

INSTITUTO FEDERAL

Paraíba

Campus João Pessoa

**INSTITUTO FEDERAL DA PARAÍBA
CAMPUS JOÃO PESSOA
CURSO SUPERIOR DE BACHARELADO EM ENGENHARIA ELÉTRICA**

GUILHERME LEITE MAIA

**DESENVOLVIMENTO DE SOFTWARE DE MONITORAMENTO PARA
CORREÇÃO DE POSTURA ODONTOLÓGICA**

JOÃO PESSOA – PB

2024

GUILHERME LEITE MAIA

**DESENVOLVIMENTO DE SOFTWARE DE MONITORAMENTO PARA
CORREÇÃO DE POSTURA ODONTOLÓGICA**

*Trabalho de Conclusão de Curso
submetido à Coordenação do Curso
Superior de Bacharelado em Engenharia
Elétrica do Instituto Federal da Paraíba
– Campus João Pessoa, como parte dos
requisitos necessários para a obtenção
do grau de Bacharel em Engenharia
Elétrica.*

Orientador:

Carlos Danilo Miranda Regis, Dr.

**JOÃO PESSOA – PB
2024**

GUILHERME LEITE MAIA

**DESENVOLVIMENTO DE SOFTWARE DE MONITORAMENTO PARA
CORREÇÃO DE POSTURA ODONTOLÓGICA**

*Trabalho de conclusão de curso
submetido à Coordenação do Curso
Superior de Bacharelado em Engenharia
Elétrica do Instituto Federal da Paraíba
como parte dos requisitos necessários
para a obtenção do grau de Bacharel em
Engenharia Elétrica.*

Trabalho Aprovado em 20/02/2024 pela banca examinadora:

Documento assinado digitalmente
 Carlos Danilo Miranda Regis
Data: 23/02/2024 11:08:50-0300
Verifique em <https://validar.iti.gov.br>

Carlos Danilo Miranda Regis, Dr.
Orientador, IFPB

Documento assinado digitalmente
 ALLYSSON MACARIO DE ARAUJO CALDAS
Data: 23/02/2024 11:16:26-0300
Verifique em <https://validar.iti.gov.br>

Allysson Macario de Araujo Caldas, Dr.
Examinador, IFPB

Documento assinado digitalmente
 PATRIC LACOUTH DA SILVA
Data: 23/02/2024 13:00:47-0300
Verifique em <https://validar.iti.gov.br>

Patric Lacouth da Silva, Dr.
Examinador, IFPB

**JOÃO PESSOA – PB
2024**

AGRADECIMENTOS

Agradeço a todos que contribuíram pela conclusão dessa atividade, que não é apenas o término de uma etapa acadêmica, mas também a realização de um sonho pessoal e profissional. Elevo meu coração em agradecimento a Deus, cuja divina providência, com a intercessão de São José, guiou cada passo dessa jornada. Sua graça e sabedoria foram as forças sobrenaturais que me permitiram alcançar esse marco significativo. A Ele dedico este trabalho como um tributo de minha fé inabalável em Sua sabedoria e misericórdia.

Ademais, é impossível não mencionar e agradecer a dois pilares fundamentais da minha jornada: meus amados pais, Edson Teixeira Maia e Maria Lindaci Leite de Andrade. Seu apoio inabalável, amor incondicional, humildade e sabedoria constante foram a força motriz por trás de cada desafio vencido e de cada triunfo alcançado. Agradeço-lhes do fundo do meu coração por estarem ao meu lado em cada passo dessa jornada. Além deles, a minha namorada, Vivianne Beatriz Sena Luciano por todo o companheirismo ao longo dessa jornada.

Aos meus estimados irmãos da Comunidade Shalom, minha gratidão é imensa. Sua irmandade, apoio moral e orações constantes enriqueceram minha jornada de forma indescritível. As lições de fé, oferta de vida e desejo do céu que compartilhamos permanecerão comigo para sempre.

Por fim, às minhas amadas ovelhas do grupo de oração Harami, deixo meu mais profundo agradecimento. Sua presença constante e orações fervorosas foram como um farol que iluminou meu caminho nos momentos mais desafiadores. Agradeço-lhes por fortalecerem minha vocação e compartilharem comigo seu caminho ao céu. Que esse marco seja o início de novas conquistas e realizações sob os cuidados e a vontade de Deus. A todos vocês, meu mais profundo obrigado!

RESUMO

O presente trabalho, a partir de um estudo teórico, propõe uma inovadora ferramenta desenvolvida em Python, utilizando a biblioteca OpenCV. Trata-se de um algoritmo para redução do hábito de sucção de dedo e/ou chupeta, utilizando a linguagem *Python*, já que a má oclusão é uma preocupação comum na odontologia pediátrica e pode estar relacionada a hábitos de sucção de dedo e/ou chupeta. A má oclusão refere-se ao desalinhamento dos dentes e ao posicionamento inadequado da mandíbula. A utilização da biblioteca OpenCV permite a criação de um algoritmo de detecção e acompanhamento dos movimentos da mão e da boca da criança, identificando quando ocorre o hábito da sucção. Através de técnicas de visão computacional, o algoritmo registra esses momentos quando a criança se encontra em frente ao computador e fornece feedback visual, auxiliando na conscientização da criança e dos pais sobre o hábito prejudicial. Este trabalho representa uma abordagem inovadora e promissora para abordar o problema da sucção de dedo e chupeta em crianças, oferecendo uma ferramenta eficaz para modificar comportamentos, promovendo a saúde bucal e prevenindo potenciais problemas de oclusão.

Palavras-chave: Má oclusão; Criança; Python; OpenCV.

ABSTRACT

This work, that was made based on a theoretical study, proposes an innovative tool developed in Python using the OpenCV library. An algorithm to reduce finger and/or pacifier sucking habits using Python language, as malocclusion is a common concern in pediatric dentistry, and may be related to thumb and/or pacifier sucking habits. Malocclusion refers to the misalignment of the teeth and wrong positioning in tradition. The using of the OpenCV library allows the creation of an algorithm to detecting and monitoring the movements of the child's hand and mouth, identifying when the sucking habit occurs. Using computer vision techniques, the algorithm records these moments when the child is in front of the computer and provides visual feedback, helping to raise awareness among children and parents about this baddly habit. This work represents an approach innovative and promising approach to addressing the problem of thumb and pacifier sucking in children, offering an effective tool to modify behaviors, promoting oral health and preventing potential occlusion problems.

Keywords: Malocclusion; Child; Python; OpenCV.

LISTA DE FIGURAS

Figura 1 - Tipo de sucção nutricional.....	11
Figura 2 - Exemplo de sucção não nutritiva.....	12
Figura 3 - Ato de chupar o dedo é considerado sucção não nutritiva.....	12
Figura 4 - Exemplo de oclusão hipoteticamente ideal.....	13
Figura 5 - Contatos bilaterais e simultâneos e estáveis.....	14
Figura 6 - Vista superior do alinhamento correto dos dentes nas arcadas.....	14
Figura 7 - Trespasse vertical e horizontal harmônicos, em que as relações entre os arcos dentários superior e inferior são equilibradas e harmônicas em ambas as dimensões vertical e horizontal.....	15
Figura 8 - Dentição sem perda de inserção e sem mobilidade.....	16
Figura 9 - Espaço tridimensional existente entre os dentes superiores e inferiores quando a mandíbula está em posição de repouso, permitindo movimentos mandibulares livres e naturais.....	16
Figura 10 - Linha imaginária da curva de Spee e Wilson.....	17
Figura 11 - Capacidade natural e desimpedida da mandíbula de realizar uma série de movimentos sem encontrar interferências ou bloqueios significativos.....	17
Figura 12 - As forças oclusais distribuídas nas áreas de trabalho e no sentido axial dos dentes referem-se às pressões ou cargas exercidas durante o processo de mastigação..	17
Figura 13 - Alinhamento adequado dos dentes, uma distribuição equilibrada das forças oclusais e uma função saudável da ATM.....	18
Figura 14 - Possíveis efeitos da sucção não nutritiva, especialmente quando prolongada.....	19
Figura 15 – Representação da mordida aberta anterior (MAA).....	20
Figura 16 - Mordidas cruzadas posteriores.....	20
Figura 17 - Tipos de Kernel.....	23
Figura 18 – Interferência da luminosidade.....	25
Figura 19 - Agrupamento das classes e interferência da luminosidade.....	27
Figura 20 - Tipo de análise LBP.....	28
Figura 21 - Interferência da luminosidade.....	28
Figura 22 - Faces cadastradas para fase de teses.....	37
Figura 23 - Teste do algoritmo em pleno funcionamento, constatando que para faces em situações normais, foram atribuídos números ímpares.....	38
Figura 24 - Teste bem-sucedido do algoritmo em pleno funcionamento, revelando que, em circunstâncias nas quais a pessoa colocar a chupeta, números pares foram associados a faces.....	38
Figura 25 - Teste efetuado com sucesso no algoritmo em pleno funcionamento, indicando que, quando a pessoa coloca o dedo na boca em circunstâncias específicas, números pares são atribuídos às faces.....	39

LISTA DE ABREVIATURAS E SIGLAS

ATM	Articulação Temporomandibular
MAA	Mordida Aberta Anterior
MIH	Máxima Intercuspidação Habitual
OPENCV	<i>Open Source Computer Vision Library</i>

SUMÁRIO

1	INTRODUÇÃO	9
2	OBJETIVOS	10
2.1	OBJETIVO GERAL	10
2.2	OBJETIVOS ESPECÍFICOS	10
3	EMBASAMENTO TEÓRICO	11
3.1	HÁBITOS DE SUCCÃO.....	11
3.2	OCLUSÃO.....	13
3.3	MALOCLUSÃO ADVINDA DO HÁBITO DA SUCCÃO DE DEDO E/OU CHUPETA	18
3.4	PYTHON	20
3.5	OPENCV	21
3.6	RECONHECIMENTO FACIAL E SUAS ETAPAS	21
3.6.1	HAAR CASCADE.....	22
3.6.2	CARACTERÍSTICAS DE HAAR.....	22
3.6.3	CLASSIFICADOR EM CASCATA.....	23
3.7	<i>EIGENFACE</i>	24
3.8	<i>FISHERFACE</i>	25
3.9	LBPHFACES	27
3.10	COMPARAÇÃO ENTRE OS TRÊS CLASSIFICADORES EIGENFACES, FISHERFACES E LBPHFACES	29
4	METODOLOGIA	30
4.1	COLETA DAS IMAGENS	30
4.2	TREINAMENTO.....	34
4.3	RECONHECIMENTO DE FACE E AÇÃO MITIGADORA	35
5	RESULTADOS FINAIS	37
6	CONCLUSÃO	40
	REFERÊNCIAS	41

1 INTRODUÇÃO

Hoje, muitos algoritmos são treinados para realizar aprendizagem e generalização, com o intuito de simular as capacidades humanas, como por exemplo, veículos autônomos, robôs, detecção de fraudes, análise de *streaming* de dados, previsão de demanda, processamento de linguagem natural, entre outras (RASCHKA, PATTERSON, NOLET, 2020).

