizadas janelas quadradas, nas quais a vizinhança $N(x, y)$ do pixel p na imagem digital f é definida como um quadrado de dimensões ímpares $W \times W = (2M - 1) \times (2M - 1)$, para $M > 0$ inteiro centrado em p . O conceito de vizinhança é empregado em operações espaciais sobre o plano da imagem.

A equação (3.4) descreve a operação espacial aplicada a uma imagem $f(x, y)$, em que T é um operador em f definido em uma vizinhança do pixel em (x, y) e $g(x, y)$ é a imagem de saída.

$$g(x, y) = T[f(x, y)] \quad (3.4)$$

A operação espacial gera um pixel s em $g(x, y)$ nas mesmas coordenadas da imagem f .

A equação (3.5) descreve essa operação que é denominada filtragem espacial.

$$s = T(f, N) \quad (3.5)$$

Esse processo de filtragem espacial efetuado pelo deslocamento de uma janela (máscara) sobre a imagem pelo avanço de cada pixel por vez é denominado convolução.

3.4.2 Filtro Gaussiano

O filtro gaussiano é um operador convolucional bidimensional utilizado para borrar ou desfocar a imagem na qual é aplicado, com o objetivo de reduzir ruídos presentes na imagem. Essa aplicação é realizada através da convolução da imagem com uma função gaussiana, como um filtro passa-baixa. Enquanto essa função expressa a distribuição normal em estatística e é descrita pela equação (3.6), que define a curva gaussiana de uma dimensão (GONZALEZ; WOODS, 2008). Sendo $G(x)$ a distribuição gaussiana dos valores de x , σ o desvio padrão dos valores x e μ a média dos valores de x .

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\mu - x)^2}{2\sigma^2}} \quad (3.6)$$

O filtro gaussiano aplicado a uma imagem é de duas dimensões, uma X e outra Y. Para obter esse filtro é necessário multiplicar a equação (3.6) definida em X pela equação (3.6) definida em Y, obtendo a equação (3.7).

$$G(x, y) = G(x)G(y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (3.7)$$

A matriz gaussiana utilizada como máscara na convolução da imagem é gerada a partir da equação (3.7), atribuindo os valores de x , y e σ . Portanto, para uma matriz de dimensão 5, com desvio padrão igual a 1 tem-se a matriz mostrada na Tabela 4.

Tabela 4 – Filtro gaussiano 5x5

0.00078631	0.00655952	0.01330347	0.00655952	0.00078631
0.00655952	0.05472049	0.11097944	0.05472049	0.00655952
0.01330347	0.11097944	0.22507904	0.11097944	0.01330347
0.00655952	0.05472049	0.11097944	0.05472049	0.00655952
0.00078631	0.00655952	0.01330347	0.00655952	0.00078631

Fonte: Autor

Então, temos como um dos objetivos do filtro gaussiano a suavização da imagem, dessa forma quanto maior o valor maior a suavidade do desvio padrão da Gaussiana usada.

3.5 SEGMENTAÇÃO

Segmentar consiste em dividir a imagem em diferentes regiões (SONKA; HLAVAC; BOYLE, 2014) e o objetivo da segmentação é encontrar a região de interesse no processamento de imagem. No caso do trabalho proposto, a região de interesse é a carta de cores e cada paleta.

Nesta seção serão apresentadas as seguintes técnicas de segmentação: limiarização global e adaptativa, detecção de bordas e as operações morfológicas de erosão e dilatação.

3.5.1 Limiarização

O *threshold* ou limiarização é um processo de segmentação de imagens baseado na diferença dos níveis de cinza, extraindo os objetos de interesse por meio de um limiar T que separa os agrupamentos de níveis de cinza (GONZALEZ; WOODS, 2008).

A equação (3.8) descreve a imagem $g(x, y)$ segmentada através da limiarização da imagem $f(x, y)$, em que T é o limiar de intensidade que segmenta a imagem em dois grupos.

$$g(x, y) = \begin{cases} 1 & , se f(x, y) \geq T \\ 0 & , se f(x, y) \leq T \end{cases} \quad (3.8)$$

Vale ressaltar que a imagem $f(x, y)$ deve ser em tons de cinza, de acordo com o descrito em 3.2.1.2, e a imagem $g(x, y)$ deve ser binária, conforme o descrito em 3.2.1.1.

A dificuldade da segmentação através do processo de limiarização advém da dificuldade de estabelecer o limiar T . Iluminação, semelhança entre níveis de cinza de objeto e

fundo, contraste inadequado e ruído são alguns dos fatores que podem tornar as distribuições de pixels de fundo e dos objetos de interesse insuficientemente diferentes, impossibilitando o estabelecimento de um único limiar global aplicável a toda a imagem (FELISBERTO, 2015).

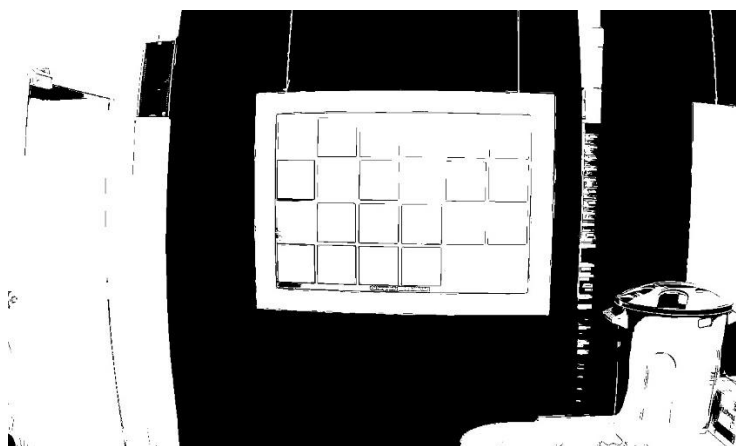
Os métodos adaptativos de limiarização são técnicas que vieram para sanar os problemas descritos, pois enquanto o método convencional utiliza um valor global T para todos os pixels, os métodos adaptativos selecionam de forma dinâmica baseados na vizinhança de cada pixel (SEZGIN; SANKUR, 2014).

O método adaptativo usado neste trabalho é o de média da intensidade dos pixels descrita na equação (3.9). Nela $m(x, y)$ é a média de intensidade da vizinhança quadrada com centro em (x, y) , M o tamanho da vizinhança e C a tolerância. Para simplificação do método o valor da tolerância é obtido de forma empírica.

$$T_M(x, y) = m(x, y) - C \quad (3.9)$$

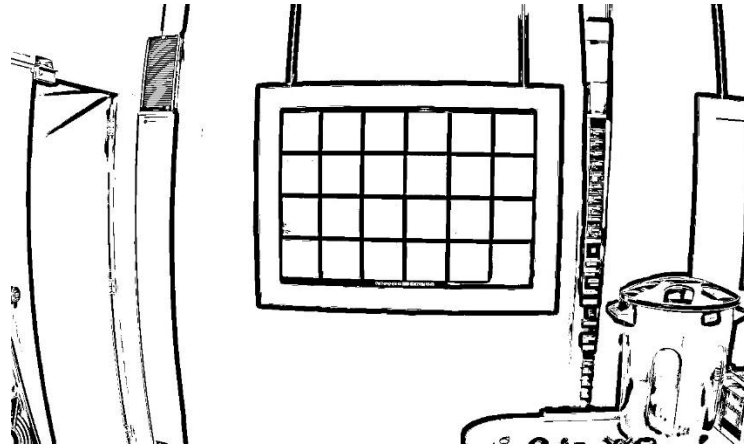
A Figura 18 e a Figura 19 ilustram a diferença entre o método adaptativo e o global. Como pode ser percebido, o resultado utilizando o método adaptativo é superior ao global, tendo em vista que grande parte da área de interesse é perdida no método global, o que não acontece no método adaptativo.

Figura 18 – Método de limiarização global



Fonte: Autor

Figura 19 – Método de limiarização adaptativa



Fonte: Autor

3.5.2 Detecção de bordas

A limiarização tem o propósito de evidenciar as principais características da imagem, porém ainda é necessária a detecção das bordas, que é a etapa da segmentação com a função de identificar grupos de pixels que formam linhas em uma imagem de interesse (GONZALEZ; WOODS, 2008).

Nesta pesquisa será usada a transformada de Hough, que é um dos métodos mais utilizados para esse propósito em sistemas e é chamado de espaço paramétrico (FELISBERTO, 2015).

$$y = ax + b \quad (3.10)$$

A equação (3.10) pode ser reescrita conforme a equação (3.11), obtendo a expressão de reta que intercepta o pixel com coordenada (x_i, y_i) , em que considera o plano ab , correspondente ao espaço de parâmetros (MATTOS; GODOY, 2016).

$$b = y - ax \quad (3.11)$$

A partir desses dados, é possível gerar um acumulador com dimensão igual ao número de parâmetros de cada pixel mapeado, localizado em (x_i, y_i) . Isso consiste no exame de cada pixel e no cálculo dos parâmetros da reta descrita pela equação (3.11). Após a determinação dos parâmetros do pixel, eles são quantizados para um valor correspondente a e b , e o

acumulador, denominado A , é incrementado na posição (a, b) . Concluída a varredura da imagem, os maiores valores no acumulador A indicam os parâmetros prováveis de uma linha (MATTOS; GODOY, 2016).

A dificuldade de representar linhas de inclinação vertical com a equação (3.10) faz com que seja mais interessante a representação através das coordenadas polares descrita em (3.12). Os parâmetros x e y são as coordenadas do pixel, ρ é a distância até a origem definida e θ é o ângulo entre o eixo x e esse ponto.

$$\rho = x \cos(\theta) + y \sin(\theta) \quad (3.12)$$

Ao invés de se utilizar um plano com os parâmetros (a, b) é utilizado um plano com parâmetros (ρ, θ) , denominado espaço de Hough (OLIVEIRA, 2018). Mapeando-se os pixels, não para o espaço paramétrico e sim para o de Hough, o espaço discretizado é analisado célula por célula e quanto mais pixels tiverem sido mapeados para determinada célula maior a chance de eles representarem uma reta (OLIVEIRA, 2018).

3.6 OPERAÇÕES MORFOLÓGICAS

A morfologia matemática é uma ferramenta para extração de componentes da imagem que sejam úteis na representação e descrição da forma da região (CARRERA, 2017) e é usada para remover imperfeições de imagens binarizadas a fim de extrair informações da sua forma e da sua estrutura (GONZALEZ; WOODS, 2008). Elas são normalmente aplicadas na remoção de imperfeições introduzidas durante o processo de segmentação.

Fundamentalmente, técnicas de processamento morfológico de imagens são como técnicas de filtragem espacial, isto é, a janela denominada de elemento estruturante é movimentada por cada pixel da imagem original para gerar um pixel na nova imagem.

A seguir serão descritas as duas operações morfológicas usadas de erosão e dilatação.

3.6.1 Erosão

A operação de erosão permite separar objetos que se tocam e remover pequenos objetos da imagem.

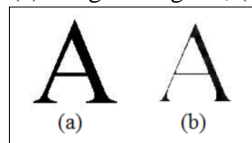
Sendo em A a imagem e em B o elemento estruturante, a operação de erosão representada por \ominus é descrita em (3.13).

$$A \ominus B = \{x | (B)_x \subseteq A\} \quad (3.13)$$

Em outras palavras, considere uma imagem binarizada na qual o fundo tem o valor 0 e os objetos 1. Na erosão a imagem é varrida pelo elemento estruturante de forma que o pixel analisado receberá o valor 1 somente se todos os pixels contidos no elemento estruturante forem 1, caso contrário o pixel recebe o valor 0 (VALMORBIDA, 2018). Os efeitos da erosão são encolhimento de objetos maiores e remoção de objetos finos e pequenos.

Na Figura 20 é demonstrada a operação de erosão em uma imagem original (a) e o resultado é obtido após o processo em uma imagem resultante (b).

Figura 20 – Erosão: (a) Imagem original; (b) Imagem resultante



Fonte: (SILVA, 2015)

3.6.2 Dilatação

Ao contrário da erosão, a dilatação é uma operação morfológica que aumenta os objetos em uma imagem binária e esta operação pode reparar pequenas falhas na imagem. Os efeitos da dilatação são aumento e união de objetos e preenchimento de buracos em seus interiores.

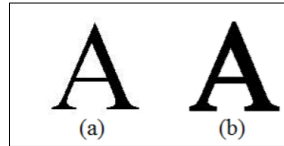
Sendo A a imagem e B o elemento estruturante, a operação de erosão representada por \oplus é descrita em (3.14).

$$A \oplus B = \{x | (B)_x \cap A \neq \emptyset\} \quad (3.14)$$

Em outros termos, considere uma imagem binarizada na qual o fundo tem o valor 0 e o objeto 1. Na dilatação a imagem é varrida pelo elemento estruturante de forma que o pixel analisado receberá o valor 0 somente se todos os pixels do elemento estruturante forem 0, caso contrário, é atribuído o valor 1 (VALMORBIDA, 2018).

Na Figura 21 é demonstrada a operação de dilatação em uma imagem original (a) e o resultado é obtido após o processo em uma imagem resultante (b).

Figura 21 – Dilatação: (a) Imagem original; (b) Imagem resultante



Fonte: (SILVA, 2015)

3.7 CONCLUSÕES

Este Capítulo descreveu os aspectos de visão computacional e processamento digital de imagem utilizados neste trabalho, desde a aquisição até a segmentação, que serão as bases teóricas necessárias para o reconhecimento das paletas de cores de uma carta de cores.

4 DESENVOLVIMENTO DE ALGORITMO DE VISÃO COMPUTACIONAL PARA COLORIMETRIA

4.1 INTRODUÇÃO

Este Capítulo descreve o desenvolvimento do algoritmo desde a conversão da imagem digital em tons de cinza até o resultado, que consiste na comparação de cores entre duas imagens.

A construção do algoritmo se baseia nos conceitos abordados referentes à visão computacional descritos no Capítulo 3, para realizar as três primeiras etapas do processo apresentadas na Figura 14, que são: aquisição, pré-processamento e segmentação; e as etapas de extração de atributos e classificação que se baseiam nos conceitos de colorimetria e foram abordadas no Capítulo 2.

Este Capítulo descreve, além da apresentação do algoritmo, os principais recursos utilizados para o seu desenvolvimento.

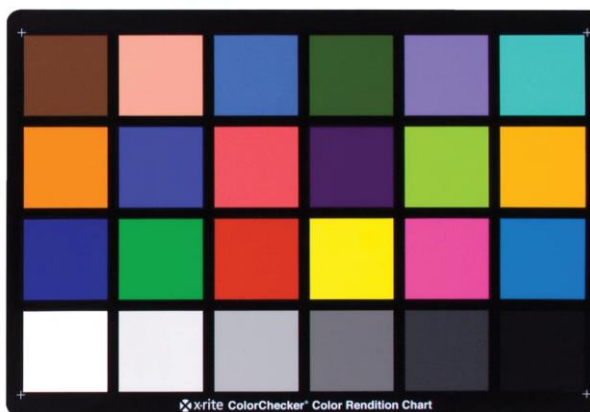
4.2 RECURSOS

Esta seção expõe os meios fundamentais empregados para o desenvolvimento da aplicação.

4.2.1 Carta de cores

A carta de cores é um cartão utilizado para calibração de cores por profissionais de fotografia. Esse cartão está dividido de forma quadriculada em 24 regiões e organizado em 4 linhas de 6 regiões (paletas de cores) (Barcellos, 2011). Na Figura 22 temos a aparência da carta de cores e na Tabela 5 é apresentado como cada paleta de cor será referida nesse trabalho.

Figura 22 – Carta de cores



Fonte: <https://www.xrite.com/>. Acesso em: 2 nov. 2020

Tabela 5 – Paletas de cores

Código	Cor
#1 MRM	Marrom
#2 RCL	Rosa Claro
#3 ACU	Azul Céu
#4 VFL	Verde Folha
#5 RXC	Roxo Claro
#6 VDA	Verde Azulado
#7 LRJ	Laranja
#8 ARX	Azul Arroxeado
#9 VMD	Vermelho Moderado
#10 RXO	Roxo
#11 VAM	Verde Amarelado
#12 ALJ	Amarelo alaranjado
#13 AZL	Azul
#14 VRD	Verde
#15 VML	Vermelho
#16 AMR	Amarelo
#17 MGT	Magenta
#18 CNO	Ciano
#19 BRC	Branco
#20 CZ8	Cinza 8
#21 CZ6	Cinza 6,5
#22 CZ5	Cinza 5
#23 CZ3	Cinza 3,5
#24 PRT	Preto

Fonte: Autor

A carta de cores foi usada como referência de cor nesta pesquisa, uma vez que a fornecedora apresenta dados confiáveis referentes as especificações das cores contidas na cartela (BARCELLOS, 2011). Além disso, esse produto tem uso consagrado na calibração de cores em fotografia e é referência em diversos trabalhos. Em razão disso ela será o objeto de comparação entre as cores de duas imagens digitais.

4.3 LINGUAGEM DE PROGRAMAÇÃO PYTHON

Considerada uma das linguagens de programação mais utilizadas no mundo, a linguagem Python caracteriza-se por apresentar uma sintaxe simples, de fácil interpretação e com grande capacidade de funcionalidades. Foi desenvolvida com o objetivo de criar uma ferramenta de programação que acelerasse o processo de desenvolvimento de softwares em comparação com as já existentes (CHUN, 2006).

O python é uma linguagem interpretada e de alto nível que pode ser utilizada de duas formas: interativa e scripts (ALMEIDA, 2018). Optamos por utilizar o modo scripts, no qual o código python é escrito e salvo em um arquivo (extensão .py), que para sua execução necessita de um interpretador.

Em virtude da facilidade, programadores em todo mundo começaram a desenvolver pacotes (bibliotecas) que ampliaram as capacidades da linguagem (ALMEIDA, 2018). Devido ao fato de ser uma programação fácil e possuir uma gama de pacotes disponíveis para comunidade (*open source*) possibilitando conferir o código implementado com a formulação proposta, essa foi a linguagem escolhida para a elaboração desta dissertação, que tem suas principais bibliotecas presentes relatadas em seu decorrer, sendo o OpenCV o pacote em destaque a ser apresentado de forma mais detalhada em 4.3.1.

4.3.1 OpenCV

O OpenCV (*Open source computer vision*) é uma biblioteca de visão computacional para aplicações em tempo real desenvolvida originalmente pela Intel em C++, porém atualmente tem suporte para outras linguagens, inclusive Python (ALMEIDA, 2018).

Essa biblioteca fornece algoritmos para diversas técnicas de processamento de imagem como: transformada de espaço de cores, normalização, filtros de suavização, detectores de bordas e plotagem de imagens. Pela relevância na comunidade científica e pelos algoritmos disponíveis, essa foi a biblioteca escolhida como base deste projeto de pesquisa.

4.3.2 PyCharm

No presente trabalho foi utilizada a versão gratuita da IDE (*Integrated development environment*) PyCharm versão 2020.1 da JetBrains. Tomemos IDE em uma tradução livre como ambiente de desenvolvimento integrado, que é um software desenvolvido para agrupar ferramentas e características de linguagem de programa com o objetivo de facilitar a implementação de códigos.

4.4 CÓDIGO

Esta seção descreve o algoritmo da aplicação implementado em Python, bem como ilustra um exemplo a fim de facilitar a explicação.

