

**CORPO DE BOMBEIROS MILITAR DO DISTRITO FEDERAL
DEPARTAMENTO DE ENSINO, PESQUISA, CIÊNCIA E TECNOLOGIA
DIRETORIA DE ENSINO
ACADEMIA DE BOMBEIRO MILITAR
Coronel Osmar Alves Pinheiro
CURSO DE FORMAÇÃO DE OFICIAIS**

Cadete BM/2 **JAPHY MONTEIRO LIMA MIRANDA**



**O USO DA VISÃO COMPUTACIONAL NO PROCESSO DE
DETECÇÃO DE INCÊNDIOS FLORESTAIS**

BRASÍLIA
2021

Cadete BM/2 **JAPHY MONTEIRO LIMA MIRANDA**

O USO DA VISÃO COMPUTACIONAL NO PROCESSO DE DETECÇÃO DE INCÊNDIOS FLORESTAIS

Memorial descritivo apresentado à disciplina Trabalho de Conclusão de Curso como requisito para conclusão do Curso de Formação de Oficiais do Corpo de Bombeiros Militar do Distrito Federal.

Orientador: Major. QOBM/Comb. PABLO FEDERICO **BAIGORRI**

BRASÍLIA
2021

Cadete BM/2 **JAPHY MONTEIRO LIMA MIRANDA**

**O USO DA VISÃO COMPUTACIONAL NO PROCESSO DE DETECÇÃO DE
INCÊNDIOS FLORESTAIS**

Memorial descritivo apresentado à disciplina Trabalho de Conclusão de Curso como requisito para conclusão do Curso de Formação de Oficiais do Corpo de Bombeiros Militar do Distrito Federal.

Aprovado em: ____/____/____.

BANCA EXAMINADORA

**ALBERTO WESLEY DOURADO DE SOUZA – Ten-Cel. QOBM/Comb.
Presidente**

**ZILTA DIAZ PENNA MARINHO - Professora
Membro**

**VINÍCIUS FIUZA DUMAS – Maj. QOBM/Comb.
Membro**

**PABLO FEDERICO BAIGORRI - Major. QOBM/Comb
.Orientador**

RESUMO

O presente trabalho tem como tema a utilização de visão computacional para auxiliar na detecção de focos de incêndios florestais. A visão computacional é um campo da inteligência artificial que treina computadores para interpretar e compreender o mundo visual, replicando partes da complexidade do sistema de visão humana e permitindo que os computadores identifiquem e processem objetos em imagens e vídeos da mesma forma que os humanos fazem. Na última década observou-se o uso de visão computacional para detecção e medição de incêndios florestais. O objetivo central do trabalho foi analisar o uso da visão computacional no processo de detecção de incêndios florestais com o uso de câmeras. Este propósito foi alcançado mediante revisão bibliográfica sobre visão computacional e sua aplicação a incêndios florestais, além de levantar os principais problemas existentes e saná-los com a implantação do sistema. A partir de um modelo desenvolvido foi possível encontrar o limiar de detecção do fogo utilizando trabalhos relacionados para realizar os testes e assim detectar focos de incêndio florestal. Pretende-se com essa iniciativa tornar o trabalho do supervisor ambiental mais eficiente e auxiliá-lo na tomada de decisões. O trabalho também inclui alguns resultados de experimentos com incêndios florestais.

Palavras-chave: Visão Computacional. Incêndios Florestais. Processamento de Imagem. Detecção por imagem.

THE USE OF COMPUTATIONAL VISION IN THE FOREST FIRE DETECTION PROCESS

ABSTRACT

The present paper has as its theme the use of computer vision to assist in the detection of forest fire outbreaks. Computer vision is a field of artificial intelligence that trains computers to interpret and understand the visual world, replicating parts of complexity of the human vision system and allowing computers to identify and process objects in images and videos just as humans do. In the last decade, the use of computer vision for detecting and measuring forest fires has been observed. The main objective of the work is to analyze the use of computer vision in the process of detecting forest fires using cameras. This purpose will be achieved through a literature review on computer vision and its application to forest fires, in addition to raising the main existing problems and remedying them with the implementation of the system. From a developed model it was possible to find the fire detection threshold using related works to perform the tests and thus detect forest fire outbreaks. This initiative is intended to make the work of the environmental supervisor more efficient and assist him in decision making. The paper also includes some results of experiments with forest fires.

Keywords: *Computer vision. Forest Fires. Image Processing. Detection by image*

1. INTRODUÇÃO

Muitas vezes as mídias nacionais e internacionais têm noticiado inúmeros incêndios florestais de diversas proporções. No Distrito Federal o Corpo de Bombeiros atende milhares de ocorrências deste tipo por ano.

Segundo Estevam (2020) o incêndio florestal pode ser definido como um fogo sem controle que se propaga consumindo os combustíveis naturais representando uma grande preocupação aos ecossistemas naturais e/ou artificiais, todavia o fogo desempenha um importante papel na manutenção de alguns ecossistemas e sua ocorrência de forma descontrolada pode representar uma fonte de perturbação permanente, acarretando perdas e danos materiais (NUNES, 2005).

