

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

JEAN HENRIQUE FERREIRA

**DETECÇÃO DE PESSOAS COM FEBRE POR TERMOGRAFIA
INFRAVERMELHA**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2016

JEAN HENRIQUE FERREIRA

**DETECÇÃO DE PESSOAS COM FEBRE POR TERMOGRAFIA
INFRAVERMELHA**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Ionildo José Sanches

PONTA GROSSA

2016



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa

Departamento Acadêmico de Informática
Bacharelado em Ciência da Computação



TERMO DE APROVAÇÃO

DETECÇÃO DE PESSOAS COM FEBRE POR TERMOGRAFIA INFRAVERMELHA

por

JEAN HENRIQUE FERREIRA

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 10 de novembro de 2016 como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Ionildo José Sanches
Prof. Orientador

André Pinz Borges
Membro titular

Erikson Freitas de Moraes
Membro titular

Prof. Dr. Augusto Foronda
Responsável pelo Trabalho de Conclusão
de Curso

Prof. Dr. Erikson Freitas de Moraes
Coordenador do curso

Dedico este trabalho a minha mãe, Taís Maria Ferreira, que sempre me apoiou e fez de tudo para que eu tivesse os recursos necessários para cumprir essa jornada.

AGRADECIMENTOS

Agradeço a todos que de alguma forma me ajudaram a desenvolver esse trabalho, seja com auxílio técnico, auxílio teórico ou disponibilização de materiais e recursos. Sou grato, também, a aqueles que me ajudaram com o incentivo, paciência e humildade durante toda essa trajetória de minha vida, sejam essas pessoas familiares, amigos, pessoas próximas ou colegas de trabalho. Faço um agradecimento especial a minha namorada Luana Carvalho Luiz, que ficou durante todo curso ao meu lado, nos bons e maus momentos, sempre me ajudando, ouvindo e apoiando.

Agradeço a todos os professores que com suas sábias palavras me fizeram adquirir o conhecimento e me instigaram a sempre querer ter maiores e melhores conquistas. Senti-me honrado ao trabalhar juntamente com o Prof. Dr. Ionildo José Sanches e o Prof. Dr. Erikson Freitas de Moraes que com suas experiências me deram lições de vida que levarei por toda vida pessoal e profissional. Também agradeço ao Prof. Dr. Ednei Felix Reis, que sempre acreditou no meu potencial e mostrou-me que em meio as dificuldades todo esforço será recompensado.

A todos os familiares, professores, amigos e colegas de trabalho que não tenham tido seus nomes citados, saibam que para cada momento da minha vida, um de vocês será lembrado e os conselhos ouvidos serão seguidos.

RESUMO

FERREIRA, Jean H. **Detecção de Pessoas com Febre por Termografia Infravermelha**. 2016. 63f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2016.

A identificação de padrões e de faces em imagens são alguns ramos da Visão Computacional e podem servir como uma ferramenta para automatizar um processo que demanda trabalho manual de um observador. Dentre os tipos de imagens existentes, a imagem infravermelha ou imagem termográfica é utilizada em áreas da medicina e engenharia. Na medicina, as imagens infravermelhas podem ser utilizadas para detectar áreas da anatomia humana com situação irregular. Com este trabalho, mostra-se uma ferramenta que irá localizar e classificar o valor de máxima temperatura de uma face em uma imagem infravermelha. Para isso, realizou-se um treinamento para aprendizagem de máquina utilizando o algoritmo *AdaBoost*, com amostras de faces em imagens infravermelhas, e na detecção facial foi utilizado o algoritmo de Viola e Jones. O funcionamento da ferramenta deu-se pela detecção da face do indivíduo seguida pela localização e classificação do valor de maior temperatura da face. Foram realizadas comparações com os resultados de detecções de obtidos com diferentes treinamentos desenvolvidos. Com estes dados, percebeu-se que classificadores mais rígidos, descartam maior quantidade de áreas da imagem, enquanto classificadores menos rigorosos resultam em maior quantidade de falsos positivos. Os resultados finais do desenvolvimento trouxeram uma metodologia de detecção de febre em indivíduos sem necessidade de contato direto, apresentando pouca diferença entre a temperatura axial e a temperatura na imagem infravermelha. Também conseguiu-se criar um arquivo de detecção de faces em imagens infravermelhas, que detectou corretamente 79,51% das faces nas imagens fornecidas.

Palavras-chave: Imagens Infravermelhas. Detecção Facial. Febre. Comissura Medial Palpebral.

ABSTRACT

FERREIRA, Jean H. **Detection of People with Fever by Infrared Thermography**. 2016. 63f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2016.

The identification of patterns and faces in images are some branches of Computer Vision and can serve as a tool to automate a process that demands the manual work of an observer. Among the types of existing images, the infrared image or thermographic image is used in areas of medicine and engineering. In medicine, infrared images can be used to detect areas of the human anatomy with irregular situation. With this work, it is shown a tool that will locate and classify the maximum temperature value of the face in an infrared image. In order to do this, a machine learning training was performed using the AdaBoost algorithm, with face samples in infrared images, and in the face detection the Viola and Jones algorithm was used. The operation of the tool was by the detection of the face of the individual followed by the location and classification of the value of the higher temperature of the face. Comparisons were made with the results of detections obtained with different training sessions. With this data, it was noticed that more rigid classifiers discard more areas of the image, while less rigorous classifiers result in a greater amount of false positives. The final results of the development brought a methodology of detection of fever in individuals without need of direct contact, showing little difference between the axial temperature and the temperature in the infrared image. It was also possible to create a face detection file in infrared images, which correctly detected 79.51% of the faces in the images provided.

Keywords: Infrared Images. Face Detection. Fever. Medial Palpebral Commissure.

LISTA DE FIGURAS

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1 - Imagem infravermelha de um voluntário | 12 |
| Figura 2 - Imagens digitais e infravermelhas em situações distintas de iluminação. (a) Imagem digital em ambiente escuro, (b) imagem infravermelha em ambiente escuro, (c) imagem digital em ambiente iluminado, (d) imagem infravermelha em ambiente iluminado..... | 13 |
| Figura 3 - Espectro eletromagnético | 14 |
| Figura 4 - Exemplo de mapeamento de imagem digital | 15 |
| Figura 5 - Utilização de histograma para escolha de limiar | 17 |
| Figura 6 - Exemplo de histograma para escolha de limiares. Sendo (a) a escolha de um limiar e (b) dois limiares..... | 18 |
| Figura 7 - Resultado da aplicação de pseudo cores usando a paleta de cores <i>rainbow</i> . (a) Imagem original, (b) imagem resultado | 19 |
| Figura 8 – Exemplos de <i>features</i> para classificação de áreas | 21 |
| Figura 9 - Modelo de rejeição em cascata do algoritmo de Viola e Jones | 22 |
| Figura 10 - Resultado da aplicação do algoritmo de Viola e Jones em uma imagem digital..... | 22 |
| Figura 11 - Modelo de informação do arquivo <i>negatives.info</i> | 25 |
| Figura 12 - Modelo de informação do arquivo <i>positives.info</i> | 25 |
| Figura 13 - Criança com febre (a) e criança sem febre (b)..... | 29 |
| Figura 14 - Comissura medial palpebral (CMP) e comissura lateral palpebral (CLP)..... | 30 |
| Figura 15 - Demonstrativo de aquisição de variáveis de testes | 31 |
| Figura 16 - Ferramenta de geração de imagens positivas e negativas | 35 |
| Figura 17 - Diagrama do treinamento realizado | 36 |
| Figura 18 – Exemplos de resultado da ferramenta com a identificação da CMP | 39 |
| Figura 19 - Comparativo de resultado entre TermaCAM Research a esquerda e a ferramenta desenvolvida a direita | 45 |
| Figura 20 - Resultados com diferentes estágios de treinamento. (a) detecção após 5 estágios e (b) após 15 estágios de treinamento..... | 47 |

LISTA DE QUADROS

| | |
|---------------------------------------------------------------------------------|----|
| Quadro 1 - Funções <code>detectMultiscale</code> da biblioteca OpenCV 3.0 | 24 |
| Quadro 2 - Exemplo do arquivo <code>negatives.info</code> | 25 |
| Quadro 3 - Exemplo do arquivo <code>positives.info</code> | 26 |
| Quadro 4 - Parâmetros da função <code>createsamples</code> | 36 |
| Quadro 5 - Parâmetros da função <code>haatraining</code> | 37 |

LISTA DE TABELAS

| | |
|----------------------------------------------------------------------|----|
| Tabela 1 - Dados dos treinamentos realizados | 40 |
| Tabela 2 - Análise do treinamento..... | 41 |
| Tabela 3 - Dados coletados de temperatura | 42 |
| Tabela 4 - Valores de resultados da ferramenta | 43 |
| Tabela 5 - Valores de sensibilidade, especificidade e acurácia | 44 |
| Tabela 6 - Valores de resposta da ferramenta em indivíduos | 45 |
| Tabela 7 - Comparativo com detecções e resultados corretos..... | 46 |
| Tabela 8 - Comparativo com detecções e resultados incorretos..... | 47 |

LISTA DE SIGLAS

| | |
|------|--------------------------------------------------------|
| FPF | <i>FLIR Public Format</i> |
| FPS | <i>Frames Per Second</i> |
| CLP | Comissura Lateral Palpebral |
| CMP | Comissura Medial Palpebral |
| DAB | <i>Discrete AdaBoost</i> |
| GAB | <i>Gentle AdaBoost Discrete AdaBoost Real AdaBoost</i> |
| HAAR | <i>HAAR-like features</i> |
| LB | <i>Logit Boost</i> |
| LBP | <i>Local Binary Patterns</i> |
| LUT | <i>Look-Up Table</i> |
| MAT | <i>MatLab</i> |
| OS | <i>Operating System</i> |
| RAB | <i>Real AdaBoost</i> |
| RGB | <i>Red, Green, Blue</i> |
| SARS | Síndrome Respiratória Aguda Grave |
| TXT | <i>TeXT</i> |
| XML | <i>eXtensible Markup Language</i> |

SUMÁRIO

| | |
|----------------------------------------------------------------------------|-----------|
| 1 INTRODUÇÃO | 6 |
| 1.1 OBJETIVOS | 8 |
| 1.1.1 Objetivo Geral | 8 |
| 1.1.2 Objetivos Específicos | 8 |
| 1.2 JUSTIFICATIVA | 9 |
| 1.3 TRABALHOS RELACIONADOS | 10 |
| 1.4 ESTRUTURA DO TRABALHO | 11 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 12 |
| 2.1 IMAGENS INFRAVERMELHAS | 12 |
| 2.1.1 Radiação Térmica | 14 |
| 2.2 PROCESSAMENTO DIGITAL DE IMAGENS | 15 |
| 2.2.1 Segmentação por Limiarização | 16 |
| 2.2.2 Processamento de Imagens em Pseudo-Cores | 18 |
| 2.3 BIBLIOTECA <i>OPENCV</i> | 19 |
| 2.3.1 Algoritmo de Reconhecimento Facial de Viola e Jones | 20 |
| 2.3.2 <i>AdaBoost Haartraining</i> e Geração de classificador em XML | 24 |
| 2.4 VARIAÇÃO DE TEMPERATURA DA FACE HUMANA | 28 |
| 2.5 CÁLCULO DE ESPECIFICIDADE, SENSIBILIDADE E ACURÁCIA | 30 |
| 3 METODOLOGIA | 32 |
| 3.1 COLETA E TRATAMENTO DE IMAGENS | 32 |
| 3.2 ARQUIVO DE IMAGEM FPF | 33 |
| 3.3 TREINAMENTO | 34 |
| 3.4 USO DA FERRAMENTA | 37 |
| 3.5 COMENTÁRIOS | 39 |
| 4 RESULTADOS | 40 |
| 4.1 ANÁLISE DO TREINAMENTO | 40 |
| 4.2 ANÁLISE DE RESULTADOS | 42 |
| 5 CONCLUSÃO | 49 |
| 5.1 TRABALHOS FUTUROS | 50 |
| REFERÊNCIAS | 51 |
| APENDICE A | 54 |

1 INTRODUÇÃO

O termo infravermelho tem origem latina e tem como significado a região abaixo do vermelho no espectro eletromagnético. Uma imagem infravermelha é uma quantificação de energia térmica emitida por um objeto captado por uma câmera com detector de radiação infravermelha. O uso de imagens infravermelhas possui as vantagens de não precisar de iluminação e contato com o corpo para sua aquisição, basta apenas uma câmera infravermelha apontada para um corpo que emita calor (BEZERRA, 2010).

A termografia é uma representação não-invasiva que converte em tempo real a energia térmica emitida pela superfície do corpo em impulsos elétricos, para serem visualizados na forma de uma imagem. As imagens infravermelhas representam em níveis de cinza ou em pseudo-cores as faixas de calor que um corpo emitiu. Nessas faixas geralmente pontos mais claros representam temperaturas maiores, e pontos mais escuros representam temperaturas menores (SANCHES, 2009).

As aplicações das imagens infravermelhas podem abranger áreas como medicina, monitoramento do meio ambiente, astronomia, monitoramento na indústria, sistemas de defesa, dentre outras áreas. Nos sistemas de defesa, tal aplicação tem importância relevante, pois as operações de detecção e reconhecimento tem maior eficácia, devido à ausência de emissão de onda eletromagnética (detectar sem ser notado) (NEVES, 2003). A medicina é uma das áreas que utiliza imagens infravermelhas, pois as variações de temperatura em um corpo podem servir para detecção de câncer, febres reumáticas, problemas vasculares e outros tipos de doença (SANCHES, 2009).

A temperatura do corpo humano é controlada por uma área do cérebro denominada hipotálamo. Essa região é responsável por manter os órgãos internos em temperaturas de aproximadamente 37°C. A medição de temperatura de um corpo humano saudável, nos locais de maior quantidade de calor, deve estar entre 36°C e 37°C. Temperaturas maiores que 37°C e menores que 37,8°C indicam que é um estado de febrícula e temperaturas maiores ou iguais a 37,8°C indicam um estado de febre. A febre é um indicativo de anomalia no corpo que pode ser provocado por doenças causadas por vírus, fungos ou bactérias (VARELLA, 2012).

“A febre é uma elevação de temperatura ocasionada na maioria das vezes em virtude da presença de agentes patógenos no organismo, como por exemplo, as

bactérias, sendo estes pirógenos exógenos” (CARVALHO, 2002). A febre é um mecanismo de auxílio a defesa imunológica do corpo que pode causar perturbações no corpo de um patógeno, auxiliando no combate à doença. Porém essa elevação de temperatura só é benéfica ao indivíduo quando não é muito alta, pois em excesso, pode causar convulsões no indivíduo (CARVALHO, 2002).

Nos últimos anos, com a ocorrência de possíveis pandemias, como a influenza H1N1, SARS (Síndrome Respiratória Aguda Grave) e o vírus Ebola, alguns países tomaram como medida preventiva a triagem de pessoas em locais de grande movimentação pública, como por exemplo, aeroportos (BONNET *et al.* 2008). Limitar a exposição de pessoas que apresentam estado febril tem sido um critério efetivo na prevenção de transmissão de agentes contagiosos. Pois devido à baixa imunidade natural do ser humano, doenças virais graves podem se espalhar com facilidade. O local do corpo humano escolhido para realizar a medição da temperatura foram os olhos, pois é o local de máxima temperatura registrada (RING *et al.*, 2013).

Segundo Haddad (2014), a região da face possui pontos termo anatômicos, isto é, a face humana possui regiões de pontos comuns com quantidade de calor igual ou semelhante comparados com outros corpos. Um desses pontos é o da comissura medial palpebral (CMP), região que localiza-se no extremo das pálpebras próxima ao nariz, e este ponto possui como características ser o local de maior temperatura frontal e ser o ponto mais termo anatômico da face.