Um problema comum em odontopediatria está relacionado à má oclusão, que é desenvolvida com os hábitos de sucção de dedo e/ou chupeta em crianças, configurando-se como a terceira prioridade de problemas de saúde bucal no Brasil (BOECK et al. apud OMS, 2013). A má oclusão refere-se ao desalinhamento dos dentes e à posição inadequada da mandíbula (OMS, 1991).

Quando uma criança desenvolve o hábito de chupar o dedo ou a chupeta, principalmente por longos períodos e com frequência, isso pode impactar negativamente no desenvolvimento dos dentes e da estrutura facial. Esses hábitos podem causar problemas, como protrusão dos dentes anteriores (dentes da frente que se projetam para a frente), mordida aberta (espaço entre os dentes superiores e inferiores quando a criança fecha a boca) e desalinhamento dos dentes (BOECK *et al.*, 2013).

Diante desses fatos, estudos para o desenvolvimento de técnicas através do reconhecimento facial tornaram-se importante para agilizar processos nas mais diversas áreas possíveis, seja para aumentar a segurança, seja para estimular o processo de acesso a sistemas, indo desde segurança biométrica até aplicativos de entretenimento.

A importância desses algoritmos reside na capacidade de identificar e autenticar indivíduos com base em suas características faciais únicas, proporcionando uma camada adicional de segurança. A capacidade de realizar análises precisas em tempo real e o desenvolvimento contínuo desses algoritmos tornam-nos uma tecnologia valiosa na nossa sociedade cada vez mais digital.

Nesse sentido, este trabalho propõe desenvolver e analisar um algoritmo para identificar o hábito de sucção de dedo e/ou chupeta, utilizando a linguagem Python, a partir do uso das funções disponíveis na biblioteca de processamento de imagem e vídeo OpenCv.

2 OBJETIVOS

2.1 OBJETIVO GERAL

O presente projeto tem como objetivo desenvolver um software de monitoramento via *Python* para correção de postura odontológica.

2.2 OBJETIVOS ESPECÍFICOS

- Desenvolver e implementar algoritmo de aprendizado de máquina, capaz de identificar o hábito de sucção de dedo e/ou chupeta;
- Realizar a aquisição de um conjunto representativo de imagens relacionadas ao comportamento de sucção de dedo e/ou chupeta para o treinamento do algoritmo;
- Aplicar técnicas de pré-processamento para melhorar a qualidade e a uniformidade das imagens, incluindo normalização e correção de iluminação;
- Elaborar um método para extrair características distintivas das imagens que possam ser utilizadas como entradas para o algoritmo de aprendizado de máquina;
- Desenvolver e implementar medidas de intervenção baseadas nos resultados do algoritmo, visando à atenuação eficaz do hábito de sucção de dedo e/ou chupeta.

3 EMBASAMENTO TEÓRICO

Neste capítulo serão apresentados os conceitos e fundamentos que constituem a base para a compreensão do trabalho proposto. Além disso, será realizada uma descrição das ferramentas utilizadas na implementação e execução da solução abordada.

3.1 HÁBITOS DE SUCÇÃO

A ação de sucção é o primeiro ato muscular coordenado e realizado por um bebê. A sucção pode ser dividida em duas categorias principais: nutricional, que fornece ao bebê nutrientes essenciais, e não nutritiva, que proporciona conforto e segurança (O'BRIEN et al., 1996).

A sucção nutricional refere-se à ação de sugar, que tem como objetivo principal a ingestão de alimentos. Isso ocorre quando o bebê ou a criança suga o seio materno ou uma mamadeira para obter leite ou outro alimento essencial para o seu crescimento e desenvolvimento. A sucção nutricional (Figura 1) é uma parte vital do processo de alimentação e desempenha um papel crucial na nutrição infantil (CUNHA et al., 1998).

Figura 1 - Tipo de sucção nutricional



Fonte: Mais Saúde revista, 2019.

Por outro lado, a sucção não nutritiva (Figuras 2 e 3) ocorre quando o bebê ou a criança suga para fins de consolo, conforto ou segurança. Isso pode se manifestar por meio do uso de chupetas, hábito de chupar o dedo ou até mesmo objetos como cobertores.

A sucção não nutritiva é um comportamento natural e comum em bebês (Figura 2) (BOECK et al., 2013).

Figura 2 - Exemplo de sucção não nutritiva



Fonte: Pediatria Descomplicada, 2018.

É comum que a sucção não nutritiva se mantenha, por exemplo; sucção de dedos e chupetas durante a infância. O hábito prolongado de chupar o dedo pode ter impactos negativos na saúde bucal da criança e no desenvolvimento arcada dentária. Alguns dos prejuízos incluem: problemas de mordida, alterações da arcada dentária, problemas no desenvolvimento da fala e dificuldade na amamentação (Figura 3) (BOECK et al., 2013).

Figura 3 - Ato de chupar o dedo é considerado sucção não nutritiva



Fonte: Pediatria Descomplicada, 2018.

3.2 OCLUSÃO

Oclusão é um termo da odontologia que se refere à forma como os dentes superiores e inferiores se articulam e se tocam durante o ato de fechar a boca. É um aspecto necessário à função mastigatória e à harmonia do sistema estomatognático, que consiste em dentes bem alinhados e bem posicionados (Figura 4) (MESOMO, LOSSO, 2004).

Figura 4 - Exemplo de oclusão hipoteticamente ideal



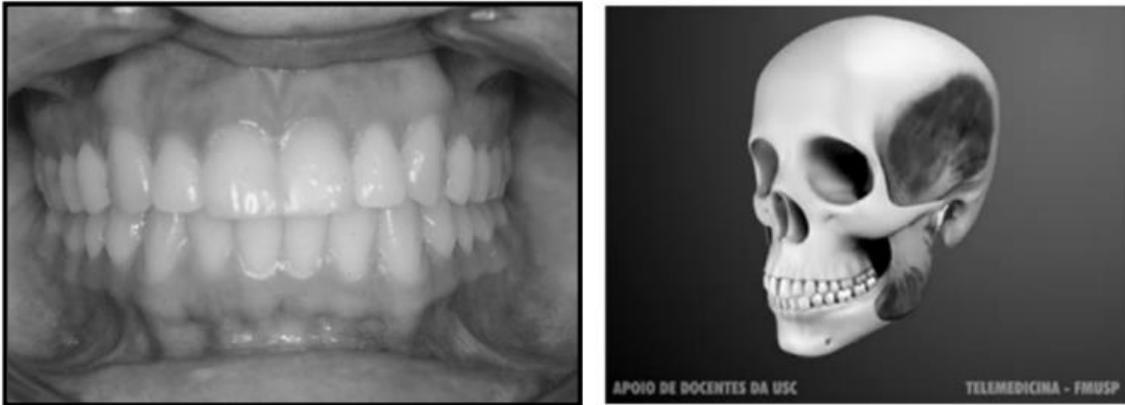
Fonte: SILVA, 2013

O estudo da oclusão envolve a análise da relação das superfícies dos dentes superiores com as dos dentes inferiores, quando a mandíbula está em posição de repouso e durante os movimentos de mastigação, fala e deglutição (MESOMO, LOSSO, 2004).

Uma oclusão ideal é aquela em que há:

a) Contatos bilaterais simultâneos e estáveis entre todos os dentes, coincidindo com a harmonia muscular (Figura 5).

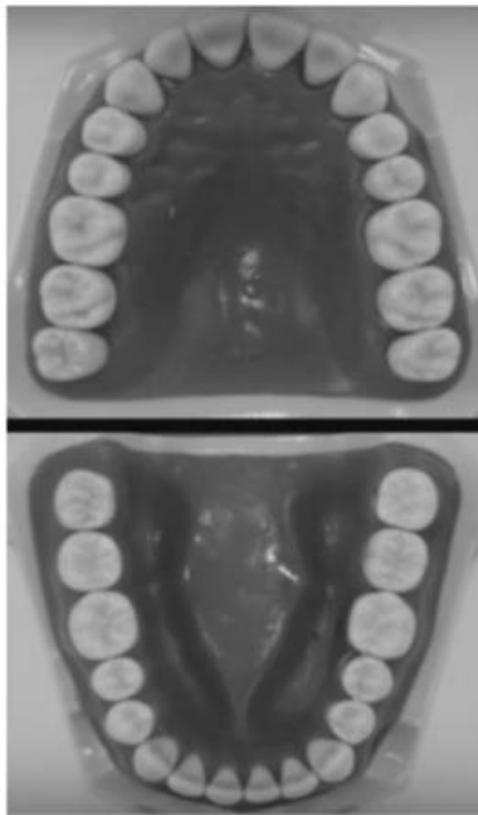
Figura 5 - Contatos bilaterais e simultâneos e estáveis.



Fonte: BATAGLION, 2020.

b) Alinhamento correto dos dentes nas arcadas (Figura 6).

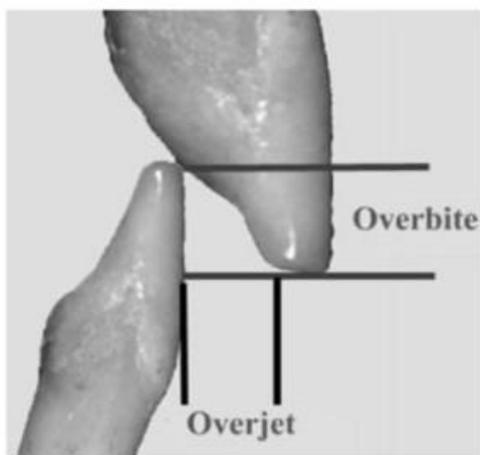
Figura 6 - Vista superior do alinhamento correto dos dentes nas arcadas



Fonte: BATAGLION, 2020.

c) *Overjet* (trespasse horizontal): é a distância horizontal em que os dentes superiores se projetam sobre os inferiores, quando em máxima intercuspidação habitual (MIH). É uma posição maxilomandibular, em que o maior número de dentes se toca, ou seja, tem o máximo de contato oclusal. Já o *Overbite* (trespasse vertical) é a distância em que se projetam verticalmente os dentes superiores dos inferiores, quando em MIH. Para ser oclusão ideal, não deve ultrapassar 1/3 da altura do incisivo central inferior (Figura 7).