4.4.1 Conversão em tons de cinza

A biblioteca OpenCV disponibiliza uma função de conversão da imagem em tons de cinza chamada *cvtColor*, que tem por parâmetro a “COLOR_RGB2GRAY”, responsável por realizar a conversão dos valores RGB de um pixel p para uma escala de cinza correspondente (0 a 255) (COSTA, 2019). Essa conversão é realizada de acordo com a Equação (3.3).

A Figura 23 mostra o trecho do código pertinente a essa operação implementada na imagem original mostrada na Figura 24 em que o resultado está ilustrado na Figura 25.

Figura 23 – Conversão para uma escala de cinza

```
1. #conversão tons de cinza
2. img_cz = cv2.cvtColor(img,cv2.COLOR_RGB2GRAY)
```

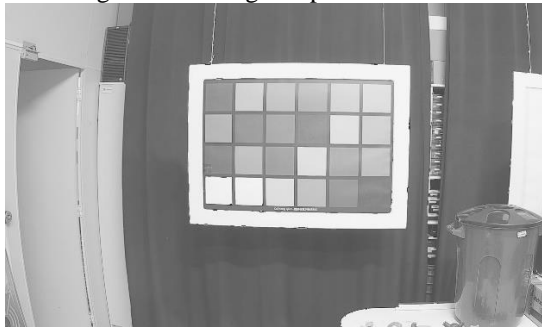
Fonte: Autor

Figura 24 – Imagem original



Fonte: Autor

Figura 25 – Imagem após a conversão



Fonte: Autor

4.4.2 Filtro Gaussiano

A função *GaussianBlur* é disponibilizada pela biblioteca OpenCV para realizar o filtro gaussiano explicado na seção 3.4.2 em que os parâmetros introduzidos são: imagem em tom de cinza a ser realizada a filtragem, dimensão da matriz gaussiana (máscara) e desvio padrão σ em x e y respectivamente.

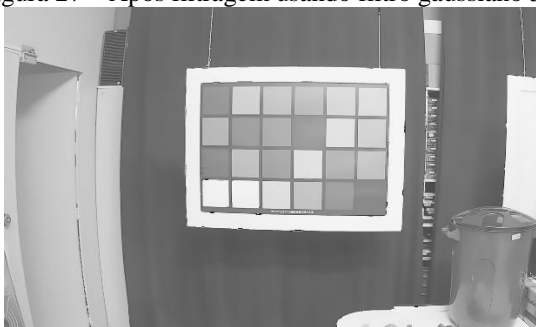
Figura 26 – Filtro gaussiano

```
3. #filtro gaussiano
4. image_g = cv2.GaussianBlur(image_cz, (5,5), 0)
```

Fonte: Autor

A Figura 26 demonstra o trecho do código referente a filtragem realizada usando uma matriz gaussiana de dimensão 5 como mostrada na Tabela 4. Na Figura 27 temos a ilustração do resultado da filtragem aplicada na imagem em tons de cinza, apresentada na Figura 25.

Figura 27 – Após filtragem usando filtro gaussiano 5x5



Fonte: Autor

4.4.3 Limiarização adaptativa

A limiarização adaptativa descrita em 3.5.1 é aplicada utilizando a função *adaptiveThreshold* da biblioteca OpenCV, e os parâmetros de entrada da função são: a imagem em tons de cinza, que será aplicada à limiarização; o valor máximo utilizado para o valor do pixel resultante; o método adaptativo utilizado; o tipo de imagem, que será a saída da função; o tamanho da janela de vizinhança e a variável C da equação (3.9).

Na Figura 28 é mostrado o trecho de código referente a essa operação, realizada nesse trabalho, em que a entrada é a Figura 27, o valor máximo do pixel é 255 e o método adaptativo é o da média, explicado na seção 3.5.1. Já a saída é uma imagem binária e o parâmetro *block_size* utilizado é calculado automaticamente pelo OpenCV, mas não será aqui detalhado. Por fim a constante C utilizada foi 3, definida empiricamente após o teste amostral em dez imagens, e o resultado implementado na Figura 27 está ilustrado na Figura 19.

Figura 28 – Limiarização adaptativa

```
5. #limiarização adaptativa
6. image_1 = adaptiveThreshold(image_g, 255,
    cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY,
    block_size, 3)
```

Fonte: Autor

4.4.4 Erosão e dilatação

As operações morfológicas de erosão e dilatação descritas em 3.6.1 e 3.6.2 são aplicadas através das funções *erode* e *dilate*, respectivamente, disponíveis na biblioteca OpenCV.

Ambas as funções têm como parâmetros de entrada a imagem a ser aplicada à função, o elemento estruturante e o número de interações, isto significa, o número de vezes que será aplicada em toda a imagem.

Neste projeto foi definido como elemento estruturante uma matriz 3×3 com valores unitários, como demonstrado em (4.1), e foi aplicada apenas uma interação por função.

Figura 29 mostra o trecho do código referente à erosão e à dilatação com os parâmetros descritos anteriormente. Por sua vez, a Figura 30 ilustra a erosão do código aplicada na Figura 27 e na Figura 31 é mostrada a dilatação após a erosão.

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.1)$$

Figura 29 – Erosão e dilatação

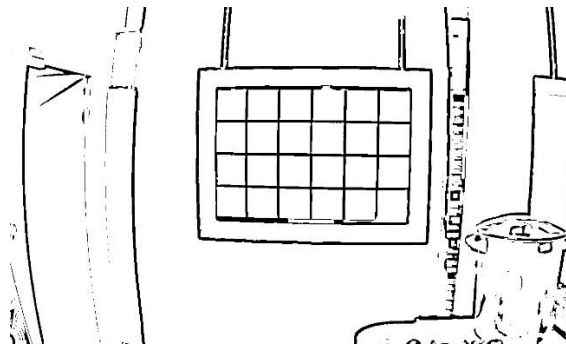
```

7. #definição do elemento estruturante B
8. B = np.ones((3, 3), np.uint8)
9. #erosão com 1 interação
10. image_e = cv2.erode(image_l, B, iterations=1)
11. #dilatação com 1 interação
12. image_d = cv2.dilate(image_e, B, iterations=1)

```

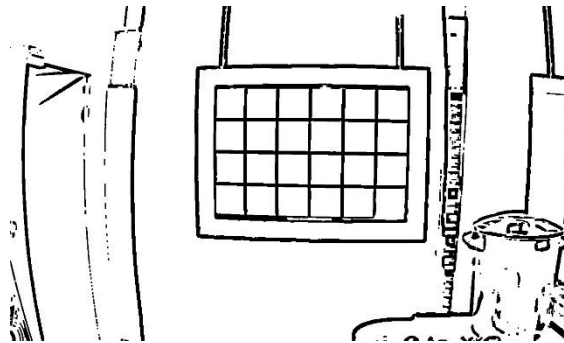
Fonte: Autor

Figura 30 – Após erosão



Fonte: Autor

Figura 31 – Após dilatação



Fonte: Autor

A limiarização adaptativa descrita em 3.5.1 é aplicada ao se utilizar a função *adaptiveThreshold* da biblioteca OpenCV, e os parâmetros de entrada da função são: a imagem em tons de cinza, que será aplicada a limiarização, e o valor máximo.

4.4.5 Segmentação

Esta subseção descreve as etapas de segmentação da imagem implementadas para encontrarmos os centros das paletas de cores.

4.4.5.1 Transformada de Hough

A transformada de Hough descrita em 3.5.2 é aplicada utilizando a função *HoughLinesP* da biblioteca OpenCV, e os parâmetros de entrada da função são: imagem binária com as bordas destacadas, em que será aplicada a transformada; ρ e θ da equação (3.12); *threshold* do acumulador (valor limite de votos); número mínimo de pontos, para identificar uma linha (*minLineLength*) e diferença máxima entre dois pontos, para ser considerada uma reta (*maxLineGap*).

A Figura 32 mostra o trecho de código referente a essa operação, implementada neste trabalho, na qual a entrada é a Figura 31, o ρ é igual a 1, o θ igual a $\frac{\pi}{180}$ e o *threshold* é de 200. Essa função permite que os dois últimos parâmetros sejam ajustados automaticamente. Os valores armazenados na variável *lines*, após a operação, estão sobrepostos na figura original para ilustração didática, conforme a Figura 33.

Figura 32 – Transformada de Hough

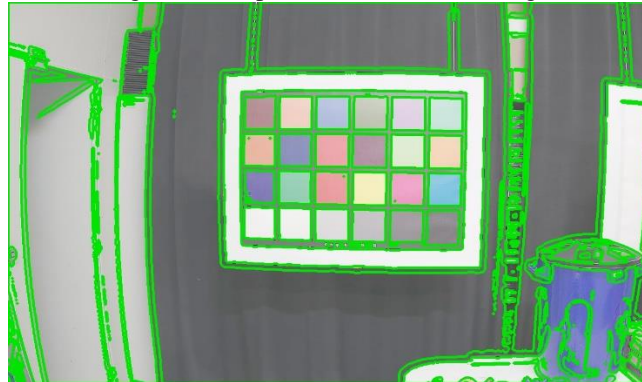
```

13. #definição do elemento estruturante B
14. lines = cv2.HoughLinesP(image_d,1,np.pi/180,200)

```

Fonte: Autor

Figura 33 – Após transformada de Hough



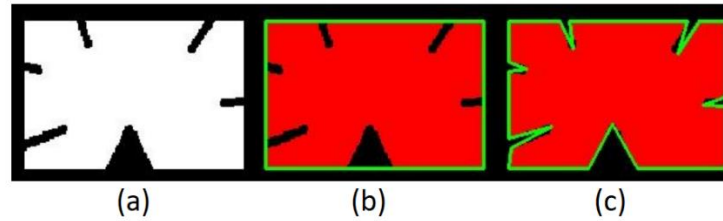
Fonte: Autor

4.4.5.2 Definição do contorno das paletas

Como pode ser observado na Figura 33, os contornos encontrados estão espalhados por toda a imagem após a transformada de Hough. A próxima etapa consiste em identificar os contornos que se assemelham a retângulos, filtrando outros não definidos. Essa identificação é em razão do objeto de interesse, que é a carta de cores retangular com as paletas na forma quadrada.

Para essa identificação será usada a função *arcLength*, disponível pela biblioteca OpenCV, que é capaz de verificar se o contorno referido forma um perímetro, ou seja, se é um contorno fechado. Após a identificação dos contornos fechados é feita uma aproximação de contorno com o objetivo de retirar imperfeições dos valores extraídos. A função *approxPolyDp* da biblioteca OpenCV aproxima formas de contorno para outra forma com menos vértices segundo o parâmetro de precisão, denominado épsilon (ϵ), passado para a função.

Para exemplificar o funcionamento de *approxPolyDp*, considere que seja necessário extrair um quadrado da imagem, porém em virtude das imperfeições a imagem apresentada seja conforme a Figura 34 (a). Usando a função de aproximação com $\epsilon = 10\%$ obtemos o contorno ilustrado pela Figura 34 (b) e a Figura 34 (c) é obtida com $\epsilon = 1\%$.

Figura 34 – Exemplo da função *approxPolyDp*

Fonte: <https://www.docs.opencv.org/>. Acesso em: 13 nov. 2020

Atentando para pequenas diferenças de cenários que nossa pesquisa apresenta e para o fato de que o objeto alvo tem um contorno bem definido (carta de cores) foi usado $\epsilon = 1\%$ no programa final após resultados satisfatórios nos testes.

A Figura 35 mostra o trecho do código referente à aproximação de contornos com os parâmetros descritos anteriormente. A Figura 36 ilustra os contornos após a aproximação realizada nos que foram extraídos na transformada de Hough, ilustrada na Figura 33.

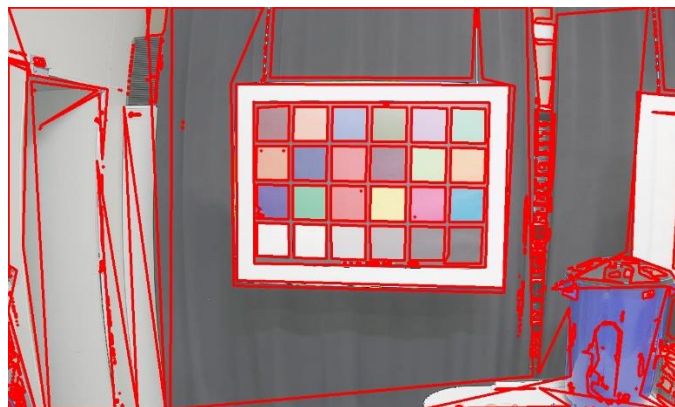
Figura 35 – Aproximação de contornos

```

15. #Filtro de contornos retangulares
16. for contour in contours:
17.     epsilon = 0.1*cv2.arcLength(contour,True)
18.     curve= cv2.approxPolyDP(contour,epsilon,True)

```

Fonte: Autor

Figura 36 – Após a aproximação de contornos com $\epsilon = 1\%$ 

Fonte: Autor

De acordo com o que foi notado na Figura 36 ainda existem contornos detectados que não são os das paletas de cores e para excluir os indesejados ao máximo foram aplicadas três regras descritas a seguir nos contornos remanescentes:

1. Deve ser menor que 5% da imagem, ponderando que temos 24 paletas presentes na imagem teremos no máximo 4.17% da imagem para cada paleta;
2. Deve ser maior que 0,05% da imagem, valor extraído de forma empírica através da distância máxima desejada;
3. A forma do contorno deve ser quadrada, pois as paletas têm esse formato. Para ser considerado um quadrado foi utilizado uma tolerância de 1,5%, ou seja, sendo R a razão entre os dois vértices do contorno, R deve ser menor que 0,985 e maior que 1,015.

A Figura 37 mostra o trecho do código da aplicação das três regras através da estrutura condicional *if*. A função *is_square* foi desenvolvida para identificar um contorno quadrado e está disponível para consulta detalhada no ANEXO A.

A Figura 38 ilustra o resultado após a aplicação dos três filtros nos contornos conforme mostrado na Figura 36. Segundo o que foi observado, os contornos que não são pertencentes a carta foram filtrados, restando apenas as paletas de cores.

Figura 37 – Três regras para filtrar contornos indesejados

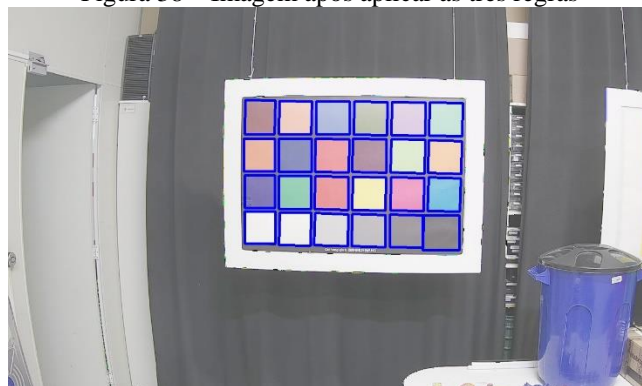
```

20. #leitura do comprimento e largura
21. width, height = image.shape[1], image.shape[0]
22.
23. #definicao das areas
24. image_area = width * height
25. minimum_area = 0.05 * image_area
26. maximum_area = 0.005 * image_area
27.
28. if minimum_area < cv2.contourArea(curve) < maximum_
    area and is_square(curve):
29.     swatches.append(as_int_array(cv2.boxPoints(cv2.
        minAreaRect(curve))))

```

Fonte: Autor

Figura 38 – Imagem após aplicar as três regras



Fonte: Autor

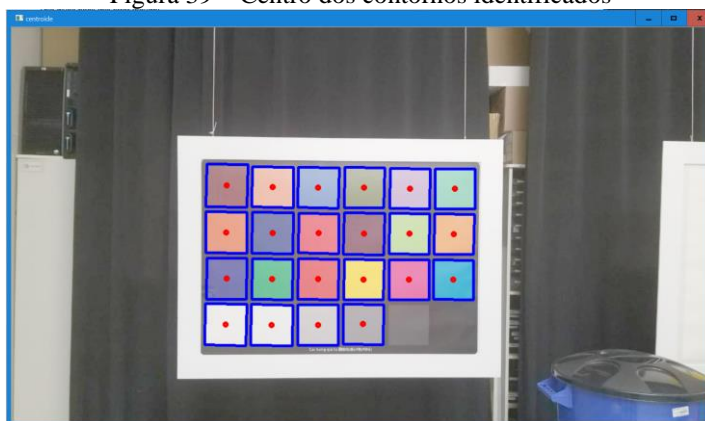
4.4.5.3 Estimativa das paletas restantes

As próximas etapas da segmentação consistem em estimar a região provável das 24 paletas de cores presentes na carta. O objetivo é conferir se o algoritmo foi capaz de detectar todas as paletas e caso não tenha atingido o objetivo, será preciso estimar as regiões restantes. Essas etapas consistirão em analisar e tomar decisões a partir das coordenadas extraídas anteriormente e não a partir do processamento da imagem. Tendo isso em consideração, os processos serão descritos, e os algoritmos apresentados quando necessário, contudo, não entraremos em detalhe sobre o código implementado.

Nota-se que após aplicar as três regras a maioria dos contornos indesejados foram retirados, restando apenas as paletas de cores. No entanto, em alguns testes foi notado que certas paletas não foram detectadas, como observado na Figura 39 em que as últimas duas paletas, que correspondem ao cinza e ao preto, não foram detectadas. Isso ocorre devido ao contraste entre o fundo da carta e essas duas paletas, que é agravado em consequência da distância da imagem. Esse fato levou à necessidade da etapa de estimativa das paletas restantes apresentada a seguir.

A base para as próximas etapas será o centro das paletas de cores, visto que a partir dele serão conferidos e extraídos os pixels amostrais de cores. As coordenadas dos centros dos contornos remanescentes foram identificadas através da média aritmética das arestas de cada contorno e a Figura 39 mostra o resultado.

Figura 39 – Centro dos contornos identificados



Fonte: Autor

O objetivo da próxima etapa consiste em agrupar todos os contornos pertencentes a mesma linha na carta de cores em ordem crescente em um vetor, respeitando a ordem da esquerda para a direita da imagem. Então, os contornos são agrupados em quatro vetores a partir da semelhança do valor da coordenada y de seu centro. Após o agrupamento eles são ordenados

em ordem crescente da coordenada x . Com esses quatro vetores é possível conferir se foram extraídos os 24 contornos referentes às paletas de cores e caso não tenha sido satisfeito esse número será imprescindível estimar os centros das paletas faltantes. A premissa do algoritmo é que tenha no mínimo 20 contornos identificados e obrigatoriamente que duas linhas (vetores) tenham os seis contornos. Caso esses requisitos não sejam satisfeitos o programa retorna a mensagem para o usuário inserir outra imagem.

Após conferir os requisitos, o primeiro passo consiste em calcular a distância entre os centroides adjacentes nas duas primeiras linhas e armazenar esses valores no vetor de média M . Então é calculada a média aritmética e o desvio padrão (D_p) usando (B.2) e (B.4). Depois de o desvio padrão definido, são descartados os valores em M que desviam em mais de 5% de D_p e o processo é repetido até que nenhum valor seja descartado. Ao fim do processo teremos em d a distância padrão entre dois centroides, que será a média dos valores remanescentes em M . Com a distância d definida, a carta de cores é percorrida com o objetivo de encontrar as paletas nas quais não estão definidos os centroides.