Um algoritmo eficaz de reconhecimento facial é o algoritmo desenvolvido por Viola e Jones (2001). Neste algoritmo, consegue-se processar uma grande quantidade de dados de maneira rápida e precisa, pois, é um método desenvolvido a partir de um aprendizado de máquina e aprimorado com técnicas de descarte de áreas desnecessárias. O algoritmo de Viola e Jones funciona em três etapas, a primeira dá-se o nome de Imagem Integral, que trata-se de uma representação intermediária em forma de retângulo onde através de sucessivas somas obtém-se as características da imagem. A segunda é o treinamento baseado em *AdaBoost*, que seleciona as principais características da imagem. A terceira é a combinação de classificadores em cascata que são utilizados para o descarte de áreas desnecessárias (VIOLA e JONES, 2001).

Neste trabalho, foi desenvolvida uma ferramenta que realiza a detecção da região da face e identifica o valor de maior temperatura dessa região. Depois dos valores coletados, é realizada uma classificação da temperatura como sendo normal,

febrícula ou febre. No desenvolvimento da ferramenta foi aplicado o treinamento de máquina utilizando imagens infravermelhas de faces, e para a detecção facial foi utilizando o algoritmo de Viola e Jones implementado na biblioteca *OpenCV*. Com isso, a ferramenta apresentou uma menor taxa de erro ao apresentar os resultados da classificação de temperatura em imagens que não continham indivíduos.

1.1 OBJETIVOS

Nesta sessão serão apresentados os objetivos geral e específicos, e os mesmos serão utilizados como guias no desenvolvimento do trabalho como um todo.

1.1.1 Objetivo Geral

Este trabalho tem como objetivo geral a classificação de indivíduos com febre a partir do reconhecimento facial e a captação do valor de máxima temperatura de pontos termoanatómicos da face.

1.1.2 Objetivos Específicos

Este trabalho tem como objetivos específicos:

- Adquirir, visualizar, tratar, armazenar e exportar imagens infravermelhas;
- Realizar treinamento de máquina com o algoritmo *AdaBoost* utilizando as imagens infravermelhas;
- Verificar o desempenho do treinamento analisando os percentuais de acertos e erros das detecções;
- Detectar faces em imagens infravermelhas utilizando o algoritmo de reconhecimento facial de Viola e Jones;
- Identificar o valor de maior temperatura da região detectada;
- Analisar e classificar o valor de máxima temperatura, para identificação de estado febril.

1.2 JUSTIFICATIVA

A temperatura do corpo humano em condições normais tende a ficar próxima dos 37°C, porém quando o corpo é agredido por um agente externo a temperatura pode aumentar em até 3 graus, caracterizando a febre. A febre é um indicativo de alerta de uma patologia que precisa de tratamento imediato, podendo ser um sinal de doença ou infecção causada por fungos, vírus ou outros parasitas (VARELLA, 2012).

Em locais de grande circulação de pessoas, como locais de entrada e saída de países, doenças contagiosas como influenzas, e infecções virais podem causar uma vasta proliferação desses agentes contagiosos. Por isso, quando há probabilidade de epidemia ou pandemia, alguns países tomam como medida preventiva a análise da temperatura na triagem das pessoas em locais de entrada e saída do país (RING *et al.*, 2015).

Uma das aplicações que vem sendo estudada na área de visão computacional é a detecção facial, que pode ser utilizada para a verificação de existência de faces, ou a biometria de uma determinada pessoa. Em paralelo a estes estudos, está a localização de padrões em imagens, onde, a partir de uma determinada imagem, pode-se localizar uma forma ou um objeto. A detecção de padrões pode ser aplicada em ferramentas de localização de doenças, automóveis, placas de trânsito ou reconhecimento de pessoas. Ao obter a região de interesse pode-se aplicar técnicas ou realizar classificações para tomadas de decisões em um meio externo.

As imagens infravermelhas estão sendo utilizadas na medicina, onde, com a aplicação de técnicas de reconhecimento de padrões pode-se localizar anormalidades na superfície da pele e uma possível detecção de doenças como o câncer. Logo, o reconhecimento facial em imagens infravermelhas também é uma técnica possível e aplicável. Esse trabalho terá como foco de estudo a aplicação da detecção facial e a localização de pontos termoanatomicos da face a partir da temperatura máxima da região detectada.

Contudo, uma ferramenta que detecta uma face em uma imagem infravermelha pode aumentar a precisão de localização dos pontos termoanatómicos de valor máximo, sendo estes, os melhores para a coleta de temperatura. Também, a detecção facial ajuda a garantir que o processo de localização de febre será realizado em uma região da imagem onde existe uma face. A utilização dos valores de temperatura máxima da face pode-se aferir, automaticamente, um valor próximo da temperatura

real do indivíduo, realizando de forma autônoma o processo de triagem de pessoas sem necessidade de contato físico.

1.3 TRABALHOS RELACIONADOS

Existem na bibliografia trabalhos relacionados à reconhecimento facial utilizando imagens infravermelhas e trabalhos que utilizam as imagens infravermelhas para detecção de febre e doenças.

Bezerra (2010) apresenta uma ferramenta de biometria facial utilizando imagens infravermelhas. Em seu trabalho, é apresentado uma e técnica que pode ser utilizada na identificação facial em imagens infravermelhas, explanando em gráficos e números as vantagens e desvantagens de diferentes tipos de classificadores.

Ring *et al.* (2008) e Ring *et al.* (2013) descrevem um guia de identificação de febre em crianças polonesas utilizando imagens infravermelhas. Com esse trabalho foi detalhando existências de diferenças significativas nas imagens de pacientes com temperaturas normais e imagens de pacientes em estado febril. Para isso, os autores realizaram uma coleta quantitativa de imagens de crianças saudáveis e crianças com doenças que tinham a febre como sintoma.

Ring (1998) e Varela (2012) escreveram seus trabalhos baseando-se na temperatura do corpo humano, descrevendo as variações de temperatura e os valores ideais para o corpo humano. Há também, uma classificação dessas temperaturas. Onde, de acordo com os valores coletados pode-se saber se uma pessoa está com febre ou não.

Carvalho (2002) apresenta o comportamento do organismo humano quando em febre. Demonstra as fases da febre e o comportamento das regiões do corpo humano quando neste estado.

Haddad (2014) realiza estudos sobre a termoanatomia da face utilizando imagens infravermelhas, demonstrando de forma quantitativa e qualitativa a presença de pontos confiáveis de termoanatomia em faces distintas.

Bonnet *et al.* (2008) descreveram em seus resultados que as utilizações de termômetros termográficos não trouxeram resultados confiáveis na detecção de febre em pessoas.

Viola e Jones (2001) desenvolveram uma técnica, aprimorada por eles mesmos em 2004, de reconhecimento facial que a partir de um aprendizado de máquina consegue detectar faces em vídeos de até 15 *frames* por segundo em tempo real.

Sanches (2009) desenvolveu uma metodologia para sobreposição de imagens de termografia infravermelha e ressonância magnética, demonstrando aplicações do uso de imagens infravermelhas na área médica. Para tal é descrito a aplicação de técnicas de processamento de imagens para o tratamento de imagens infravermelhas.

Usando como base principal a bibliografia citada, busca-se utilizar os métodos de processamento de imagens, visão computacional, termografia, termoanatomia e temperatura corporal para desenvolver uma ferramenta que agregue esses conceitos. Contudo, o diferencial deste trabalho será integrar o reconhecimento facial com a detecção e a classificação da temperatura de pontos específicos da face, para a classificação automática das temperaturas da face.

1.4 ESTRUTURA DO TRABALHO

Este trabalho organiza-se em cinco capítulos. O Capítulo 2 aborda a revisão bibliográfica sobre imagens infravermelhas com suas ciências e usos, processamento digital de imagens, a aplicação da biblioteca *OpenCV* (*Open Computer Vision*) com o algoritmo de Viola e Jones na detecção facial, e a variação da temperatura da face humana. No Capítulo 3 são apresentados a descrição da metodologia que foi aplicada no desenvolvimento do trabalho. O Capítulo 4 contém os resultados e as análises realizadas com os dados obtidos da realização do treinamento e da execução da ferramenta desenvolvida. No Capítulo 5 são listadas as conclusões obtidas com a realização do trabalho e algumas sugestões de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será apresentada uma descrição sobre imagens infravermelhas e sua utilização em pessoas para análise de temperatura. A seguir, serão apresentadas algumas técnicas de processamento digital de imagens e sua utilização em imagens infravermelhas. Por fim, será realizada uma descrição da biblioteca *OpenCV* e o algoritmo de reconhecimento facial proposto por Viola e Jones (2001) e a apresentação dos assuntos relacionados à febre e a variação de temperatura da face humana.

2.1 IMAGENS INFRAVERMELHAS

A imagem infravermelha, representada na Figura 1, é o registro da distribuição da radiação, em forma de temperatura, emitida pela superfície de um corpo. As imagens infravermelhas utilizam uma técnica não invasiva, ou seja, não necessita de contato direto para ser gerada. A geração de uma imagem infravermelha dá-se pela conversão da radiação emitida pela superfície de um corpo em pulsos elétricos que possam ser visualizados através de uma imagem em escala de cinza ou em pseudo-cores (SANCHES *et al.*, 2013).



Fonte: Autoria própria.

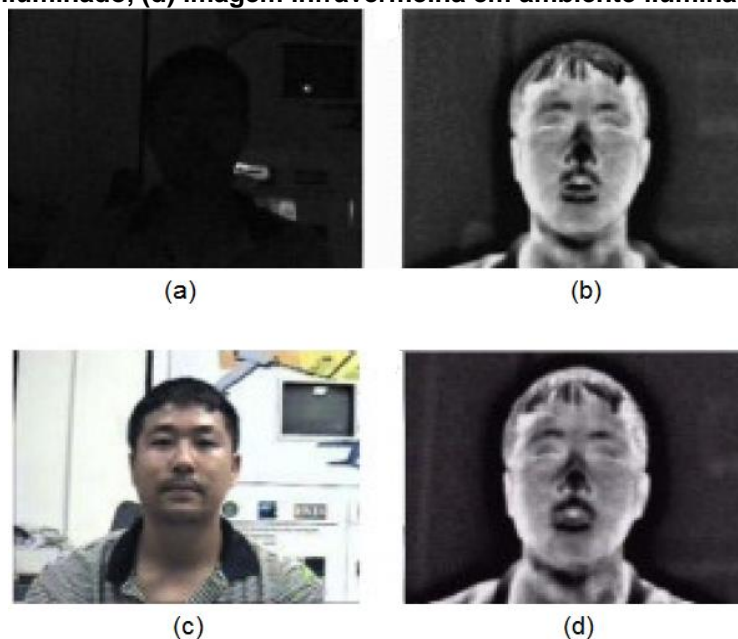
Ao visualizar uma imagem infravermelha, o observador não irá ver o mapeamento da radiação de luz visível de um corpo, pois a imagem infravermelha

apresenta a distribuição de pontos de temperatura do corpo. A análise de uma imagem infravermelha mostrará o mapeamento de diferenças de temperaturas de uma região em relação a outra. Em níveis de cinza, geralmente os pontos mais claros representam níveis de maior quantidade de calor, enquanto pontos escuros representam pontos de menor calor. Já as imagens infravermelhas com cores, mostram ao observador uma cor distinta para cada valor de temperatura, facilitando a distinção de temperatura entre uma região da imagem e outra.

Para gerar uma imagem infravermelha é necessária uma câmera infravermelha ligada e apontada para um corpo que possua temperatura maior que o zero absoluto ($-273,15^{\circ}\text{C}$ ou 0K). A existência ou a ausência da iluminação são fatores que não influenciam a geração de uma imagem infravermelha, pois o único fator físico para originar uma imagem infravermelha é a radiação térmica emitida pela superfície do corpo (BEZERRA, 2010).

A Figura 2 ilustra, na prática, como uma imagem infravermelha se comporta em meios iluminados e escuros. A Figura 2(a) e Figura 2(c) são imagens geradas a partir de câmeras digitais com ausência de iluminação e em ambiente iluminado, respectivamente. As Figura 2(b) e Figura 2(d) são as imagens infravermelhas sem iluminação e com iluminação, respectivamente.

Figura 2 - Imagens digitais e infravermelhas em situações distintas de iluminação. (a) Imagem digital em ambiente escuro, (b) imagem infravermelha em ambiente escuro, (c) imagem digital em ambiente iluminado, (d) imagem infravermelha em ambiente iluminado.



Fonte: Adaptação de BEZERRA (2010).

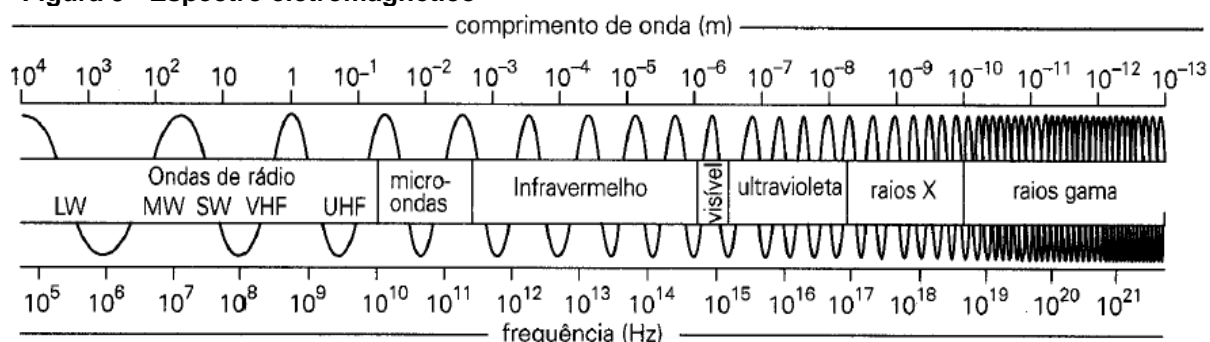
As imagens infravermelhas são geralmente utilizadas na engenharia e na medicina. Na engenharia, as imagens infravermelhas podem ser utilizadas como uma ferramenta de auxílio à variação de temperatura em equipamentos elétricos. Na medicina as imagens infravermelhas servem como um auxílio à detecção de doenças relacionadas ao câncer, doenças vasculares, dermatológicas, etc.

Os dois principais fatores relacionados às imagens infravermelhas são a resolução da imagem ou quantidade de *pixels* (*Picture element*) e a sensibilidade do sensor térmico, que determina a precisão ou ordem de erro. Sensores de sensibilidade moderada apresentam diferenças de até 0,1°C e sensores de boa qualidade apresentam diferenças de até 0,025°C na temperatura obtida (DIAKIDES *et al.*, 2006).

2.1.1 Radiação Térmica

O espectro de radiação eletromagnética (Figura 3) é um de ondas eletromagnéticas com suas frequências e comprimentos de onda. As ondas eletromagnéticas são caracterizadas pelo comprimento de onda e pela oscilação dos campos elétricos e magnéticos. As faixas de frequência de frequência entre $4,4 \cdot 10^{14} \text{Hz}$ e $7,5 \cdot 10^{14} \text{Hz}$ são visíveis a olho humano, e para estas dá-se o nome de luz visível. Mas, a maioria das faixas de frequência do espectro eletromagnético não são visíveis a olho nu, e o infravermelho é um exemplo de faixas não visíveis (LEITE e PRADO, 2012).

Figura 3 - Espectro eletromagnético



Fonte: Adaptação de LEITE e PRADO (2012).