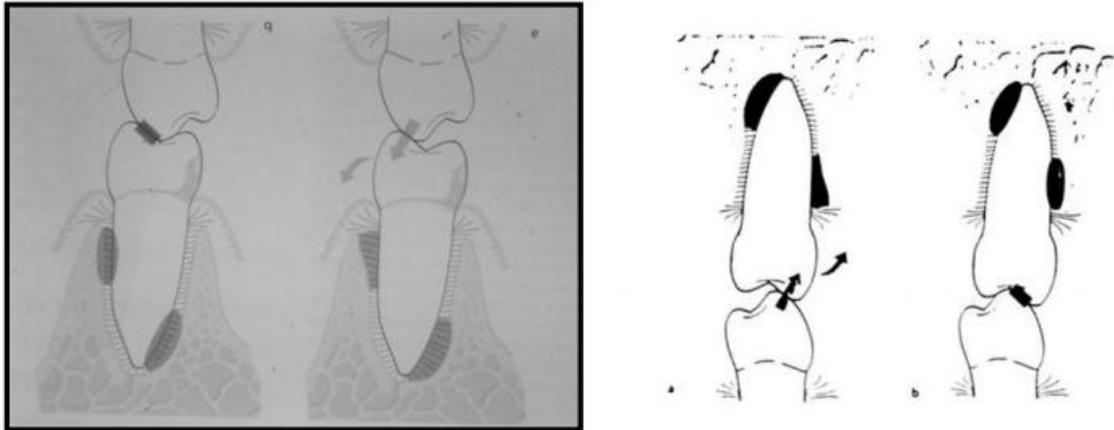
Figura 7 - Trespasse vertical e horizontal harmônicos, em que as relações entre os arcos dentários superior e inferior são equilibradas e harmônicas em ambas as dimensões vertical e horizontal



Fonte: QUINTESSENCE apud BATAGLION, 2020.

d) Relação dente-periodonto normal, sem trauma oclusal (Figura 8): refere-se a uma condição em que os dentes estão em uma posição e oclusão adequada, ou seja, sem a presença de impactos adversos causados por forças excessivas durante oclusão ou mastigação. Já a periodonto é a parte de sustentação dos dentes, que está destacada na cor preta envolvendo-os (Figura 8), pois quando eles estão com uma oclusão não ideal, sobrecarregam-se, já que isso afeta a mobilidade e interfere no sulco gengival, que é o espaço entre a gengiva e o dente em condições ordinárias; o sulco gengival é raso e não permite que a sonda penetre profundamente.

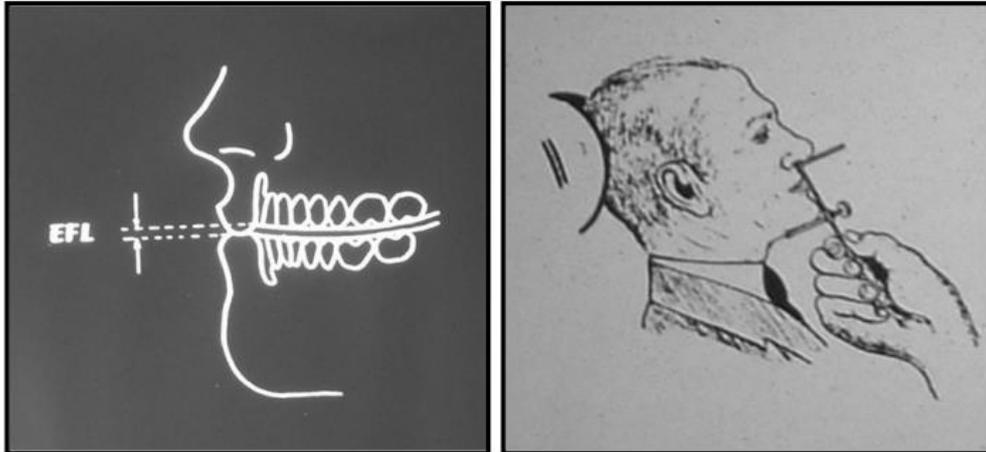
Figura 8 - Dentição sem perda de inserção e sem mobilidade



Fonte: LINDHE; BATAGLION apud BATAGLION, 2020.

e) Espaço funcional livre fisiológico (Figura 9): é a distância que naturalmente existe entre os dentes superiores e inferiores quando a mandíbula está em seu estado de repouso.

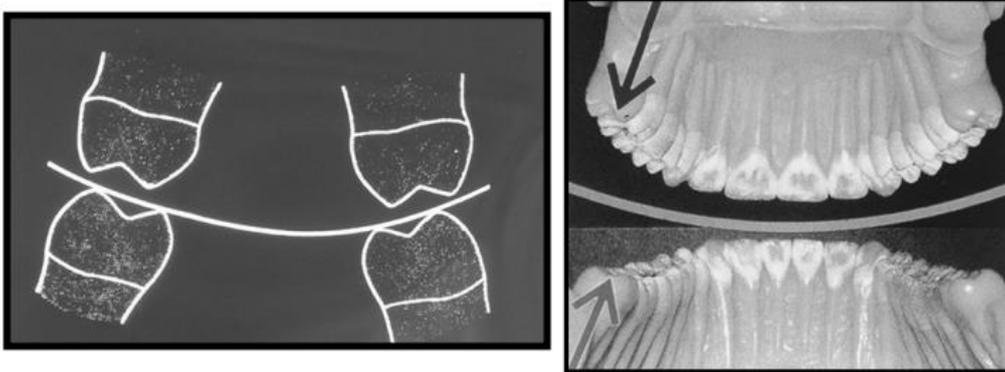
Figura 9 - Espaço tridimensional existente entre os dentes superiores e inferiores quando a mandíbula está em posição de repouso, permitindo movimentos mandibulares livres e naturais



Fonte: BATAGLION, 2020.

f) Curvas de Compensação (*SPEE* e *WILSON*) em harmonia (Figura 10): a curva de *Spee* é uma linha imaginária que vai da cúspide do canino até os dentes posteriores pela parte vestibular, e a de *Wilson* é a inclinação dos molares inferiores para lingual e dos superiores para vestibular, quando os arcos são observados em oclusão por uma vista de frente.

Figura 10 - Linha imaginária da curva de Spee e Wilson.



Fonte: BATAGLION, 2020.

g) Movimentos mandibulares livres, sem interferências ou bloqueios (Figura 11).

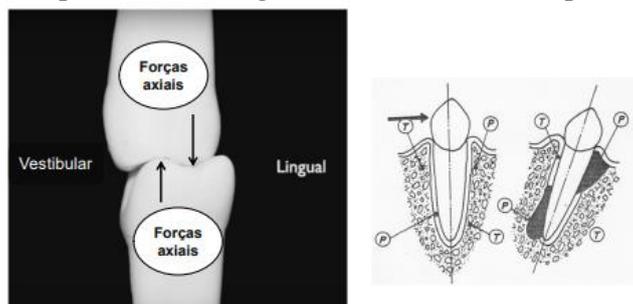
Figura 11 - Capacidade natural e desimpedida da mandíbula de realizar uma série de movimentos sem encontrar interferências ou bloqueios significativos



Fonte: BATAGLION, 2020.

h) Forças oclusais distribuídas nas áreas de trabalho e no sentido axial (longo eixo) dos dentes (Figura 12).

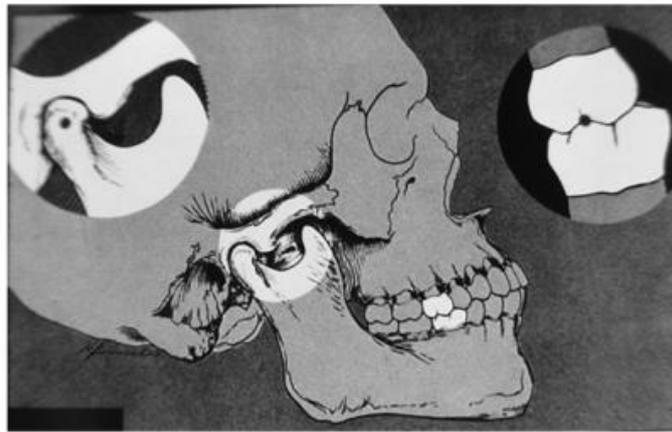
Figura 12 - As forças oclusais distribuídas nas áreas de trabalho e no sentido axial dos dentes referem-se às pressões ou cargas exercidas durante o processo de mastigação



Fonte: BATAGLION, 2020.

i) Perfeita harmonia entre a oclusão dentária e a articulação temporomandibular (ATM) (Figura 13). Sendo necessária uma perfeita harmonia, os movimentos da mandíbula, como abrir, fechar e movimentos laterais, são realizados suavemente e sem causar estresse.

Figura 13 - Alinhamento adequado dos dentes, uma distribuição equilibrada das forças oclusais e uma função saudável da ATM



Fonte: NEFF apud BATAGLION, 2020.

A busca por uma oclusão equilibrada e funcional é um dos objetivos da odontologia, visando à saúde e ao bem-estar do paciente. Isso pode ser alcançado por meio de tratamentos ortodônticos, restaurações dentárias e outras intervenções, sempre com a finalidade de assegurar uma oclusão correta e harmônica.

3.3 MALOCLUSÃO ADVINDA DO HÁBITO DA SUCÇÃO DE DEDO E/OU CHUPETA

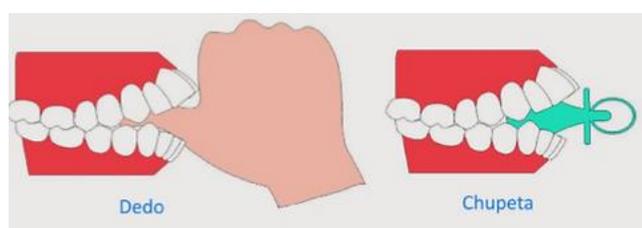
A maloclusão é caracterizada por desvios de normalidade das arcadas dentárias. Estudos apontam que esses desvios causam uma série de problemas fisiológicos nos indivíduos. Uma das causas da maloclusão é a sucção discriminada do dedo e/ou chupeta (Suliano et al., 2007).

O uso de chupetas em crianças é uma das formas adotadas para confortar o bebê, por exemplo, quando submetido a procedimentos dolorosos ou estressantes, relacionados a momento de agitação, irritação ou choro intenso. Outros métodos além do uso da chupeta também são usados na tentativa de proporcionar uma maior calma ao bebê,

isso porque o ato de sucção é um instinto natural de um recém-nascido (FIGUEIREDO; VERÍSSIMO; LIMA; 2022).

Contudo, são igualmente observáveis efeitos adversos significativos decorrentes do hábito da sucção de dedo e/ou chupeta (Figura 14). Por causa de tal hábito, depois dos três anos, a maloclusão poderá ser perpetuada pela função anormal estabelecida (Cunha et al., 1998; Pastor et al., 2000). A prevalência de maloclusões entre crianças com hábitos de sucção varia de 43% a 90%, sendo o hábito de sucção de chupeta mais frequente (Calisti et al., 1960; Svedmyr, 1979).

Figura 14 - Possíveis efeitos da sucção não nutritiva especialmente quando prolongada



Fonte: ODONTOYAMA, 2016.

Em uma pesquisa realizada pela Universidade Federal do Paraná, Mesomo e Losso (2024) apresentaram que as crianças com hábito prolongado de chupeta têm uma alta frequência de mordida aberta anterior (MAA), condição ortodôntica caracterizada pela falta de contato entre os dentes superiores e inferiores quando a boca está fechada, resultando em um espaço visível entre os arcos dentários (Figura 15), independentemente do tipo de chupeta usada, sendo ela convencional ou ortodôntica.