Após testes com o algoritmo de detecção de contorno, notou-se que os erros acontecem geralmente nos contornos da paleta preta e cinza, segundo a Figura 38. Por esse motivo, a paleta na cor ciano foi estabelecida como a origem e a primeira conferência será feita percorrendo o eixo y com o objetivo de conferir se o centro da paleta preta está definido. Lembrando que o centro da paleta ciano é o centroide com maior valor na coordenada x e y devido a sua posição na carta, como pode ser observado na Figura 40.

Figura 40 – Paleta ciano é a origem para percorrer a carta



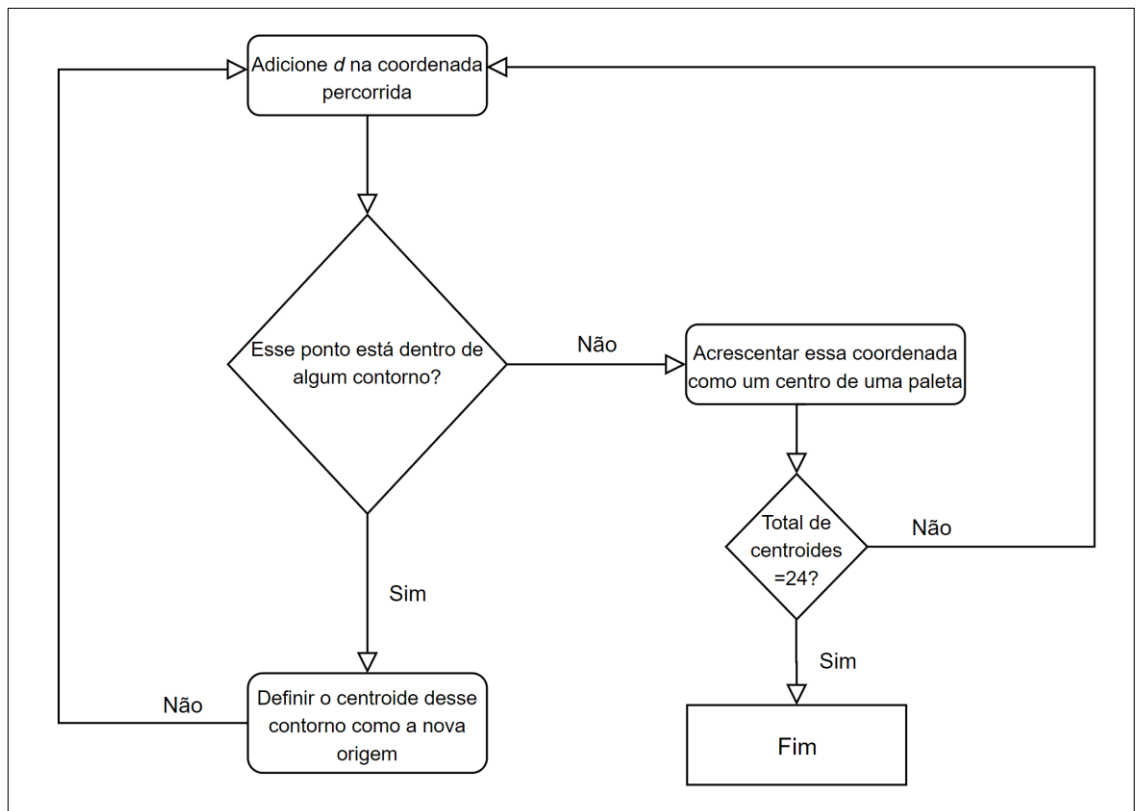
Fonte: Autor

É adicionado d a coordenada y com o objetivo de conferir se há um contorno envolvendo esse ponto, e caso não haja o centro será definido através desse novo ponto. A Figura 41 mostra o algoritmo usado para a conferência, que é realizada 3 vezes no sentido y , para verificar a paleta preta e cinco vezes no sentido x , a partir da paleta preta, com o objetivo de conferir as paletas cinzas. Caso ainda não tenham sido localizadas as 24 paletas, é feita a

conferência de forma interativa a partir das paletas à esquerda da ciano, usando o mesmo algoritmo descrito na Figura 41, porém descontado uma interação na verificação no eixo x e no eixo y. Ao fim desse processo temos as coordenadas de todos os centros das 24 paletas de cores que serão utilizadas para extrair os pixels de amostragem.

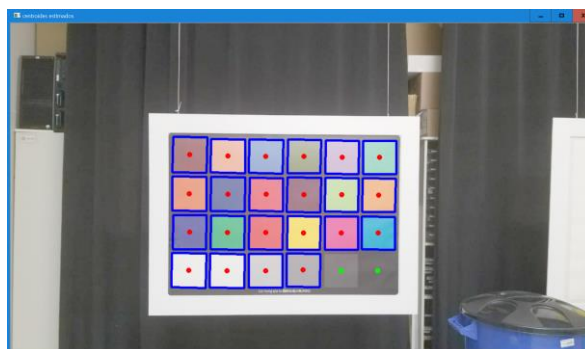
A Figura 42 mostra os centros definidos pelo processamento de imagem em vermelho e os estimados em verde.

Figura 41– Fluxograma para percorrer a carta de cores



Fonte: Autor

Figura 42 – Todos os centros das paletas definidos



Fonte: Autor

4.4.5.4 Extração dos pixels de amostragem

Para a comparação de cores entre imagens será fundamental utilizar um valor de referência para cada paleta de cores. Ele corresponderá ao valor médio de cor dos 100 pixels, sendo um quadrado de aresta de 10 pixels centrado no centroide encontrado de cada paleta.

Ao final, teremos 24 valores no formato RGB, ou seja, em três canais de 8 bits cada de acordo 3.2.1.3, que representam a cor de cada paleta.

4.4.6 Espaço CIELAB

Esta seção descreve o algoritmo implementado para a transformação de um pixel de sRGB para CIELAB. Essa transformação se faz necessária tendo em vista que o método de comparação CIE2000 usado nesta pesquisa requer que as cores comparadas estejam no espaço CIELAB, segundo explicado em 2.3.6.

A primeira etapa consiste em normalizar os valores do pixel para um intervalo entre 0 e 1. Consoante ao que foi explanado em 3.2.1.3, a normalização de um valor em 8 bits é realizada efetuando a divisão por 255 (valor máximo do canal). Após a normalização, é realizado a transformada do espaço sRGB para o CIE XYZ. A Figura 43 mostra o código implementado da função *srgb2xyz*, responsável pela transformação, usando as equações (2.46) à (2.49).

Figura 43 – Código da função *srgb_para_xyz*

```

31. #matriz padrao para conversao
32. M = np.array([[0.412453, 0.357580, 0.180423],
33.               [0.212671, 0.715160, 0.072169],
34.               [0.019334, 0.119193, 0.950227]])
35.
36. #funcao responsavel por fazer conversao de srgb par
a XYZ
37. def srgb_para_xyz(rgb):
38.     for value in rgb:
39.         value = float(value) / 255
40.         mask = rgb > 0.04045
41.         rgb[mask] = np.power((rgb[mask] + 0.055) / 1.05
5, 2.4)
42.         rgb[~mask] /= 12.92
43.     return rgb @ M.T.astype(arr.dtype)

```

Fonte: Autor

Com a imagem no espaço CIE XYZ é possível fazer a transformação para o espaço CIELAB. Para fazer essa conversão foi posta em execução a função *xyz_para_cielab*

desenvolvida pelo autor, em que os parâmetros passados são: o pixel no espaço CIE XYZ e os valores *White-point* de acordo a Tabela 1.

Na Figura 44 temos o código implementado usando as equações (2.13) a (2.16).

Figura 44 – Código da função *srgb_para_xyz*

```

45. #funcaoresponsavel por fazer conversao de srgb para
    XYZ
46. def xyz_para_cielab(xyz, white_point):
47.     xyz = xyz / white_point
48.
49.     mask = xyz > 0.008856
50.     xyz[mask] = np.cbrt(xyz[mask])
51.     xyz[~mask] = 7.787 * xyz[~mask] + 16. / 116.
52.
53.     x, y, z = xyz[..., 0], xyz[..., 1], xyz[..., 2]
54.
55.     L = (116. * y) - 16.
56.     a = 500.0 * (x - y)
57.     b = 200.0 * (y - z)
58.
59.     return np.concatenate([x[..., np.newaxis] for x
    in [L, a, b]], axis=-1)

```

Fonte: Autor

4.4.7 CIE2000

Após a implementação da transformada de um pixel para sua representação no espaço CIELAB, o algoritmo de comparação pode ser desenvolvido usando o modelo CIE2000.

Segundo o que foi exposto em 2.3.6, para a comparação entre as cores é essencial que seus valores estejam no espaço CIELAB, com esses valores usamos as equações de (2.27) a (2.45) para encontrarmos o ΔE_{00} entre duas cores. Essa implementação é mostrada na Figura 45.

Figura 45 – Código da função deltaE_2000

```

1. Def deltaE_2000(lab1, lab2, kL=1, kC=1, kH=1):
2.
3.     lab1 = np.asarray(lab1)
4.     lab2 = np.asarray(lab2)
5.
6.     L1, a1, b1 = np.rollaxis(lab1, -1)[:3]
7.     L2, a2, b2 = np.rollaxis(lab2, -1)[:3]
8.
9.     c7 = C_linha ** 7
10.    G = 0.5 * (1 - np.sqrt(c7 / (c7 + 25 ** 7)))
11.    scale = 1 + G
12.    C1, h1 = _cart2polar_2pi(a1 * scale, b1)
13.    C2, h2 = _cart2polar_2pi(a2 * scale, b2)
14.
15.    # Definicao de L_term = delta L / S1 (primeiro termo)
16.    L_linha = 0.5 * (L1 + L2)
17.    tmp = (L_linha - 50) ** 2
18.    SL = 1 + 0.015 * tmp / np.sqrt(20 + tmp)
19.    L_term = (L2 - L1) / (kL * SL)
20.
21.    # Definicao de C_term = delta C / Sc
22.    C_linha = 0.5 * (C1 + C2) # new coordiantes
23.    SC = 1 + 0.045 * C_linha
24.    C_term = (C2 - C1) / (kC * SC)
25.
26.    # Definicao de h
27.    h_diff = h2 - h1
28.    h_sum = h1 + h2
29.    CC = C1 * C2
30.
31.    # delta h
32.    dH = h_diff.copy()
33.    dH[h_diff > np.pi] -= 2 * np.pi
34.    dH[h_diff < -np.pi] += 2 * np.pi
35.    dH[CC == 0.] = 0. # if r == 0, dtheta == 0
36.    dH_term = 2 * np.sqrt(CC) * np.sin(dH / 2)
37.
38.    # h linha
39.    H_linha = h_sum.copy()
40.    mask = np.logical_and(CC != 0., np.abs(h_diff) >
    np.pi)
41.    H_linha[mask * (h_sum < 2 * np.pi)] += 2 * np.pi
42.    H_linha[mask * (h_sum >= 2 * p.pi)] -= 2 * np.pi
43.    H_linha[CC == 0.] *= 2
44.    H_linha *= 0.5
45.    T = (1 -
46.         0.17 * np.cos(H_linha - np.deg2rad(30)) +
47.         0.24 * np.cos(2 * H_linha) +
48.         0.32 * np.cos(3 * H_linha + np.deg2rad(6))
49.         -
50.         0.20 * np.cos(4 * H_linha - np.deg2rad(63))
51.         )

```

```

50.     SH = 1 + 0.015 * C_linha * T
51.
52.     #definicao de dH_term = delta H / SH
53.     H_term = dH_term / (kH * SH)
54.
55.     # definicao de Rt multiplicado pelo deltaC/Sc e
    deltaH/Sh (ultimo termo)
56.     c7 = C_linha ** 7
57.     Rc = 2 * np.sqrt(c7 / (c7 + 25 ** 7))
58.     dtheta = np.deg2rad(30) * np.exp(-
    ((np.rad2deg(H_linha) - 275) / 25) ** 2)
59.     R_t = -np.sin(2 * dtheta) * Rc * C_term *
    H_term
60.     # definicao do delta e
61.     dE2 = L_term ** 2
62.     dE2 += C_term ** 2
63.     dE2 += H_term ** 2
64.     dE2 += R_t
65.     resposta = np.sqrt(np.maximum(dE2, 0))
66.     if unroll:
67.         resposta = resposta[0]
68.     return ans
69.

```

Fonte: Autor

4.5 CONCLUSÕES

Este Capítulo descreveu o desenvolvimento do algoritmo desenvolvido em Python. Foram apresentados os principais trechos dos códigos e explicadas as principais bibliotecas. O algoritmo usou os conceitos apresentados de visão computacional no Capítulo 3 para fazer o reconhecimento das paletas de cores e definir a região central de cada paleta. Os conceitos de colorimetria do Capítulo 2 foram utilizados para desenvolver o algoritmo capaz de implementar a transformação de um pixel do sistema sRGB para o CIELAB, necessário para fazer a comparação entre cores no sistema CIE2000.

5 SIMULAÇÕES E RESULTADOS

5.1 INTRODUÇÃO

Com o objetivo de avaliar o desempenho das estratégias propostas por este trabalho de conclusão de curso, foi realizada uma série de simulações cujos resultados e avaliações mais relevantes são mostrados neste Capítulo. Os casos apresentados buscam ilustrar as principais características do comparador de cores desenvolvido ao longo desta pesquisa, assim como reproduzir algumas situações de sucesso e falha. Os nomes dos fabricantes das câmeras serão omitidos ao longo das simulações para preservar as marcas.

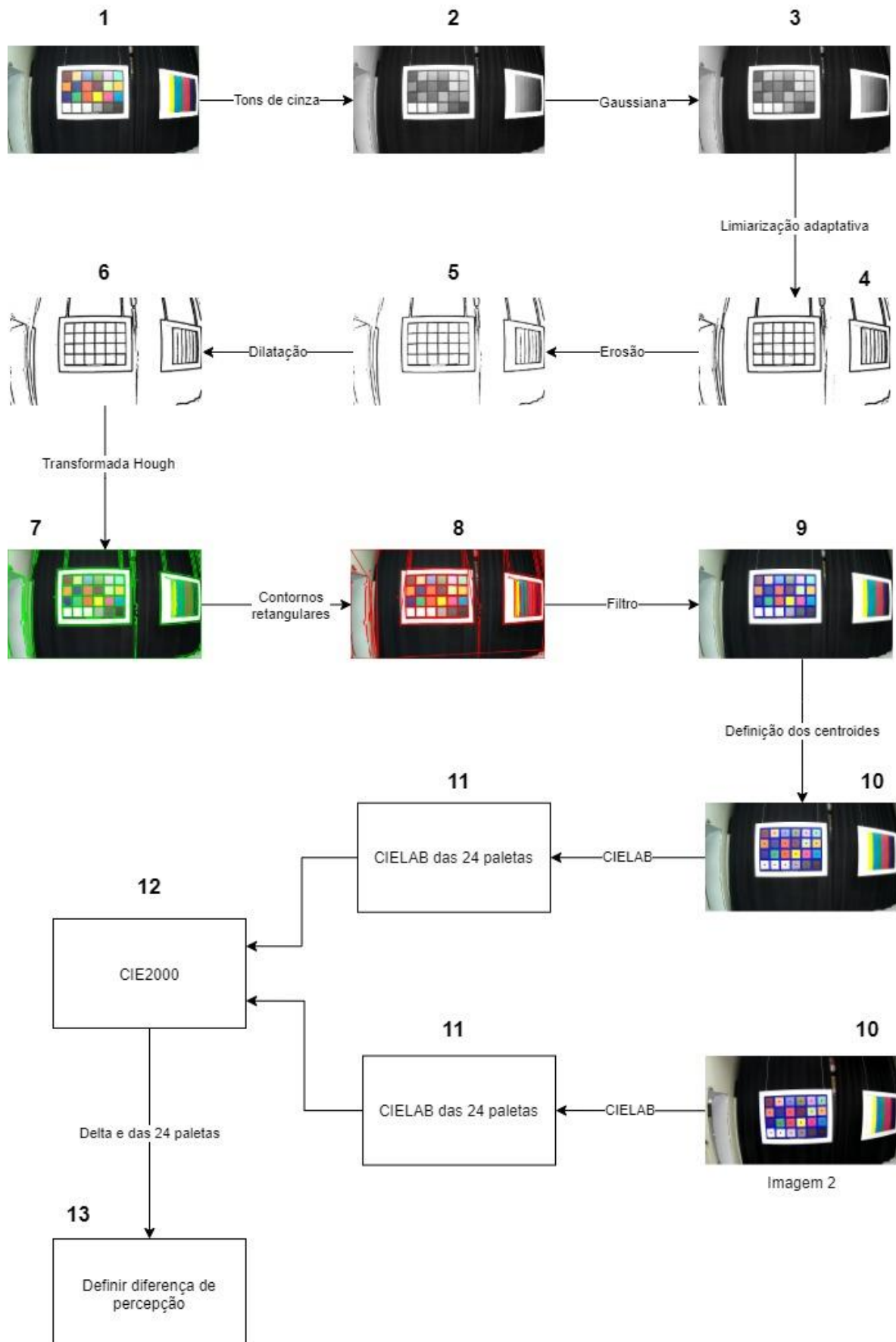
5.2 RESULTADOS DAS ETAPAS

A Figura 46 mostra os resultados de todos os processos em forma de fluxograma, divididos em treze etapas, desde a imagem original capturada pela câmera até a definição da diferença de cores entre outra imagem. O fluxograma da Figura 46 é uma analogia ao esquemático mostrado de um sistema de visão computacional da Figura 14.

A etapa um é a de aquisição de imagem, explanada no item 3.3, as etapas de dois à seis representam o pré-processamento discutido teoricamente em 3.4 e 3.6 e a implementação está em 4.4.1, 4.4.2 e 4.4.4. A segmentação é composta pelas etapas de sete à dez e foram descritas em 3.5 e a implementação demonstrada em 4.4.5.

A extração de atributos representa a transformada do pixel para o espaço CIELAB, detalhado no item 2.3.3 e concretizado em 4.4.6. A etapa de classificação consiste em realizar o cálculo entre a diferença de cores usando o método CIE2000, explicado em 2.3.6 e implementado em 4.4.7 e através da Tabela 2 podemos classificar a diferença encontrada.

Figura 46 – Processo completo



Fonte: Autor

5.3 LOCALIZAÇÃO DA CARTA DE CORES

O teste para verificação da carta de cores consiste em certificar a acuracidade da aplicação em encontrar a região de interesse que são as vinte e quatro paletas de cores, com o intuito de validar os casos em que o software funciona e as situações em que ele não alcança o objetivo. Esses testes possibilitam identificar falsos positivos e negativos na utilização da aplicação.

Nesse contexto a aplicação deve reconhecer corretamente a carta de cores no cenário. Como a localização é independente do sensor e da câmera, foram utilizadas imagens obtidas de três diferentes dispositivos intitulados nesse trabalho como X, Y e Z, conforme descrito a seguir:

- **Dispositivo X:** câmera de segurança IP com sensor com resolução de 5MP e lente varifocal (2,7 mm à 13,5 mm);
- **Dispositivo Y:** câmera de segurança IP com sensor com resolução de 2MP e lente fixa de 3,6 mm;
- **Dispositivo Z:** câmera de um *smartphone* com sensor com resolução de 14MP e lente fixa de 26 mm.