A radiação infravermelha é uma fração do espectro eletromagnético cuja frequência é menor que a luz visível a olho humano e maior que as ondas de microondas e, portanto, possui comprimento de onda menor que as microondas e maior que a luz visível (SANCHES, 2009). A faixa do espectro eletromagnético onde

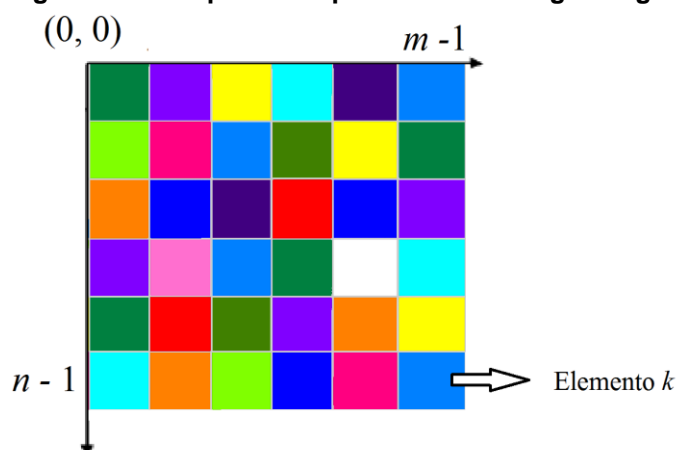
estão presentes as frequências do infravermelho iniciam-se em 0,75 micrometros (μm) e encerram-se em 1000 μm . Quanto mais quente um objeto se encontra maior quantidade de radiação infravermelha ele irá emitir, e esta é invisível a olho humano (AMALU, 2002).

Todo corpo ou objeto com uma temperatura superior ao zero absoluto emite radiação infravermelha. Isso dá-se pela movimentação dos átomos das moléculas deste corpo, gerando radiação em forma de calor. A pele humana emite radiação infravermelha de comprimentos de onda que variam entre 2 μm e 20 μm , mas na média possui picos entre 9 μm e 10 μm (AMALU, 2002).

2.2 PROCESSAMENTO DIGITAL DE IMAGENS

Uma imagem digital pode ser definida como uma estrutura bidimensional de largura N e comprimento M onde cada elemento k pertencente a uma posição (m_i, n_j) da matriz $M \times N$ contém um *pixel*. Cada *pixel* contém informações relacionadas a cor ou nível de cinza do elemento k da imagem digital (GONZALEZ e WOODS, 2010). A Figura 4 exemplifica uma imagem digital como uma matriz de *pixels* em um plano cartesiano.

Figura 4 - Exemplo de mapeamento de imagem digital



Fonte: Autoria própria.

A partir de uma imagem digital existem algumas técnicas do processamento de imagens que podem ser aplicadas, para coleta de informações ou modificações dessa imagem. Toda técnica aplicada tem o seu próprio objetivo que pode ser desde

melhorar a distribuição das cores como separar áreas ou objetos específicos para um observador ter uma melhor análise.

2.2.1 Segmentação por Limiarização

A segmentação é uma técnica onde aplicam-se algoritmos para dividir uma imagem de modo a separar todos os objetos de interesse compostos em seu interior. Para realizar uma segmentação precisa, cabe ao analista da imagem conseguir detectar os padrões do ambiente em estudo, juntamente com o realce que os possíveis objetos podem apresentar (GONZALEZ e WOODS, 2010).

Uma das técnicas mais simples da segmentação de imagens em escala de cinza é a limiarização, que consiste em determinar um valor específico L (limiar) que será utilizado como base para a realização de uma varredura *pixel a pixel* por toda imagem. Durante a realização da varredura, é coletado a intensidade do tom de cinza do *pixel* e de acordo com o valor do limiar L , este mesmo pixel pode ter a sua cor alterada. A limiarização pode auxiliar na separação dos objetos de interesse presentes e fundo na imagem (SANCHES, 2009).

A fundamentação matemática da limiarização pode ser dada pela Equação (1):

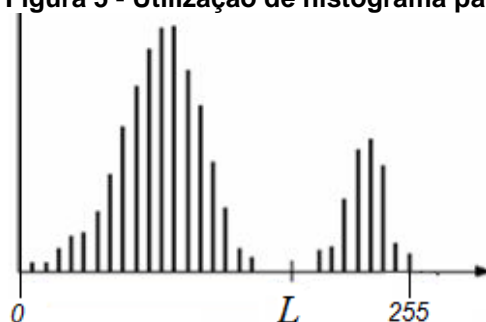
$$g(m, n) = \begin{cases} a, & \text{se } f(m, n) < L \\ b, & \text{se } f(m, n) \geq L \end{cases} \quad (1)$$

Onde:

- a, b são tons de cinza diferentes entre si;
- $f(m, n)$ é o valor do tom de cinza do *pixel* na linha m e coluna n ;
- L é o valor do limiar;
- $g(m, n)$ é o novo valor do *pixel* na linha m e coluna n .

A escolha do valor de limiar pode ser dada a partir de uma representação gráfica de distribuição de dos tons de cinza pela sua quantidade na imagem (histograma). De acordo com o formato do histograma, pode-se escolher um ou mais valores que melhor separem os objetos de uma imagem. A Figura 5 demonstra um histograma onde o eixo horizontal representa os tons de cinza e o eixo vertical a quantidade de *pixels* daquele tom.

Figura 5 - Utilização de histograma para escolha de limiar



Fonte: Adaptação de GONZALEZ e WOODS (2010).

A binarização é a aplicação da limiarização utilizando dois tons, aplicando o primeiro tom quando uma cor for menor que o limiar e aplicando o segundo tom quando a cor for maior ou igual ao limiar (SANCHES, 2009).

Outra maneira de segmentação é a limiarização múltipla Figura 6, onde adota-se mais de um valor de limiar para dividir a imagem. Para uma limiarização múltipla utilizando dois limiares necessita-se de três tons para segmentar a imagem. Segundo Gonzalez e Woods (2008) uma imagem segmentada por limiarização múltipla de dois limiares é dada pela Equação (2):

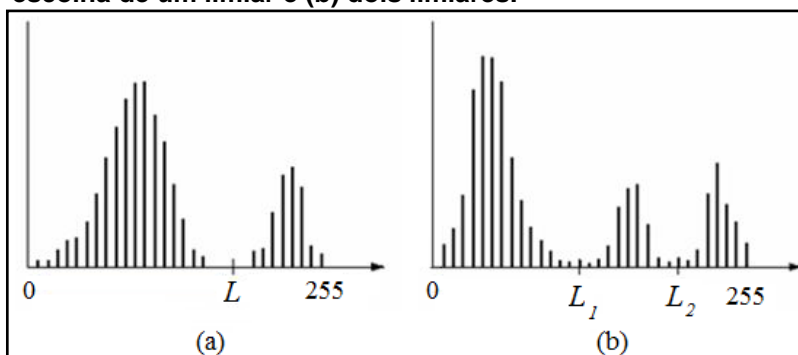
$$g(m, n) = \left\{ \begin{array}{ll} a, & \text{se } f(m, n) < L_1 \\ b, & \text{se } L_1 \leq f(m, n) \leq L_2 \\ c, & \text{se } f(m, n) > L_2 \end{array} \right\} \quad (2)$$

Onde:

- a , b e c são tons de cinza diferentes entre si;
- $f(m, n)$ é o valor do tom de cinza do *pixel* na linha m e coluna n ;
- L_1 é o valor do primeiro limiar;
- L_2 é o valor do segundo limiar;
- $g(m, n)$ é o novo valor do *pixel* na linha m e coluna n .

Na Figura 6 é representada a escolha de limiares L_1 e L_2 de acordo com o histograma.

Figura 6 - Exemplo de histograma para escolha de limiares. Sendo (a) a escolha de um limiar e (b) dois limiares.



Fonte: Adaptação de GONZALEZ e WOODS (2010).

A limiarização utilizando imagens infravermelhas é dada pela escolha do valor de temperatura, mapeada por um tom de cor correspondente. A aplicação da limiarização em imagens infravermelhas resulta na eliminação de pontos com valores de temperaturas específicos.

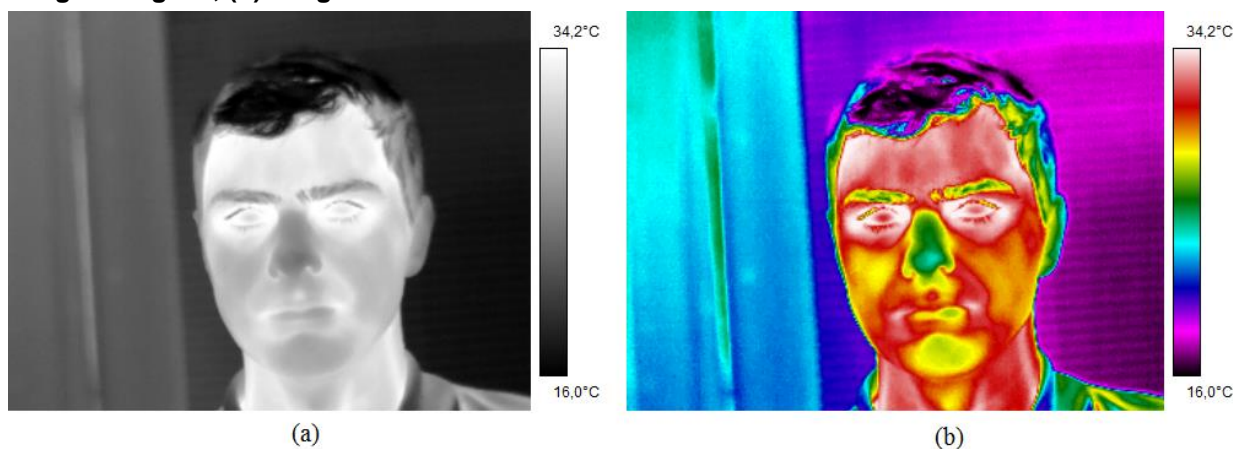
2.2.2 Processamento de Imagens em Pseudo-Cores

Algumas câmeras infravermelhas geram o mapeamento de temperaturas utilizando cores, diferenciando os níveis de temperatura de uma região ou outra. Porém, as cores das imagens infravermelhas não representam a cor original do corpo mapeado, representando então apenas uma diferenciação dos níveis de temperatura de regiões distintas. Segundo Gonzalez e Woods (2010), o olho humano consegue distinguir milhares de cores, mas é limitado em duas dezenas de tons de cinza.

A transformação de uma imagem em pseudo-cores pode ser implementada distribuindo o nível monocromático de um *pixel* em três canais, sendo eles vermelho, verde e azul. Contudo, os tons de cinza que pode variar de 0 a 255 em um nível, podem ser transformados em três níveis que variam de 0 a 255, podendo representar 16.777.216 cores diferentes (2^{24}). Uma técnica utilizada para colorir uma imagem em escala de cinza é a LUT (*Look-Up Table*) que mapeia todos os 256 tons de cinza e referencia cada tom com uma cor distinta (SANCHES, 2009).

A Figura 7(a) mostra uma imagem infravermelha em tons de cinza, onde os pontos mais claros são pontos de maior quantidade de calor. Já a Figura 7(b) é a aplicação de pseudo-cores a partir da Figura 7(a). Como exemplo, foi aplicada a paleta de cores *rainbow* (arco-íris).

Figura 7 - Resultado da aplicação de pseudo cores usando a paleta de cores *rainbow*. (a) Imagem original, (b) imagem resultado



Fonte: Autoria própria.

As câmeras infravermelhas utilizam da técnica de pseudo-cores como forma de auxiliar o observador da imagem a distinguir as regiões de temperaturas diferentes. Conhecendo a paleta de cores e o valor de temperatura correspondente de um ponto, um observador intuitivamente consegue localizar as regiões mais quentes e frias de uma imagem.

2.3 BIBLIOTECA *OPENCV*

Visão computacional é uma área vasta que pode ser definida como a transformação de um arquivo imagem ainda dentro de uma câmera de vídeo para tomada de decisões ou aplicação de representações. Tais transformações possuem uma maneira específica de aplicação e devem ser utilizadas com um objetivo específico (KAEHLER e BRADSKI, 2008).

Uma diferença entre o processamento de imagens digitais e a visão computacional é que a visão computacional busca resultados interpretados pelo cérebro humano, e o processamento de imagem aplica fórmulas matemáticas em imagens digitais representadas por matrizes. (PARKER, 2011).

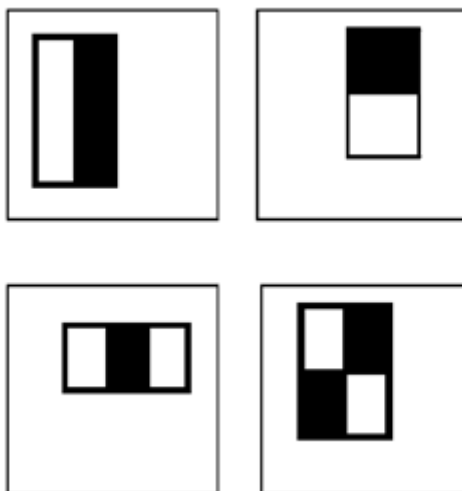
A biblioteca *Open Computer Vision (OpenCV)* é uma biblioteca computacional desenvolvida inicialmente pela Intel Research em linguagem C e C++, de código fonte aberto utilizada na área de visão computacional. Esta biblioteca foi desenvolvida e aprimorada com o foco de auxiliar na resolução de problemas de visão computacional

em tempo real de execução em máquinas com um ou mais processadores. A biblioteca *OpenCV* pode ser utilizada em sistemas operacionais Microsoft Windows, Linux e Apple MacOS (KAEHLER e BRADSKI, 2008).

Uma das áreas de visão computacional é a detecção de padrões, a partir de um treinamento de máquina os sistemas computacionais são capazes de detectar as formas aprendidas. Para o treinamento de máquina pode-se utilizar técnicas da Inteligência Artificial como Redes Neurais Artificiais ou aplicação de algoritmos de busca de similaridade em vizinhança, onde um sistema consegue detectar padrões e características em uma imagem. A biblioteca *OpenCV* já possui algumas técnicas de treinamento para aprendizagem de máquinas implementadas, inclusive a implementação do algoritmo de reconhecimento Viola e Jones em imagens digitais. Também, essa biblioteca já fornece arquivos de treinamentos realizados anteriormente, como por exemplo, arquivos de detecção facial e detecção de olhos (KAEHLER e BRADSKI, 2008).

2.3.1 Algoritmo de Reconhecimento Facial de Viola e Jones

Viola e Jones (2001) desenvolveram um algoritmo de detecção facial em tempo real baseado na varredura de características e não na varredura *pixel-a-pixel* utilizada geralmente. Para a classificação das regiões de uma imagem o algoritmo utiliza os chamados *Haar-like features*, que são imagens que possuem regiões diferentes utilizando-se dois tons. Na Figura 8 (a), (b), (c) e (d) estão alguns exemplos de *Haar-like features* que são utilizados pelo algoritmo de Viola e Jones. A realização da classificação é dada em uma imagem integral, pela subtração do somatório dos *pixels* dos dois tons após a aplicação de um limiar.