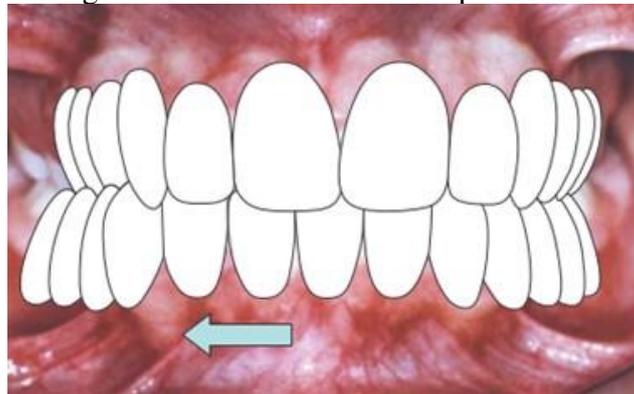
Já crianças usuárias de chupetas ortodônticas apresentaram maior frequência de mordidas cruzadas posteriores, condição em que os dentes superiores não se alinham corretamente com os dentes inferiores na parte de trás da boca, resultando em uma sobreposição inadequada quando a mandíbula está fechada, quando comparado aos usuárias de chupetas do modelo convencional (Figura 16). Quando avaliados os efeitos do uso sobre a oclusão, a chupeta ortodôntica não pareceu apresentar vantagens sobre a chupeta convencional em hábitos prolongados.

Figura 15 – Representação da mordida aberta anterior (MAA)



Fonte: BUENO, George, 2022.

Figura 16 - Mordidas cruzadas posteriores



Fonte: BUENO, George, 2022.

3.4 PYTHON

A Python é uma das várias linguagens de programação que existem, uma linguagem muito simples e que está ficando bastante conhecida nos últimos anos. Algumas das empresas que utilizam essa linguagem são: Autodesk, Globo, Google, Zope, YouTube, NASA, Air Canada, Industrial Light & Magic, entre outras. Isso ocorre porque a linguagem de programação Python possui um alto nível, interpretada e de propósito geral.

Desenvolvida por Guido van Rossum e lançada pela primeira vez em 20 de fevereiro de 1991, ganhou considerável popularidade devido à sua sintaxe clara e legibilidade, tornando-a acessível até mesmo para programadores iniciantes, sendo suficiente para ser usada em uma ampla variedade de aplicações.

Os elementos que fazem do Python uma escolha excelente para a aprendizagem são:

- É fácil de aprender – o tempo necessário para aprender Python é menor do que para muitas outras linguagens, e isso significa que é possível iniciar a programação propriamente dita mais rapidamente;
- É fácil de usar para escrever novo software – é possível escrever código mais rapidamente ao usar Python;
- É fácil de obter, instalar e implantar – Python é gratuito, aberto e multiplataforma.

3.5 OPENCV

A biblioteca de software OpenCV (*Open Source Computer Vision Library*) destaca-se como uma das ferramentas mais empregadas para atividades relacionadas à visão computacional e ao processamento de imagens (MARENGONI; STRINGHINI, 2009). Foi desenvolvida inicialmente pela Intel em 2000 e permite desenvolvimento de algoritmos de processamento de imagem: detectar e reconhecer rostos, identificar objetos, calibração de câmera, rotulagem de imagem, análise estrutural e classificar ações humanas em vídeos (MARENGONI; STRINGHINI, 2009).

Além da biblioteca OpenCV, outras bibliotecas comuns para trabalhar com reconhecimento de imagens incluem TensorFlow, PyTorch, Keras, entre outras (BARROS, 2023).

3.6 RECONHECIMENTO FACIAL E SUAS ETAPAS

O reconhecimento facial toma como referência traços do rosto de pessoas e é uma técnica bastante utilizada, tendo como objetivo identificar e fornecer informações referentes àqueles que a utilizarão. É basicamente dividida em três etapas, que são: Detecção da face, Extração de características e Reconhecimento da face (BRAGA, 2013).

A detecção da face é a etapa mais simples, pelo fato de ainda não ser necessário reconhecer a pessoa, mas apenas a presença do rosto. Esse reconhecimento delimita a

região da imagem em que está contida o rosto do indivíduo; após essa delimitação, essa região é separada, e o restante da imagem é descartado (BRAGA, 2013).

Após identificar a face, o algoritmo de reconhecimento facial vai extrair as características da face detectada. Nesse aspecto, podemos destacar as seguintes características faciais: espaço entre os olhos, largura da boca, tamanho do nariz, cor do cabelo, humor e faixa etária. Essas características são definidas por pontos que são intitulados de *landmarks* ou pontos fiduciais, os quais definem as características mencionadas anteriormente (BRAGA, 2013).

Já na etapa do reconhecimento, os pontos detectados na etapa anterior são comparados com um banco de dados. A construção desse banco de dados se dá por meio de um treinamento, no qual as faces são cadastradas em um banco de dados, e de posse desse banco de dados, o algoritmo faz uma comparação entre o banco de dados com os pontos detectados na face. Caso exista uma compatibilidade entre os pontos, ou seja, entre as características, o algoritmo finaliza o reconhecimento (BRAGA, 2013).

3.6.1 HAAR CASCADE

A primeira etapa para o reconhecimento facial é a detecção da face humana, para que a partir dessa imagem, possam ser captadas as características necessárias para o reconhecimento fácil. O classificador *Haar Cascade* foi proposto no ano de 2001 em um artigo com o título de “Detecção rápida de objetos usando uma cascata impulsionada de recursos simples”. O objetivo inicial, como o próprio título fala, era o reconhecimento de objetos de uma forma clara e simples. No caso estudado, esse objeto passa a ser um rosto, presença ou não presença de um rosto na imagem. Esse classificador introduz quatro conceitos fundamentais para realizar a detecção dos objetos: as características de Haar; as imagens integrais; o algoritmo de aprendizagem supervisionado *Adaboost* e os classificadores em cascata (OPENCV, 2016).

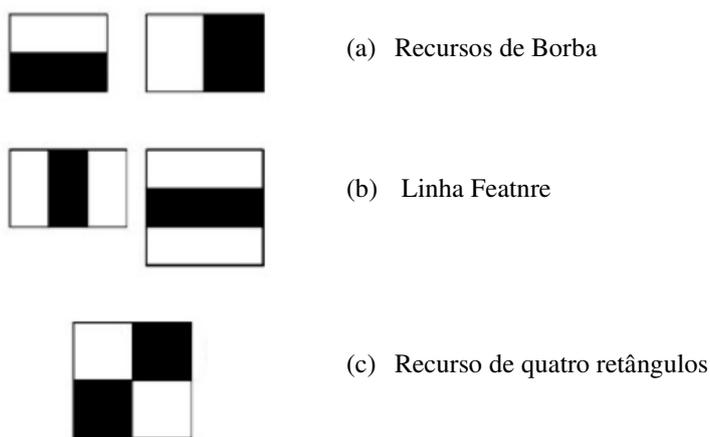
3.6.2 CARACTERÍSTICAS DE HAAR

Esse algoritmo trabalha com o princípio de aprendizagem de máquina, ou seja, é preciso apresentar para a máquina qual imagem possui um rosto (imagens positivas) e quais imagens não possuem um rosto (imagens negativas) para treinar o classificador.

Para isso, são utilizados recursos Haar mostrados na Figura 18, para a extração de padrões e texturas, utilizadas como entrada para o classificador em cascata (OPENCV, 2016).

Esse algoritmo percorre dois ou mais retângulos unidos de coloração branca e preta. Cada uma dessas características resulta em um único valor obtido através da subtração da soma dos pixels nos retângulos brancos pela soma dos pixels dos retângulos pretos, em que os pixels serão analisados e armazenados para estudo (OPENCV, 2016). São utilizados todos os tamanhos possíveis de cada kernel para calcular, ou seja, muitos recursos são necessários para isso. Por exemplo, uma janela 24x24 resulta em mais de 160.000 recursos. Para cada cálculo de recurso, precisamos encontrar a soma dos pixels sob os retângulos branco e preto, ou seja, algo bastante lento e não apropriado para aplicações em tempo real.

Figura 17 - Tipos de Kernel



Fonte: Opencv, 2020.

3.6.3 CLASSIFICADOR EM CASCATA

Os classificadores em cascata referem-se a uma abordagem sequencial no processo de detecção de objetos, com o objetivo de otimizar a eficiência e reduzir o tempo de processamento. Nessa etapa final do algoritmo, as regiões irrelevantes, que não possuem características semelhantes à procura são descartadas rapidamente, evitando uma análise mais detalhada dessas áreas, para então reservar os classificadores mais específicos para regiões mais propensas a conterem o objeto de interesse. Essa abordagem

em cascata é frequentemente utilizada em algoritmos de detecção de objetos, visando otimizar o desempenho do sistema, especialmente em cenários nos quais a maioria das regiões não contém o objeto de interesse.

3.7 EIGENFACE

Essa técnica foi desenvolvida por Sirovich e Kirby (1987), com o objetivo de representar, de forma eficiente, imagens de faces através da análise de componentes principais (PCA – *Principal Component Analysis*). Basicamente, cada face pode ser representada como a combinação linear de diversas Eigenfaces. Uma vez que as faces são representadas como combinações lineares de Eigenfaces, é possível reconstruir uma face a partir de um conjunto relativamente pequeno de Eigenfaces. Isso torna a técnica útil para reconhecimento facial e outras tarefas de processamento de imagem (BRAGA, 2013).

Esse tipo de algoritmo usa a matriz da foto para que se possa realizar o reconhecimento facial por comparação, ou seja, é utilizada a quantidade de linhas e colunas para que seja criado o vetor EigenFace. Em uma foto com 220 linhas e 220 colunas, ou seja, 220x220, que também pode ser representada por 220², teremos um vetor de dimensão de 48400. O somatório é feito basicamente como detalhado abaixo (BRAGA, 2013).

1. Com um conjunto de M faces, temos

$$T = (T_1, T_2, T_3, \dots, T_M) \quad (3)$$

em que T_1 são os pixels de cada face do conjunto.

2. A média das imagens é calculada para centralizar o conjunto de dados (ψ)

$$T = \frac{\sum_{i=1}^M T_i}{M} \quad (4)$$

3. É criada uma matriz com as faces de treino com os pixels dispostos em linhas e as M faces do conjunto de treinamento dispostas em colunas.