Tendo em vista que a posição da carta na imagem deve respeitar a angulação descrita, isto é, deve sempre estar na mesma posição, o parâmetro de maior impacto nesse teste é o fundo do cenário. Por esse motivo, esse será o alvo maior dessa avaliação.

A seguir são descritos os casos de testes realizados variando o tamanho da carta na imagem, assim como os comentários dos resultados expostos nas tabelas. Além de uma imagem ilustrativa de uma amostra, em cada caso é apresentado uma tabela na qual é explicitada a quantidade de imagens (amostras) testadas e a assertividade total dos testes, seguindo a equação (5.1).

$$Assertividade = \frac{n^{\circ} \text{ de contornos detectados}}{24 \times n^{\circ} \text{ de amostras}} \times 100 \quad (5.1)$$

Caso 1: Imagem frontal com a carta de cores ocupando aproximadamente entre 20% e 30% da imagem com fundo preto. A quantidade de imagens testadas e a assertividade em cada teste podem ser observadas na Tabela 6. A Figura 47 ilustra a imagem original do teste e a Figura 48, a Figura 49 e a Figura 50 apresentam os resultados de uma amostra da detecção dos contornos nos testes 1, 2 e 3 respectivamente.

Tabela 6 – Caso 1 do teste de localização da carta

CASO 1		
Dispositivo	Amostras	Assertividade
X	10	100%
Y	10	100%
Z	10	100%

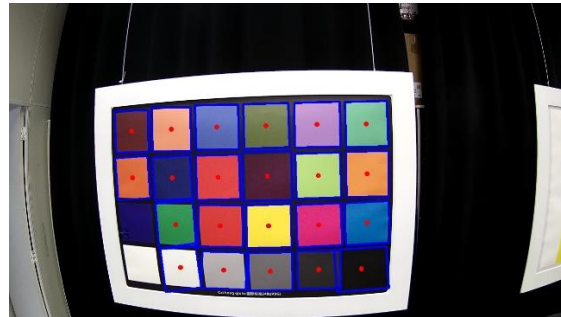
Fonte: Autor

Figura 47 – Imagem original do caso 1, carta representa 30% da imagem



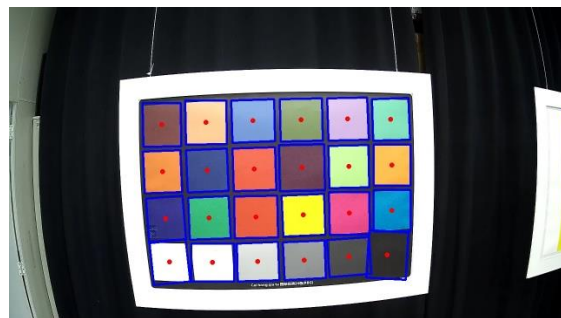
Fonte: Autor

Figura 48 – Foram encontrados os 24 contornos no caso 1 no dispositivo X



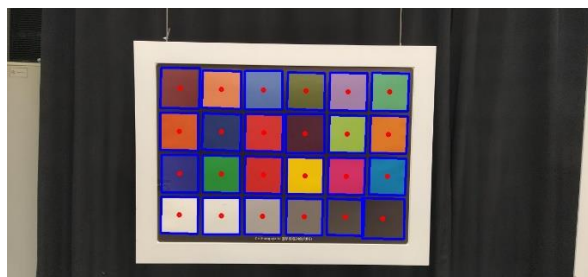
Fonte: Autor

Figura 49 – Foram encontrados os 24 contornos no caso 1 no dispositivo Y



Fonte: Autor

Figura 50 – Foram encontrados os 24 contornos no caso 1 no dispositivo Z



Fonte: Autor

Caso 2: Imagem frontal com a carta de cores ocupando aproximadamente entre 10% e 20% da imagem com fundo preto. A quantidade de imagens testadas e a assertividade em cada teste podem ser observadas na Tabela 7. A Figura 51, a Figura 52 e a Figura 53 apresentam os resultados de uma amostra da detecção dos contornos nos testes 1, 2 e 3 respectivamente.

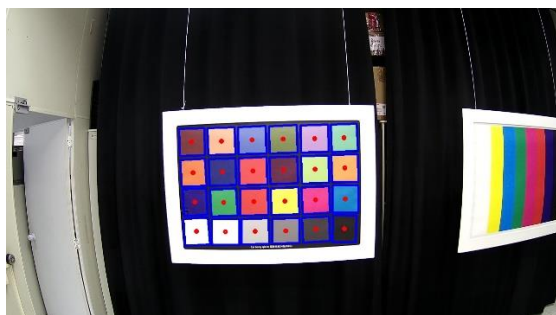
Vale notar que nesse cenário não houve nenhum caso no qual foi necessário usar o algoritmo de estimação de paletas, posto que todas as paletas foram encontradas após a segmentação. Esse cenário de teste foi tido como ideal, por isso foi realizado uma amostragem maior.

Tabela 7 – Caso 2 do teste de localização da carta

CASO 2		
Dispositivo	Amostras	Assertividade
X	20	100%
Y	20	100%
Z	20	100%

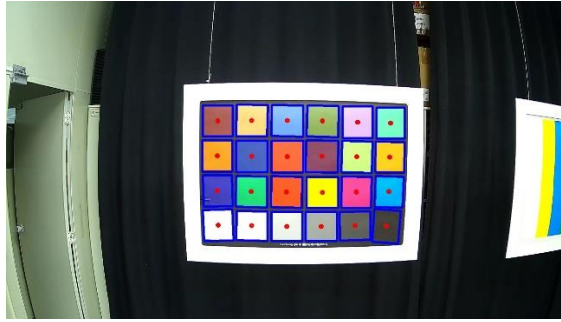
Fonte: autor

Figura 51 – Foram encontrados os 24 contornos no caso 2 no dispositivo X



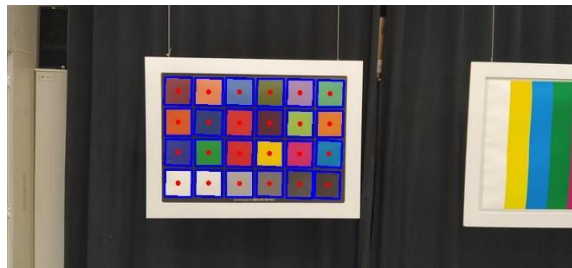
Fonte: Autor

Figura 52 – Foram encontrados os 24 contornos no caso 2 no dispositivo Y



Fonte: Autor

Figura 53 – Foram encontrados os 24 contornos no caso 2 no dispositivo Z



Fonte: Autor

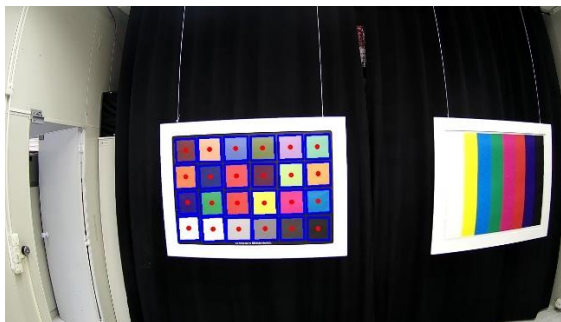
Caso 3: Imagem frontal com a carta de cores ocupando aproximadamente entre 2% e 10% da imagem com fundo preto. A quantidade de imagens testadas e a assertividade em cada teste podem ser observadas na Tabela 8. A Figura 54, a Figura 55 e a Figura 55 apresentam os resultados de uma amostra da detecção dos contornos nos testes 1, 2 e 3 respectivamente.

Tabela 8 – Caso 3 do teste de localização da carta

CASO 3		
Dispositivo	Amostras	Assertividade
X	2	100%
Y	2	100%
Z	2	100%

Fonte: Autor

Figura 54 – Foram encontrados os 24 contornos no caso 3 no dispositivo X



Fonte: Autor

Figura 55 – Foram encontrados os 24 contornos no caso 3 no dispositivo Y



Fonte: Autor

A seguir será descrito o caso 4 dos testes realizados em que foram variados os cenários.

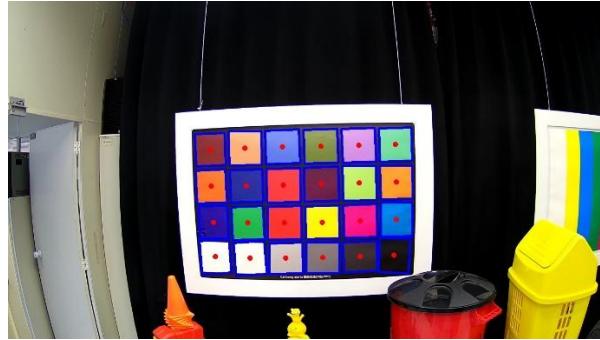
Caso 4: Imagem frontal com a carta de cores ocupando aproximadamente entre 10% e 20% da imagem com fundos aleatórios em ambiente interno. A quantidade de imagens testadas e a assertividade em cada teste podem ser observadas na Tabela 9. A Figura 56, a Figura 57 e a Figura 58 apresentam os resultados de uma amostra da detecção dos contornos nos testes 1, 2 e 3 respectivamente.

Tabela 9 – Caso 4 do teste de localização da carta

CASO 4		
Dispositivo	Amostras	Assertividade
1	2	100%
2	2	100%
3	2	98%

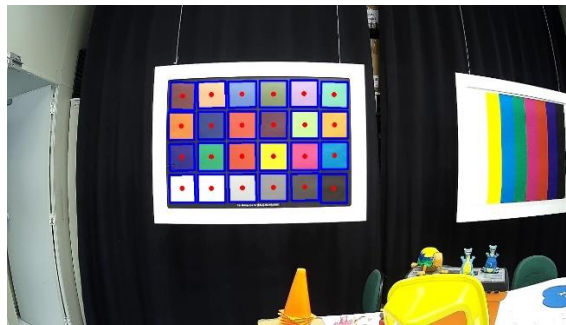
Fonte: Autor

Figura 56 – Foram encontrados os 24 contornos no caso 4 no dispositivo X



Fonte: Autor

Figura 57 – Foram encontrados os 24 contornos no caso 4 no dispositivo Y



Fonte: Autor

Figura 58 – Foram encontrados os 24 contornos no caso 4 no dispositivo Z



Fonte: Autor

Os resultados após a análise dos resultados encontrados nos quatro casos de testes realizados são satisfatórios, tendo em consideração que houve uma assertividade próxima de 100% em todos os casos. Vale ressaltar que a aplicação fez o uso do algoritmo de estimação de

paletas para garantir a identificação das 24 cores corretamente, evidenciado a necessidade desse algoritmo.

5.4 COMPARAÇÃO ENTRE CORES

O teste de comparação de cores é o objetivo final deste estudo, que consiste em estimar a diferença de cores entre duas imagens usando o método CIE2000. Esses testes foram divididos em três etapas: validação inicial, comparação entre imagens do mesmo dispositivo e dispositivos diferentes.

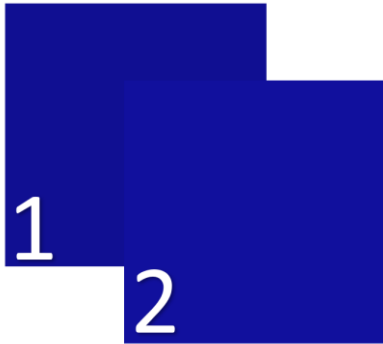
Os testes de validação inicial tiveram como propósito validar a aplicação usando a comparação apenas entre imagens de dispositivos com características definidas e em baixa escala. Esses testes foram necessários para validar a aplicação a seguir para a próxima etapa que são os testes em maior escala.

A segunda etapa dos testes consiste em fazer comparações entre imagens do mesmo dispositivo. O teste tem como finalidade medir a acuracidade e precisão da aplicação, tendo como premissa que imagens originadas do mesmo dispositivo devem ter diferença nula entre as cores.

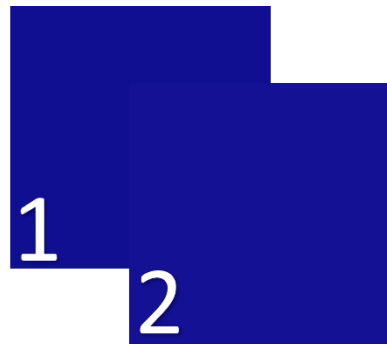
A última etapa simula a aplicação em diferentes condições de iluminações variando a iluminância e o iluminante. Esses testes são realizados comparando a imagem de diferentes dispositivos e avaliando as diferenças causadas no ΔE pela variação da iluminação.

Em todas as etapas foram realizados apenas testes com as câmeras de segurança, uma vez que são as que temos o controle de matéria-prima necessário para os testes.

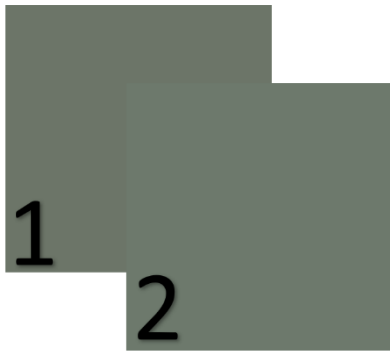
Como foi descrito ao longo do trabalho o objetivo do método proposto é fazer a análise de comparação de cores de forma analítica e não subjetiva. Porém, para efeitos didáticos serão apresentadas a seguir quatro cores sobrepostas e o resultado ΔE entre elas usando o método proposto (Figura 59 a 66). A percepção na diferença de cores pode ser diferente para cada pessoa e por isso as imagens e os resultados serão apresentados sem nenhum comentário ou impressões do autor desse trabalho. Para melhor percepção da diferença é aconselhável aproximar a imagem digitalmente através do zoom na sobreposição dos quadrados.

Figura 59 – $\Delta E = 1,6$ para a cor azul (#13)

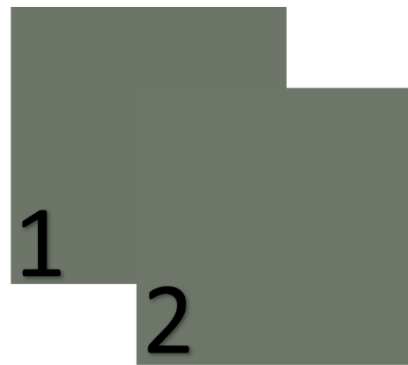
Fonte: Autor

Figura 60 – $\Delta E = 0,5$ para a cor azul (#13)

Fonte: Autor

Figura 61 – $\Delta E = 2,0$ para a cor cinza (#22)

Fonte: Autor

Figura 62 – $\Delta E = 0,5$ para a cor cinza (#22)

Fonte: Autor

Figura 63 – $\Delta E = 1,8$ para a cor vermelha (#22)

Fonte: Autor

Figura 64 – $\Delta E = 0,5$ para a cor vermelha (#15)

Fonte: Autor

Figura 65 – $\Delta E = 1,9$ para a cor verde (#14)

Fonte: Autor

Figura 66 – $\Delta E = 0,3$ para a cor verde (#22)

Fonte: Autor

5.4.1 Validação inicial

Foram realizados cinco casos diferentes de testes nessa etapa, simulando comparações durante a avaliação de qualidade no processo produtivo de câmeras.

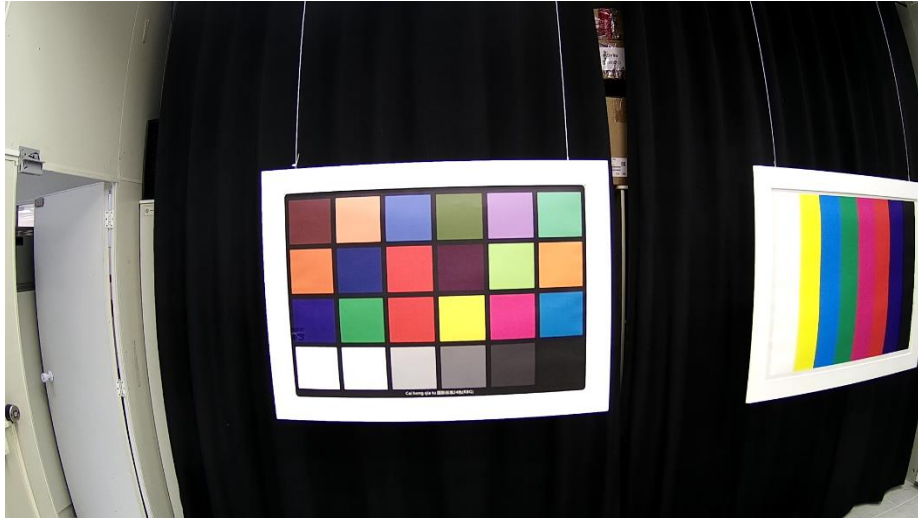
Os quatro primeiros casos (5 a 8) são comparações entre dispositivos do mesmo modelo, porém com variações em sua produção, ou seja, a expectativa é de que os resultados sejam variações mínimas entre as cores das imagens. O quinto caso (9) é uma simulação forçada de diferença de cores, onde foi imposta uma diferença na saturação em uma das imagens para avaliar a capacidade do método de sinalizar essa alteração. A seguir são descritos os testes e os resultados dos cinco testes.

Caso 5: As amostras do teste são do mesmo lote de fabricação, isto é, produzidos com os mesmos fabricantes de matéria-prima. A seguir é descrito os resultados dos três testes realizados nesse caso.

A Tabela 10 apresenta a diferença entre as cores usando dois dispositivos do tipo X, ou seja, duas câmeras com lentes varifocais, no caso de teste 5. Na tabela são apresentadas as cores da carta no espaço CIELAB para as duas imagens capturadas (Figura 67 e Figura 68) e a diferença entre elas é apresentada usando a fórmula CIE2000. Como era esperado, por se tratar de uma câmera fabricada com a mesma matéria-prima, não há nenhuma diferença maior do que um, sendo a posição 23 a que apresenta maior diferença (0,826).

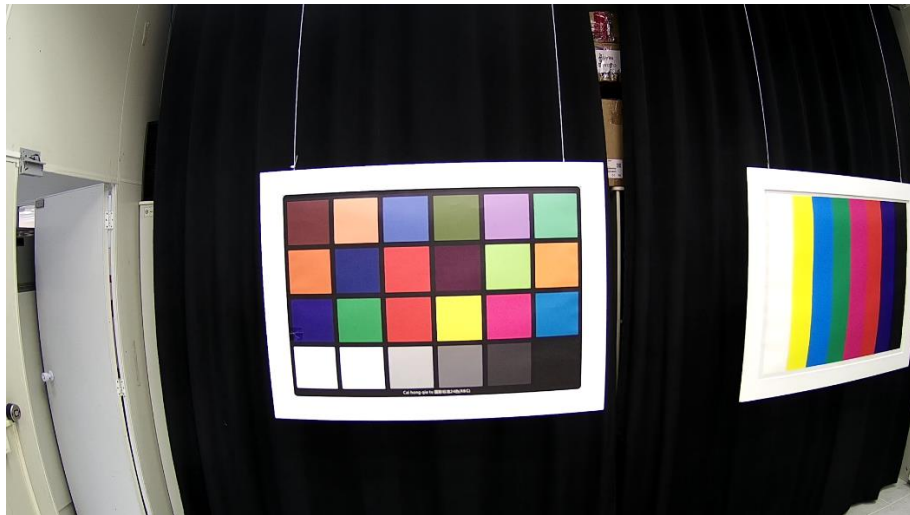
Outros dois testes no mesmo cenário são apresentados na Tabela 12 denominados como teste 2 e 3, sendo que apresentam os mesmos resultados esperados descritos anteriormente.