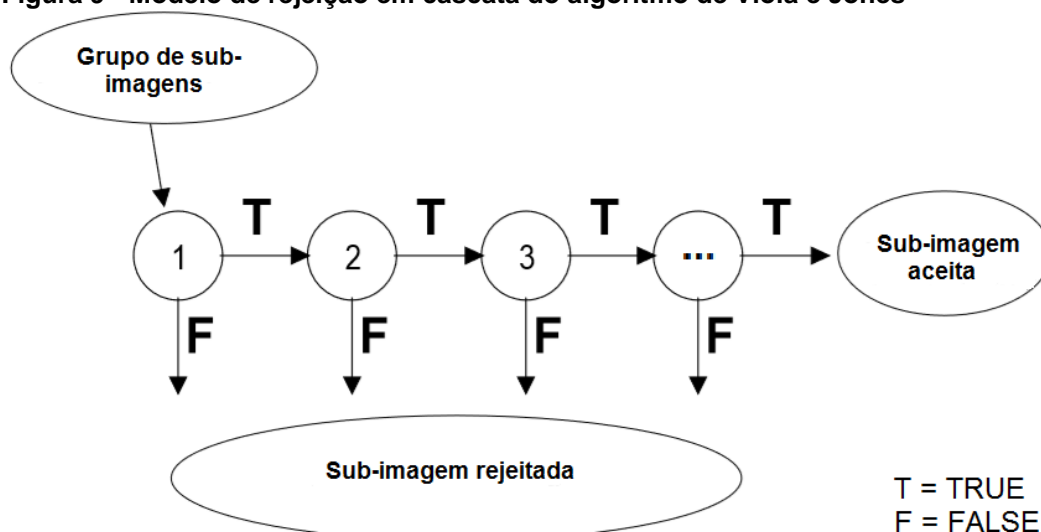
Figura 8 – Exemplos de *features* para classificação de áreas

Fonte: Adaptação de VIOLA e JONES (2004).

O algoritmo de Viola e Jones após o treinamento de imagens positivas e negativas consegue detectar, em tempo real, a existência de objetos em vídeos de até 15 quadros por segundo. O treinamento de máquina é baseado no algoritmo *AdaBoost*, que constrói um classificador selecionando um número pequeno de características específicas dos objetos de interesse. A eficácia do algoritmo dá-se pelo fato de ser possível criar vários classificadores menores, que executados sequencialmente, rejeitam com facilidade as áreas consideradas negativas. Classificadores mais simples são mais utilizados para a rejeição de áreas de fundo, enquanto classificadores complexos são utilizados para alcançar taxas pequenas de falsos positivos. Basicamente o trabalho do classificador é tentar localizar a região de fundo da imagem, e quando isso ocorre o classificador rejeita a Imagem Integral (VIOLA e JONES, 2004).

De forma geral, o algoritmo de detecção realiza a leitura de uma árvore de decisão degenerada, chamada de cascata, criada a partir dos classificadores, exemplificada na Figura 9. A detecção é realizada pela rejeição em cascata das imagens integrais, ou seja, uma vez que o classificador rejeita uma imagem integral, esta é descartada e não passa pelos demais classificadores. Logo, uma imagem integral só será aceita se não for rejeitada por todos os classificadores que for submetida (VIOLA e JONES, 2004).

Figura 9 - Modelo de rejeição em cascata do algoritmo de Viola e Jones



Fonte: Adaptação de Viola e Jones (2004).

Na Figura 10 está presente o resultado da aplicação de detecção facial utilizando o algoritmo de Viola e Jones em imagens digitais.

Figura 10 - Resultado da aplicação do algoritmo de Viola e Jones em uma imagem digital



Fonte: VIOLA e JONES (2004).

A função na biblioteca *OpenCV* que invoca o algoritmo de Viola e Jones é a *detectMultiScale*, e esta faz parte de um objeto da classe *CascadeClassifier*, que é a classe que representa o resultado do treinamento realizado. A função *detectMultiScale* possui três formas de ser invocada na biblioteca *OpenCV* 3.0, e em todas não há retorno por parte da função, sua resposta dá-se pelo ponteiro de vetor de detecções

(OPENCV, 2016). As descrições dos parâmetros que podem ser utilizados na *detectMultiScale* estão listadas a seguir, e as chamadas das funções são apresentadas no Quadro 1.

- *InputArray image*: objeto que representa o conteúdo da imagem onde será realizada a detecção;
- *Vector <Rect> &objects*: vetor de objetos de posições de retângulos, que guardam as coordenadas de todas as detecções;
- *double scaleFactor*: fator de escala, esse fator altera o tamanho da imagem no momento que será realizada a detecção;
- *int minNeighbors*: quantidade mínima de vizinhança para validar a detecção.
- *int flags*: Parâmetro de utilização de quantidade de execuções de um treinamento;
- *Size minSize*: objeto de tamanho que estabelece um tamanho mínimo de imagem para validar uma detecção;
- *Size maxSize*: objeto de tamanho que estabelece um tamanho máximo de imagem para validar uma detecção;
- *vector<int> &numDetections*: vetor de número de objetos, onde será guardado o número de vizinhanças possivelmente detectadas;
- *std::vector<int> &rejectLevels*: vetor onde serão armazenados os níveis rejeitados;
- *std::vector<double> &levelWeights*: vetor onde serão armazenados os pesos dos níveis de objetos rejeitados;
- *bool outputRejectLevels*: se for passado como verdadeiro retorna o valor de *levelWeights* ou *rejectLevels*, conforme a função.

Quadro 1 - Funções detectMultiScale da biblioteca OpenCV 3.0

```

void CascadeClassifier::detectMultiScale(InputArray image,
                                         vector<Rect> &objects,
                                         double scaleFactor = 1.1,
                                         int minNeighbors = 3,
                                         int flags = 0,
                                         Size minSize = Size(),
                                         Size maxSize = Size())

void CascadeClassifier::detectMultiScale(InputArray image,
                                         vector<Rect> &objects,
                                         vector<int> &numDetections,
                                         double scaleFactor = 1.1,
                                         int minNeighbors = 3,
                                         int flags = 0,
                                         Size minSize = Size(),
                                         Size maxSize = Size())

void CascadeClassifier::detectMultiScale(InputArray image,
                                         std::vector<Rect> &objects,
                                         std::vector<int> &rejectLevels,
                                         std::vector<double> &levelWeights,
                                         double scaleFactor = 1.1,
                                         int minNeighbors = 3,
                                         int flags = 0,
                                         Size minSize = Size(),
                                         Size maxSize = Size(),
                                         bool outputRejectLevels = false)

```

Fonte: Adaptação de OPENCV (2016).

Por se tratar de um algoritmo de detecção facial em tempo real, o algoritmo de Viola e Jones está presente na biblioteca *OpenCV* e pode ser utilizado para detecção de qualquer objeto. Dependendo do objetivo de detecção, quantidade de imagens positivas e negativas, o treinamento para o algoritmo pode demorar dias, porém a aplicação do algoritmo tende a ser rápida.

2.3.2 AdaBoost *Haartraining* e Geração de classificador em XML

Para a realização do treinamento de máquina baseado no algoritmo *AdaBoost* são primeiramente necessários um grupo de imagens positivas e um grupo de imagens negativas. As imagens positivas devem conter exatamente, e exclusivamente, o objeto em que deseja-se detectar, já as imagens negativas devem conter imagens que assemelham-se as áreas de fundo das imagens que serão feitas as varreduras. Após ter os dois grupos de imagens são criados dois arquivos: um

representando as imagens negativas (*negatives.info*) e outro representando as imagens positivas (*positives.info*) (OPENCV, 2016).

O conteúdo do arquivo *negatives.info* deve ser o local e nome da primeira imagem negativa na primeira linha, o local e nome da segunda imagem na segunda linha, seguindo essa lógica para todas as imagens do grupo de imagens negativas (OPENCV, 2016). A Figura 11 exemplifica a montagem do arquivo e o Quadro 2 contém um exemplo de arquivo representante do grupo de imagens negativas.

Figura 11 - Modelo de informação do arquivo *negatives.info*



Fonte: Autoria própria.

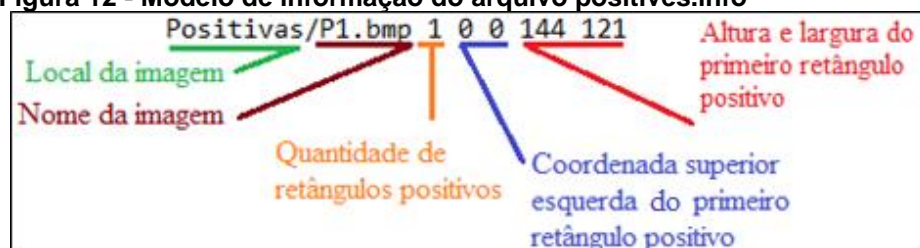
Quadro 2 - Exemplo do arquivo *negatives.info*

```
Negativas/N1.bmp
Negativas/N2.bmp
Negativas/N3.bmp
```

Fonte: Autoria própria.

Para o arquivo *positives.info* é necessário conter nas linhas o local, nome, quantidade de áreas retangulares positivas dentro da imagem e as coordenadas superior esquerda seguida da altura e largura de cada retângulo correspondentes as regiões positivas. A mesma lógica deve ser seguida até finalizar todo o grupo de imagens positivas (OPENCV, 2016). A Figura 12 exemplifica a montagem de um arquivo representante do grupo de imagens positivas e o Quadro 3 representa um exemplo de arquivo representante do grupo de imagens positivas.

Figura 12 - Modelo de informação do arquivo *positives.info*



Fonte: Autoria própria.

Quadro 3 - Exemplo do arquivo positives.info

```
Positivas/P1.bmp 1 0 0 144 121
Positivas/P2.bmp 1 0 0 120 88
Positivas/P3.bmp 2 0 0 144 121 190 255 240 300
```

Fonte: Autoria própria.

Com os arquivos *positives.info*, *negatives.info* e os grupos de imagens positivas e negativas já é possível prosseguir com as etapas de treinamento do algoritmo. A próxima etapa é a criação de *samples* (amostras) que é a geração de um arquivo binário (*samples.vec*) com as informações de quantidades de imagens para o treinamento. Para gerar esse arquivo é necessário utilizar uma função da biblioteca *OpenCV* que na versão 3.0 chama-se *createsamples* e possui os seguintes parâmetros:

- -vec: nome do arquivo binário a ser gerado (e.g. *samples.vec*);
- -img: arquivo com as imagens positivas (e.g. *positives.info*);
- -bg: arquivo com as imagens negativas (e.g. *negatives.info*);
- -num: quantidade de amostras que serão geradas;
- -bgcolor: cor de fundo predominante;
- -inv: se for especificado troca uma cor pela sua inversa;
- -randinv: se especificado troca aleatoriamente uma cor pela sua inversa;
- -maxidev: quantidade de *pixels* de desvio das amostras;
- -maxxangle: ângulo máximo de rotação no eixo *x*;
- -maxyangle: ângulo máximo de rotação no eixo *y*;
- -maxzangle: ângulo máximo de rotação no eixo *z*;
- -show: utilizado para depuração (*debugger*);
- -w: largura em *pixels* da amostra;
- -h: altura em *pixels* da amostra.

Depois de finalizado a geração do arquivo *samples.vec* pode-se iniciar o processo de treinamento. Esse processo pode demorar muito tempo, dependendo da quantidade de níveis estágios estabelecidos e o desempenho da detecção em que se deseja no treinamento. A realização do treinamento também é realizada através de

uma chamada de função da biblioteca *OpenCV*. Na versão 3.0 para sistema operacional Microsoft Windows é a função *haartraining* e para Linux é a *opencv_traincascade* (OPENCV, 2016). Os parâmetros dessa função são:

- -data: diretório onde serão gravados os arquivos do treinamento (*Data*);
- -vec: arquivo binário com as amostras (*samples.vec*);
- -bg: arquivo com as imagens negativas (*negatives.info*);
- -numPos: número de amostras positivas;
- -numNeg: número de amostras negativas;
- -numStages: número de estágios;
- -precalcValBufSize: tamanho em *megabytes* de *buffer* pré-calculado;
- -precalcIdxBufSize: tamanho em *megabytes* de índices do *buffer* pré-calculados;
- -baseFormatSave: formato de criação dos arquivos de treinamento;
- -numThreads: número máximo de *threads* a serem usadas;
- -featureType: modo de varredura pode ser HAAR (*HAAR-like features*) ou LBP (*Local Binary Patterns*);
- -w: largura em *pixels* da amostra;
- -h: altura em *pixels* da amostra;
- -bt: tipo de classificador que pode ser DAB (*Discrete AdaBoost*), RAB (*Real AdaBoost*), LB (*Logit Boost*) ou GAB (*Gentle AdaBoost*);
- -minHitRate: quantidade mínima de estágios por classificados;
- -maxFalseAlarmRate: quantidade máxima de classificadores falsos;
- -weightTrimRate: estabelece um peso como medida de descarte;
- -maxDepth: profundidade máxima de uma árvore de decisão fraca;
- -maxWeakCount: contagem máxima de árvores fracas na varredura;
- -mode: modo de treinamento que pode ser BASIC (básico), CORE (núcleo) ou ALL (*BASIC e CORE*).

Após a realização do treinamento, todos os arquivos gerados ficarão na pasta estabelecida (*Data*), e deve-se então, realizar a conversão desses arquivos em um arquivo XML (*eXtensible Markup Language*) para a aplicação realizar a leitura e a detecção dos objetos desejados (OPENCV, 2016). A função da biblioteca *OpenCV*

versão 3.0 que realiza essa conversão é a *convert_cascade*, e tem os seguintes argumentos:

- --size = WxH, onde W é a largura em *pixels* e H é a altura em *pixels*;
- Diretório dos arquivos (e.g. *Data*);
- Nome do arquivo XML para geração (e.g. *haartraining.xml*);

A seguir será realizada uma breve descrição da variação da temperatura na face humana, a anatomia de alguns pontos e alguns motivos de haver o aumento de temperatura e como o valor pode ser classificado. Ao final é explicado como são feitos os cálculos que comprovam a qualidade de um diagnóstico a partir de testes realizados.

2.4 VARIAÇÃO DE TEMPERATURA DA FACE HUMANA

A temperatura do corpo humano pode variar nas temperaturas entre 36°C e 37°C. A região interna do corpo humano possui, na maioria do tempo, temperaturas constantes, as variações que ocorrem são em torno de 0,6°C. (CARVALHO,2002). Porém, nas extremidades do corpo como nariz, dedos ou mãos tendem a ser pontos onde a temperatura não condiz com a real do indivíduo. Para medir a temperatura corporal utilizando um termômetro, deve-se colocar o mesmo em alguma cavidade que tenha o menor contato possível com o ambiente externo. Alguns locais ideais de medição de temperatura corporal são as dobras das axilas, interior da boca ou na cavidade onde se inicia o intestino grosso (VARELA, 2012).

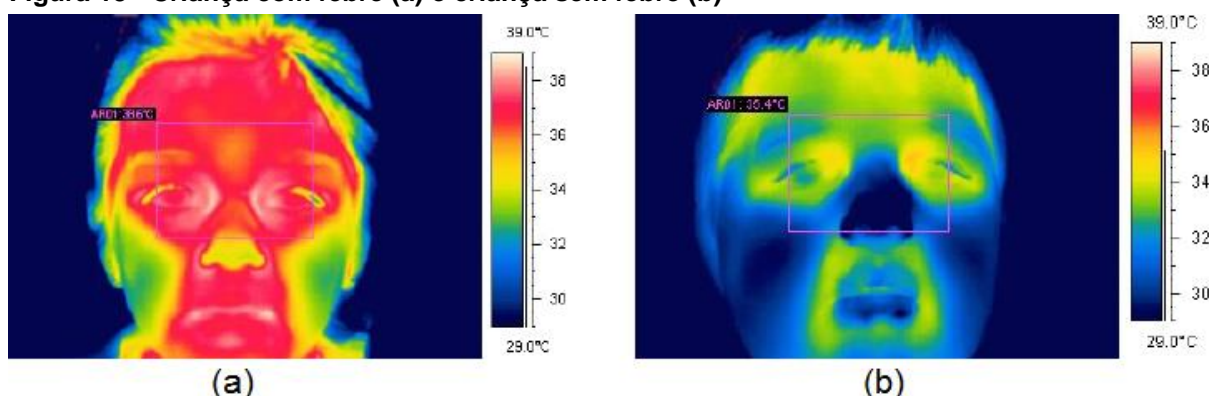
Quando um corpo humano possui uma temperatura maior que 37,8°C diz-se que o mesmo está com febre. Essa febre pode ser um sintoma de algum tipo de infecção viral, bacteriana ou doenças relacionadas a hemorragias, reumatismos entre outras (VARELA, 2012). Porém, algumas dessas doenças virais, principalmente influenzas, são doenças contagiosas e se espalhada para uma quantidade grande de pessoas causam uma contaminação em cadeia, podendo se tornar uma pandemia (RING *et al*, 2008).