Os incêndios florestais trazem grandes prejuízos ao Distrito Federal a cada ano, por sua característica de longos períodos de estiagem com baixa umidade e elevada temperatura, somando a uma vegetação que favorece a disseminação de incêndios, mobiliza uma grande soma de esforços e recursos do setor público nas operações de prevenção e combate (LAGARES, 2007, p.7).

A rápida detecção de incêndios é uma forma bastante eficiente de proteção contra incêndios porque possibilita seu combate logo no início. Contudo, atualmente, a detecção de incêndios florestais é um processo moderadamente caro e em alguns casos é ineficaz (SEPIR, 2015). As técnicas atuais de detecção utilizam a rede de satélites com diferentes sensores ópticos e observadores humanos.

Uma técnica de rápida detecção é a utilização da visão computacional, que pode ser definida como a extração de informação de uma imagem ou vídeo relevantes para a solução de problemas, não é mais vista como um problema isolado de análise de imagens, e sim como um método de interação entre máquina e meio ambiente obtendo informações úteis sobre os objetos e áreas que estão em um mundo físico real (CHI YUAN, 2017).

A chama e a fumaça presente em um incêndio florestal são as principais características para o seu reconhecimento visual. Os meios de detecção automática de incêndios são desenvolvidos para encontrar estas características nas imagens obtidas. Em um contexto florestal, a propagação de fumaça pelo ar

é um dos primeiros indicadores visuais da existência de um incêndio e pode ser usada como um ponto de partida para uma detecção mais rápida (ALVES, 2018). No entanto, em comparação com a chama, as características visuais da fumaça, como a cor e o gradiente, são menos perceptíveis, o que torna mais difícil a diferenciação de situações semelhantes. A chama possui uma gama de cores mais perceptíveis em relação ao plano de fundo, que vai desde o amarelo, laranja até ao vermelho.

A importância do trabalho se dá pelo grave problema ambiental que constituem os incêndios florestais, cujas consequências comprometem o equilíbrio dos fatores que dão suporte à vida. As consequências da ação do fogo sem controle são numerosas, comprometendo a vegetação, a fauna, o solo, os cursos d'água, o ar atmosférico, a população e a biodiversidade. Um impacto muito visível dos incêndios florestais está relacionado à poluição e à qualidade do ar. A fumaça e as cinzas produzidas durante as queimadas contêm partículas muito pequenas, causando graves problemas sociais, econômicos, ambientais e problemas de saúde como: a inflamação nas vias aéreas superiores, asma e até câncer.

Por fim, o Corpo de Bombeiros Militar do Distrito Federal (CBMDF) traz em seu Plano Estratégico 2017-2024 com o objetivo 10 no tema inovação, “Desenvolver pesquisas e a gestão do conhecimento, no qual a estratégia consiste em realizar pesquisas na área operacional bombeiro militar”, sendo necessário a busca por ações com vistas à obtenção de produtos e processos, que permitam o desenvolvimento da instituição (CBMDF, 2016).

Com base no que foi exposto, qual é a eficácia e a eficiência do uso da visão computacional na detecção de incêndios florestais?

Desta forma este estudo com fins de mitigar os incêndios florestais no seu início tem como objetivo analisar a efetividade do uso da visão computacional no processo de detecção de incêndios florestais, contribuindo para a Corporação combatê-los com eficiência, evitando assim os problemas citados. Além deste tem-se também outros como: levantar os recursos empregados no combate e os históricos de incêndios florestais atendidos no CBMDF, apresentar o conceito de visão computacional, implementar o algoritmo de detecção de incêndios

florestais usando a visão computacional e por último avaliar o funcionamento do algoritmo implementado por meio de testes e comparações com modelos existentes.

A metodologia da pesquisa se classifica como método dedutivo e de natureza aplicada, pois a partir dos estudos sobre a visão computacional foi apresentada uma forma de solução específica para detecção de incêndios florestais. É de abordagem quali-quantitativa porque tem a finalidade de analisar, em valores numéricos, a quantidade de militares e viaturas, a quantidade de lançamentos e uso de aeronaves, que foram empregados na Operação Verde Vivo de 2020 com o intuito de justificar o trabalho e analisar as vantagens que o processo de detecção de incêndios traz para o melhor atendimento do CBMDF a ocorrências de incêndios florestais.

Para a consecução deste trabalho foi realizada uma pesquisa bibliográfica exploratória, para obter dados necessários para o estudo, segundo Gil (2002), por se utilizar de uma revisão bibliográfica visando elencar o que vem sendo escrito a respeito dos problemas envolvendo visão computacional.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. Incêndio Florestal

Os incêndios florestais podem iniciar-se de forma espontânea ou por ações humanas, fazem mais do que simplesmente reduzir a cobertura vegetal, atuam na mudança dos microclimas, além de causar danos ambientais, materiais e humanos.