Figura 67 – Imagem 1 do caso 5 usando o dispositivo do tipo X



Fonte: Autor

Figura 68 – Imagem 2 do caso 5 usando o dispositivo do tipo X



Fonte: Autor

Tabela 10 – Caso 5 do teste usando amostras do dispositivo 1

Cor	Imagem 1			Imagem 2			CIE2000
	L	a	b	L	a	b	
#1	12,278	22,642	10,832	12,329	22,785	10,757	0,114
#2	71,499	36,23	44,85	71,56	36,01	44,85	0,138
#3	39,399	26,725	-56,179	39,532	26,904	-56,272	0,139
#4	36,68	-12,349	27,2	36,953	-12,231	27,032	0,233
#5	58,762	40,887	-32,766	59,019	40,945	-33,294	0,33
#6	69,91	-42,348	21,648	70,064	-42,477	21,48	0,158
#7	55,246	63,403	59,19	55,245	63,403	58,966	0,33
#8	6,406	19,933	-32,584	6,625	20,53	-33,294	0,345
#9	52,963	75,066	54,623	53,466	75,736	54,971	0,502
#10	7,638	18,275	0,066	8,003	18,782	0,099	0,342
#11	86,011	-39,057	62,648	86,501	-39,298	62,899	0,33
#12	67,632	42,306	64,657	67,89	41,703	64,506	0,336
#13	4,409	22,411	-34,55	4,564	22,918	-34,955	0,245
#14	43,292	-44,013	34,481	43,53	-44,073	34,35	0,229
#15	49,513	72,438	55,459	49,581	71,323	55,411	0,073
#16	97,185	-21,182	91,064	97,19	-21,145	90,746	0,063
#17	51,287	76,903	25,358	51,951	77,591	25,308	0,683
#18	30,466	21,48	-55,387	21,022	21,487	-55,789	0,472
#19	0,051	-0,012	0,0150	0,030	-0,052	0,015	0,062
#20	0,999	-0,03	0,106	1,200	0,102	0,211	0,251
#21	71,227	4,406	3,684	71,801	4,869	3,84	0,69
#22	31,796	4,439	1,562	32,831	4,384	1,544	0,826
#23	4,747	1,623	0,42	5,334	2,032	0,333	0,656
#24	1,32	0,251	0,145	1,477	0,304	0,12	0,123

Fonte: Autor

Caso 6: Os dispositivos 1 e 2 são de lotes diferentes de fabricação, porém possuem o mesmo fornecedor da matéria-prima. Em concordância com o que foi percebido na Tabela 12 não houve diferenças superiores a um entre a diferença de cores das imagens.

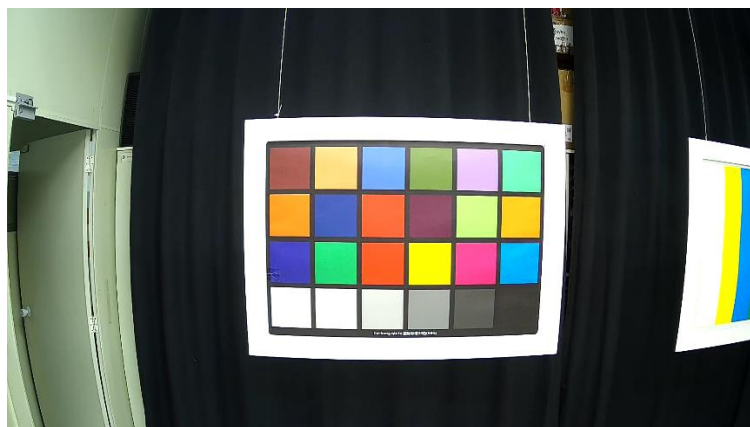
Caso 7: Os dispositivos 1 e 2 são de fornecedores de lentes diferentes, porém com a mesma especificação. Como era esperado, por possuir especificação idêntica de lentes, a diferença entre as cores se manteve dentro da taxa aceitável, segundo o que foi notado na Tabela 12.

Caso 8: Os dispositivos 1 e 2 são de fornecedores diferentes de sensor, entretanto possuem a mesma especificação. O sensor, por ter maior impacto no aspecto de cores, teve uma diferença maior que os casos anteriores, o que pode ser notado detalhadamente na Tabela 11.

Foram observadas dez cores com diferenças superiores a um e inferiores a dois, isso representa uma diferença mais significativa que os casos anteriores. Esse caso será mais explorado nos testes apresentados no tópico 5.4.3.

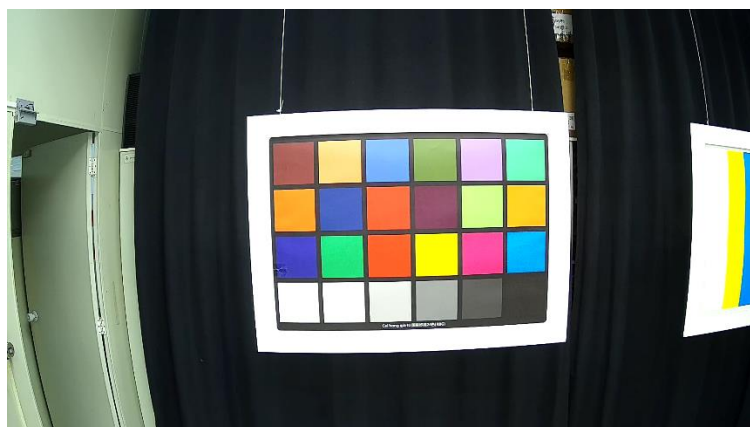
Caso 9: As imagens 1 e 2 são oriundas do mesmo dispositivo, todavia é configurado com uma diferença de 3 unidades (em uma escala de 100) na saturação da cor. O objetivo é extrapolar as diferenças e medir a capacidade da aplicação em sinalizá-las. Em conformidade com o que foi percebido na Tabela 12 a maioria das diferenças entre as cores é superior a dois, o que representa uma fácil percepção, como pode ser contemplado na Figura 71 e na Figura 72.

Figura 69 – Imagem 1 do caso 8



Fonte: Autor

Figura 70 – Imagem 2 do caso 8



Fonte: Autor

Tabela 11 – Caso 8 do teste usando a câmara do teste 1

Cor	Imagem 1			Imagem 2			CIE2000
	L	a	b	L	a	b	
#1	25,049	33,993	23,627	25,792	34,108	22,641	0,825
#2	81,803	10,357	70,481	83,285	7,639	68,869	1,894
#3	57,404	18,711	-66,518	58,887	15,454	-64,133	1,872
#4	52,01	-31,659	44,359	53,483	-33,697	44,241	1,706
#5	74,693	41,857	-38,4	74,939	41,189	-38,019	0,29
#6	89,215	-69,409	34,969	89,331	-67,996	30,772	1,44
#7	61,855	56,268	70,245	62,49	54,613	70,315	0,93
#8	21,194	36,288	-60,064	22,226	37,566	-62,145	0,927
#9	55,148	74,798	64,834	55,299	74,389	64,642	0,173
#10	17,563	28,205	3,569	17,785	27,707	2,99	0,444
#11	92,419	-49,001	75,522	92,438	-48,793	74,025	0,372
#12	76,273	20,927	79,398	76,597	20,205	79,523	0,486
#13	18,13	47,524	-67,602	19,764	50,289	-71,415	1,489
#14	72,632	-70,928	59,534	75,03	-72,677	60,323	1,817
#15	54,285	77,153	66,266	54,391	76,858	66,151	0,126
#16	97,14	-21,554	94,479	97,14	-21,554	94,472	0,001
#17	54,595	80,742	25,737	54,776	81,089	21,123	1,989
#18	21,284	21,284	-72,434	54,551	18,785	-70,807	1,584
#19	0,097	-0,021	0,087	0,108	0,09	0,004	0,185
#20	1,238	0,164	0,147	1,008	0,273	0,144	0,208
#21	92,52	-8,401	9,867	93,128	-7,977	8,611	0,959
#22	48,128	-5,735	5,73	49,378	-7,175	5,642	1,969
#23	12,741	-1,856	3,999	12,812	-1,131	2,33	1,696
#24	3,53	0,018	1,067	3,59	-0,248	1,088	0,395

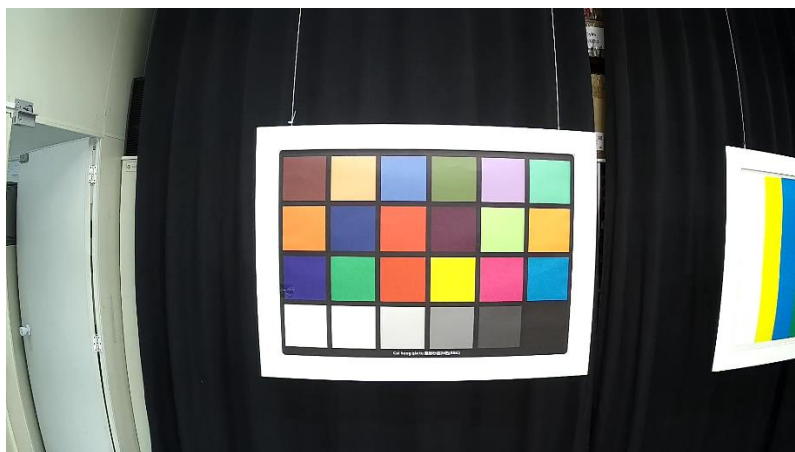
Fonte: Autor

Tabela 12 – Resultado dos casos de comparação

Cor	Caso 5			Caso 6	Caso 7	Caso 8	Caso 9
	Teste 1	Teste 2	Teste 3				
#1	0,114	0,122	0,129	0,197	0,204	0,825	6,452
#2	0,138	0,472	0,232	0,294	0,241	1,894	4,29
#3	0,139	0,152	0,143	0,201	0,183	1,872	3,827
#4	0,233	0,079	0,128	0,097	0,141	1,706	5,367
#5	0,33	0,616	0,421	0,493	0,341	0,29	5,151
#6	0,158	0,177	0,225	0,212	0,229	1,44	8,023
#7	0,33	0,341	0,398	0,376	0,21	0,93	1,587
#8	0,345	0,118	0,236	0,259	0,153	0,927	7,433
#9	0,502	0,164	0,422	0,365	0,355	0,173	1,659
#10	0,342	0,22	0,35	0,319	0,299	0,444	5,228
#11	0,33	0,143	0,23	0,253	0,214	0,372	3,451
#12	0,336	0,097	0,193	0,116	0,102	0,486	0,939
#13	0,245	0,118	0,23	0,234	0,159	1,489	8,914
#14	0,229	0,106	0,174	0,164	0,135	1,817	13,514
#15	0,073	0,13	0,093	0,034	0,092	0,126	2,209
#16	0,063	0,048	0,084	0,113	0,061	0,034	0,154
#17	0,683	0,147	0,34	0,252	0,149	1,989	1,201
#18	0,472	0,088	0,302	0,185	0,118	1,584	12,447
#19	0,128	0,147	0,098	0,107	0,166	0,127	0,439
#20	0,115	0,182	0,132	0,192	0,193	0,201	0,512
#21	0,69	0,334	0,496	0,438	0,031	0,959	2,474
#22	0,826	0,411	0,57	0,472	0,372	1,969	3,16
#23	0,656	0,238	0,438	0,459	0,235	1,696	1,547
#24	0,123	0,08	0,147	0,126	0,073	0,395	0,42

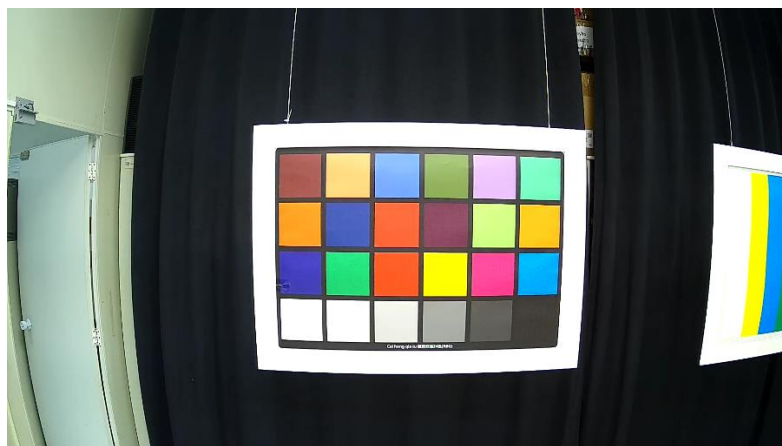
Fonte: Autor

Figura 71 – Imagem 1 do caso 9



Fonte: Autor

Figura 72 – Imagem 2 do caso 9



Fonte: Autor

Apesar da etapa um ter sido um sucesso em questões de resultados esperados, ela apenas habilita a aplicação a seguir para os testes em maior escala que serão apresentados a seguir para conseguirmos ter uma conclusão efetiva do método proposto.

5.4.2 Comparação entre imagens do mesmo dispositivo

Considerando o método proposto nesse trabalho como uma mensuração da diferença entre duas cores, sempre haverá incertezas de medição associadas ao resultado. Estas podem estar relacionadas a erros sistemáticos que levam os resultados sempre para o mesmo lugar, alterando na exatidão, ou também podem ser originadas aleatoriamente, diretamente relacionada à precisão da medida (TAYLOR, 1997).

A precisão de uma medida está diretamente relacionada com a repetibilidade dos resultados (MAZZOLA, 2021). Ou seja, em idênticas condições, o quão próximos entre si estão os valores obtidos. Assim sendo, faremos testes comparando imagens repetidamente em condições idênticas e através do desvio padrão estimaremos a precisão de nossa aplicação.

Em contrapartida, a exatidão ou acurácia está relacionado ao grau de concordância entre o valor medido e o valor verdadeiro. Portanto, usaremos o fato de que a diferença de cores entre imagens originadas do mesmo dispositivo no mesmo cenário é nula, ou seja, sabemos o valor verdadeiro da comparação, para dimensionar a acurácia da aplicação desenvolvida.

Foram realizados dois tipos de teste com o objetivo de definir a acuracidade e a precisão da aplicação, conforme descritos a seguir:

1. Teste com a câmeras na mesma posição;
2. Teste com pequenas variações na posição da câmera;

Os testes apresentados nas próximas seções serão referentes a dois tipos de modelos de câmera de segurança IP conforme descritos a seguir:

- **Dispositivo A:** câmera de segurança IP com sensor com resolução de 2MP e lente fixa de 3,6 mm;
- **Dispositivo B:** câmera de segurança IP com sensor com resolução de 4MP e lente fixa de 3,6 mm;

5.4.2.1 Teste na mesma posição

O primeiro teste consiste em fazer comparativos entre imagens do mesmo dispositivo instalado na mesma posição e condições de cenário, apenas em instantes distintos. Foram extraídas trinta imagens nessas condições, sendo realizadas comparações entre todas, ou seja, foram feitas 435 comparativos para cada paleta de cor e câmera testada.

A Tabela 13 e Tabela 14 mostram os resultados ΔE em ordem crescente de média para cada paleta da carta de cores desse teste usando os dispositivos A e B respectivamente. Notamos por elas que obtivemos um resultado satisfatório, pois as máximas e as médias estiveram próximas a zero ($< 0,1$) para todas as cores, como era esperado, sendo que a média geral para o dispositivo A foi de 0,0251 e para o dispositivo B de 0,0290.

Observamos que para ambos os modelos as seguintes cores apresentaram as cinco maiores médias nos ΔE calculados, consequentemente as piores exatidões: cinza 3.5 (#21), cinza 5 (#22), magenta (#17), verde amarelado (#11) e verde azulado (#6).

Tabela 13 – Resultado (ΔE) do dispositivo A na mesma posição

Cor	Mínima	Máxima	Média	Desvio Pad.
#4	0,0000	0,0711	0,0145	0,0092
#14	0,0000	0,086	0,015	0,0102
#18	0,0000	0,0878	0,0151	0,0102
#12	0,0001	0,1301	0,0152	0,011
#19	0,0002	0,1246	0,0152	0,0106
#15	0,0001	0,0888	0,0153	0,011
#1	0,0000	0,0634	0,0154	0,0096
#3	0,0001	0,0796	0,0155	0,0098
#10	0,0000	0,0892	0,0155	0,0111
#13	0,0001	0,0663	0,0155	0,01
#24	0,0001	0,0855	0,0155	0,0107
#2	0,0001	0,0886	0,0156	0,0111
#7	0,0001	0,1746	0,0158	0,0122
#9	0,0000	0,0779	0,0159	0,0099
#20	0,0000	0,1170	0,0159	0,011
#16	0,0000	0,1403	0,016	0,0121
#8	0,0000	0,0791	0,0161	0,0108
#5	0,0001	0,082	0,0162	0,0096
#23	0,0001	0,0602	0,0168	0,0089
#6	0,0000	0,0994	0,0487	0,0274
#11	0,0001	0,1228	0,0602	0,0353
#17	0,0004	0,1289	0,0620	0,0368
#22	0,0002	0,1386	0,0685	0,0389
#21	0,0000	0,1356	0,0686	0,0396
Geral	0,0000	0,1746	0,0251	0,0267

Fonte: Autor

Tabela 14 – Resultado (ΔE) do dispositivo B na mesma posição

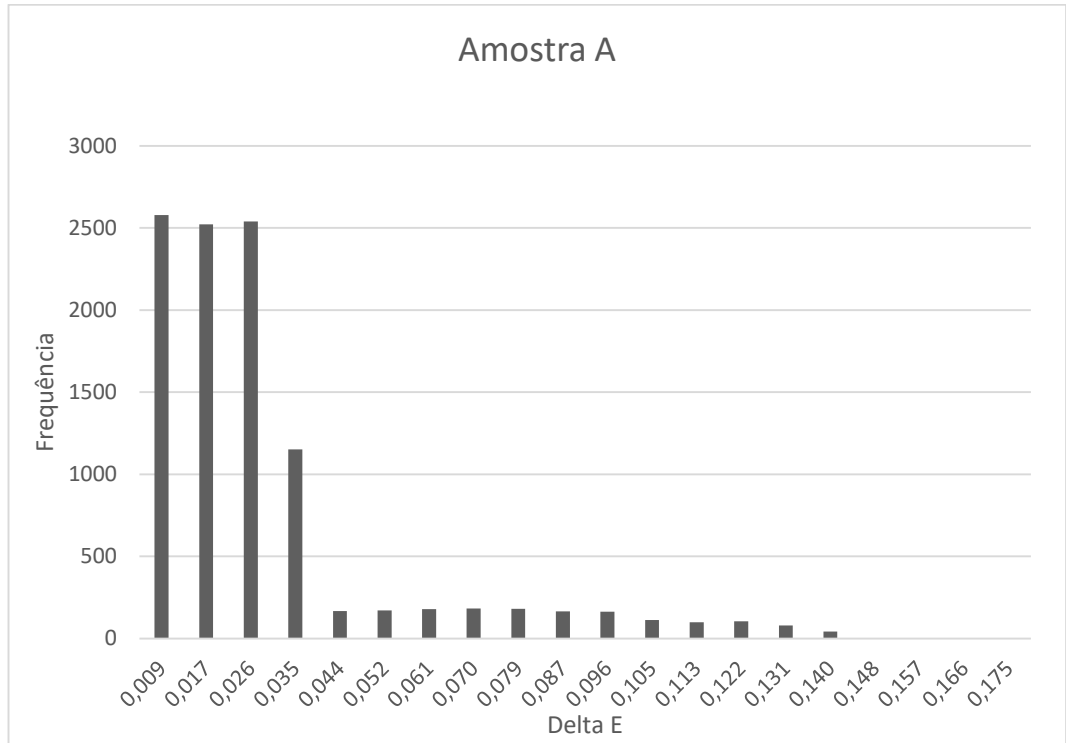
Cor	Mínima	Máxima	Média	Desvio Pad.
#18	0,0000	0,0807	0,0151	0,0101
#2	0,0001	0,0500	0,0152	0,0147
#3	0,0002	0,0899	0,0152	0,0101
#15	0,0001	0,0826	0,0152	0,0098
#8	0,0000	0,0843	0,0153	0,0105
#14	0,0001	0,0644	0,0153	0,0189
#12	0,0000	0,0517	0,0155	0,0089
#1	0,0001	0,0513	0,0157	0,0144
#16	0,0001	0,0801	0,0158	0,0107
#24	0,0000	0,0300	0,0160	0,0088
#19	0,0001	0,0897	0,0162	0,0111
#13	0,0001	0,0399	0,0195	0,0115
#9	0,0000	0,0398	0,0202	0,0112
#4	0,0001	0,0412	0,0212	0,0117
#7	0,0001	0,0428	0,0216	0,0122
#5	0,0000	0,0894	0,0251	0,0108
#23	0,0000	0,0640	0,0313	0,0185
#20	0,0001	0,0800	0,0315	0,0097
#10	0,0002	0,0696	0,0353	0,0202
#6	0,0006	0,1109	0,0575	0,0305
#11	0,0000	0,1188	0,0581	0,0355
#21	0,0000	0,1216	0,0624	0,0355
#22	0,0006	0,1269	0,0635	0,0378
#17	0,0002	0,1285	0,0688	0,0372
Geral	0,0000	0,1285	0,0290	0,0267

Fonte: Autor

A Figura 73 ilustra o gráfico de distribuição dos ΔE encontrados no ensaio com o dispositivo A para todas as cores, ou seja, para as 11160 comparações. Nele é possível notar que 84% dos ΔE estiveram inferiores a 0,038, mostrando que em geral as comparações estiveram próximas de zero, consequentemente uma exatidão aceitável.