Nos últimos anos, com a ocorrência de possíveis pandemias, como a influenza H1N1 e o vírus Ebola, alguns países tomaram como medida preventiva a triagem de pessoas em locais de grande movimentação pública. Porém, o que pode ocorrer é o processo de triagem ser manual e colocar em risco a saúde do profissional que está realizando esse processo de triagem (SANCHES *et al.*, 2016).

A utilização de câmeras termográficas vem sendo uma alternativa para a verificação de temperatura de um indivíduo, e possui a vantagem de não precisar de contato direto do profissional com o indivíduo para a tomada do valor da temperatura. Com a utilização de uma câmera termográfica com precisão de temperatura menor ou igual a 0,2°C e uma fixação de paleta de cores pode-se perceber somente olhando na imagem se um paciente está ou não com febre (RING *et al.*, 2008).

A Figura 13 contém duas imagens com a utilização de paleta de cores fixa, com crianças de mesma idade, onde a criança na Figura 13(a) está com estado febril e a criança na Figura 13(b) está com temperatura considerada normal.

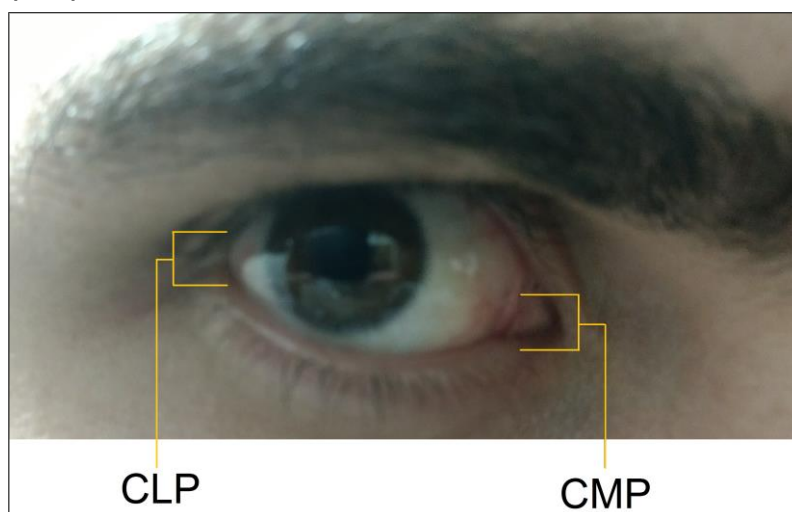
Figura 13 - Criança com febre (a) e criança sem febre (b)



Fonte: RING *et al.*, 2008.

A face humana adulta possui 28 pontos de termoanatomia e tais pontos são simétricos entre si, ou seja, os pontos termoanatômicos encontrados na região esquerda da face são os mesmos da direita. Esses pontos da face, em um ambiente controlado e com a aplicação de procedimentos padrão para coleta dos dados obtiveram valores significativamente parecidos em seres humanos diferentes. Então, concluiu-se que a face humana possui pontos de características comuns em corpos diferentes. Alguns dos pontos termoanatômicos da face são a Comissura Medial Palpebral (CMP) e a Comissura Lateral Palpebral (CLP), como foram demarcados na Figura 14 (HADDAD, 2008).

Figura 14 - Comissura medial palpebral (CMP) e comissura lateral palpebral (CLP)



Fonte: Autoria própria.

Segundo Ring *et al.* (2013), analisando uma imagem termográfica, pode-se perceber que a região dos olhos é o local da face com a maior quantidade de calor. Segundo Haddad (2008) as regiões das comissuras palpebrais são pontos termoanatómicos e mesmo em indivíduos diferentes apresentarão, em sua maioria, o mesmo comportamento térmico. Segundo Sanches *et al.* (2016) a utilização de imagens infravermelhas permitiu identificar a temperatura da CMP e compará-las com a temperatura axilar do voluntário.

2.5 CÁLCULO DE ESPECIFICIDADE, SENSIBILIDADE E ACURÁCIA

Com a variedade de doenças e o quantitativo de sintomas comuns entre estas surgiu, para os médicos, a necessidade de realização de diagnósticos precisos e de qualidade a partir de resultados de exames. Para tal, a partir de hipóteses de doenças, são realizados alguns exames específicos e sensíveis, onde estes podem comprovar ou não as hipóteses levantadas (GUIMARÃES, 1985).

Segundo Guimarães (1985), sensibilidade é a capacidade de um teste selecionar, dentre um conjunto de indivíduos sintomáticos, os que realmente têm a patologia estudada. Especificidade é a capacidade de selecionar, dentre um grupo de

indivíduos sintomáticos, os que não possuem a patologia em questão. Já a acurácia é a capacidade de um teste conseguir obter resultados esperados.

Na Figura 15 estão demonstrados a forma de aquisição das variáveis necessárias para os cálculos de sensibilidade, especificidade e acurácia.

Figura 15 - Demonstrativo de aquisição de variáveis de testes

| | | Teste | | |
|--------|-----|---------------------|---------------------|------------------|
| | | Sim | Não | |
| Doença | Sim | Verdadeiro positivo | Falso negativo | Total de doentes |
| | Não | Falso positivo | Verdadeiro negativo | Total de sadios |
| | | Total de positivos | Total de negativos | Total geral |

Fonte: Adaptação de GUIMARÃES (2014).

Os cálculos de sensibilidade, especificidade e acurácia dão-se respectivamente pela Equação (3), Equação (4) e Equação (5):

$$Sensibilidade = 100 * \frac{verdadeiros\ positivos}{(verdadeiros\ positivos + falsos\ negativos)} \% \quad (3)$$

$$Especificidade = 100 * \frac{verdadeiros\ negativos}{(verdadeiros\ negativos + falsos\ positivos)} \% \quad (4)$$

$$Acurácia = 100 * \frac{(verdadeiros\ positivos + verdadeiros\ negativos)}{total\ geral} \% \quad (5)$$

Como cada diagnóstico de patologia possui seus valores de acurácia, sensibilidade e especificidade, cabe ao especialista decidir, dentre um conjunto de fatores, quais exames pode confirmar ou rejeitar a maior quantidade de hipóteses (GUIMARÃES, 1985).

3 METODOLOGIA

O desenvolvimento da ferramenta passou pelas etapas de coleta e tratamento das imagens, treinamento com imagens infravermelhas e uso da ferramenta. Essas etapas foram dependentes e contíguas entre si.

3.1 COLETA E TRATAMENTO DE IMAGENS

Com uma câmera infravermelha FLIR T440 em um ambiente controlado, foram adquiridos vídeos de 30 FPS e imagens de voluntários. A justificativa das imagens serem de um ângulo frontal dá-se pelo fato destas serem específicas do tipo esperado para o resultado final da ferramenta. Para cada voluntário, foi coletado também a temperatura axial, tendo-se o valor aproximado de temperatura corporal de cada um.

O ambiente em que foram coletadas as imagens estava controlado com temperatura de $22^{\circ}\text{C} \pm 1^{\circ}\text{C}$. As imagens obtidas possuem a resolução de 320×240 *pixels* sendo, cada um deles, uma representação numérica da quantidade de calor com precisão térmica de $0,1^{\circ}\text{C}$ nos valores de temperaturas. Com o software da fabricante da câmera, foi feita uma conversão dos vídeos obtidos para imagens nos formatos *BMP* e *FPF*. Foram selecionadas as imagens de melhor foco, nitidez e menor quantidade de ruído, as demais foram descartadas.

Após o processo de coleta das imagens estar finalizado foram obtidas um total de 1.558 imagens infravermelhas, estas contendo uma única visão frontal de uma face. A partir disso, foi aplicado a detecção facial com o algoritmo de Viola e Jones com o treinamento de detecção facial *haarcascade_frontalface_default.xml*, padrão da biblioteca *OpenCV* versão 3.0. Os resultados obtidos foram:

- 64,25% de falsos negativos (1001 não detecções);
- 3,32% de falsos positivos (52 detecções);
- 32,41% de verdadeiros positivos (509 detecções);

Após a aplicação da detecção facial utilizando o arquivo de treinamento *haarcascade_frontalface_default.xml*, notou-se que o percentual de falsos negativos era elevado e o percentual de verdadeiros positivos era baixo. Com o objetivo de melhorar o percentual de verdadeiros positivos e diminuir os percentuais de falsos negativos e falsos positivos, decidiu-se realizar um novo treinamento, utilizando imagens infravermelhas em escala de cinza. Como a quantidade de imagens infravermelhas disponíveis era de apenas 1.558, resolveu-se aumentar esse quantitativo aplicando diferentes paletas de cores em tons de cinza em nas imagens. Tal aplicação gera uma imagem em memória diferente, entretanto garante que as propriedades da imagem sejam mantidas. Essa técnica foi aplicada cinco vezes e resultaram nos grupos de temperaturas A, B, C, D e E. Cada grupo possui um intervalo de temperatura diferente, sendo estes:

- A: de 25°C a 37°C;
- B: de 25°C a 38°C;
- C: de 25°C a 39°C;
- D: de 25°C a 40°C;
- E: de 25°C a 41°C.

Para cada grupo de temperaturas, um tom de cinza representa o mesmo intervalo de valores de temperaturas em todas as outras imagens desse mesmo grupo. E para aumentar a quantidade de imagens, foi aplicado em cada uma destas as temperaturas de todos os grupos, multiplicando a quantidade de imagens disponíveis pela quantidade de grupos. Para melhor organizar as imagens resultantes, todas as imagens foram colocadas em um único diretório, e foi renomeado uma-a-uma com o nome do grupo e o número da imagem. Exemplo: A1, A2, B1, B2, etc.

3.2 ARQUIVO DE IMAGEM FPF

As imagens infravermelhas adquiridas com câmeras infravermelhas da FLIR podem ser analisadas, importadas e exportadas pelo *software* ThermaCAM Researcher. Na versão 2.9 desse sistema, as imagens infravermelhas podem ser exportadas nos formatos de vídeo AVI (*Audio Video Interleave*), imagem no formato

BMP e como binários nos formatos *FPF* (*Flir Public Format*) e *MAT* (*MatLab*) (FLIR, 2000).

Os arquivos de extensão **.fpf** são binários compostos por um cabeçalho de 892 *bytes* seguido de uma matriz de valores do tipo ponto flutuante de precisão simples (32 bits) (IEEE-754) que representam unitariamente um ponto da imagem. As imagens infravermelhas podem ser exportadas para o tipo *FPF*, a partir do *software* *ThermaCAM Researcher 2.9*, porém esse sistema não realiza a importação com esse formato (FLIR, 2000).

No desenvolvimento do trabalho, as imagens infravermelhas utilizadas na realização do treinamento foram exportadas em escala de cinza utilizando o formato de imagem digital *BMP*. Para a realização da detecção facial e febre, as imagens infravermelhas foram exportadas no formato de arquivo *FPF*.

As definições, utilizando a linguagem C/C++ estão listadas no APENDICE A. A quantidade de valores de temperatura que um arquivo do tipo *FPF* possui é obtida a partir do produto das dimensões da altura pela largura da imagem (Equação 6). E esses valores são listados sequencialmente após os valores do cabeçalho do arquivo.

$$\left\{ \begin{array}{l} FPFHEADER_T::FPFIMAGE_DATA_T::ySize * \\ FPFHEADER_T::FPF_IMAGE_DATA_T::xSize \end{array} \right\} \quad (6)$$

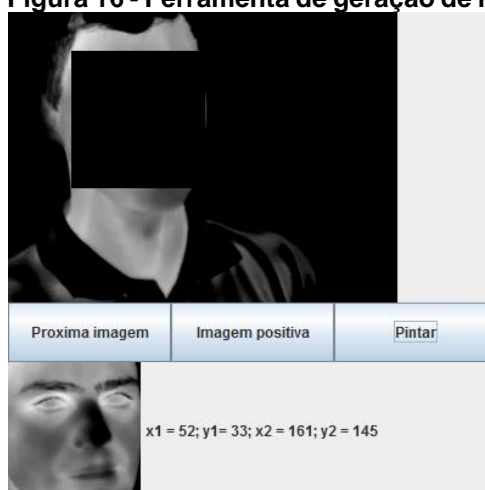
3.3 TREINAMENTO

As ferramentas utilizadas nas fases do treinamento foram a linguagem Java, a biblioteca *OpenCV* versão 3.0, a linguagem *VBScript* (*Visual Basic Script*) e alguns comandos *shell*. A linguagem Java foi utilizada para criar uma ferramenta de corte e a geração dos arquivos de texto necessários para a realização do treinamento. A biblioteca *OpenCV* foi utilizada como ferramenta para a realização do treinamento com os arquivos de texto e as imagens positivas e negativas geradas com a ferramenta desenvolvida em Java. A linguagem *VBScript* e comandos *shell* foram utilizados para criação de *scripts* de forma a automatizar o processo de treinamento, garantindo a correta sequência de execução das tarefas de forma organizada.

Para a automação do processo de criação de imagens positivas, negativas e o recorte de imagens, foi desenvolvida uma ferramenta em linguagem Java. Utilizou-

se desta pois é uma linguagem de alto nível, de rápido desenvolvimento e atenderia as necessidades de forma eficiente. O funcionamento da ferramenta dá-se pela abertura de um arquivo *TXT* com os nomes das imagens que seriam recortadas, seguido pela abertura uma-a-uma de cada imagem. Após carregada, cada imagem era mostrada e passava pelo recorte da região positiva de forma manual. Após o recorte, era salvo a imagem positiva e preenchida esta região com a cor preta, cor padrão de fundo na biblioteca *OpenCV*. Depois de recortado e preenchido todas as regiões positivas de preto de cada imagem, era então feito recortes aleatórios na imagem restante, gerando assim as imagens negativas. Após esse ciclo ser executado em todas as imagens disponíveis, a ferramenta realizava a geração dos arquivos de texto necessários para a primeira etapa do treinamento na biblioteca *OpenCV*. A Figura 16 mostra a ferramenta de corte de imagens com uma imagem carregada já com o recorte da região positiva.