Segundo Araújo (2019) o incêndio florestal ocorre quando alguma ameaça atua em um ambiente que se encontra vulnerável, em resumo acontece quando combustíveis inflamáveis de uma região florestal são expostos a uma fonte de calor.

A maioria das ocorrências de incêndios florestais acontecem quando se utiliza o fogo como ferramenta de manejo, pela agricultura de corte e pecuária extensiva, ou seja, resultantes de ações humanas de uso e ocupação de terra, onde o domínio do fogo escapa alcançando a cobertura vegetal mais propensa (LIMA, 2017).

Figura 1 – Bombeiros Militares do CBMDF em combate ao incêndio florestal



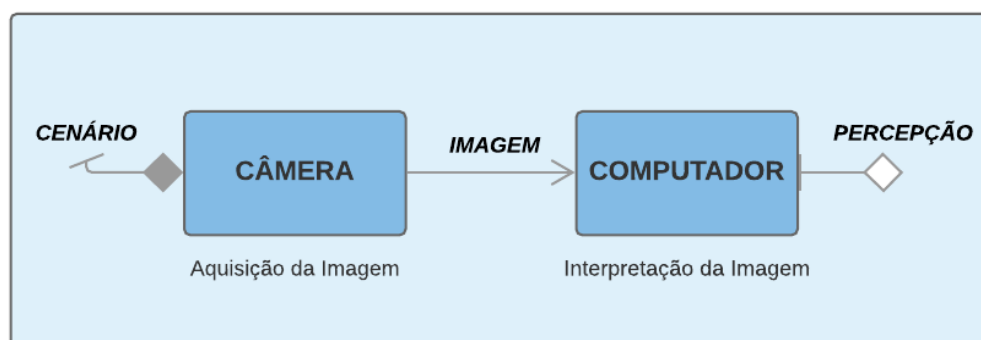
Fonte: PORTAL G1 (2021)

2.2. Visão Computacional

A visão computacional é um campo de estudo dentro da área da computação que trabalha na operação de extração de informação útil a partir de imagens. Sua função é fazer com que computador “perceba” o ambiente, permitindo que o computador realize tarefas com base no que está sendo “visto” (CHI YUAN, 2017).

Segundo SZELISKI (2010) a visão computacional pode ser definida como a área da ciência que estuda e procura desenvolver teorias e métodos em que as informações úteis contidas em imagens são extraídas automaticamente, seguindo algumas etapas que preparam a imagem capturada para interpretação do computador.

Figura 3 – Representação do Sistema de Visão Computacional



Fonte: o autor

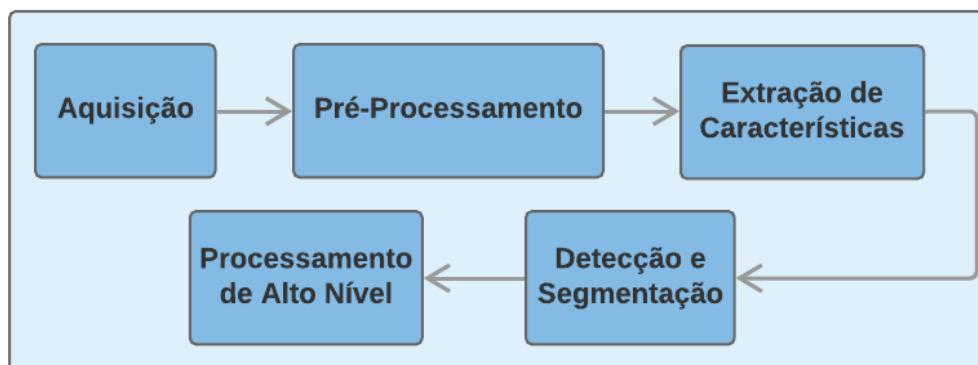
2.3. Processamento de Imagens

O processamento de imagens é a área da computação que se refere à manipulação e operações com imagens possuindo o foco na interpretação da extração de informações, novas representações e conclusões feitas pela Visão Computacional, processando e transformando dados das imagens de forma inteligente e interpretativa (COUTO, 2012).

Segundo Morris (2004) a partir da imagem capturada é feito o processamento digital da imagem compreendendo as seguintes etapas:

- Aquisição de imagem;
- Pré-processamento;
- Extração de características;
- Detecção e segmentação;
- Processamento de alto nível.

Figura 4 – Representação das etapas do Processamento de Imagens



Fonte: o autor

As etapas do processamento de imagens são descritas a seguir:

Aquisição de imagem: todas as imagens digitais são criadas por um ou mais sensores. De acordo com cada sensor pode-se ter vários tipos de imagens, como bidimensional, tridimensional ou ainda uma sequência de imagens. Cada pixel tem um valor agregado e cada valor é responsável por uma intensidade de luz em uma ou em várias faixas de cor.

Pré-processamento: antes da aplicação de qualquer método de visão computacional é necessário verificar se a imagem satisfaz as condições do método a ser utilizado. Para isso podemos citar alguns exemplos: remapeamento (para assegurar o sistema de coordenadas), redução de ruídos (para assegurar que as informações são verdadeiras) e aumento de contraste (para assegurar que as informações relevantes serão detectadas).