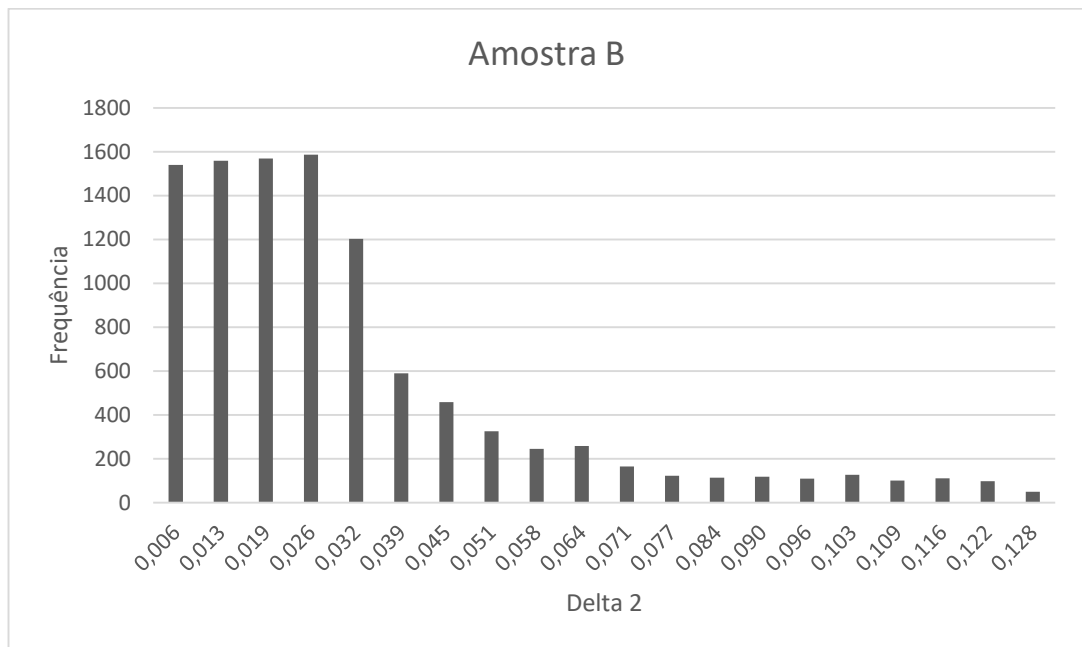
Os mesmos resultados podem ser observados para o dispositivo B na Figura 74. Notamos que apesar da distribuição estar mais dispersa que no modelo A, temos que 81% dos ΔE são inferiores a 0,045, evidenciando também uma exatidão satisfatória.

Figura 73 – Histograma dos ΔE de todas as cores do dispositivo A no teste da mesma posição



Fonte: Autor

Figura 74 – Histograma dos ΔE de todas as cores do dispositivo B no teste da mesma posição



Fonte: Autor

5.4.2.2 Teste em posições diferentes

Tendo em consideração a dificuldade de posicionar duas câmeras exatamente na mesma posição ao realizar o teste de comparação entre duas câmeras, o segundo teste tem como objetivo mensurar o impacto dessa variação. Esse teste consiste em fazer comparativos entre imagens do mesmo dispositivo instalado em posições ligeiramente diferentes. Foram extraídas dez imagens nessas condições, sendo realizadas comparações entre todas, ou seja, foram feitas 45 comparativos (ΔE) para cada cor e câmera testada.

A Tabela 15 e a Tabela 16 mostram os resultados em ordem crescente de média para cada paleta da carta de cores desse teste usando o dispositivo A e B respectivamente. Notamos por elas que a média ainda se manteve próxima a zero para todas as cores, como era esperado, porém os valores são aproximadamente o dobro do teste em posição fixa, o que é justificado devido a maior variação nas imagens comparadas, causada pela ligeira diferença de posição.

Pelo fato das médias no teste em posições variadas serem superiores aos calculados no ensaio de posição fixa e os testes comparativos definitivos serem feitos com essa variação aleatória de posição, esses valores representarão a acurácia das medidas de ΔE estimados pelo método desenvolvido.

A equação (5.3) representa o valor ΔE considerando esta tolerância descrita, onde o ΔE_{medido} é o valor mensurado através do software e M_{cor} a média encontrada para a cor no teste de posições variadas do mesmo dispositivo.

$$\Delta E = \Delta E_{medido} \pm M_{cor} \quad (5.2)$$

Tabela 15 – Resultado (ΔE) do dispositivo A em posições variadas

Cor	Mínima	Máxima	Média	Desvio Pad.
#14	0,0007	0,0510	0,0248	0,0165
#13	0,0012	0,0513	0,0259	0,0139
#4	0,0004	0,0524	0,0281	0,0152
#9	0,0004	0,0615	0,0291	0,0172
#7	0,0049	0,0528	0,0302	0,0140
#2	0,0006	0,0715	0,0319	0,0213
#18	0,0074	0,1401	0,0319	0,0443
#24	0,0051	0,0599	0,0319	0,0174
#15	0,0011	0,0635	0,0322	0,0205
#1	0,0003	0,0684	0,0351	0,0205
#12	0,0005	0,1122	0,0474	0,0347
#8	0,0006	0,1014	0,0506	0,0321
#19	0,0015	0,1255	0,0617	0,0403
#3	0,0029	0,1154	0,0639	0,0365
#20	0,0072	0,1314	0,0641	0,0379
#5	0,0015	0,1298	0,0655	0,0355
#23	0,0001	0,0791	0,0684	0,0257
#10	0,0001	0,1332	0,0689	0,0424
#16	0,0010	0,1641	0,0728	0,0489
#6	0,0001	0,1548	0,0750	0,0507
#17	0,0007	0,1785	0,0869	0,0565
#11	0,0171	0,1716	0,0909	0,0459
#22	0,0079	0,1854	0,0956	0,0583
#21	0,0011	0,1723	0,0969	0,0546
Geral	0,0001	0,1854	0,0554	0,043

Fonte: Autor

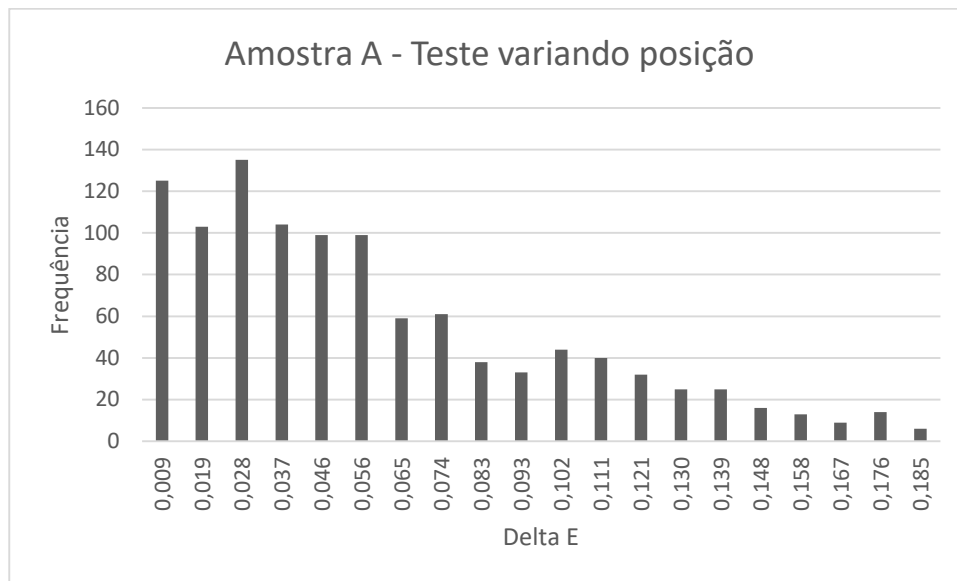
Tabela 16 – Resultado (ΔE) do dispositivo B em posições variadas

Cor	Mínima	Máxima	Média	Desvio Pad.
#14	0,0001	0,0536	0,0259	0,0161
#13	0,0011	0,0509	0,0285	0,0144
#9	0,0004	0,0589	0,0298	0,0166
#2	0,0000	0,0699	0,0304	0,0212
#1	0,0008	0,0602	0,0306	0,0188
#24	0,0014	0,0631	0,0321	0,0209
#15	0,0012	0,0669	0,0350	0,0181
#16	0,0006	0,0666	0,0356	0,0200
#18	0,0087	0,1417	0,0363	0,0427
#4	0,0013	0,1138	0,0561	0,0327
#7	0,0009	0,1145	0,0565	0,0313
#8	0,0012	0,1037	0,0575	0,0349
#19	0,0008	0,1272	0,0585	0,0365
#3	0,0026	0,1127	0,0586	0,0337
#12	0,0008	0,1132	0,0621	0,0308
#23	0,0050	0,1181	0,0624	0,0310
#10	0,0097	0,1317	0,0674	0,0357
#20	0,0002	0,1292	0,0683	0,0385
#5	0,0036	0,1308	0,0694	0,0392
#11	0,0055	0,1674	0,0763	0,0515
#21	0,0076	0,1838	0,0915	0,0513
#6	0,0009	0,1567	0,0920	0,0486
#17	0,0012	0,1665	0,0932	0,0509
#22	0,0021	0,1826	0,0994	0,0544
Geral	0,0000	0,1838	0,0576	0,0413

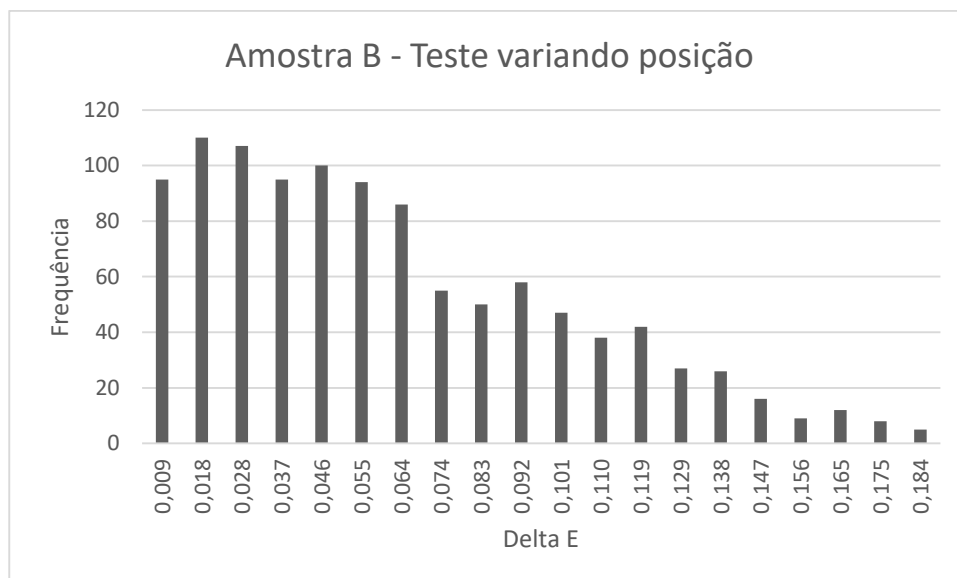
Fonte: Autor

Vale ressaltar ainda que as cinco cores observadas como as menos exatas no teste anterior se mantiveram com as maiores médias no ΔE (apesar de ordem diferente), reforçando a análise anterior.

A Figura 75 e a Figura 76 ilustra o gráfico de distribuição dos ΔE encontrados no ensaio de posição variada com o dispositivo A e B respectivamente. Nota-se uma maior distribuição dos valores ΔE nesse teste em relação ao ensaio em posição fixa apresentado na Figura 73 e Figura 67. Porém em ambos os testes os desvios padrões calculados ainda foram inferiores a 0,06, indicando uma precisão satisfatória.

Figura 75 – Histograma dos ΔE de todas as cores do dispositivo A em posições variadas

Fonte: Autor

Figura 76 – Histograma dos ΔE de todas as cores do dispositivo B em posições variadas

Fonte: Autor

5.4.3 Comparação entre imagens de dispositivos diferentes

Os testes finais serão realizados para mensurar os impactos do iluminante na comparação de cores originados de dispositivos distintos e, conseqüentemente, definir o melhor cenário para fazer a comparação. Para isso serão feitos dois tipos de testes que serão discutidos a seguir:

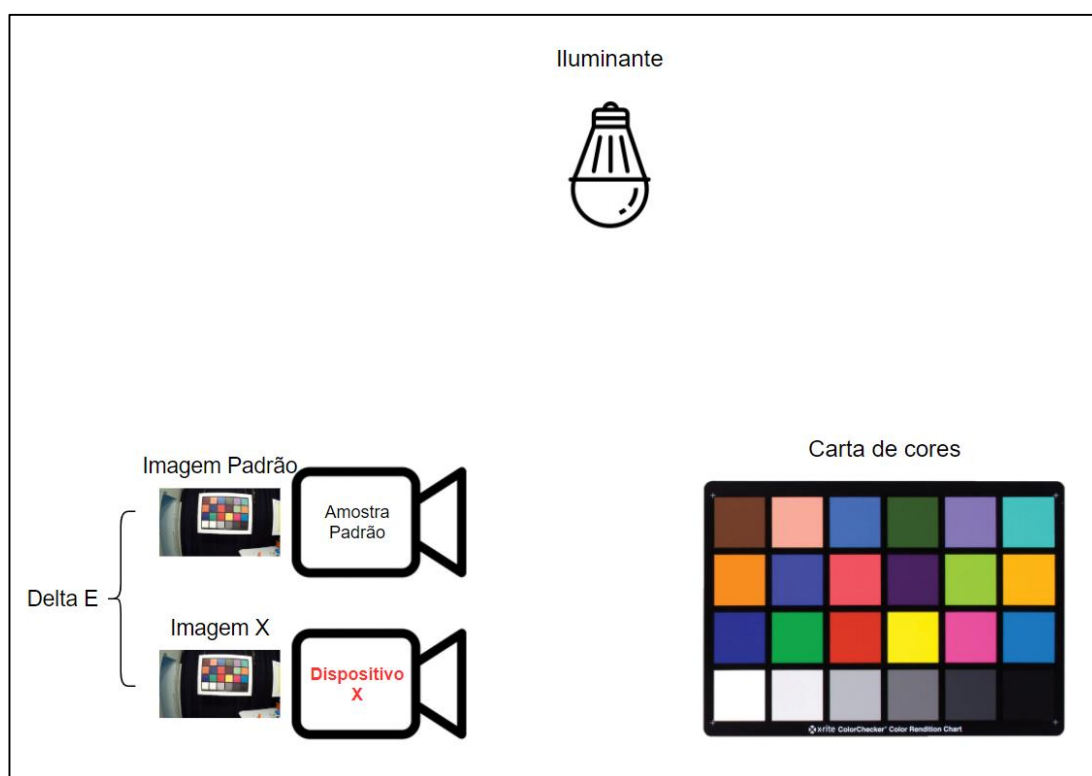
1. Teste de iluminância;
2. Teste com diferentes iluminantes.

5.4.3.1 Teste de iluminância

Esse teste consiste em fazer comparativos entre a imagem de uma câmera padrão e outros dez dispositivos do mesmo modelo em cenários com a iluminância controlada conforme ilustrado na Figura 77. Essa comparação será feita em seis diferentes condições de iluminância de uma lâmpada D65: 50, 100, 150, 200 e 250 lux.

Entre esses dez dispositivos que serão comparados com a amostra padrão, nove são do mesmo modelo e lote de fabricação (dispositivo 1 ao 9) e outro de lote com diferente fabricante de sensor (dispositivo 10).

Figura 77 – Ilustrativo do teste entre imagens de dispositivos diferentes



Fonte: Autor

Na Tabela 17 e Tabela 18 verificamos os resultados dos ΔE da comparação entre a amostra padrão e os dez dispositivos em 250 e 200 lux, respectivamente. Como pode ser notado em destaque nas tabelas, apenas os dispositivos 9 e 10 apresentaram valores de ΔE superiores a dois, caracterizando uma diferença de fácil percepção entre as cores.

Com o objetivo de verificar a diferença encontrada entre os resultados nos dois valores de iluminância, os valores ΔE para cada cor e dispositivo em 250 lux foram subtraídos dos encontrados em 200 lux. A Tabela 19 expõe o resultado dessa subtração, onde é possível notar que para nenhum valor a diferença foi superior a tolerância definida anteriormente em (5.2), ou seja, para essas duas iluminâncias os valores ΔE são equivalentes.

Tabela 17 – Comparativo (ΔE) entre a amostra padrão com iluminância de 250 lux.