Figura 16 - Ferramenta de geração de imagens positivas e negativas



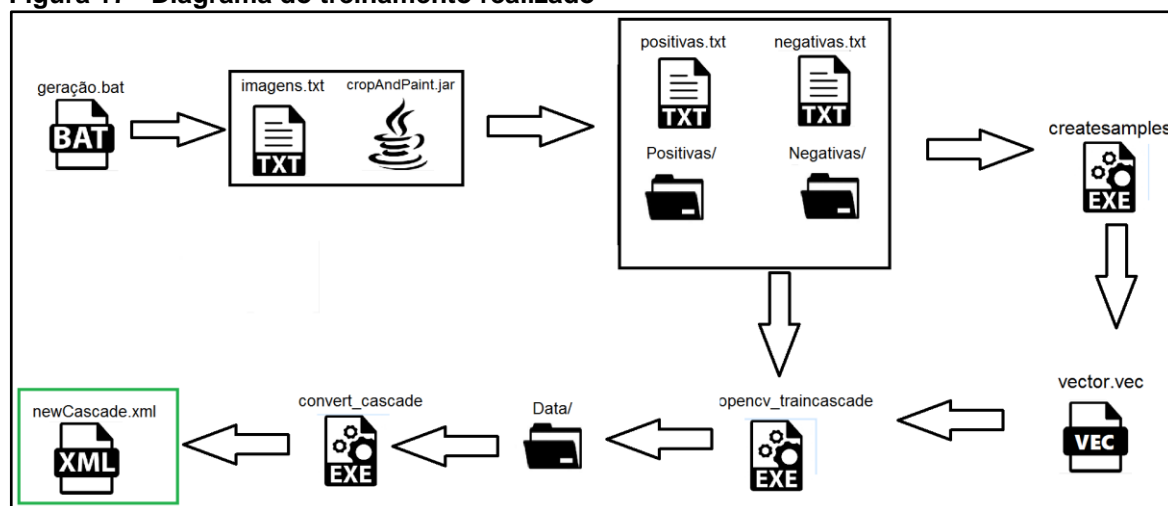
Fonte: Autoria própria.

Como as etapas do treinamento e a geração do arquivo *XML* são, basicamente chamadas de funções da biblioteca *OpenCV*, foi criado um arquivo de comandos em lote, impondo ao sistema operacional a execução das funções. As tarefas executadas pelo arquivo em lote são: *backup* dos arquivos de treinamentos anteriores, criação de diretórios de organização, invocação da ferramenta de corte, realização de treinamento e geração do arquivo *XML*. Antes de iniciar o treinamento e após a conclusão do treinamento foi adicionado um *script* que coletava a hora atual

do relógio controlado pelo sistema operacional. Esses dados de hora serviram para calcular o tempo aproximado da realização do treinamento.

Visualmente, o processo de realização do treinamento está apresentado na Figura 17, onde o processo inicia-se invocando o arquivo *geração.bat* e finaliza-se com a geração do arquivo *newCascade.xml*.

Figura 17 - Diagrama do treinamento realizado



Fonte: Autoria própria.

Os parâmetros utilizados na criação das amostras estão destacados no Quadro 4, os dados da geração do treinamento estão presentes no Quadro 5 e os argumentos da função de geração de XML foram: `--size=25X25 data newHaarcascade.xml`. Os parâmetros das funções da biblioteca OpenCV que não foram informados no Quadro 4 e no Quadro 5, foi adotado os valores padrões fornecidos pela biblioteca *OpenCV*. As realizações dos treinamentos resultaram em arquivos XML, onde nestes estão mapeados os classificadores de detecção necessárias para o algoritmo delimitar as áreas positivas e negativas de uma imagem.

Quadro 4 - Parâmetros da função *createsamples*

| | |
|------|---------------|
| -vec | vector.vec |
| -img | positivas.txt |
| -bg | negativas.txt |
| -num | 2000 |
| -w | 25 |
| -h | 25 |

Fonte: Autoria própria.

Quadro 5 - Parâmetros da função *haatraining*

| | |
|-------|---------------|
| -data | Cascades/data |
| -vec | vector.vec |
| -bg | negativas.txt |
| -npos | 7790 |
| -mem | 1024 |
| -mode | ALL |
| -w | 25 |
| -h | 25 |

Fonte: Autoria própria.

3.4 USO DA FERRAMENTA

Para realizar a detecção facial e a classificação do valor de máxima temperatura da face, foi implementada uma ferramenta que executa uma sequência de ações a partir de arquivos informados na entrada. Tais arquivos são: o arquivo *TXT* com o nome das imagens em formato *FPF*, o arquivo de treinamento em formato *XML* e o arquivo de paleta de cores, em formato *RGB*, para visualização do resultado. Como resposta, a ferramenta fornece as imagens geradas, o valor de máxima temperatura das áreas detectadas de cada imagem, os quantitativos de detecções, e a quantidade de indivíduos nas classes febre, febrícula e normal. No desenvolvimento da ferramenta buscou-se criar as funções de forma isolada e independente, separando cada funcionalidade em uma função diferente, pois dessa forma se poderia trabalhar com várias quantidades de imagens de uma só vez.

A rotina da ferramenta dá-se, primeiramente, pela leitura e validação dos arquivos de entrada, verificando a existência e consistência dos mesmos. Se algum arquivo de entrada for inválido, o sistema encerra sua execução. O primeiro arquivo a ser validado é o *XML* (classificador), que quando válido é convertido para um objeto na memória. O segundo arquivo a ser validado é o arquivo *RGB* (paleta de cores), que quando válido é convertido para uma estrutura de mapeamento, que servirá para trocar os tons de cinza por tons de cores. Posteriormente é feita a leitura do arquivo *TXT* (lista de imagens) onde linha a linha coleta-se o nome da imagem *FPF* para o seu carregamento e execução das funções da ferramenta.

A primeira função a ser executada após a leitura do arquivo *TXT* é a leitura do arquivo *FPF* correspondente, este será utilizado para a criação da imagem em que serão executadas as ações de detecção facial e de febre. Inicialmente, a imagem é criada com 256 tons de cinza, sendo os tons mais claros os valores de maior temperatura e os mais escuros os valores de menor temperatura. Com a imagem na memória, é realizado a limiarização, removendo as temperaturas maiores que 43°C e menores que 25°C, com intuito de remover parte do fundo e regiões desnecessárias.

Para realizar a detecção com o algoritmo de Viola e Jones, foram estabelecidos tamanhos de *features* que variam nas dimensões de 20% a 80% do tamanho da imagem original. Após a imagem estar carregada e tratada é realizada a detecção da face, onde são retornados os pontos superior esquerdo e inferior direito da região retangular da face encontrada. Considerando as regiões detectadas é realizada uma varredura para a localização do valor e a posição do ponto de maior temperatura. Ao obter o valor máximo de temperatura é realizado a sua classificação. Tendo todas as regiões e pontos de temperaturas encontrados, é realizado na imagem a aplicação da paleta de cores e desenhado as regiões de face e ponto de temperatura máxima. O processo na imagem é finalizado quando é feita a gravação da imagem e incrementado o valor da classe de temperatura em que a temperatura máxima desta imagem se encontra.

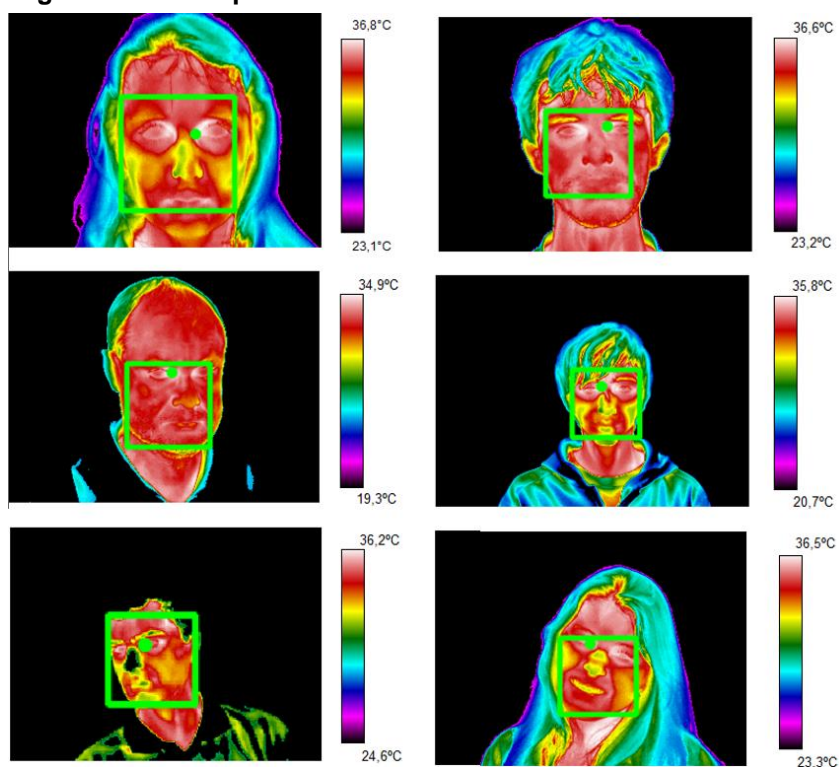
Ao término dos procedimentos da ferramenta tem-se o retorno com o valor e a classificação do ponto de máxima de temperatura de cada detecção, e juntamente com as imagens gravadas pode-se verificar a performance exercida pela ferramenta.

Com os valores de temperaturas da imagem infravermelha e a axial do voluntário, é possível verificar se a classificação de temperatura da ferramenta corresponde ou não a verdadeira classificação do indivíduo. E com as imagens gravadas com o local da detecção pode-se definir as taxas de verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos. O tempo de execução da ferramenta varia de acordo com a quantidade de imagens informadas na entrada e a quantidade de detecções que ela realiza, pois para cada detecção é feita a busca de maior valor de temperatura. Porém, enquanto o tempo gasto para treinamento das imagens demora algumas horas, o tempo para a detecção dessas mesmas imagens são cerca de alguns minutos.

Na Figura 18 estão apresentados alguns resultados da ferramenta quando utilizado o treinamento de 15 estágios. O quadro no interior da face representa a

detecção realizada, e o ponto na CMP indica o valor de maior temperatura dessa região. A paleta de cores que foi aplicada foi a *rainbow*, porém os pontos de temperatura menores que 25°C e maiores que 43°C foram pintados de preto.

Figura 18 – Exemplos de resultado da ferramenta com a identificação da CMP



Fonte: Autoria própria.

3.5 COMENTÁRIOS

Nesta sessão foram apresentados os métodos de coleta, tratamento das imagens infravermelhas para a realização do treinamento e execução da detecção facial. Também foi realizada a exportação e visualização das imagens infravermelhas após a detecção facial.

Na sessão a seguir, serão mostrados os resultados obtidos após a aplicação das etapas descritas anteriormente, utilizando um conjunto de imagens infravermelhas que não foram utilizadas na realização do treinamento.

4 RESULTADOS

Neste capítulo serão apresentados os resultados e as análises dos resultados da ferramenta obtidos após a aplicação da metodologia descrita no capítulo 3. Para tal, foram realizados sete treinamentos, para cada um foram utilizadas 7.790 imagens positivas e 20.000 imagens negativas, com esse total de imagens foram estabelecidas 2.000 amostras de tamanho 25x25 *pixels*. O diferencial de cada treinamento foi a quantidade de estágios, onde adotou-se as quantidades de 3, 5, 7, 11, 13, 15 e 20 estágios, esses valores foram escolhidos pois desejava-se comparar a duração e o desempenho nas detecções de cada treinamento.

4.1 ANÁLISE DO TREINAMENTO

Na realização do treinamento, a escolha dos parâmetros implica diretamente na qualidade e na duração do treinamento, dentre estes, observou-se que a quantidade de estágios e de amostras afetam diretamente esses atributos. A quantidade de estágios por treinamento, o tempo aproximado gasto pelo algoritmo, e a quantidade de nós da árvore de decisão são detalhados na Tabela 1.

Tabela 1 - Dados dos treinamentos realizados

| Execução <i>haartraining</i> | | | |
|-------------------------------------|------------------------|------------------------------|---------------------------|
| Treinamento | Quantidade de estágios | Duração aproximada | Quantidade de nós gerados |
| 1 | 5 | 0 dias 02 horas e 15 minutos | 4 |
| 2 | 7 | 0 dias 07 horas e 55 minutos | 6 |
| 3 | 9 | 0 dias 12 horas e 15 minutos | 8 |
| 4 | 11 | 0 dias 16 horas e 00 minutos | 9 |
| 5 | 13 | 1 dia 00 horas e 40 minutos | 11 |
| 6 | 15 | 1 dia 12 horas e 35 minutos | 13 |
| 7 | 20 | 9 dias 02 horas e 15 minutos | 17 |

Fonte: Autoria própria.

Para analisar os arquivos XML obtidos foi desenvolvido uma ferramenta que realiza a detecção e como saída devolve a quantidade de detecções e uma imagem com marcação nas áreas detectadas. Cada imagem utilizada possuía exatamente

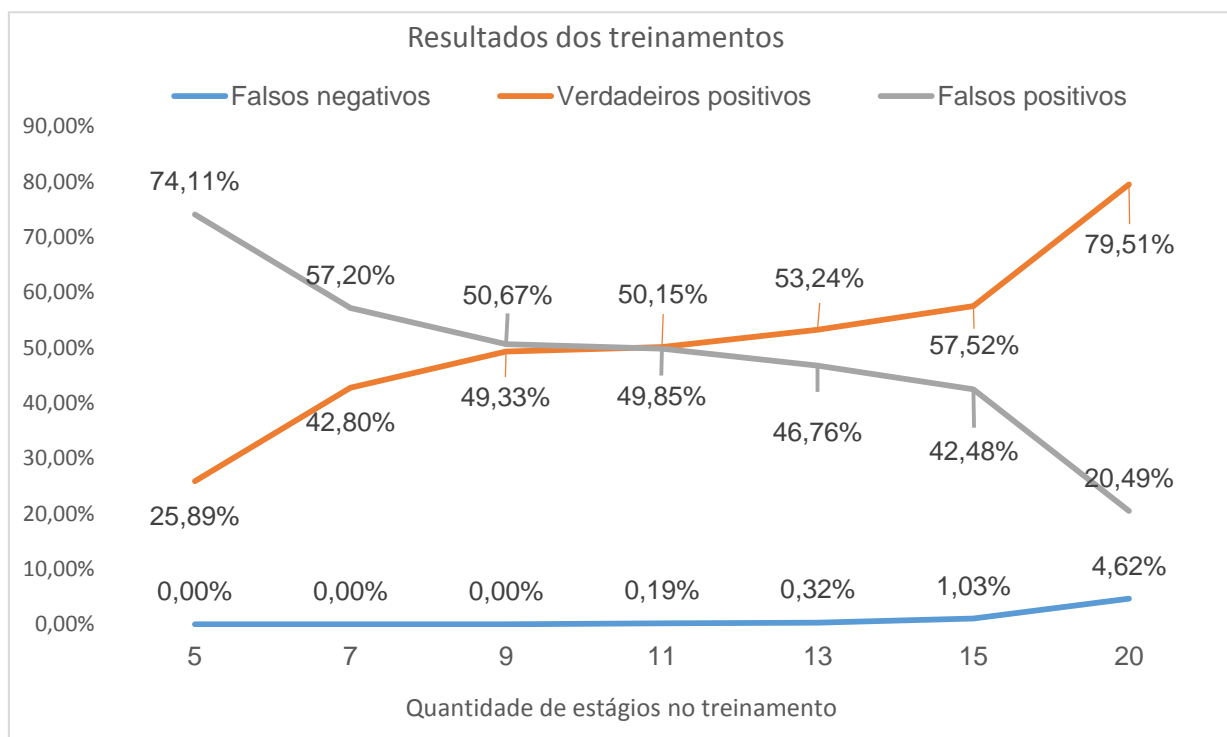
uma face com a visão frontal parcial ou total, e o resultado perfeito dar-se-ia pela única detecção exata da região onde se encontrava a face. Na detecção pode-se ter como resultados áreas positivas (verdadeiros positivos), áreas positivas detectadas erroneamente (falsos positivos) e áreas positivas que não foram detectadas (falsos negativos). Na Tabela 2 estão os dados obtidos com a realização da detecção com todos os arquivos obtidos nos treinamentos e o Gráfico 1 demonstra, visualmente esses valores.