Extração de características: existem várias características matemáticas que se pode obter em uma imagem em diversos níveis de complexidade. Como exemplo básico inclui-se a detecção de bordas, cantos ou pontos. Exemplos mais complexos incluem detecção de textura, formato e movimento.

Detecção e segmentação: durante uma parte do processo sempre se toma uma determinada decisão em relação às regiões de interesse que serão processadas no futuro. Como exemplo citamos algumas: seleção de regiões de interesse específicas e segmentação de uma ou mais regiões que contém um objeto de interesse.

Processamento de alto nível: nesta etapa a entrada é definida como um conjunto pequeno de dados, ou seja, no caso de monitoramento de

foco de incêndio, o dado é apenas o objeto de interesse já detectado (MORRIS, 2004 apud SILVA, 2008, p.14).

Na etapa da segmentação, que é o processo de dividir uma imagem digital em múltiplas regiões (conjunto de *pixels*) ou objetos, com o objetivo de simplificar ou mudar a representação de uma imagem para facilitar a sua análise, cabe ainda discutir que são selecionadas as partes destaques com base na extração de características que possam ser usadas para distinguir as regiões de um objeto (FERNANDES, 2020). O processo de extração é denominado descrição e a estrutura que permite armazenar e manipular essa seleção é chamada de representação (SZELISKI, 2010).

2.4. Sistema de Cores

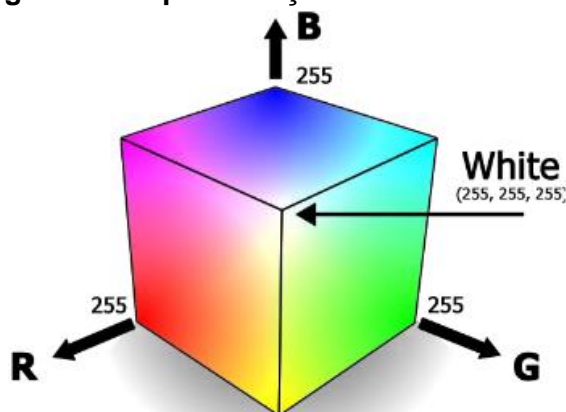
A escolha de um sistema de cores para a identificação de objetos em imagens é a etapa de maior importância para o sucesso do processamento de imagens facilitando seu reconhecimento em um determinado padrão. Existem diferentes tipos, nos quais destacam-se o RGB (*Red, Green, Blue*) e o HSV (*Hue, Saturation e Value*) no processamento digital de imagens (COUTO, 2012).

2.4.1. Sistema RGB

O Sistema RGB é um modelo aditivo no qual uma imagem consiste em três planos ou canais de imagem independentes: vermelho, verde e azul e cada cor obtida corresponde ao somatório de suas intensidades. Neste sistema, o branco resulta da soma dos valores máximos de cada componente, o preto da soma dos mínimos e um nível de cinza dos outros casos em que as três componentes tomam o mesmo valor (GARCÍA *et al*, 2015).

Para especificar uma cor particular, cada valor indica a quantidade de cada um dos componentes presentes em cada pixel, com valores mais altos correspondendo a *pixels* mais brilhantes (FERNANDES, 2020). Este sistema de cores é o mais comum e amplamente utilizado porque baseia-se na fisiologia da visão humana.

Figura 5 – Representação do sistema RGB



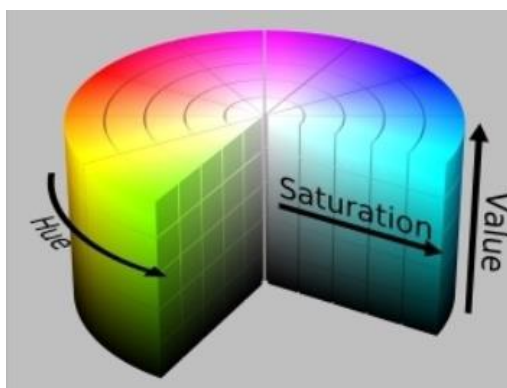
Fonte: CUNHA (2016)

2.4.2. Sistema HSV

O sistema de cores HSV traduzido para o português significa Matiz, Saturação e Valor respectivamente. O sistema HSV procura delimitar o sistema de cores de forma mais clara e divide o valor do *pixel* nas suas componentes. A matiz representa a informação básica da cor propriamente dita, saturação é uma medida de quão pura essa cor é, quanto menor a saturação mais clara ou mais branca essa cor será. A componente valor está relacionada à quantidade de brilho ou luminância (GARCÍA *et al*, 2015).

Devido a uma fraca correlação entre as suas componentes, o sistema HSV é uma opção bastante oportuna no momento em que se procura independência a variações de contraste para isolar uma cor, ou isolar um nível de iluminação, por exemplo (COUTO, 2012).

Este tipo de modelo de cor emula de perto os modelos de percepção de cor humana. Enquanto em outros modelos de cor, como RGB, uma imagem é tratada como um resultado aditivo de três cores de base.