4. Realiza-se uma subtração de cada imagem T com imagem média (ψ), gerando uma nova matriz

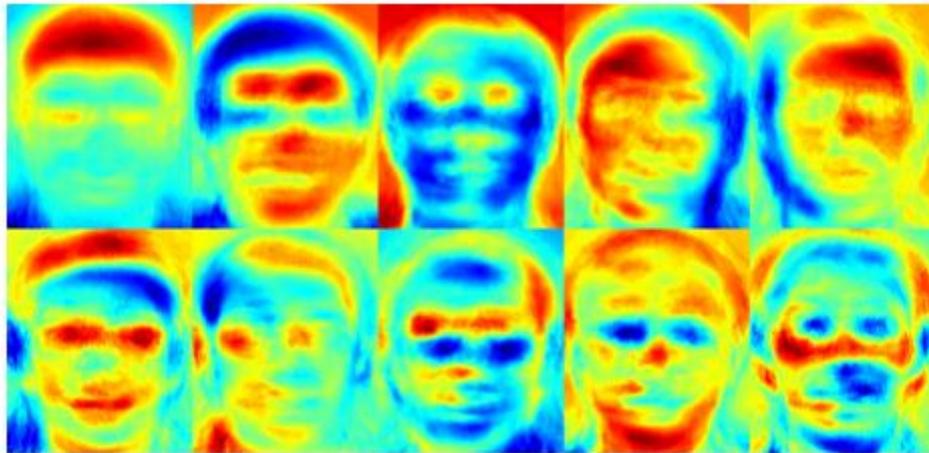
$$\Phi = T - \psi. \quad (5)$$

5. Sendo M menor que a dimensionalidade (largura * altura das imagens de treinamento) das imagens em Φ , assim calculando a matriz de covariância (C),

$$C = T - \psi. \quad (6)$$

O método de reconhecimento facial por vetor, mais conhecido como *EigenFace*, sofre bastante variação com os tons de cinza da foto analisada, ou seja, a luminosidade irá interferir na identificação do indivíduo. A Figura 19 mostra bem como ocorre essa variação de acordo com a luminosidade do ambiente.

Figura 18 – Interferência da luminosidade



Fonte:OpenCV, 2020.

3.8 FISHERFACE

Esse método de reconhecimento facial foi idealizado e construído pelo estatístico Sir RA Disher e foi publicado com sucesso no seu artigo para classificar flores em 1936.

Esse algoritmo é uma adaptação do código *EigenFaces*, no qual são utilizados apenas os componentes principais que diferenciam as pessoas entre si (OPENCV, 2020).

Esse método encontra grupos semelhantes e os classifica. Após a classificação, as classes com maior semelhança ficam perto umas das outras, e as classes com menos semelhanças ficam o mais distante possível umas das outras. Assim, não serão feitas apenas comparações com um vetor fixo, como é o caso do *EngeinFaces*. A abordagem é mais dinâmica e flexível. Em vez disso, as imagens são agrupadas com base em suas características semelhantes, usando técnicas de clusterização ou agrupamento. Isso significa que as imagens são comparadas umas com as outras, em vez de serem comparadas com um único vetor fixo (OPENCV, 2020).

Esse método se mostra mais eficaz e mais rápido para processar e detectar cada face devido ao seu agrupamento facilitar o cálculo computacional. Alguns somatórios matemáticos descrevem bem esse processo, são eles:

Seja um vetor aleatório com amostras retiradas de classes: X_c .

$$X = \{X_1, X_2, \dots, X_c\} \quad (7)$$

$$X_{Ea} = \{x_1, x_2, \dots, x_n\} \quad (8)$$

As matrizes de dispersão e SW são calculadas como: SB

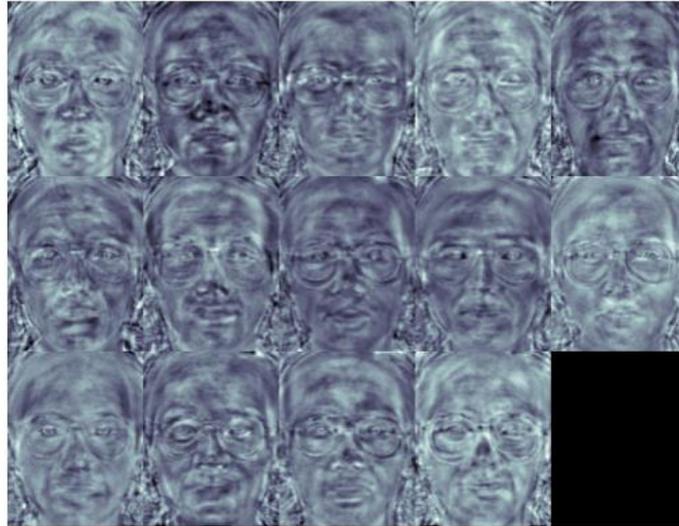
$$S_B = \sum_{i=1}^c N_{Eu} (\mu_{Eu} \mu)^T \quad (9)$$

$$S_W = \sum_{i=1}^c \sum_{x_j \in X_{Eu}} (x_j - \mu_{Eu})(x_j - \mu_{Eu})^T \quad (10)$$

$$\mu = \frac{1}{M} \sum_{i=1}^N x_{Eu} \quad (11)$$

O método FisherFace não sofre grandes variações com a luminosidade do ambiente devido ao seu método de agrupamento das classes mais semelhantes ficarem o mais próximas possível uma da outra. Com isso, a captura das características faciais é mais fidedigna, como pode ser observado na Figura 19.

Figura 19 - Agrupamento das classes e interferência da luminosidade

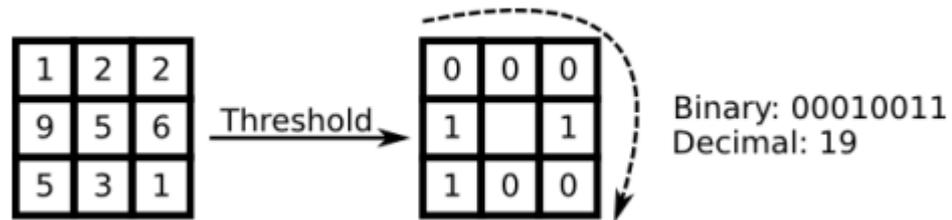


Fonte:Opencv, 2020.

3.9 LBPHFACES

O Local Binary Pattern (LBP) é um algoritmo de processamento de imagem frequentemente usado para extração de características. Esse método se mostra bem robusto devido a sua forma de implementação. Basicamente o funcionamento desse algoritmo é analisar os pixels vizinhos do pixel escolhido, como melhor ilustrado na Figura 20. Assim, o pixel de comparação a ser formado não dependerá apenas dele mesmo, mas sim dos pixels que dividem fronteira com ele.

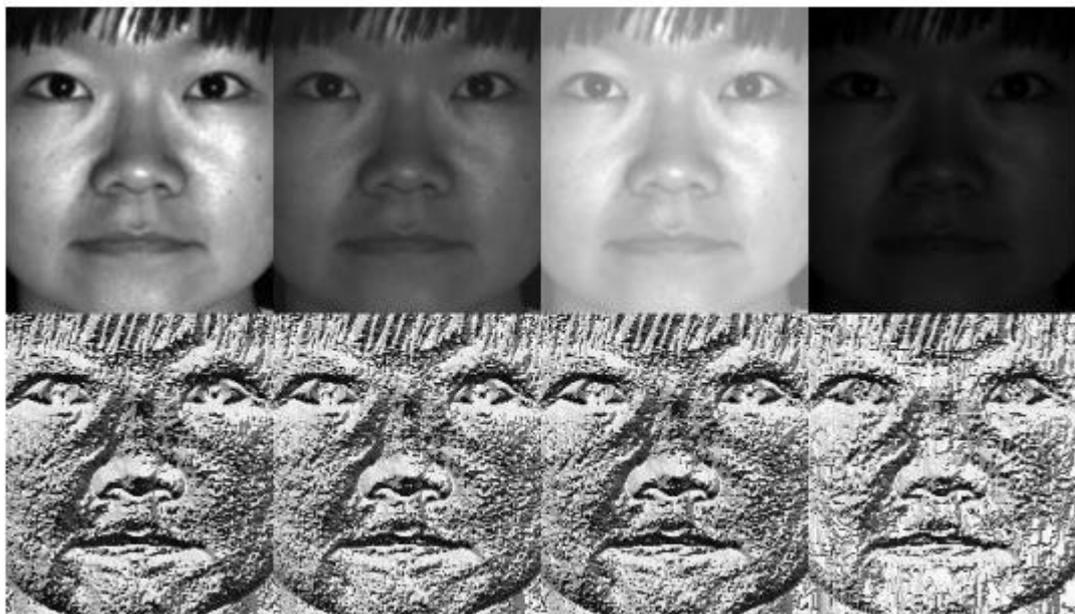
Figura 20 - Tipo de análise LBP



Fonte:Opencv, 2020.

Com isso, podemos observar que o algoritmo LBP proporciona uma menor variação dos pixels em relação à variação da luminosidade, consequentemente a assertividade desse algoritmo acaba sendo bastante alta (Figura 21). Isso se dá pelo fato de o algoritmo operar localmente em pequenas regiões da imagem, comparando cada pixel com seus vizinhos imediatos. Isso significa que ele é menos sensível a variações globais de iluminação, pois se concentra nas diferenças locais de intensidade entre os pixels

Figura 21 - Interferência da luminosidade



Fonte:Opencv, 2020.

3.10 COMPARAÇÃO ENTRE OS TRÊS CLASSIFICADORES EIGENFACES, FISHERFACES E LBPHFACES

Uma comparação entre os três métodos populares de reconhecimento facial dependerá das características específicas do problema de reconhecimento facial, das condições de aquisição de imagem e dos requisitos de desempenho. Resultados encontrados mostram o Eigenfaces como o método com processamento mais rápido e simples, porém baixa robustez na detecção quando há diferenças entre as iluminações dos ambientes de treinamento e reconhecimento. Já o Fisherfaces requer uma carga computacional maior, porém apresenta-se menos sensível às variações de iluminações. O LBPHFaces, por sua vez, é o método com processamento mais robusto em relação a variações de iluminação, lida bem com variações locais e é menos sensível a mudanças na pose.