Cor	Disp. 1	Disp. 2	Disp. 3	Disp. 4	Disp. 5	Disp. 6	Disp. 7	Disp. 8	Disp. 9	Disp. 10
#1	0,1541	0,1432	0,0043	0,2653	0,1307	0,1223	0,4569	0,5317	0,2653	0,2603
#2	0,4952	0,4367	0,8682	0,2541	0,6220	0,7370	0,4310	0,6577	1,8664	1,9514
#3	0,3583	0,2511	0,1742	0,5683	0,3307	0,3595	0,1424	0,4865	0,3460	0,4772
#4	0,2340	0,0940	0,0367	0,3294	0,1418	0,6554	0,6117	0,6125	0,3839	0,4112
#5	0,3965	0,2352	0,1538	0,8174	1,0647	0,2147	0,3619	0,5987	1,3104	1,4219
#6	0,4161	0,2425	0,3070	1,5028	1,5441	0,4964	1,0736	1,5548	1,9743	2,7660
#7	0,3704	0,5274	0,4203	0,6482	0,1935	0,7656	0,4321	0,6887	1,2562	1,3523
#8	0,4261	0,1046	0,0109	0,7626	0,6800	0,7280	0,0850	0,3024	0,2499	0,4919
#9	0,3276	0,3896	0,3052	0,4665	0,8114	0,5907	0,7065	0,8118	1,4676	1,5002
#10	0,6059	0,4553	0,5431	0,4012	0,5520	0,4882	0,5268	0,4526	0,4556	0,5195
#11	1,4814	0,3128	0,3811	0,4696	0,4712	0,9237	0,6338	0,6286	2,3269	2,5880
#12	0,5255	0,4126	0,3502	0,3799	0,3773	0,3812	0,3212	0,3964	1,9954	2,6415
#13	0,2735	0,3696	0,3338	0,4772	0,1746	0,0999	0,2157	0,2271	0,6007	0,5527
#14	0,0293	0,0150	0,0499	0,7836	0,0794	0,2646	0,2705	0,1295	0,4761	0,9167
#15	0,1002	0,1219	0,1201	0,4143	0,5899	0,3508	0,2830	0,5783	0,8012	0,9454
#16	0,3857	0,1931	0,1254	0,7456	0,4811	0,5197	1,0063	0,6357	2,6212	2,7015
#17	0,1832	0,2434	0,3103	0,5939	0,6798	0,5901	0,4172	0,4619	1,1584	1,9097
#18	0,1340	0,1930	0,0376	0,2296	0,3774	0,0226	0,3490	0,3233	0,1656	0,3615
#19	0,6476	0,6728	0,2721	1,0474	1,1975	0,3740	0,1954	0,2516	2,6374	2,2576
#20	1,4791	1,2802	0,7542	0,8157	0,4366	1,1755	1,5928	0,2628	2,5645	2,4043
#21	0,3552	0,2847	0,5226	1,4852	1,6067	0,7645	0,5957	1,4254	2,4776	2,7587
#22	0,2013	0,8986	0,6487	1,0700	1,1384	0,1811	0,1345	0,6117	1,3977	1,6124
#23	0,1066	0,3806	0,0765	0,1886	0,2139	0,2674	0,3158	0,2748	0,2915	0,3468
#24	0,0239	0,0546	0,0787	0,1500	0,1467	0,1480	0,1431	0,1418	0,1885	0,1819

Fonte: Autor

Tabela 18 – Comparativo (ΔE) entre a amostra padrão com iluminância de 200 lux

Cor	Disp. 1	Disp. 2	Disp. 3	Disp. 4	Disp. 5	Disp. 6	Disp. 7	Disp. 8	Disp. 9	Disp. 10
#1	0,1507	0,1405	0,0043	0,2709	0,1264	0,1261	0,4635	0,5269	0,2697	0,2570
#2	0,4907	0,4334	0,8719	0,2504	0,6252	0,7317	0,4278	0,6627	1,8626	1,9448
#3	0,3622	0,2579	0,1777	0,5615	0,3346	0,3631	0,1458	0,4800	0,3419	0,4809
#4	0,2288	0,0932	0,0364	0,3338	0,1348	0,6592	0,6184	0,6182	0,3765	0,4147
#5	0,4000	0,2396	0,1570	0,8113	1,0705	0,2179	0,3651	0,6039	1,3156	1,4295
#6	0,4124	0,2480	0,3130	1,5060	1,5474	0,4933	1,0683	1,5591	1,9673	2,7587
#7	0,3664	0,5320	0,4244	0,6449	0,1972	0,7621	0,4276	0,6828	1,2491	1,3587
#8	0,4212	0,1083	0,0103	0,7581	0,6847	0,7314	0,0784	0,3093	0,2535	0,4877
#9	0,3309	0,3838	0,3020	0,4711	0,8172	0,5970	0,7007	0,8086	1,4755	1,4932
#10	0,6019	0,4509	0,5369	0,4046	0,5574	0,4919	0,5200	0,4480	0,4516	0,5160
#11	1,4865	0,3053	0,3845	0,4729	0,4651	0,9176	0,6375	0,6242	2,3230	2,5945
#12	0,5336	0,4211	0,3552	0,3843	0,3816	0,3735	0,3267	0,3919	2,0004	2,6483
#13	0,2782	0,3619	0,3296	0,4824	0,1790	0,1055	0,2243	0,2312	0,6106	0,5600
#14	0,0294	0,0156	0,0577	0,7791	0,0715	0,2584	0,2810	0,1249	0,4700	0,9115
#15	0,0956	0,1158	0,1251	0,4198	0,5820	0,3555	0,2883	0,5675	0,7948	0,9405
#16	0,3924	0,1822	0,1178	0,7552	0,4920	0,5130	1,0009	0,6292	2,6282	2,6961
#17	0,1708	0,2361	0,3167	0,5801	0,6910	0,5832	0,4089	0,4514	1,1724	1,9177
#18	0,1420	0,1995	0,0375	0,2371	0,3702	0,0217	0,3564	0,3170	0,1764	0,3529
#19	0,6328	0,6653	0,2650	1,0360	1,1885	0,3640	0,2037	0,2442	2,6288	2,2657
#20	1,4873	1,2643	0,7648	0,8254	0,4487	1,1652	1,5764	0,2492	2,5802	2,3928
#21	0,3471	0,2727	0,5315	1,4953	1,6165	0,7539	0,5831	1,4376	2,4694	2,7665
#22	0,1823	0,9173	0,6337	1,0841	1,1209	0,1908	0,1434	0,6205	1,3817	1,6005
#23	0,1085	0,3929	0,0884	0,1976	0,2230	0,2781	0,2992	0,2885	0,2826	0,3591
#24	0,0236	0,0539	0,0984	0,1444	0,1457	0,1426	0,1443	0,1402	0,1990	0,1820

Fonte: Autor

Tabela 19 – Resultado da subtração do ΔE em 250 e 200 lux

Cor	Disp. 1	Disp. 2	Disp. 3	Disp. 4	Disp. 5	Disp. 6	Disp. 7	Disp. 8	Disp. 9	Disp. 10
#1	0,0034	0,0027	0,0000	-0,0057	0,0043	-0,0038	-0,0066	0,0048	-0,0044	0,0033
#2	0,0045	0,0033	-0,0037	0,0037	-0,0032	0,0054	0,0032	-0,0050	0,0038	0,0066
#3	-0,0039	-0,0068	-0,0034	0,0068	-0,0039	-0,0036	-0,0034	0,0065	0,0041	-0,0037
#4	0,0051	0,0008	0,0002	-0,0044	0,0070	-0,0038	-0,0067	-0,0057	0,0074	-0,0035
#5	-0,0036	-0,0044	-0,0031	0,0061	-0,0058	-0,0032	-0,0032	-0,0052	-0,0052	-0,0076
#6	0,0037	-0,0055	-0,0060	-0,0032	-0,0033	0,0031	0,0052	-0,0043	0,0070	0,0073
#7	0,0040	-0,0045	-0,0041	0,0033	-0,0037	0,0034	0,0045	0,0059	0,0072	-0,0064
#8	0,0049	-0,0037	0,0006	0,0046	-0,0047	-0,0035	0,0066	-0,0069	-0,0035	0,0042
#9	-0,0033	0,0058	0,0032	-0,0046	-0,0058	-0,0063	0,0058	0,0032	-0,0078	0,0069
#10	0,0040	0,0044	0,0062	-0,0034	-0,0054	-0,0036	0,0068	0,0045	0,0041	0,0036
#11	-0,0051	0,0075	-0,0034	-0,0033	0,0061	0,0061	-0,0037	0,0043	0,0039	-0,0066
#12	-0,0081	-0,0085	-0,0050	-0,0044	-0,0043	0,0076	-0,0055	0,0045	-0,0050	-0,0068
#13	-0,0047	0,0077	0,0041	-0,0051	-0,0044	-0,0057	-0,0087	-0,0041	-0,0099	-0,0073
#14	0,0000	-0,0006	-0,0078	0,0046	0,0078	0,0062	-0,0104	0,0047	0,0061	0,0052
#15	0,0046	0,0061	-0,0050	-0,0055	0,0078	-0,0047	-0,0054	0,0108	0,0064	0,0049
#16	-0,0067	0,0109	0,0077	-0,0096	-0,0110	0,0068	0,0055	0,0065	-0,0070	0,0054
#17	0,0125	0,0073	-0,0064	0,0138	-0,0113	0,0069	0,0083	0,0105	-0,0140	-0,0081
#18	-0,0079	-0,0065	0,0001	-0,0076	0,0073	0,0009	-0,0075	0,0063	-0,0108	0,0086
#19	0,0147	0,0076	0,0071	0,0115	0,0090	0,0100	-0,0083	0,0074	0,0086	-0,0081
#20	-0,0083	0,0158	-0,0106	-0,0097	-0,0121	0,0104	0,0164	0,0135	-0,0156	0,0114
#21	0,0081	0,0120	-0,0089	-0,0101	-0,0099	0,0105	0,0126	-0,0121	0,0081	-0,0078
#22	0,0190	-0,0187	0,0151	-0,0141	0,0175	-0,0097	-0,0089	-0,0089	0,0159	0,0119
#23	-0,0018	-0,0123	-0,0120	-0,0090	-0,0091	-0,0107	0,0166	-0,0138	0,0089	-0,0123
#24	0,0003	0,0008	-0,0197	0,0056	0,0010	0,0053	-0,0012	0,0016	-0,0104	-0,0001

Fonte: Autor

Os resultados encontrados em 150 lux se assemelham as simulações em 200 lux descritas anteriormente, isto é, para essas duas iluminâncias os valores ΔE são equivalentes considerando a tolerância definida.

Os testes realizados em 100 lux apresentaram as primeiras divergências entre o ΔE calculado em comparação com 250 lux. A Tabela 20 mostra essa comparação e destaca em vermelho os valores onde o valor é divergente, em outras palavras, onde a diferença é superior a tolerância. Como pode ser observado o dispositivo 10 foi o que apresentou mais cores onde o ΔE diverge dos calculados em 250 lux, sendo o cinza 5 (#21) o que apresentou a maior diferença. Vale ressaltar que, em todas as sete cores divergentes o valor ΔE em 100 lux foi menor do que em 250 lux, fato observado nos próximos testes com baixas iluminâncias.

Tabela 20 – Resultado da subtração do ΔE em 250 e 100 lux

Cor	Disp. 1	Disp. 2	Disp. 3	Disp. 4	Disp. 5	Disp. 6	Disp. 7	Disp. 8	Disp. 9	Disp. 10
#1	-0,0135	-0,0055	-0,0170	-0,0128	0,0101	0,0158	-0,0128	-0,0224	-0,0198	-0,0199
#2	0,0248	0,0102	-0,0132	0,0105	0,0149	-0,0116	0,0180	-0,0195	-0,0147	0,0104
#3	-0,0116	-0,0232	-0,0145	0,0179	-0,0194	-0,0113	0,0141	0,0154	-0,0123	-0,0115
#4	-0,0178	0,0108	0,0085	0,0107	0,0162	0,0114	0,0108	-0,0239	-0,0110	0,0159
#5	0,0151	-0,0104	-0,0179	0,0125	0,0108	0,0129	0,0161	0,0185	0,0159	-0,0245
#6	0,0143	-0,0214	-0,0122	-0,0220	0,0119	0,0214	0,0240	-0,0145	0,0152	-0,0165
#7	-0,0122	-0,0107	-0,0174	-0,0104	0,0250	-0,0165	-0,0121	-0,0107	0,0117	0,0188
#8	-0,0190	0,0241	-0,0200	0,0108	-0,0105	0,0105	-0,0183	-0,0147	0,0238	0,0194
#9	0,0113	0,0112	-0,0202	-0,0139	-0,0113	0,0250	0,0202	-0,0132	0,0166	0,0138
#10	0,0334	0,0234	0,0310	0,0073	0,0539	0,0262	0,0552	0,0197	0,0213	0,0229
#11	-0,0120	-0,0105	0,0154	0,0181	-0,0256	-0,0155	0,0129	0,0168	-0,0202	0,0192
#12	0,0144	0,0233	-0,0179	-0,0231	0,0128	0,0131	-0,0121	0,0137	-0,0138	-0,0162
#13	-0,0129	0,0198	-0,0232	0,0237	0,0126	-0,0195	-0,0211	0,0125	-0,0127	0,0137
#14	-0,0248	0,0073	0,0176	-0,0165	0,0160	0,0184	-0,0234	0,0049	0,0152	-0,0184
#15	-0,0138	0,0219	-0,0309	0,0150	0,0137	-0,0232	-0,0144	-0,0159	-0,0230	0,0173
#16	0,0150	0,0199	0,0316	0,0236	-0,0224	-0,0238	0,0166	0,0150	0,0147	-0,0142
#17	0,0231	0,0201	0,0274	-0,0185	-0,0270	0,0314	-0,0183	-0,0279	0,0167	0,0283
#18	-0,0202	0,0173	0,0013	-0,0185	0,0363	0,0177	0,0217	-0,0119	0,0184	0,0175
#19	-0,0089	-0,0073	0,0384	0,0643	0,0512	-0,0107	-0,0181	-0,0243	0,0708	0,0714
#20	0,0055	0,0603	0,0394	-0,0177	0,0616	-0,0040	0,0562	-0,0471	0,0222	0,0771
#21	0,0477	0,0464	0,0015	0,1087	0,0459	0,0689	0,0713	0,0389	0,0850	0,1660
#22	-0,0159	0,0052	0,0717	0,0862	0,0227	0,0434	0,0336	0,0544	0,0985	0,1394
#23	-0,0183	0,0046	0,0367	0,0676	0,0596	-0,0093	0,0602	-0,0047	0,0545	0,0887
#24	-0,0240	-0,0071	-0,0249	-0,0113	0,0069	0,0395	-0,0394	-0,0309	-0,0318	-0,0261

Fonte: Autor

Em baixas iluminâncias houve uma aproximação na semelhança das cores, mesmo para as identificadas com alto índice de diferença nos testes de alta iluminância. A Tabela 21 mostra esse fato ao fazer a comparação dos ΔE em 250 e 50 lux, destacado em vermelho estão os valores onde o valor é divergente considerando a tolerância. Como pode ser observado, a grande maioria das cores tiveram resultados distantes em 50 lux ao comparar aos encontrados no teste de 250 lux. Sendo que cores identificadas anteriormente como divergente do padrão, foram estimadas com ΔE inferior a 1 em baixa iluminância, o que representaria um falso positivo de semelhança de cor.

Tabela 21 – Resultado da subtração do ΔE em 250 e 50 lux

Cor	Disp. 1	Disp. 2	Disp. 3	Disp. 4	Disp. 5	Disp. 6	Disp. 7	Disp. 8	Disp. 9	Disp. 10
#1	0,0305	0,0268	-0,0047	0,0546	0,0202	0,0143	0,1068	0,1183	0,1226	0,1240
#2	0,1157	0,0989	0,2070	0,0563	0,1442	0,1772	0,0974	0,1564	0,9270	0,9705
#3	0,0762	0,0515	0,0323	0,1340	0,0733	0,0781	0,0260	0,1132	0,1624	0,2339
#4	0,0464	0,0096	-0,0009	0,0746	0,0254	0,1554	0,1408	0,1436	0,1825	0,1989
#5	0,0905	0,0437	0,0264	0,1975	0,2571	0,0412	0,0835	0,1371	0,6493	0,7061
#6	0,0916	0,0467	0,0627	0,3603	0,3790	0,1150	0,2611	0,3803	1,2592	2,2104
#7	0,0844	0,1157	0,0941	0,1529	0,0409	0,1815	0,0981	0,1578	0,6197	0,6710
#8	0,0994	0,0100	-0,0017	0,1736	0,1531	0,1697	0,0111	0,0678	0,1171	0,2412
#9	0,0744	0,0904	0,0694	0,1026	0,1863	0,1336	0,1697	0,1921	0,7265	0,7454
#10	0,1433	0,1025	0,1286	0,0913	0,1213	0,1066	0,1240	0,1026	0,2194	0,2546
#11	0,3589	0,0711	0,0833	0,1054	0,1078	0,2186	0,1489	0,1426	1,6254	1,8084
#12	0,1150	0,0944	0,0750	0,0797	0,0837	0,0870	0,0651	0,0914	1,5923	2,1111
#13	0,0527	0,0769	0,0758	0,1070	0,0264	0,0091	0,0443	0,0487	0,2940	0,2702
#14	-0,0137	-0,0198	0,0034	0,1770	0,0082	0,0498	0,0569	0,0232	0,2317	0,4532
#15	0,0134	0,0210	0,0157	0,0945	0,1389	0,0790	0,0555	0,1288	0,3943	0,4664
#16	0,0822	0,0375	0,0186	0,1745	0,1113	0,1192	0,2337	0,1483	1,3033	1,3425
#17	0,0335	0,0507	0,0654	0,1330	0,1590	0,1310	0,0919	0,1037	0,5726	0,9412
#18	0,0219	0,0385	-0,0072	0,0474	0,0838	-0,0157	0,0748	0,3050	0,0697	0,1736
#19	0,1468	0,1580	0,0441	0,2507	0,2800	0,0774	0,0314	0,0469	1,3107	1,1162
#20	0,5052	0,4334	0,2418	0,2734	0,1409	0,3921	0,5334	0,0668	1,2673	1,1909
#21	0,1083	0,0864	0,1678	0,4985	0,5508	0,2541	0,1800	0,4804	1,7284	1,9234
#22	0,0290	0,1924	0,1410	0,2482	0,2630	0,0291	0,0020	0,1316	0,6829	0,7963
#23	0,0042	0,0763	-0,0026	0,0223	0,0392	0,0522	0,0521	0,0523	0,1287	0,1565
#24	-0,0111	-0,0164	0,0041	0,0051	0,0201	0,0108	0,0206	0,0151	0,0840	0,0681

Fonte: Autor

Com o objetivo de comparar os resultados nas cinco iluminâncias para um dispositivo definido como divergente ao padrão estabelecido, foi calculada a razão entre a diferença entre os ΔE e a tolerância, conforme a equação (5.3). Uma razão superior a 1 representa um comparativo fora da tolerância de erro, ou seja, o ΔE é diferente do calculado em 250 lux.