Figura 6 - Representação do sistema HSV

Fonte: GARCÍA (2015)

Conforme é demonstrado na figura acima a matiz varia da faixa de 0 a 360, a saturação e o brilho variam entre 0 e 100%. Para a programação utilizando a biblioteca OpenCV, a matiz varia entre 0 e 180 para que seja possível utilizar uma variável de 8 bits, e saturação e brilho possuem uma faixa de variação entre 0 e 255 (GARCÍA *et al*, 2015).

2.5. Microcontrolador

O microcontrolador muitas vezes é confundido com “microprocessador”, talvez pela semelhança de nomes. Ele é um circuito integrado compacto projetado para comandar uma operação específica em um sistema embarcado. Um microcontrolador típico inclui um processador, memória e periféricos de entrada e saída em um único chip. Em resumo, ele nada mais é do que um computador pequeno capaz de realizar determinadas tarefas de maneira eficaz e com a vantagem de possuir tamanho compacto (KIM, 2018).

2.6. Câmera OV2640 – Imagem Digital

A câmera possui um sensor de imagem OV2640 sendo um dispositivo CMOS (*Complementary metal-oxide-semiconductor*) de chip único e processador de imagem em um pacote compacto. O OV2640 oferece imagens *full-frame*, *sub-sample*, dimensionadas ou em janela de 8/10 *bits* em uma ampla gama de formatos, controlados por meio da interface *Serial Camera Control Bus* (SCCB) (OMNIVISION, 2021).

Este produto tem uma matriz de imagens capaz de operar em até 15 quadros por segundo (fps) com total controle do usuário sobre a qualidade da imagem, formatação e transferência de dados de saída. Todas as funções de processamento de imagem necessárias, incluindo controle de exposição, gama, equilíbrio de branco, saturação de cores e muito mais, também são programáveis por meio da interface SCCB (OMNIVISION, 2021).

Figura 7 – Câmera OV2640 junto com o Módulo ESP32- CAM



Fonte: DFROBOT (2021)

2.7. Módulo ESP32-CAM

É um microcontrolador ESP32 de desenvolvimento com uma câmera *wi-fi* embutida de baixo custo, permitindo a criação de projetos de câmeras IP para *streaming* de vídeo com diferentes resoluções. O módulo suporta as câmeras OV2640 e OV7640. Tem flash embutido, *CPU* de 32 bits de baixa potência, *clockspeed* de 160 MHz, vários modos de hibernação embutidos e também com SRAM. É composto por pinos de configuração e por *GPIOs* (*General Purpose Input / Output*), interface padrão de entrada e saída usada para conectar periféricos a ESP32-CAM (DFROBOT, 2021).

O ESP32-CAM pode ser amplamente utilizado em várias aplicações de IoT (*Internet of Things*). É adequado para dispositivos domésticos inteligentes, controle sem fio industrial, monitoramento sem fio, identificação sem fio, sinais de sistema de posicionamento sem fio e outras aplicações. A placa não vem com um conector USB, então para fazer o carregamento do código de programação é necessário o módulo FTDI por meio dos pinos U0R e U0T (pinos seriais) (DFROBOT, 2021).

Figura 8 – Módulo ESP32 - CAM



Fonte: Studio Peters (PETERS, 2021)

A figura 9 mostra os componentes da placa ESP32-CAM de ambos os lados (DFROBOT, 2021).

Chip ESP32-S: O módulo é um chip principal que contém duas CPUs LX6 de 32 bits de alto desempenho com uma arquitetura de pipeline de 7 estágios e usado para todo o processamento e funcionamento.

Bloco Antena: O IPEX impresso conecta antenas GSM para transmitir sinais.

Capacitor Tantalum: O capacitor de tantalum é usado principalmente em módulos de tamanho pequeno. Eles são duráveis e fornecem filtragem de fonte de alimentação para a quantidade de sinal ideal.

Botão Reset: Ao ser pressionado, o botão Reset reinicia o código executado no módulo.

Chip regulador de tensão: O chip regulador de tensão no módulo mantém a tensão de saída apesar das flutuações na alimentação de entrada. Ele regula a tensão para 3,3 volts.