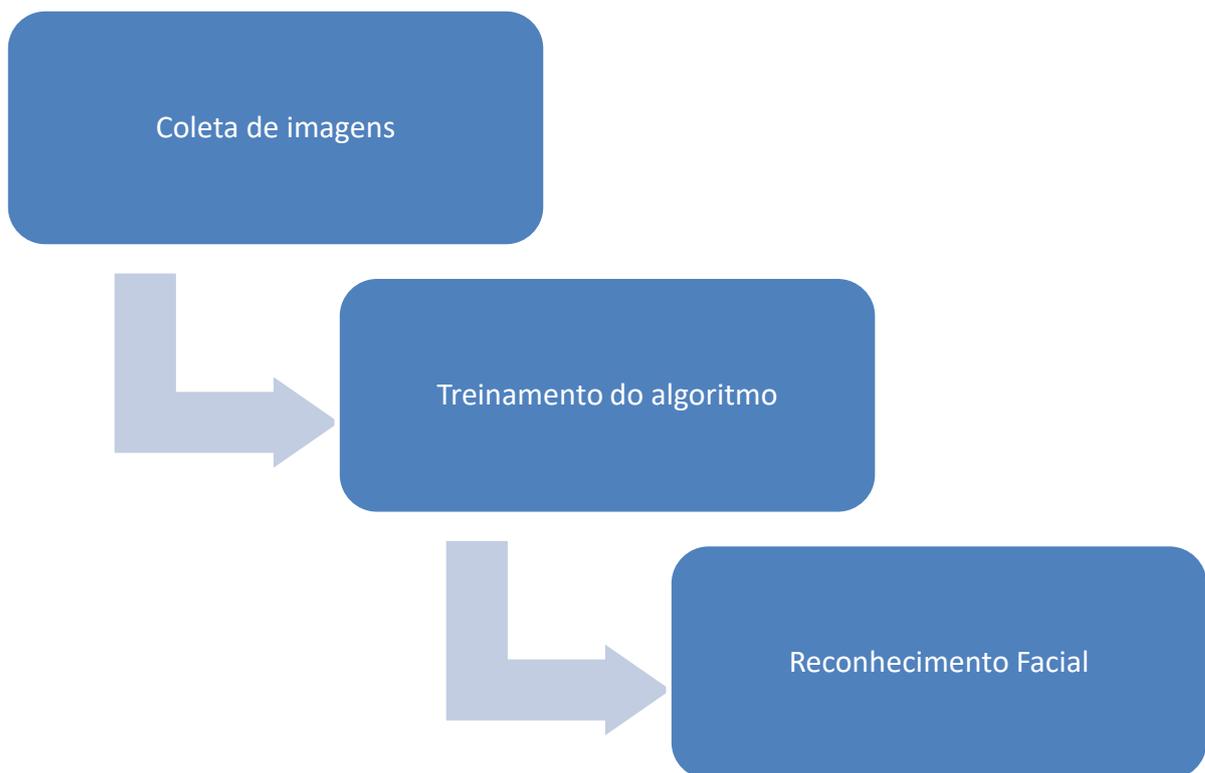
Algo importante a ser analisado em cada classificador é a confiança do reconhecimento. Sabe-se que para o reconhecimento da pessoa que está aparecendo na tela, é necessário comparar dois histogramas e devolver a imagem com o histograma mais próximo. Esse histograma é calculado a partir da distância entre dois histogramas, ou seja, o da imagem cadastrada e o da imagem que está aparecendo na tela. O OpenCv permite também retornar a distância calculada, que pode ser usada como medida de “confiança” ou nível de confiança. Quanto mais próximo de 1 esse nível de confiança for, mais preciso será o reconhecimento.

4 METODOLOGIA

Neste capítulo, abordam-se as fases essenciais para a concretização do projeto proposto. Inicialmente, o foco recai sobre o desenvolvimento do algoritmo, empregando a linguagem de programação Python, com o apoio da biblioteca OpenCV e a integração dos módulos cv2, numpy e os.

Adicionalmente, adota-se o classificador LBPH Face Recognizer create, por apresentar uma maior assertividade em projeto realizado pelo autor anteriormente.

Com o intuito de otimizar a execução do projeto, o trabalho é dividido em etapas distintas, a saber: detecção de faces, coleta de imagens, treinamento do algoritmo, reconhecimento facial e implementação de medidas mitigadoras, visando ao hábito de sucção de dedo e/ou chupeta, conforme o fluxograma abaixo.



4.1 COLETA DAS IMAGENS

Nesta etapa são coletadas imagens faciais, com o objetivo de preparação para o processo de treinamento. Para garantir uma abordagem completa e representativa, é necessária uma coleção que abranja diferentes localizações faciais.

As imagens para o treinamento são fundamentais para um reconhecimento eficiente, como foram se padronizado as expressões e posições. Além disso, é importante que o ambiente esteja iluminado e não tenha nenhuma luz incidindo no rosto, variações na expressão (feliz, triste, com e sem óculos), ângulo (olhando levemente para cima, baixo, esquerda, direita) e com o dedo e/ou chupeta na boca. Para essa implementação, foi utilizado a função `cv2.imwrite()`.

Algoritmo 1: Cadastramento das imagens em escala cinza para realizar o treinamento por meio da webcam

```

1 import cv2
2 import numpy as np
3 import time
4 import os
5
6 print ("-----")
7 -----\n")
8 print ("                               ByeBye Bubu
9 \n")
10 print ("-----")
11 -----\n")
12 print ("                               Escolha as opções abaixo
13 \n")
14 print ("-----")
15 -----\n")
16 print ("[1] CADASTRAMENTO DE FACES")
17 print ("[2] RODAR PROGRAMA")
18 input("Digite a opção desejada:")
19
20
21
22 classificador =
23 cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
24 classificadorOlho = cv2.CascadeClassifier("haarcascade_eye.xml")
25
26 camera = cv2.VideoCapture(0) #Define o objeto de captura de
27 vídeo, nesse caso, webcam do notebook.
28
29 amostra = 1 #Controla a quantidade de fotos que serão tiradas
30
31 numeroAmostras = 75 #Ou seja, serão tiradas 50 fotos
32
33 aux = 0
34
35 aux1 = 1
36
37 aux2 = 2
38

```

```

39 aux3 = 4
40
41 classe = 0
42
43 print ("\n ⚠ Atenção: Nesta etapa são coletadas imagens faciais
44 com o objetivo de preparação para o processo de treinamento. \n")
45
46 nome = input("\n Digite o nome completo da criança:")
47
48 print ("\n")
49 print("Por favor, leia as informações abaixo:\n")
50 print ("1° ambiente esteja iluminado e não tenha nenhuma luz
51 incidindo no rosto;\n")
52 print ("2° Ângulo: Olhando levemente para cima, baixo, esquerda,
53 direita;\n")
54 print ("3° Variações na expressão: Feliz, Triste, Com e sem
55 óculos);\n")
56 print ("Tudo mencionado acima, é necessário que a criança esteja
57 com a chupeta ou o dedo na boca);\n")
58
59
60 time.sleep(5)
61
62 largura, altura = 520 , 520 #Controla o tamanho da foto que será
63 tirada
64
65 print ("***CAPTURANDO AS FACES DA CRIANÇA "+ str(nome) +" SEM
66 NENHUM OBJETO***\n")
67
68 while (True): #Loop Infinito
69     conectado, imagem = camera.read()
70
71     imagemCinza = cv2.cvtColor(imagem, cv2.COLOR_BGR2GRAY)
72 #Conversão na escala Cinza
73
74     FacesDetectadas = classificador.detectMultiScale(imagemCinza,
75
76 scaleFactor=1.05,
77
78 minSize=(50,50))
79     for (x, y, l, a) in FacesDetectadas:#Coloca os retângulos nas
80 faces detectadas
81         cv2.rectangle(imagem, (x,y), (x + l, y + a), (0,0,255),
82 2)
83         regioao = imagem [y:y + altura, x:x + l]
84         regioaoCinzaOlho = cv2.cvtColor(regiao,
85 cv2.COLOR_BGR2GRAY)
86         olhosDetectados =
87 classificadorOlho.detectMultiScale(regiaoCinzaOlho)
88         for (ox, oy, ol, oa) in olhosDetectados:
89             cv2.rectangle(regiao, (ox, oy), (ox + ol, oy + oa),
90 (0,255,0), 2)
91             if (cv2.waitKey(1) & 0xFF == ord('q')): #Fica
92 esperando a tecla q se digitada
93                 if np.average(imagemCinza) > 110:
94                     if (amostra >= 1) and (amostra <= 25):

```

```

95             imagemFace = cv2.resize(imagemCinza
96 [y:y + a, x:x + l],(largura, altura))
97             classe = aux1
98             cv2.imwrite("Fotos/crianca." +
99 str(nome) + "." + str(amostra) + "." + str (classe) + ".jpg",
100 imagemFace)
101             elif (amostra >= 26) and (amostra <= 50):
102                 imagemFace =
103 cv2.resize(imagemCinza[y:y + a, x:x + l], (largura, altura))
104                 classe = aux2
105                 cv2.imwrite("Fotos/crianca." +
106 str(nome) + "." + str(amostra) + "." + str (classe) + ".jpg",
107 imagemFace)
108
109             else:
110                 imagemFace =
111 cv2.resize(imagemCinza[y:y + a, x:x + l], (largura, altura))
112                 classe = aux3
113                 cv2.imwrite("Fotos/crianca." +
114 str(nome) + "." + str(amostra) + "." + str (classe) + ".jpg",
115 imagemFace)
116
117             print ("[Foto " + str(amostra) + " capturada
118 com sucesso]")
119
120             print (amostra)
121             print (classe)
122             amostra += 1
123
124
125         cv2.imshow("Face", imagem)
126
127         cv2.waitKey(1)
128
129         if ((amostra == 26) and (aux == 0)) :
130             print ("\n***CAPTURANDO AS FACES DA CRIANÇA " + str(nome)
131 + " COM A CHUPETA***\n")
132             aux += 1
133         if ((amostra == 51) and (aux == 1)):
134             print ("\n***CAPTURANDO AS FACES DA CRIANÇA " + str(nome)
135 + " COM O DEDO NA BOCA***\n")
136             aux += 1
137         if (amostra >= numeroAmostras + 1 ):
138             break
139
140     print ("\nFaces capturadas com sucesso")
141
142     aux1 = aux1 + 2
143     aux2 = aux2 + 2
144     aux3 = aux3 + 3
145
146     camera.release ()
147     cv2.destroyAllWindows ()
148
149     print ('Deseja retornar ao início? "Sim", "Não"')

```

Inicialmente a captura da imagem é feita através da Webcam, e o formato em que foram salvas é nomedacrianca.nº.da foto.tipo da foto.jpg, conforme expresso no algoritmo

da figura 27, com 220 x 220 pixels em escala cinza. A conversão das imagens para a escala cinza se dá pelo fato de as funções de reconhecimento só permitirem as imagens nesse formato, além de permitir um melhor desempenho para o reconhecimento.

Durante o desenvolvimento do algoritmo, percebeu-se que se fazem necessários os cadastramentos de três tipos de imagens: faces em condições normais, ou seja, sem chupeta e sem dedo na boca, faces com chupeta na boca e faces com o dedo na boca. As faces sem chupeta ou sem dedo na boca são para que o algoritmo entenda a diferença entre as situações. Para as faces sem chupeta foram consideradas fotos tipos 1,3,5,7..., ou seja, números ímpares, já para as fotos com chupeta e dedo na boca, são consideradas fotos do tipo 2,4,6,8..., ou seja, números pares.

4.2 TREINAMENTO

Finalizada a etapa de captura de fotos das crianças, ou seja, alimentado o banco de dados, os dados estão prontos para iniciar o processo de treinamento. Como a biblioteca do OpenCV já disponibiliza alguns classificadores treinados e prontos para serem utilizados, isso facilita bastante essa tarefa. Inicialmente isso é feito com o auxílio do método `getImagemComClasses`, o qual irá percorrer toda a pasta que possui as imagens salvas através da etapa de cadastramento e com isso vai retornar vetores com o número do tipo da foto, ou seja, números ímpares e números pares. Logo em seguida, com a ajuda do pacote `os.path.join`, essas informações serão armazenadas no local em que se pode ter acesso às imagens. Feito isso, é necessário novamente converter em escala de cinza através da função `cv2.cvtColor(imagem, cv2.COLOR_BGR2GRAY)`.