A Tabela 22 mostra o resultado dessa comparação para o dispositivo 10 (câmera mais distante do padrão) usando 250 lux como referência. Como pode ser observado pela tabela apenas nas iluminâncias de 100 e 50 lux houveram valores calculados acima de 1, sendo a escala de cinza impactada desde a iluminância de 100 lux.

$$Razão = \frac{\Delta E \text{ 250 lux} - \Delta E \text{ X lux}}{Tolerância} \quad (5.3)$$

Tabela 22 – Resultado da razão entre a diferença de ΔE e a tolerância para o dispositivo 10 modelo A

Cor	250 - 50 Lux	250 - 100 Lux	250 - 150 Lux	250 - 200 Lux
#1	3,88	0,57	0,18	0,09
#2	30,75	0,33	0,11	0,21
#3	3,81	0,18	0,09	0,06
#4	7,56	0,57	0,19	0,12
#5	10,93	0,37	0,11	0,12
#6	7,41	0,22	0,08	0,10
#7	22,56	0,62	0,21	0,21
#8	4,95	0,38	0,09	0,08
#9	25,94	0,47	0,11	0,24
#10	3,85	0,33	0,06	0,05
#11	8,58	0,21	0,05	0,07
#12	11,19	0,34	0,11	0,14
#13	10,91	0,53	0,35	0,28
#14	18,69	0,74	0,17	0,21
#15	14,88	0,54	0,23	0,15
#16	18,67	0,20	0,11	0,07
#17	11,14	0,33	0,08	0,09
#18	5,89	0,55	0,49	0,27
#19	18,50	1,16	0,13	0,13
#20	18,93	1,20	0,14	0,18
#21	8,62	1,71	0,08	0,08
#22	8,54	1,46	0,18	0,12
#23	2,78	1,30	0,20	0,18
#24	3,57	0,82	0,48	0,00

Os testes descritos anteriormente foram repetidos usando o modelo B de câmera IP, ou seja, uma câmera padrão foi comparada com outros dez dispositivos do mesmo lote. Dentre esses dez dispositivos, nove apresentaram característica semelhantes nas cores e uma apresentou diferenças significativas (superior a 2). A Tabela 23 mostra o resultado sumarizado analogamente a Tabela 22, como pode ser observado novamente houve discrepância nos valores encontrados em baixa iluminância, sendo que foram encontrados duas cores a mais nessa condição em comparação ao teste com o modelo A.

Tabela 23– Resultado da razão entre a diferença de ΔE e a tolerância para o dispositivo 10 modelo B

Cor	250 - 50 Lux	250 - 100 Lux	250 - 150 Lux	250 - 200 Lux
#1	4,49	1,12	0,63	0,49
#2	6,96	0,48	0,50	0,50
#3	6,41	0,49	0,37	0,49
#4	6,70	0,60	0,97	0,90
#5	6,46	1,04	0,31	0,43
#6	2,85	0,10	0,31	0,97
#7	6,23	0,26	0,42	0,40
#8	9,12	0,96	0,04	0,55
#9	14,07	0,11	0,24	0,22
#10	11,44	0,00	0,95	0,18
#11	15,55	0,19	0,00	0,77
#12	13,07	0,26	0,34	0,51
#13	18,47	0,56	0,85	0,48
#14	12,69	0,69	0,42	0,15
#15	19,71	0,34	0,71	0,87
#16	14,49	0,69	0,46	0,87
#17	9,45	0,75	0,62	0,02
#18	9,50	0,59	0,43	0,64
#19	22,08	1,52	0,57	0,11
#20	23,25	1,36	0,08	0,04
#21	12,02	1,96	0,16	0,95
#22	17,40	1,53	0,94	0,66
#23	8,39	1,09	0,13	0,90
#24	4,51	0,29	0,36	0,48

Os resultados em baixa iluminação demonstraram que a aplicação desenvolvida não é capaz de mensurar precisamente a diferença de cores em cenários com iluminações inferiores a 100 lux.

5.4.3.2 Teste com diferentes iluminantes

Com o objetivo de verificar os efeitos do iluminante, o teste anterior foi refeito para 150, 200 e 250 lux usando o iluminante D50 e comparando com os resultados encontrados usando D65. Essa comparação foi realizada calculando a diferença entre os ΔE usando os dois iluminantes e comparando com a tolerância, conforme a equação (5.4).

$$\text{Razão} = \frac{\Delta E \text{ com D65 em } X \text{ lux} - \Delta E \text{ com D50 em } X \text{ lux}}{\text{Tolerância}} \quad (5.4)$$

Para as três iluminâncias não foi observado nenhuma discrepância entre os ΔE encontrados para nenhum dos dispositivos. A Tabela 24 mostra o resultado do dispositivo 10 do modelo A usando a equação (5.4). Como pode ser observado não há nenhum valor superior a 1 que caracterize discordância nos valores encontrados usando o iluminante D65 e D50.

Tabela 24 – Resultado da razão entre a diferença de ΔE usando D65 e D50 e a tolerância no modelo A disp. 10

Cor	150 Lux	200 Lux	250 Lux
#1	0,12	0,82	0,79
#2	0,50	0,67	0,51
#3	0,77	0,75	0,96
#4	0,77	0,28	0,39
#5	0,50	0,29	0,36
#6	0,48	0,14	0,26
#7	0,29	0,58	0,35
#8	0,41	0,30	0,33
#9	0,59	0,87	0,63
#10	0,72	0,40	0,63
#11	0,30	0,08	0,83
#12	0,79	0,63	0,30
#13	0,04	0,41	0,47
#14	0,37	0,87	0,57
#15	0,63	0,71	0,16
#16	0,77	0,42	0,26
#17	0,46	0,53	0,46
#18	0,11	0,66	0,81
#19	0,71	0,59	0,28
#20	0,97	0,83	0,68
#21	0,08	0,61	0,59
#22	0,40	0,53	0,65
#23	0,40	0,76	0,38
#24	0,92	0,41	0,30

Fonte: Autor

5.5 CONCLUSÕES

Este Capítulo, após definir as métricas utilizadas e propor as simulações de comparações entre imagens, apresenta e analisa os resultados dos testes realizados. Foi avaliado inicialmente o algoritmo de localização da carta de cores, onde evidenciamos a importância do algoritmo de estimação de paletas para atingir a assertividade de cem por cento.

Após a validação da localização da carta de cores, foram realizadas comparações entre imagens do mesmo dispositivo para mensurar a acuracidade e a precisão do método proposto.

Essa comparação foi realizada em duas condições: dispositivo na mesma posição e em posições diferentes. Em ambos os testes o ΔE da comparação entre cores estiveram próximas de zero, evidenciando uma precisão satisfatória para a aplicação. Vale ressaltar que os valores encontrados no teste com variação na posição foram aproximadamente o dobro do teste de posição fixa, esse resultado é esperado devido a maior variação nas imagens comparadas causada pela mudança na posição.

Os testes finais foram realizados com o objetivo de mensurar os impactos do iluminante no comparativo de cores. A primeira simulação consiste em fazer o comparativo entre a imagem de uma câmera padrão e outros dispositivos em um cenário com a iluminância controlada. Comparando os resultados em cinco diferentes iluminâncias, concluímos que em iluminâncias abaixo de 100 lux há uma aproximação na semelhança das cores, o que causa um falso positivo na detecção de uma cor fora do padrão estabelecido. Por fim, foi realizado o teste usando o iluminante do tipo D50, onde foi observado os mesmos resultados anteriores usando uma lâmpada D65, ou seja, seria possível realizar o comparativo de cores usando ambos os iluminantes.

Após aproximadamente 40000 comparações (1600 para cada paleta) usando modelos de câmeras de segurança, concluímos que o método proposto é capaz de mensurar a diferença de cores e sinalizar caso haja uma discrepância significativa nas condições propostas.

6 CONCLUSÃO

6.1 CONSIDERAÇÕES FINAIS

Oferecer um produto de qualidade é essencial para as empresas se manterem fortes no mercado. No segmento de câmeras, padrão de cores é sinônimo de qualidade de produto e, além de ser uma excelente diferenciação, pode também ser entendido como uma arma muito competitiva.

Nesse contexto, esta pesquisa tem por finalidade apresentar uma aplicação capaz de assegurar o padrão de cores em câmeras de segurança eletrônica de forma automática e analítica, fazendo a comparação entre a amostra padrão definida no desenvolvimento do produto e o dispositivo proveniente da produção em massa. Através do modelo de cor CIELAB e da fórmula de diferença de cor CIE 2000 definida pela CIE é possível estabelecer um modelo analítico matemático para a comparação entre essas duas imagens digitais.

O sistema de visão computacional é responsável por reconhecer as paletas de cores da carta padrão. Dessa forma o processo se torna automático, levando em conta que não é necessário um usuário entrar com os dados referente às cores. Essa aplicação foi desenvolvida em Python e validada mediante testes realizados em diferentes situações.

Assim, atentando a aproximadamente 40000 comparações de cores realizados concluímos que os resultados obtidos se mostraram satisfatórios e condizentes com os resultados esperados.

Deduz-se, então, que o objetivo geral proposto em *desenvolver uma aplicação capaz de fazer a comparação entre as cores de duas câmeras com o objetivo de assegurar um padrão estabelecido* foi alcançado. Os conceitos utilizados para realizar a aplicação servem como base para diversas outras aplicações, como por exemplo o controle de imagem em *display* de monitores.

6.2 TRABALHOS FUTUROS

O modelo apresentado utilizado por este estudo constitui-se em uma alternativa válida e eficiente para a comparação entre cores de duas imagens digitais.

Seguem alguns tópicos que merecem atenção para trabalhos futuros:

- Propor, através dessa aplicação, quais os parâmetros de cores configuráveis (saturação, contraste e brilho) devem ser alterados para se chegar a menor distância entre duas cores;
- Implementar algoritmos de aprendizado de máquinas para encontrar a carta de cores na cena, que deve ser testada em conjunto com o algoritmo existente, e ao final eleger o mais eficiente para a aplicação;
- Fazer uma aplicação com interface para o usuário buscando melhorar a usabilidade do programa (sugestão de que seja uma interface web para ser possível acessar via internet).

REFERÊNCIA

ALMEIDA, Luiz Eduardo Pita Mercês. **Estudo de Caso em Reconhecimento Automático de Placas Veiculares Usando Linguagem de Programação Python**. 2010. 90 f. TCC (Graduação) - Curso de Engenharia Elétrica, Universidade Federal da Paraíba, João Pessoa, 2010.

BARCELLOS, Robson. **Novo Método de Mapeamento de Espaço de Cor Através de Redes Neurais Artificiais Especializada**. 2011. 121 f. Tese (Doutorado) - Curso de Engenharia Elétrica, Universidade de São Paulo, São Carlos, 2011.

CARRERA, A. D.. **Segmentação de Multidões Para um Sistema de Monitoramento de Banhistas com Foco no Baixo Custo Computacional**. 2017. 85 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Federal Rural de Pernambuco, Recife, 2017.

CHUN, Wesley J.. **Core Python Programming**. 2. ed. [S. L.]: Prentice Hall, 2006. 1120 p.

COSTA, Vambaster José da. **Reconhecimento de Padrões Faciais: Uma Síntese**. 2019. 98 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Tecnológica Federal do Paraná, Santa Helena, 2019.

DALCIN, Sabrina Borba. **Imagetest: Uma Proposta de Metodologia para Aplicação de Testes de Software no Domínio de Transformações no Espaço de Cores**. 2007. 93 f. Dissertação (Mestrado) - Curso de Engenharia de Produção, Universidade Federal de Santa Maria, Santa Maria, 2007.

DELTA COLOR (org.). **Introdução à Calorimetria e ao Controle de Qualidade de Cores**. Disponível em: <http://deltacolor.com>. Acesso em: 20 nov. 2020.

FAIRCHILD, Mark D.. **Color Appearance Models**. 2. ed. Rochester: John Wiley & Sons, Ltd, 2005.

FELISBERTO, Thiago Zanivan. **Robo Solucionador de Sudoku**. 2015. 144 f. TCC (Graduação) - Curso de Engenharia da Computação, Universidade Federal de Santa Catarina, Araranguá, 2015.

Souza, J. S., & Cardoza, J. A.. **Sensores de Imagem Digitais CCD E CMOS**. Instituto Federal de Educação, Ciência e Tecnologia, Palmas, 2012.

FERNANDES, Antonio Carlos. **Protótipo de Visualizador para Modelos de Cor para Medições de Objetos em Espectrofotômetros por Reflectância**. 2002. 88 f. TCC (Graduação) - Curso de Ciências da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2002.

GAZZANIGA, Michael S. **Cognitive Neuroscience: The Biology of the Mind**. [S.L]: W. W. Norton, 2014.

GUALTIERI, Mirella. **Visão de Cores e Sensibilidade ao Contraste em Indivíduos com Diabete Melito: Avaliação Psicofísica e Eletrofisiológica**. 2004. 159 f. TCC (Graduação) - Curso de Engenharia Elétrica, Universidade de São Paulo, São Paulo, 2004.

HICKMAN, C. P.; LARSON, A.. **Princípios Integrados de Zoologia**. Rio de Janeiro: Guanabara Koogan S.A, 2004. 846 p.

KRONBAUER, A. L., SCHOR, P., & CARVALHO, L. A. (2008). **Medidas da Visão e Testes Psicofísicos**. Arquivos Brasileiros de Oftalmologia, São Paulo.

L'ÉCLAIRAGE C. I. **Parametric Effects in Colour-Difference Evaluation**. (C. 2.4.2, Ed.) CIE, 1993.

LENT, Roberto. **Cem Bilhões de Neurônios Conceitos Fundamentais de Neurociências**. 2. ed. [S.L]: Atheneu, 2010. 786 p

LOPES, João Manuel. **Cor e Luz**. 47 f. TCC (Graduação) - Curso de Engenharia de Informática de Computadores, Universidade Técnica de Lisboa, Lisboa, 2004.

MAIA, Núbia Cristina de Freitas. **Fundamentos Básicos da Oftalmologia e suas Aplicações**. Palmas: Eduft, 2018. 122 p.

GODOY, Paulo Cesar Cardoso; MATTOS, Natanael Fernandes de. **Processamento de Imagens Aplicado à Análise das Condições do Pavimento Asfáltico de Aeroportos**. Revista de Iniciação Científica de Ulbra, Canoas, v. 1, n. 14, p. 1-14, jun. 2016.

MEYER, B.: ZOLLINGER, H.R..**Colorimetria**. [s.l.]: Clariant, 1998

NOGUEIRA, Ricardo César de Almeida. **Análise de Conversão de Imagem Colorida para Tons de Cinza Via Contraste Percebido**. 2016. 46 f. TCC (Graduação) - Curso de Engenharia da Computação, Universidade Federal de Pernambuco, Recife, 2016.

OKUNO, Emico. **Física das Radiações**. [S.L]: Oficina de Textos, 2010. 296 p.

OLIVEIRA, Walber Rodrigues de. **Sistema de Visão Computacional para Reconhecimento de Modelos CAD Complexos**. 2018. 45 f. TCC (Graduação) - Curso de Engenharia da Computação, Universidade Federal de Pernambuco, Recife, 2016.

PASCALE, D., **A Review of RGB Color Space**. Montreal: Babel Color Company; 2003. 35p.

PEREIRA, R. C.. **Aperfeiçoamento do Cálculo do Índice de Reprodução de Cor para Fontes Luminosas de Acordo com a Metodologia Internacional Atual**. 2015. 49 f. TCC (Graduação) - Curso de Metrologia e Qualidade, Instituto Nacional de Metrologia, Duque de Caxias, 2015.

RIDOLFI, Lorenzo Francesco Giovanni Gino Maria. **Construção de Espaços de Cor Euclidianos e Perceptualmente Uniformes com Base na Fórmula CIEDE2000**. 2012. 93 f. Tese (Doutorado) - Curso de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2012.

SAMPAIO, J. A. ; FRANÇA, S. C. A.; BRAGA, P. F. A.. **Tratamento de Minérios: Práticas Laboratoriais**. Rio de Janeiro: CETEM, 2007. 570p.

Sezgin, M. e B. Sankur. **Survey Over Image Thresholding Techniques and Quantitative Performance Evaluation**. Journal of Electronic imaging, v. 13, n. 1, 2004, p. 146-166.

SILVA, Eduardo. **Detecção De Pedestres no Trajeto a ser Percorrido pelo Automóvel**. 2015. 93 f. TCC (Graduação) - Curso de Engenharia da Computação, Universidade do Vale do Itajaí, São José, 2015.

SIMÃO, José Eduardo de Lima. **Análise Modal Utilizando Câmeras Fotográficas como Sensores de Frequência sem Contato**. 2020. 78 f. Dissertação (Mestrado) - Curso de Engenharia Mecânica, Universidade Tecnológica Federal do Paraná, Córnelio Procopio, 2020.

SOLOMON, Chris; BRECKON, Toby. **Fundamentals of Digital Image Processing: practical approach with examples in matlab**. [S.L]: Wiley-Blackwell, 2011. 355 p.

SONKA, Milan; HILAVAC, Vaclav; BOYLE, Roger. **Image Processing, Analysis, and Machine Vision**. 4. ed. [S.L]: Cengage Learning, 2013. 930 p.

VALMORBIDA, Alan Deivis. **Desenvolvimento de um Sistema de Processamento de Imagens para Inspeção Automatizada Utilizando FPGA**. 2018. 91 f. TCC (Graduação) - Curso de Engenharia Mecatrônica, Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Florianópolis, 2018.

GONZALEZ, Rafael C.; WOODS, Richard E.. **Digital Image Processing**. 3. ed. [S.L]: Pearson Universidades, 2008.

MAZZOLA, Christian; **Técnicas de medição da permissividade complexa de dielétrico sólidos através da formatação de amostras como elementos capacitivos de dimensões conhecidas**. 2021. 179 f TCC (Graduação) – Curso de Engenharia Elétrica, Universidade Federal de Santa Catarina, Santa Catarina, Florianópolis, 2021.

ANEXO A – FUNÇÕES IMPLEMENTADAS

```
52. #funcao para
53. Def is_square(contour, tolerance=0.015):
54.     return cv2.matchShapes(contour, np.array([[0, 0
55.         ], [1, 0], [1, 1], [0, 1]]),
                               cv2.CONTOURS_MATCH_I2, 0
                               .0) < tolerance
```


ANEXO B – FUNÇÕES IMPLEMENTADAS

A distância d entre o ponto A com coordenada (x_a, y_a) e o ponto B com coordenadas (x_b, y_b) é definida em (B.1) e será usada para o cálculo entre pontos na imagem.

$$d = \sqrt{(x_B - x_a)^2 + (y_b - y_a)^2} \quad (\text{B.1})$$

Se x_1, x_2, \dots, x_n são valores dos dados, então podemos calcular a média aritmética MA e a equação (B.2).

$$MA = \frac{x_1, x_2, \dots, x_n}{n} = \frac{\sum_{i=1}^n x_i}{n} \quad (\text{B.2})$$

A variância (S) é definida como o desvio quadrático médio da média e é calculada de acordo com (B.3), e o desvio padrão (D_p) é a raiz quadrada da variância definido em (B.4).

$$S^2 = \frac{\sum_{i=1}^n (x_i - MA)^2}{n - 1} \quad (\text{B.3})$$

$$D_p = \sqrt{S^2} \quad (\text{B.4})$$