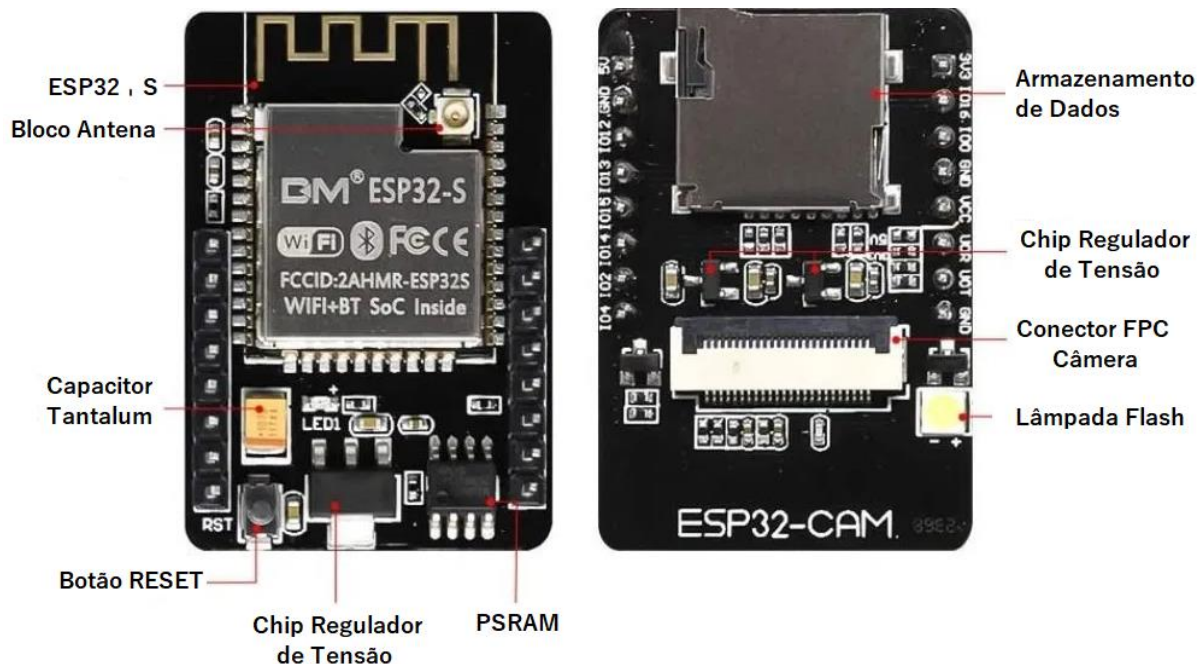
PSRAM: Uma memória de acesso pseudoaleatório de baixo consumo de 4MB está incorporada no módulo para processamento rápido das instruções. Isso ajuda a câmera a funcionar sem problemas.

Suporte para cartão de memória: a série ESP32 é incorporada com um suporte para cartão micro-SD para armazenar os dados. Toda a transmissão ocorre por meio da Interface Periférica Serial.

Conector FPC: para montar a câmera, o módulo ESP32 contém um conector de circuito impresso flexível. A conexão é precisa e responsável pela confiabilidade do sinal.

Lâmpada Flash: A lâmpada do flash produz pulsos elétricos que funcionam como um flash para a câmera para que ela possa capturar imagens nítidas.

Figura 9 – Características e Especificações Gerais



Fonte: O Autor, Adaptado DFROBOT (2021)

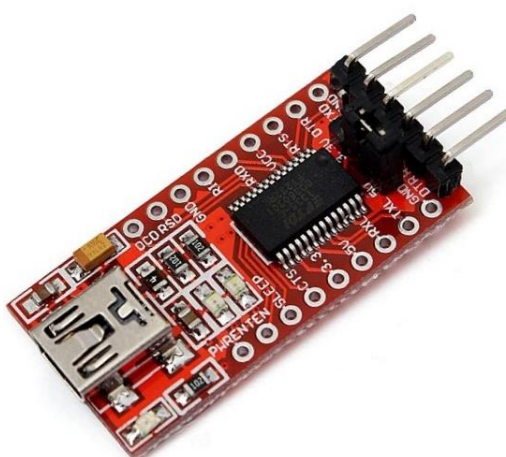
2.8. Módulo FTDI *Programmer* - Conversor UART - USB

A *Future Technology Devices International Limited*, comumente conhecida por sua sigla FTDI, é uma empresa escocesa de dispositivos semicondutores de capital fechado, especializada em tecnologia *Universal Serial Bus* (USB). Ela desenvolve, fabrica e oferece suporte a dispositivos e seus cabos relacionados e drivers de software para a conversão de transmissões seriais RS-232 ou TTL para sinais USB, a fim de fornecer suporte para dispositivos eletrônicos com computadores modernos, em resumo, o módulo FTDI é utilizado para fazer a comunicação de um circuito eletrônico com o computador via porta USB (FILIPEFLOP, 2021).

2.8.1. O conversor UART – USB

O módulo conversor é utilizado para comunicação entre o computador e dispositivos como microcontroladores, módulos e outros equipamentos que utilizam comunicação via interface serial. Ele usa seus pinos TXD e RXD para recepção e transmissão de dados recebidos pela USB. A energia de entrada pode ser feita com tensões de 3,3V a 5,25V, geralmente para isso é utilizada a tensão disponível pela porta USB (5V). O chip possui um pino nomeado VCCIO, que vai compatibilizar o nível de tensão nos pinos TX e RX com o microcontrolador, que geralmente usa 5V ou 3,3V (JOSEMAR, 2015).