Algoritmo 2: Treinamento

```
1 import cv2
2 import os
3 import numpy as np
4
5 lbph = cv2.face.LBPHFaceRecognizer_create()
6
7 def getImagemComClasses():
8     caminhos = [os.path.join('Fotos', f) for f in
9 os.listdir('Fotos')]
10     #print(caminhos)
```

```

11     faces = []
12     classes = []
13     for caminhoImagem in caminhos:
14         imagemFace = cv2.cvtColor(cv2.imread(caminhoImagem),
15 cv2.COLOR_BGR2GRAY)
16         classe = int (os.path.split(caminhoImagem)[-
17 1].split('.')[3])
18
19     return np.array(classes), faces
20
21 classes, faces = getImagemComClasses()
22 #print (faces)
23
24 print ("Treinando...")

    lbph.train(faces, classes)
    lbph.write('classificadorLBPH.yml')

```

4.3 RECONHECIMENTO DE FACE E AÇÃO MITIGADORA

Realizados todos os procedimentos mencionados anteriormente, com ênfase especial na etapa de treinamento, veem-se que o algoritmo adquiriu características necessárias para o reconhecimento. Em seguida, a próxima etapa é o algoritmo funcionar em tempo real para a identificação de quando a criança está utilizando a chupeta ou está com o dedo na boca. Para isso, foi utilizado o algoritmo abaixo.

Algoritmo 3: Reconhecimento da face

```

1  import cv2
2  import pyautogui
3  import time
4
5  camera = cv2.VideoCapture (0)
6  detectorFace =
7  cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
8  reconhecedor = cv2.face.LBPHFaceRecognizer_create()
9  reconhecedor.read("classificadorLBPH.yml")
10 largura, altura = 220,220
11 font = cv2.FONT_HERSHEY_COMPLEX_SMALL
12 camera = cv2.VideoCapture (0)
13
14 while (True):
15     conectado, imagem = camera.read ()
16     imagemCinza = cv2.cvtColor(imagem, cv2.COLOR_BGR2GRAY)
17     facesDetectadas = detectorFace.detectMultiScale(imagemCinza,
18

```

```

19
20 scaleFactor=1.5,
21
22 minSize=(30,30))
23
24     for (x, y, l, a) in facesDetectadas:#Coloca os retângulos nas
25 faces detectadas
26         imagemFace = cv2.resize(imagemCinza[y:y + a, x:x +
27 l],[largura,altura])
28         cv2.rectangle(imagem, (x,y), (x+l,y+a), (0,0,255),2)
29         classe, confianca = reconhecedor.predict(imagemFace)
30         cv2.putText(imagem,str(classe), (x,y+(a+30)), font, 2,
31 (0,0,255))
32
33     cv2.imshow("Face", imagem)
34     if classe == 4:
35         pyautogui.click(700, 350)
36         time.sleep(5)
37         pyautogui.click(700, 350)
38         time.sleep(0.5)
39         pyautogui.click(700, 350)
39
40     if cv2.waitKey(1) == ord('q'):
41         break
42     time.sleep(5)
43     camera.release()
44     cv2.destroyAllWindows

```

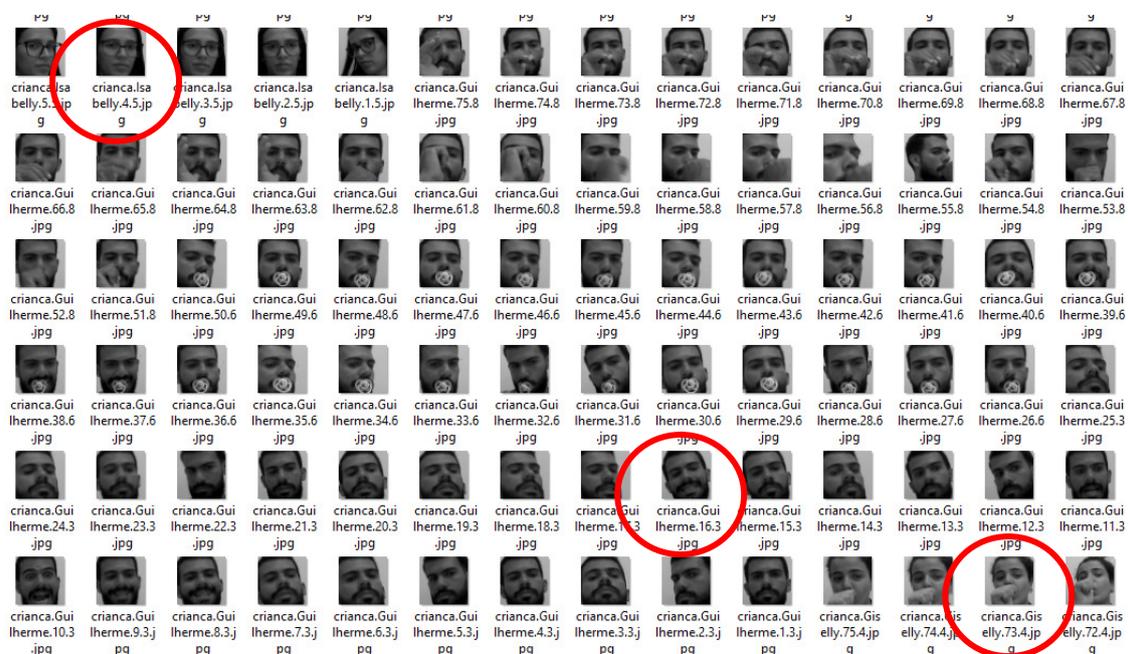
O funcionamento do algoritmo é simples: no momento em que a criança coloca o dedo ou a chupeta na boca, enquanto está assistindo a um vídeo, o mouse apertará o centro da tela com um click. Então se aguarda um intervalo de cinco segundos, durante o qual o vídeo permanece pausado, e após os cinco segundos, o vídeo retorna à execução. Após outros cinco segundos, será feita uma nova análise.

5 RESULTADOS FINAIS

Diante do exposto, os testes realizados pelo desenvolvedor do algoritmo neste estudo permitiram que o projeto atingisse o seu objetivo, ou seja, foi constatado, testado e verificado que quando se coloca o dedo ou a chupeta, o vídeo ao qual o bebê está assistindo durante a análise é pausado.

Nesta seção são apresentados os resultados obtidos quando utilizado o algoritmo. Além disso, serão apresentados dados a respeito da eficiência dele. Para efeito de testes, foram cadastradas 3 (três) pessoas, conforme a figura 22.

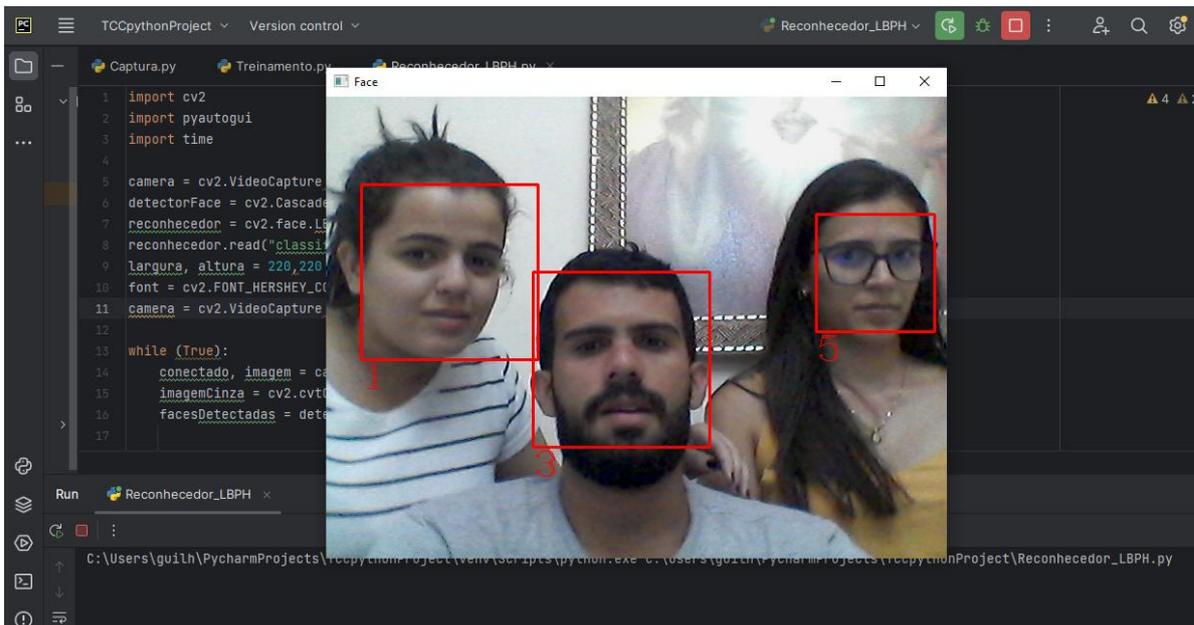
Figura 22 - Faces cadastradas para fase de testes



Fonte: Arquivo Pessoal

Conforme mencionado anteriormente, para faces cadastradas em circunstâncias normais, ou seja, sem a presença de chupetas ou dedos na boca, foram atribuídos números ímpares. Percebe-se que na figura 23 o algoritmo em pleno funcionamento atribui números ímpares para as faces sem a presença de chupetas ou dedos na boca.

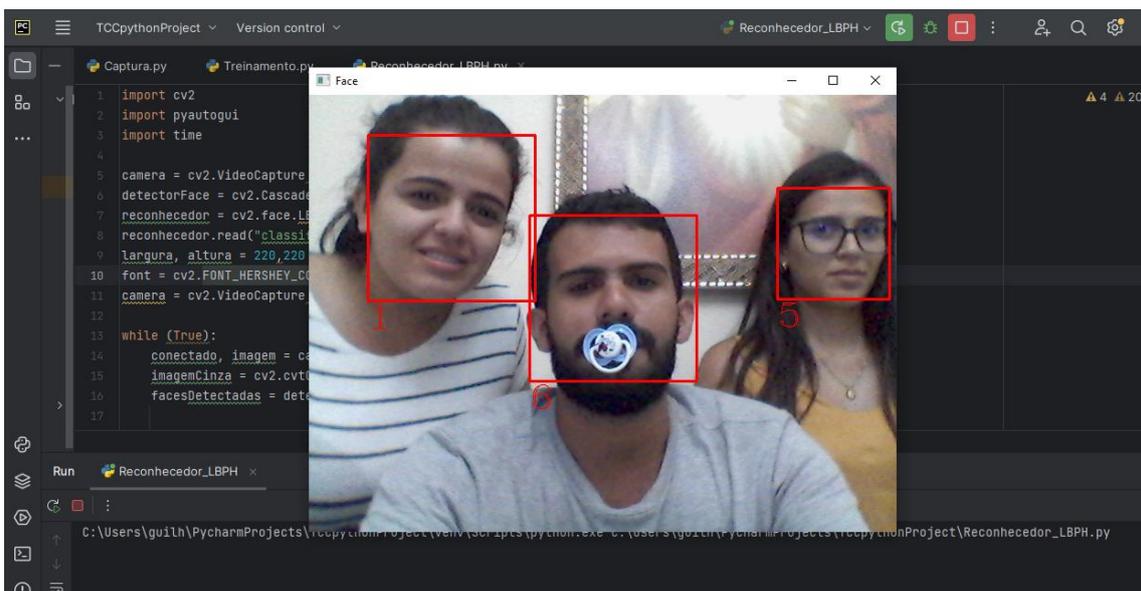
Figura 23 - Teste do algoritmo em pleno funcionamento, constatando que para faces em situações normais, foram atribuídos números ímpares



Fonte: Arquivo Pessoal

Como anteriormente indicado, já para as faces cadastradas em circunstâncias com a presença de chupetas, foram atribuídos números pares. Observa-se que na figura 24, o algoritmo está operando eficientemente.