Tabela 2 - Análise do treinamento

| Aplicação haartraining | | | | | | | |
|------------------------|-----------|------------------|--------|------------------|-------|-----------------------|--------|
| Treinamento | Detecções | Falsos positivos | | Falsos negativos | | Verdadeiros positivos | |
| 1 | 5.331 | 3.951 | 77,11% | 0 | 0,00% | 1.380 | 25,89% |
| 2 | 3.446 | 1.971 | 57,20% | 0 | 0,00% | 1.475 | 42,80% |
| 3 | 3.154 | 1.598 | 56,67% | 0 | 0,00% | 1.556 | 49,33% |
| 4 | 3.075 | 1.533 | 49,85% | 3 | 0,19% | 1.542 | 50,15% |
| 5 | 2.883 | 1.348 | 46,76% | 5 | 0,32% | 1.535 | 53,24% |
| 6 | 2.641 | 1.122 | 42,48% | 16 | 1,03% | 1.519 | 57,52% |
| 7 | 1.864 | 382 | 20,49% | 72 | 4,62% | 1.482 | 79,51% |

Fonte: Autoria própria.

Gráfico 1 - Resultados do treinamento



Fonte: Autoria própria.

Nota-se analisando a Tabela 2 e o Gráfico 1 pode-se perceber que a medida que aumenta-se a quantidade de estágios no treinamento o percentual de verdadeiros positivos e o percentual de falsos negativos aumentam. O percentual de falsos positivos diminui na mesma proporção que o percentual de verdadeiros positivos cresce. A medida que crescem a quantidade de estágios no treinamento, aumenta a quantidade de nós árvore que o algoritmo de Viola e Jones (2001) irá utilizar para aceitar ou rejeitar uma Imagem Integral. Com isso, a tomada de decisão fica mais criteriosa, e aumenta a taxa de verdadeiros positivos, diminui a taxa de falsos positivos e aumenta a quantidade de imagens que são rejeitadas como um todo.

4.2 ANÁLISE DE RESULTADOS

Utilizando a temperatura axial de um grupo de voluntários, foi realizada uma comparação com a temperatura medida através de um termômetro e o valor da temperatura máxima obtida na área detectada pelo algoritmo de Viola e Jones na imagem infravermelha. Estes valores estão apresentados na Tabela 3 e pode-se notar que a maioria das variações de temperatura ficaram em valores entre $-0,25^{\circ}\text{C}$ e $0,48^{\circ}\text{C}$. Quanto menor a diferença entre a temperatura axial e a temperatura na imagem infravermelha, melhor serão os resultados da classificação da ferramenta.

Tabela 3 - Dados coletados de temperatura

| Valores de temperatura coletados pelo termômetro e pelo termograma | | | |
|--------------------------------------------------------------------|--------------------|------------------------|-----------|
| Voluntário | Temperatura axilar | Temperatura termograma | Diferença |
| 1 | 36,4°C | 36,88°C | 0,48°C |
| 2 | 36,8°C | 36,63°C | -0,17°C |
| 3 | 35,5°C | 35,95°C | 0,45°C |
| 4 | 40,0°C | 39,75°C | -0,25°C |
| 5 | 39,8°C | 39,57°C | -0,23°C |
| 6 | 41,2°C | 41,06°C | 0,14°C |

Fonte: Autoria própria.

Os valores de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos coletados nos resultados de cada uma nas sete realizações de treinamento, foram descritos na Tabela 4. Os valores verdadeiros positivos indicam que o indivíduo tinha febre ou febrícula e a ferramenta detectou corretamente. Os

valores de falsos positivos indicam que o paciente não tinha febre ou febrícula e a ferramenta deu como resposta uma dessas opções. Os dados de verdadeiros positivos indicam que o indivíduo não tinha febre nem febrícula e a ferramenta assim detectou. Os dados de falsos negativos indicam os pacientes com estado de febre ou febrícula que a ferramenta classificou como normal.

Tabela 4 - Valores de resultados da ferramenta

| Relação de valores do resultado da ferramenta | | | | |
|------------------------------------------------------|-----------------------|------------------|-----------------------|------------------|
| Treinamento | Verdadeiros positivos | Falsos positivos | Verdadeiros negativos | Falsos negativos |
| 1 | 5 | 19 | 5.299 | 8 |
| 2 | 5 | 19 | 3.420 | 2 |
| 3 | 7 | 17 | 3.130 | 0 |
| 4 | 9 | 18 | 3.048 | 0 |
| 5 | 10 | 19 | 2.853 | 0 |
| 6 | 5 | 17 | 2.617 | 0 |
| 7 | 4 | 13 | 1.847 | 0 |

Fonte: A autoria própria.

Dentre os valores na Tabela 4, os dados de falsos negativos são melhores quando são menores. Isto dá-se pelo fato de que, se um indivíduo com uma doença contagiosa que possui febre como sintoma não é detectado, este pode espalhar esta doença para mais pessoas. Os valores de verdadeiro positivo e verdadeiro negativo apresentam que a ferramenta está classificando as temperaturas dos indivíduos com a imagem infravermelha conforme a classificação da sua temperatura real. Os valores de falsos positivos indicam que uma pessoa que não possui a doença possa vir a passar por uma triagem, igual os pacientes que têm a probabilidade de possuir a doença. Na Tabela 5 estão listados os valores de sensibilidade, especificidade e acurácia conforme os dados da Tabela 4.

Tabela 5 - Valores de sensibilidade, especificidade e acurácia

| Relação de valores do resultado da ferramenta | | | |
|------------------------------------------------------|---------------|----------------|----------|
| Treinamento | Sensibilidade | Especificidade | Acurácia |
| 1 | 38,46% | 99,64% | 99,49% |
| 2 | 71,43% | 99,45% | 99,39% |
| 3 | 100,00% | 99,46% | 99,46% |
| 4 | 100,00% | 99,41% | 99,41% |
| 5 | 100,00% | 99,34% | 99,34% |
| 6 | 100,00% | 99,35% | 99,36% |
| 7 | 100,00% | 99,30% | 99,30% |

Fonte: Autoria própria

Quanto mais elevado o valor da sensibilidade, mais chances o método tem de encontrar, em um grupo de suspeitos, um indivíduo com a patologia, no caso febre. Do mesmo modo, quanto mais alto o valor da acurácia, mais chances o método tem de acertar se o indivíduo possui ou não a patologia. E quanto maior o valor da especificidade mais chances o método tem de identificar um indivíduo sem a patologia em um grupo de sintomáticos. Destacam-se na Tabela 5 os valores a partir do 3º treinamento, pois são os dados que não possuem valores de falsos negativos e com isso, tiveram maior sensibilidade, com valores elevados de especificidade e acurácia.

Os resultados finais da ferramenta se dão por:

- Resultado correto: dá-se por exatamente uma detecção de face no termograma, e a classificação da temperatura encontrada na imagem infravermelha deve ser a mesma classificação de temperatura do valor axial.
- Resultado incorreto: é quando existem duas ou mais detecções no termograma, ou quando a classificação de temperatura da região detectada é diferente da classificação da temperatura do valor axial.
- Sem resultado: é quando não há nenhuma detecção facial, implicando em nenhuma classificação de temperatura.

Na Tabela 6 estão listados os resultados obtidos com imagens de voluntários que tiveram a o valor de temperatura axilar medido. Para tal, foram utilizadas 1.558 imagens infravermelhas, sendo estas, 3 de indivíduos com febre e 1558 de indivíduos com temperatura normal.

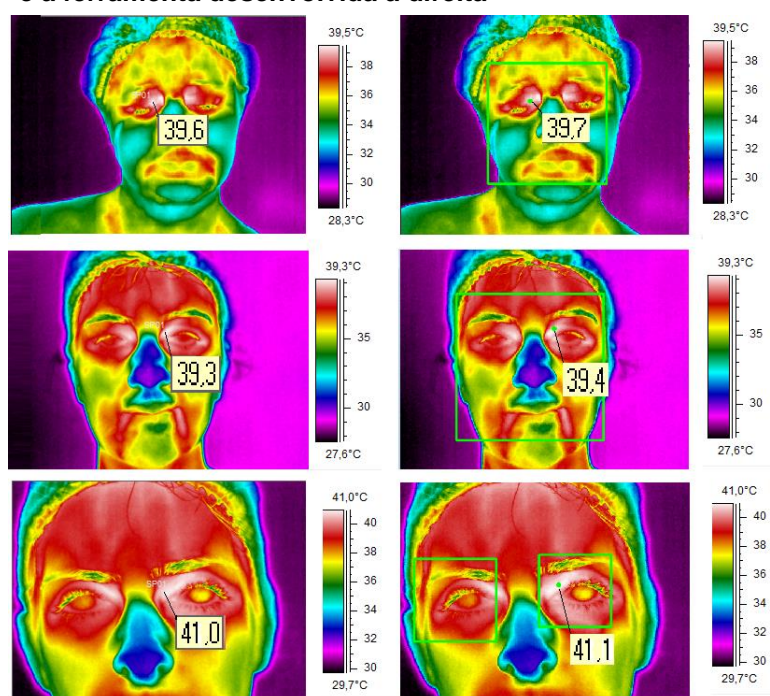
Tabela 6 - Valores de resposta da ferramenta em indivíduos

| Relação de resultados da ferramenta em indivíduos | | | | | | |
|---------------------------------------------------|-------------------|--------|---------------------|--------|---------------|-------|
| Estágios do treinamento | Resultado correto | | Resultado incorreto | | Sem resultado | |
| 5 | 121 | 7,77% | 1437 | 92,23% | 0 | 0,00% |
| 7 | 590 | 37,87% | 968 | 62,13% | 0 | 0,00% |
| 9 | 662 | 42,49% | 896 | 57,51% | 0 | 0,00% |
| 11 | 683 | 43,84% | 872 | 55,97% | 3 | 0,19% |
| 13 | 745 | 47,82% | 808 | 51,86% | 5 | 0,32% |
| 15 | 791 | 50,77% | 751 | 48,20% | 16 | 1,03% |
| 20 | 1146 | 73,56% | 340 | 21,82% | 72 | 4,62% |

Fonte: Autoria própria.

Ao verificar os dados da Tabela 6 nota-se que os percentuais de resultados corretos aumentavam quando a quantidade de estágios do treinamento aumentava. Porém, nos três últimos treinamentos, percebeu-se que os percentuais de sem resultado também cresciam conforme a quantidade de estágios do treinamento. A Figura 19 demonstra o comparativo das imagens dos indivíduos com febre realizado de forma manual no software TermaCAM Research (FLIR, 2000) com a ferramenta desenvolvida neste trabalho.

Figura 19 - Comparativo de resultado entre TermaCAM Research a esquerda e a ferramenta desenvolvida a direita



Fonte: Autoria própria.

A execução e os resultados da ferramenta tiveram um tempo de resposta rápido quando em comparação com o tempo de duração do treinamento, pois foram aplicadas 1.558 imagens e o tempo decorrido foi de aproximadamente dois minutos. Os maiores percentuais de verdadeiros positivos foram obtidos com um número maior número de estágios na etapa de treinamento e os menores percentuais de falsos negativos foram com obtidos com o treinamento utilizando uma menor quantidade de estágios. Isso dá-se pelo fato da detecção ser realizada pela rejeição em cascata, onde quanto mais treinado mais o algoritmo fica rigoroso ao aceitar as imagens, e quando menos treinado a rejeição é menor. Nos casos onde controlava-se o ambiente, a ferramenta conseguiu fornecer um resultado de classificação de febre com baixa taxa de erro utilizando métodos de coleta de imagens sem contato físico direto com o paciente.

Para uma melhor análise de desempenho de acerto na detecção e no resultado da ferramenta, foram unificados na Tabela 7, os principais dados da Tabela 1, Tabela 2, e foi dividido os dados da Tabela 6 em indivíduos com febre e sem febre. Com isso, pode-se visualizar o desempenho de acertos do treinamento e do resultado final da ferramenta pela quantidade de estágios e pela quantidade de nós da árvore de decisão gerada.

Tabela 7 - Comparativo com detecções e resultados corretos

| Comparação de dados de acerto de detecção e resultados | | | | | | | | |
|--------------------------------------------------------|-------------------------|-----------------------------|-----------------------------------|--------|---------------------|---------|-----------|--------|
| Treinamento | Estágios do treinamento | Quantidade de nós da árvore | Verdadeiros positivos na detecção | | Resultados corretos | | | |
| | | | | | Com febre | | Sem febre | |
| 1 | 5 | 4 | 1.380 | 25,89% | 0 | 0,00% | 121 | 7,78% |
| 2 | 7 | 6 | 1.475 | 42,80% | 2 | 66,67% | 588 | 37,81% |
| 3 | 9 | 8 | 1.556 | 49,33% | 2 | 66,67% | 660 | 42,44% |
| 4 | 11 | 9 | 1.542 | 50,15% | 2 | 66,67% | 681 | 43,79% |
| 5 | 13 | 11 | 1.535 | 53,24% | 3 | 100,00% | 742 | 47,72% |
| 6 | 15 | 13 | 1.519 | 57,52% | 3 | 100,00% | 788 | 50,68% |
| 7 | 20 | 17 | 1.482 | 79,51% | 3 | 100,00% | 1143 | 73,50% |

Fonte: Autoria própria.

O mesmo processo realizado na Tabela 7 foi desenvolvido na Tabela 8, no entanto os resultados apresentados, mostram os valores incorretos na detecção e no resultado da ferramenta a partir do treinamento e da quantidade de nós gerados.

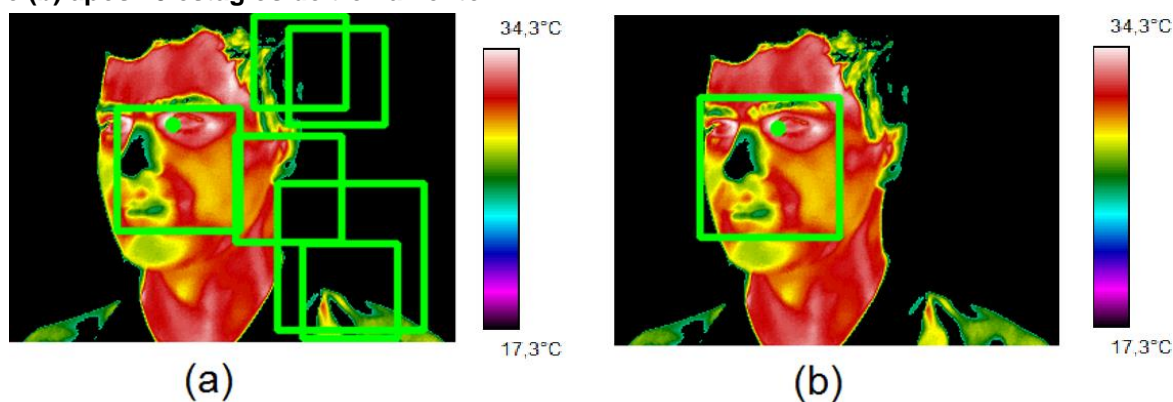
Tabela 8 - Comparativo com detecções e resultados incorretos

| Comparação de dados incorretos na detecção e nos resultados | | | | | | | | |
|-------------------------------------------------------------|-------------------------|-----------------------------|------------------------------|--------|-----------------------|---------|-----------|--------|
| Treinamento | Estágios do treinamento | Quantidade de nós da árvore | Falsos positivos na detecção | | Resultados incorretos | | | |
| | | | | | Com febre | | Sem febre | |
| 1 | 5 | 4 | 3.951 | 77,11% | 3 | 100,00% | 1434 | 92,22% |
| 2 | 7 | 6 | 1.971 | 57,20% | 1 | 33,33% | 967 | 62,19% |
| 3 | 9 | 8 | 1.598 | 56,67% | 1 | 33,33% | 895 | 57,56% |
| 4 | 11 | 9 | 1.533 | 49,85% | 1 | 33,33% | 871 | 56,01% |
| 5 | 13 | 11 | 1.348 | 46,76% | 0 | 0,00% | 808 | 51,96% |
| 6 | 15 | 13 | 1.122 | 42,48% | 0 | 0,00% | 751 | 48,30% |
| 7 | 20 | 17 | 382 | 20,49% | 0 | 0,00% | 340 | 21,86% |

Fonte: Autoria própria.