Figura 10 – Módulo FTDI - Chip FT232R



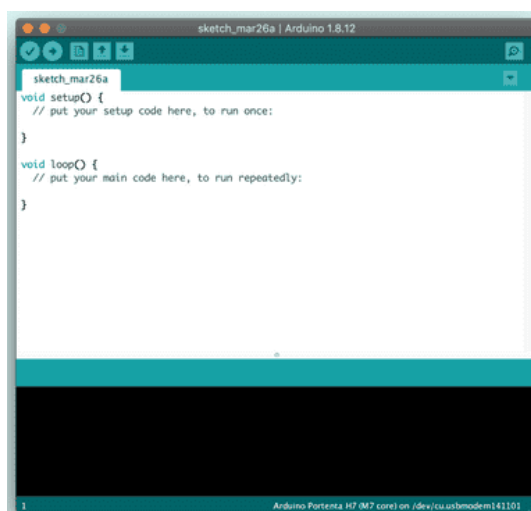
Fonte: FILIPEFLOP (2021)

2.9. Arduino Software IDE

O Arduino Software IDE (*Integrated Development Environment*) de código aberto facilita a escrita de código e o *upload* para a placa. É nele que o desenvolvedor fará a programação do algoritmo desejado. Este software pode ser usado com qualquer placa Arduino, além de auxiliar na implementação e no carregamento de códigos para outras placas (ARDUINO, 2021).

O Arduino IDE contém um editor de texto para escrever código, uma área de mensagem, um console de texto, uma barra de ferramentas com botões para funções comuns. Ele se conecta ao hardware Arduino para fazer *upload* de programas e se comunicar com eles (ARDUINO, 2021).

Figura 11 – O ambiente de desenvolvimento integrado do Arduino



Fonte: ARDUINO (2021)

2.10. Programação

Programação é o processo de criação de um conjunto de instruções que informam a um computador como realizar uma tarefa, em resumo é como se fosse uma “receita de bolo” que o computador deve seguir. A programação pode ser feita usando uma variedade de linguagens de programação de computador, como C, C++, Java, Python, entre outras.

As linguagens de programação são um tipo de padrão de codificação e são usadas na programação de computadores para implementar algoritmos. A utilizada na plataforma Arduino IDE suporta as linguagens C e C++ usando regras especiais de estruturação de código, que podem ser divididas em três

partes principais: estruturas, valores (variáveis e constantes) e funções (ARDUINO, 2021).

O Editor de texto fornece uma biblioteca de software do projeto, que fornece muitos procedimentos comuns de entrada e saída. O usuário precisa compreender as duas funções básicas para escrever o código, para iniciar o esboço e o loop do programa principal, que são compilados e vinculados a uma função do programa chamada de “*main*” em um programa executivo cíclico executável com a cadeia de ferramentas, também incluído no editor de texto Arduino IDE (ARDUINO, 2021).

2.11. Biblioteca OpenCV

A OpenCV (*Open Source Computer Vision Library*) é uma biblioteca de visão computacional de código aberto que inclui centenas de algoritmos, distribuída sob a licença BSD (*Berkeley Software Distribution*), isto quer dizer que pode ser utilizada e alterada sem a necessidade da aquisição de uma licença. A OpenCV foi criada com o intuito de acelerar o uso da visão computacional em produtos comerciais, fornecendo uma infraestrutura simples para a criação de aplicações em visão computacional de maneira rápida e eficiente (OPENCV, 2021).

Sua estrutura é modular, o que significa que o pacote inclui várias bibliotecas compartilhadas ou estáticas (OPENCV, 2021). Dentre as aplicações mais utilizadas tem-se a identificação de bordas, segmentação de imagem, reconhecimento de rostos e gestos, detecção de movimento entre outros (GARCÍA *et al*, 2015).

Desde seu lançamento em janeiro de 1999, a OpenCV tem sido usada em muitas aplicações, produtos e esforços de pesquisa. Estas aplicações incluem a união de imagens em mapas via satélite e web, alinhamento de escaneamento de imagens, redução de ruídos em imagens médicas, análise de objetos, segurança e sistemas de detecção de intrusão, monitoramento automático e sistemas de segurança, sistemas de inspeção de fabricação, calibração de câmeras, aplicações militares, veículos aéreos não tripulados, terrestres e submarinos (BRADSKI e KAEHLER, 2008, p. 2).

3. PROCESSO DE CONSTRUÇÃO

Este capítulo descreve as etapas de construção do módulo de captura, sendo o elemento primário de obtenção de imagens. Este é formado pelo seu hardware de captura (câmeras, dispositivo de interface e placa controladora) e por algoritmos de controle (configurador de interface, transmissão de dados, decodificação de imagens).

Para atingir o primeiro objetivo específico deste trabalho explorou-se inicialmente as informações existentes nos relatórios do Grupamento de Proteção Ambiental (GPRAM), procurando informações a fim de justificar a execução do projeto. Essas foram obtidas por meio de memorando e Relatório da Operação Verde Vivo de 2020, destacam-se entre elas:

1. Histórico anual do número de incêndios florestais no DF a partir de 2010
2. Dados estatísticos da Operação Verde Vivo 2020:
 - Quantidade de Militares empregados;
 - Quantidade de Viaturas empregadas;
 - Área queimada em hectares;

Dando sequência para a conclusão dos objetivos específicos, foi realizada a busca de referências bibliográficas que deram suporte científico e técnico referente à visão computacional e biblioteca OpenCV, que são o alicerce do desenvolvimento deste projeto.