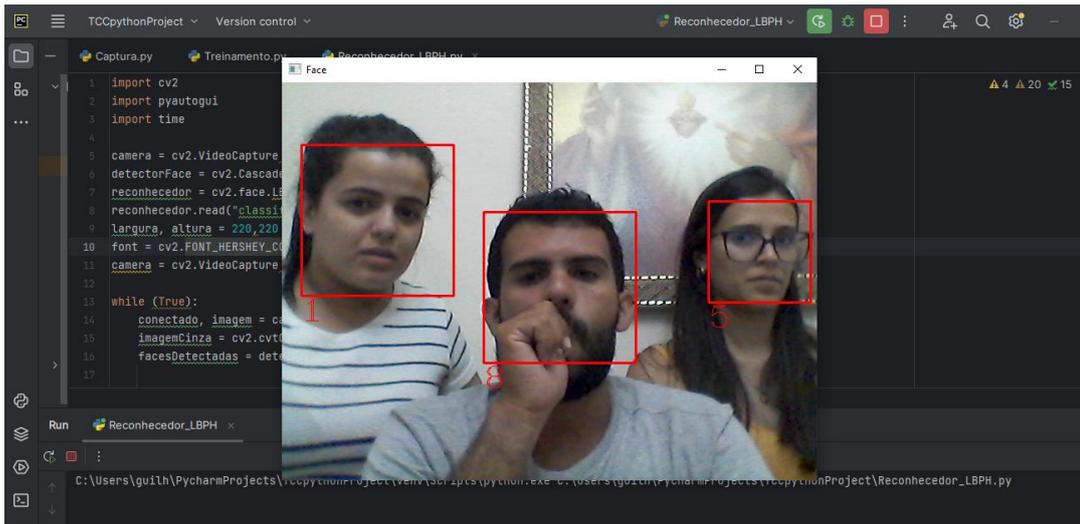
Figura 24 - Teste bem-sucedido do algoritmo em pleno funcionamento, revelando que, em circunstâncias nas quais a pessoa colocar a chupeta, números pares foram associados a faces



Fonte: Arquivo pessoal

E no que diz respeito às faces cadastradas em circunstâncias com a presença do dedo na boca, foram atribuídos números pares. Observa-se que na figura 25, o algoritmo está operando eficientemente.

Figura 25 - Teste efetuado com sucesso no algoritmo em pleno funcionamento, indicando que, quando a pessoa coloca o dedo na boca em circunstâncias específicas, números pares são atribuídos às faces.



Fonte: Arquivo pessoal

6 CONCLUSÃO

O algoritmo desenvolvido para a detecção da sucção de dedo ou chupeta obteve êxito em testes realizados. A capacidade do sistema em pausar automaticamente o vídeo durante tais comportamentos revelou-se uma abordagem eficaz na redução desses hábitos prejudiciais, sendo assim uma ferramenta eficaz para diminuir o uso de dedo ou chupeta, que causa grandes problemas ortodônticos.

Aprimoramentos adicionais são necessários no algoritmo, especialmente em aspectos específicos, por exemplo, sensibilidade à iluminação, capacidade de geração de relatório de cada criança e a expansão do sistema para plataformas Android e IOS e até mesmo a implementação através de um sistema embarcad. Porém, o algoritmo junto a sua implementação se mostrou eficaz, representando um avanço promissor na abordagem de problemas ortodônticos decorrentes do hábito de sucção, proporcionando um impacto positivo na saúde bucal das crianças.

REFERÊNCIAS

- ABNT. (2002). **NBR 10520 - Informação e documentação - Citações em documentos - Apresentação**. Associação Brasileira de Normas Técnicas (p. 7).
- ABNT. (2002). **NBR 6023 - Informação e documentação - Referências - Elaboração**. Associação Brasileira de Normas Técnicas (p. 24).
- ABNT. (2003). **NBR 6028 - Informação e documentação - Resumo - Apresentação. (p. 2)**. Associação Brasileira de Normas Técnicas.
- ABNT. (2005). **NBR 6034 - Informação e documentação - Índice - Apresentação**. Associação Brasileira de Normas Técnicas (p. 4).
- ABNT. (2011). **NBR 14724 - Informação e documentação — Trabalhos acadêmicos — Apresentação**. Associação Brasileira de Normas Técnicas (p. 11).
- BATAGLION, César. **Critérios da Oclusão Funcional Ideal**. Faculdade de Odontologia de Ribeirão Preto - USP Departamento de Odontologia Restauradora Disciplina: Disfunção Temporomandibular e Dor Orofacial. Disponível em: < https://edisciplinas.usp.br/pluginfile.php/5404958/mod_resource/content/1/Slides%20aula%20Crit%C3%A9rios%20da%20Oclus%C3%A3o%20Ideal%202020%20gradua%C3%A7%C3%A3o.pdf >. Acessado em 06 de dez de 2023.
- BOCK, A. M. B.; FURTADO, O.; TEIXEIRA, M. L. T. **Psicologias: uma introdução ao estudo de Psicologia**. 13 ed. São Paulo: Saraiva, 2008.
- BOECK, Eloisa Marcantonio; PIZZOL, Karina Eiras Dela Coleta; BARBOSA, Eduarda Geralda Pinheiro; PIRES, Naiara Cristina de Almeida; LUNARDI, Nádia. **Prevalência de má oclusão em crianças de 3 a 6 anos portadoras de hábito de sucção de dedo e/ou chupeta**. Departamento de Ortodontia, Faculdade de Odontologia, UNIARA – Centro Universitário de Araraquara, 14807-120 Araraquara - SP, Brasil.
- BUENO, George. **Mordida cruzada posterior: o que é e como corrigir**. Disponível em: < <https://blog.dentalcremer.com.br/mordida-cruzada-posterior-o-que-e-e-como-corrigir/> >. Acesso em: 28 de dez. de 2023.
- CALISTI, LJP; COHEN, MM; FALES, MH. **Correlation between malocclusion, oral habits, and socio-economic level of preschool children**. J Dent Res 1960; 39(3):450-4.
- CUNHA, Sandra Regina Thomasinho da; CORRÊA, Maria Salete Nahás Pires; OLIVEIRA, Patricia Montanari Leber C; SCHALKA, Mariangela Milena Santos. **Hábitos bucais. In: Corrêa MSNP. Odontopediatria na primeira infância**. 1ª ed. São Paulo: Santos; 1998. p.561-76.
- FIGUEIREDO, Lanna Lidia Monteiro; VERÍSSIMO, Matheus Harllen Gonçalves; LIMA, Ramon Rodrigues de. **Estudo da Maloclusão Advinda do Uso de Chupetas**

KAULFUSS, Marco Aurélio. **BEHAVIORISMO: CONCEITOS E PRECONCEITOS**. Disponível em: <file:///C:/Users/guilh/OneDrive/%C3%81rea%20de%20Trabalho/TCC/QTQkycuoQSwB6PR_2017-1-21-10-43-24.pdf >. Acessado em 16 de jan de 2024.

Ortodônticas: uma Revisão Integrativa da Literatura. Disponível em: <file:///C:/Users/vivia/Downloads/Archi+v11+n4+-+25+-+2022+-+Original+25.pdf>. Acessado em 19 de dez de 2023.

MESOMO, C; LOSSO, EM. **Avaliação dos efeitos do uso prolongado de chupetas convencionais e ortodônticas sobre a dentição decídua**. Rev Ibero-am Odontopediatr Odontol Bebê 2004; 7(38):360-4.

O'BRIEN, H.; LACHAPELLE, D.; GAGNON, P.; LAROCQUE, I; ROBERT; L.; **Nutritive and nonnutritive sucking habits: a review**. ASDC J Dent Child 1996; 63(5):321-7.

ODONTOYAMA. **Você sabe se você tem mordida aberta?** Disponível em: <https://odontoyama.com.br/mordida-aberta/>. Acesso em: 24 de dez. de 2023.

Organização Mundial da Saúde. **Levantamento epidemiológico básico de saúde bucal: manual de instruções**. 3ª ed. São Paulo: Santos; 1991.

PASTOR, IMO; FRANCO, FCM; LEITE, K. **O uso da chupeta – implicações no desenvolvimento infantil**. Rev Fac Odonto UFBA 2000; 20:82-7.

RASCHKA, Sebastian; PATTERSON, Joshua; and NOLET, Corey. **Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence**. Disponível em: <https://arxiv.org/abs/2002.04803>. Acessado em: 20 de dez de 2023.

SILVA, Jéssica. **Oclusão**. Disponível em: <https://jssodonto.wordpress.com/2013/04/09/occlusao/>. Acessado em: 17 de dez de 2023.

SULIANO, André Alencar et al. **Prevalência de maloclusão e sua associação com alterações funcionais do sistema estomatognático entre escolares**. Cadernos de Saúde Pública, v. 23, n. 8, p. 1913-1923, 2007.

SVEDMYR, B.; SUCKING, Dummy. **A study of its prevalence, duration and malocclusion consequences**. Swed Dent J 1979; 3:205-10.

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus João Pessoa
	Av. Primeiro de Maio, 720, Jaguaribe, CEP 58015-435, Joao Pessoa (PB)
	CNPJ: 10.783.898/0002-56 - Telefone: (83) 3612.1200

Documento Digitalizado Ostensivo (Público)

TCC - Guilherme Maia

Assunto:	TCC - Guilherme Maia
Assinado por:	Guilherme Leite
Tipo do Documento:	Projeto
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Guilherme Leite Maia, ALUNO (20172610023) DE BACHARELADO EM ENGENHARIA ELÉTRICA - JOÃO PESSOA**, em 26/02/2024 11:22:28.

Este documento foi armazenado no SUAP em 26/02/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1093194

Código de Autenticação: a829bb8d8b