Para uma real aplicação da ferramenta, é necessário investigar qual treinamento utilizar, pois como observado, a sensibilidade da ferramenta diminui à medida que a detecção fica mais criteriosa. Também, nestes casos, quanto maior o percentual de falsos positivos, menor o percentual de falsos negativos, ou seja, quanto maior o descarte de regiões, maior a probabilidade de uma detecção ser de uma face. Porém, o custo computacional para localização da maior temperatura é menor quando tem-se menor quantidade de detecções, e a probabilidade de a ferramenta dar um resultado da região de fundo da imagem é menor. Na Figura 20 contém resultados de diferentes treinamentos, onde na Figura 20 (a) está o resultado de um treinamento com 5 estágios e na Figura 20 (b) o resultado com 15 estágios de treinamento.

Figura 20 - Resultados com diferentes estágios de treinamento. (a) detecção após 5 estágios e (b) após 15 estágios de treinamento.



Fonte: Autoria própria.

Contudo, para a correta escolha de qual tipo de treinamento utilizar, deve-se levar em conta se estão sendo controlados o ambiente, a temperatura e a distância

em que as imagens serão coletadas. Porém, o tempo de resposta e a qualidade das análises devem ser levados em consideração, pois numa aplicação real, a ferramenta pode servir como base para seleção de indivíduos.

5 CONCLUSÃO

Durante o desenvolvimento do trabalho, as aquisições das imagens infravermelhas seguiram o mesmo procedimento, e para todas estas não houve nenhum contato físico direto com o paciente. Esse método demonstra-se eficiente para a coleta de temperatura de pessoas, pois em caso de uma doença contagiosa, o profissional da saúde terá menos contato com o paciente. Também, o comparativo entre a temperatura coletada com o termômetro e a temperatura máxima da detecção realizada não teve grandes diferenças. Entretanto, a automatização do processo manual de transferência das imagens da câmera para a ferramenta realizar a análise e a classificação, economizaria mais tempo.

Antes da implementação da ferramenta que daria o resultado final, foram desenvolvidas ferramentas auxiliares, que ajudaram a deixar o processo de realização do treinamento mais rápido e com menos chance de falha humana. Tais ferramentas não tomaram muito tempo de desenvolvimento, e deram praticidade e eficiência na realização do treinamento. O maior tempo decorrido na execução do trabalho foi diante a realização do treinamento, uma vez que a resposta era afetada diretamente com a quantidade de imagens fornecidas e a diferença entre a quantidade de estágios. Porém, notou-se que quanto maior a quantidade de estágios do treinamento, maior é a probabilidade de uma detecção ser um verdadeiro positivo, entretanto, as imagens com menor quantidade de características são mais descartadas.

A escolha da biblioteca OpenCV 3.0 facilitou o processo de detecção facial que fora necessário para diminuir a taxa de diagnósticos em imagens onde não existiam pessoas. Essa facilidade deu-se pelo fato da detecção necessitar, basicamente, de uma imagem para realizar a detecção, atributos para tomadas de decisões e um arquivo gerado a partir de treinamento de máquina utilizando imagens. A realização do treinamento foi essencial para o desenvolvimento do trabalho, uma vez que a implementação da ferramenta era focada na coleta e classificação da temperatura de valor máximo, e não em um método para detectar faces. Outro fator que influenciou os novos treinamentos, foi o fato de os treinamentos de detecção facial, disponíveis na biblioteca OpenCV, não trazerem bom percentual de acerto quando aplicado em imagens infravermelhas.

O processo de implementação da ferramenta foi simples, uma vez que foram realizadas pesquisas específicas de como os algoritmos se comportariam e quais

seriam suas respostas. A maior problemática no desenvolvimento foi a escolha dos parâmetros para a detecção, que uma vez estabelecidos tinham que ser mantidos para a realização dos testes de todos os treinamentos. Entretanto o valor de cada parâmetro depende do tipo de imagem que está sendo utilizada e do objetivo de detecção que se deseja.

Todavia, o desenvolvimento desse trabalho demonstra uma metodologia de reconhecimento facial e uma alternativa para a aplicação de triagem em pessoas sem o contato físico. Na detecção facial, os resultados tiveram uma taxa de 79,51% de verdadeiros positivos, quando aplicado o treinamento de maior quantidade de estágios. Nos resultados de detecção de febre, as classificações tiveram percentual de acerto de 73,56%, no treinamento com maior número de estágios e de 7,77% no treinamento com menor quantidade de estágios.

5.1 TRABALHOS FUTUROS

A seguir são apresentadas algumas sugestões de trabalhos futuros:

- Realizar treinamentos com uma quantidade maior de imagens infravermelhas positivas e negativas de um número maior de voluntários;
- Realizar um gráfico comparativo, demonstrando a curva de aprendizado entre distintos treinamentos, baseando-se em conceitos de Inteligência Artificial;
- Comparar o desempenho de treinamentos utilizando outras formas de detecção;
- Detecção de febre no momento da aquisição da imagem pela câmera infravermelha.

REFERÊNCIAS

AMALU, William C. **A Review of Brest Thermography**. IAC, International Academy of Clinical Thermography. 2002.

BEZERRA, Teogénes E. **Reconhecimento de Faces com Imagens Infravermelhas**. 2010. 55f. Monografia (Bacharelado em Engenharia da Computação) - Escola Politécnica de Pernambuco. Universidade de Pernambuco. Recife, 2010.

BONNET, Pascale; DEFRENNE, Stéphanie; HAUSFATER, Pierre; ZHAO, Yan. RIOU, Bruno. **Cutaneous Infrared Thermometry for Detecting Febrile Patients**. Emerging Infectious Diseases n.8 v.14, No. 8, 2008.

CARVALHO, Araken R. **Mecanismo da Febre**. 2002. 37. Monografia (Licenciatura em Ciências Biológicas) - Centro Universitário de Brasília. Faculdade de Ciências da Saúde. Brasília, 2002.

DIAKIDES, N. A.; DIAKIDES, M.; LUPO, J. C.; PAUL, J. L.; BALCERAK, R. **Advances in Medical Infrared Imaging**, In: Medical Devices and Systems, The Biomedical Engineering Handbook, Edited by Joseph D. Bronzino, Third Edition, CRC Press, USA, 2006.

FLIR. ThermCam Researcher Operating Manual. 1. ed. Portland, USA. 2000-2001.

GONZALEZ, Rafael. C.; WOODS, Richard. E. **Processamento de imagens digitais**. 3ª edição. São Paulo: Pearson Education do Brasil, 2010. 624p.

GUIMARÃES, Maria. C. S. **Exames de Laboratório: Sensibilidade, Especificidade, Valor Preditivo Positivo**. Revista da Sociedade Brasileira de Medicina Tropical n.18 v. 2, pp117-120, 1985.

HADDAD, Denise S. **Estudo da Distribuição Térmica da Superfície Cutânea Facial por Meio de Termografia Infravermelha: Termoanatomia da Face**. 2014. 108f. Tese (Doutorado em Odontologia) – Universidade de São Paulo, Faculdade de Odontologia. São Paulo, 2014.

KAEHLER, Adrian; BRADSKI, Gary. **Learning OpenCV: Computer Vision with the OpenCV Library**. Edição 1. Sebastopol: O' Reilly Media, 2008. 556p.

LEITE, Diego O.; PRADO, Rogério J. **Espectroscopia no Infravermelho: Uma Apresentação para o Ensino Médio**. Revista Brasileira de Ensino de Física, n. 2, v. 34, pp 2504.1-2504.9, 2012.

NEVES, Sergio R. **Algoritmos para Segmentação de Imagens Infravermelhas**. 2003. 131f. Tese (Doutorado em Ciências) – Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2003.

OPENCV, OpenCV 3.0.0-dev-documentation. Disponível em <<http://docs.opencv.org/3.0-beta/modules/refman.html>>. Acesso em: 18 out. 2016.

PARKER, J. R. **Algorithms for Image Processing and Computer Vision**. Edição 2. Indianapolis: Wiley Publishing, 2011. 480p.

RING, E. F. J. **Progress in the Measurement of Human Body Temperature**. IEEE, Engineering in Medicine and Biology Magazine, v .17, n. 4, pp.19-24, 1998.

RING, E. F. J.; KALICKI, B.; ZUBER, J.; RUSTECKA, A.; VARDASCA, R. “New Standards for Fever Screening with Thermal Imaging Systems”, In: **Journal of Mechanics in Medicine and Biology**. Szaserow, Poland, p. 209-230, 2013.

RING, E. F. J.; JUNG, A.; ZUBER, J.; RUTOWSKI, P.; KALICKI, B.; BAJWA, U. “Detecting Fever in Polish Children by Infrared Thermography”, In: **9th International Conference on Quantitative Infrared Thermography**. Krakow, Poland, pp. 125-128, 2008.

SANCHES, Ionildo J. **Sobreposição de Imagens de Termografia e Ressonância Magnética: Uma Nova Modalidade de Imagem Médica Tridimensional**. 2009. 168f. Tese (Doutorado em Engenharia Biomédica) – Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2009.

SANCHES, I. J.; GAMBA, H. R.; SOUZA, M. A.; NEVES, E. B.; NOHAMA, P. **Fusão 3D de Imagens de MRI/CT e Termografia**, Revista Brasileira de Engenharia Biomédica, v. 29, n. 3, pp. 298-308, 2013.

SANCHES, Ionildo. J.; MORAIS, Erikson F.; BRIOSCHI, Marcos L.; FERREIRA, Jean H. Sistema de Detecção de Febre por Imagens. **In: XXV Congresso Brasileiro de Engenharia Biomédica (CBEB-2016)**. Foz do Iguaçu-PR. CBEB, Outubro, 2016.

VARELLA, Drauzio. Doenças e Sintomas: Febre. Publicado em 31 maio 2012. Disponível em <<http://drauziovarella.com.br/letras/f/febre>>. Acesso em: 19 set. 2015.

VIOLA, Paul; JONES, Michael J. Rapid Object Detection Using a Boosted Cascade of Simple Features. **In: Computer Vision and Pattern Recognition**. Kuai: CVPR, 2001, p.511-518. v.1.

VIOLA, Paul; JONES, Michael J. Robust Real-Time Face Detection. **In: International Journal of Computer Vision**. Netherlands: CVPR, 2004, pp.137-154. v.1.

APENDICE A - Formato de arquivo FPF

Nesta sessão será descrito o formato de arquivo *FPF* segundo o manual de operação do *ThermCam Research* (FLIR, 2000).

```
// Cabeçalho
typedef struct{
    FPF_IMAGE_DATA_T imgData;    // Estrutura da imagem
    FPF_CAMDATA_T    camData;    // Estrutura da câmera
    FPF_OBJECT_PAR_T objPar;     // Estrutura de parâmetros da imagem
    FPF_DATETIME_T   datetime;   // Estrutura de data da imagem
    FPF_SCALING_T    scaling;     // Estrutura de escala da imagem
    int              spareLong[32]; /* = 0 (long when the
                                   arquitetura is x32 or x86) */
} FPFHEADER_T;

// Estrutura da imagem
typedef struct{
    char          fpfID[32];    // "FLIR Public Image Format"
    unsigned int  version;      // = 2 */
    unsigned int  pixelOffset; /* Offset to pixel values from start of
                                fpfID.*/
    unsigned short ImageType; /* Temperature    = 0,
                               Diff Temp       = 2,
                               Object Signal   = 4,
                               Diff Object Signal = 5*/
    unsigned short pixelFormat; /* 0 = short integer = 2 bytes
                                1 = long integer   = 4 bytes
                                2 = float         = 4 bytes
                                3 = double        = 8 bytes */
    unsigned short xSize;
    unsigned short ySize;
    unsigned int   trig_count; /* external trig counter (long when the
                                arquitetura is x32 or x86)*/
    unsigned int   frame_count; /* frame number in sequence (long
                                when the arquitetura is x32 or x86)*/
    int            spareLong[16]; /* = 0 (long when the arquitetura is x32 or
                                x86)*/
} FPF_IMAGE_DATA_T;

// Estrutura da câmera
#define FPF_CAMERA_TYPE_LEN    31 // Camera name string
#define FPF_CAMERA_PARTN_LEN  31 // Camera part number string
#define FPF_CAMERA_SN_LEN     31 // Scanner serial number string
#define FPF_LENS_TYPE_LEN     31 // Lens name string
#define FPF_LENS_PARTN_LEN    31 // Lens part number string
#define FPF_LENS_SN_LEN       31 // Lens serial number string
#define FPF_FILTER_TYPE_LEN   31 // Filter name string
#define FPF_FILTER_PARTN_LEN  31 // Filter part number string
```

```

#define FPF_FILTER_SN_LEN      31 // Filter serial number string

typedef struct{
    char  camera_name[FPF_CAMERA_TYPE_LEN+1];
    char  camera_partn[FPF_CAMERA_PARTN_LEN+1];
    char  camera_sn[FPF_CAMERA_SN_LEN+1];
    float camera_range_tmin;
    float camera_range_tmax;
    char  lens_name[FPF_LENS_TYPE_LEN+1];
    char  lens_partn[FPF_LENS_PARTN_LEN+1];
    char  lens_sn[FPF_LENS_SN_LEN+1];
    char  filter_name[FPF_FILTER_TYPE_LEN+1];
    char  filter_partn[FPF_FILTER_PARTN_LEN+1];
    char  filter_sn[FPF_FILTER_SN_LEN+1];
    int   spareLong[16]; /* = 0 (long when using x86)*/
} FPF_CAMDATA_T;

// Estrutura de parâmetros da imagem
typedef struct{
    float emissivity;      // 0 - 1
    float objectDistance; // Meters
    float rflTemp;        // Reflected temperatura in Kelvin. Version = 2
    //float ambTemp;      // Ambient temperature in Kelvin. Version = 1
    float atmTemp;        // Atmospheric temperature in Kelvin
    float relHum;         // 0 - 1 */
    float compuTao;       // Computed atmospheric transmission 0 - 1
    float estimTao;       // Estimated atmospheric transmission 0 - 1
    float refTemp;        // Reference temperature in Kelvin
    float extOptTemp;     // Kelvin
    float extOptTrans;    // 0 - 1
    int   spareLong[16]; // = 0 (long when using x86)
} FPF_OBJECT_PAR_T;

// Estrutura de data da imagem
typedef struct{
    int Year;
    int Month;
    int Day;
    int Hour;
    int Minute;
    int Second;
    int MilliSecond;
    int spareLong[16]; // = 0 (long when using x86)
} FPF_DATETIME_T;

```

```
// Estrutura de escala da imagem
typedef struct{
float tMinCam;      // Camera scale min, in current output
float tMaxCam;      // Camera scale max
float tMinCalc;     // Calculated min (almost true min)
float tMaxCalc;     // Calculated max (almost true max)
float tMinScale;    // Scale min
float tMaxScale;    // Scale max
int  spareLong[16]; /* = 0 (long when the architecture is x32 or
                    x86) */
} FPE_SCALING_T;
```