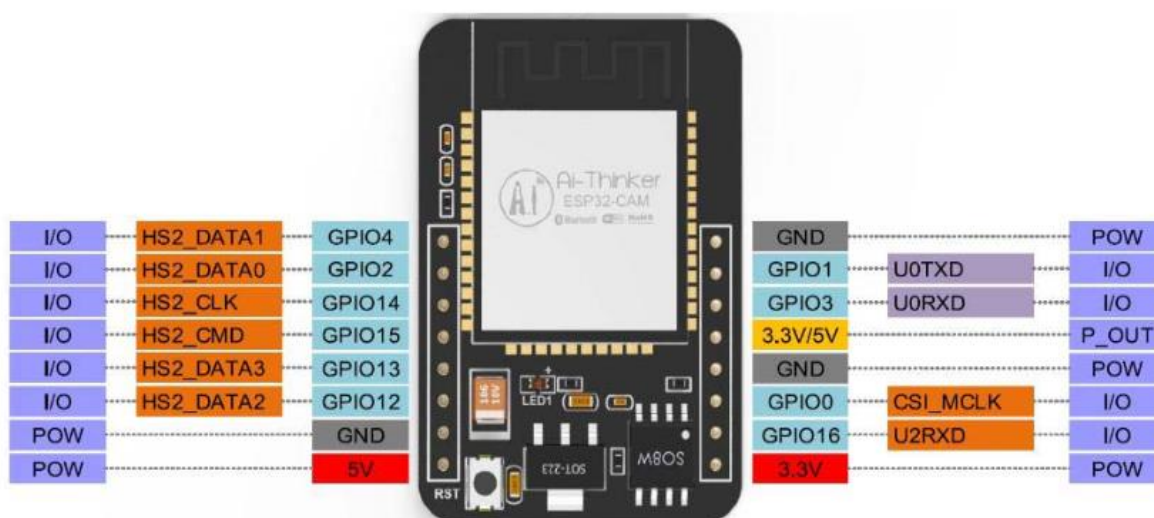
Para o melhor entendimento do objetivo principal que é identificar focos de incêndios florestais usando visão computacional foi necessário a construção de um protótipo de captura de imagens utilizando uma câmera e do desenvolvimento de um software composto de conjunto de algoritmos, responsável pelo ajuste de parâmetros de captura das imagens, segmentação dos dados e detecção de objetos.

A apresentação do processo de construção foi dividida em três etapas denominadas: Montagem do produto; Programação e Testes.

3.1. Montagem do Produto

A construção da parte física (*hardware*) do produto foi baseada na utilização da plataforma ESP32-CAM de projetos de IoT. Ela é uma placa de desenvolvimento com um chip ESP32-S, uma câmera OV2640 e vários GPIOs, que são pinos de conexão, para conectar a outros dispositivos. A figura abaixo mostra o diagrama da pinagem do módulo.

Figura 12 – Diagrama de Pinagem do ESP32-CAM



Fonte: DFROBOT (2021)

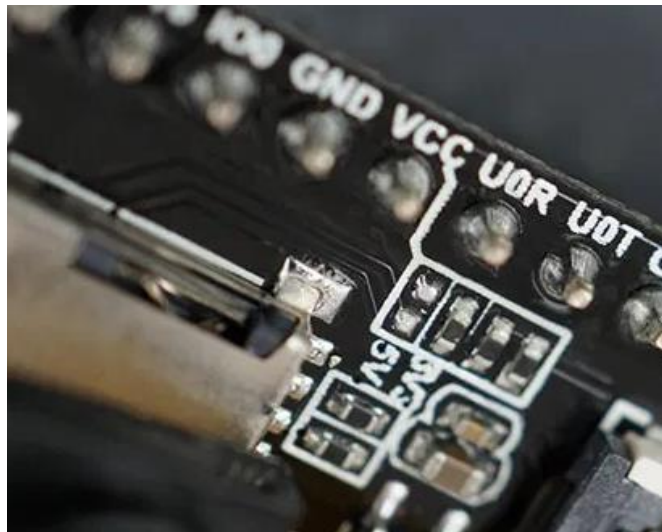
Pinos de energia

O ESP32-CAM vem com três pinos GND (*ground*) e dois pinos de alimentação: 3,3V e 5V. O fornecimento de energia pode ser feito por meio dos pinos de 3,3V ou 5V. No entanto, percebeu-se durante alguns testes que a saída de 3,3V não fornece uma alimentação estável para a placa.

3.1.1. Configurando o ESP32-CAM

O ESP32-CAM não vem com um conector USB, então foi necessário um programador FTDI para carregar o código por meio dos pinos U0R e U0T (pinos seriais).

Figura 13 – Referência do Pinos de U0R e U0T de conexão com FTDI



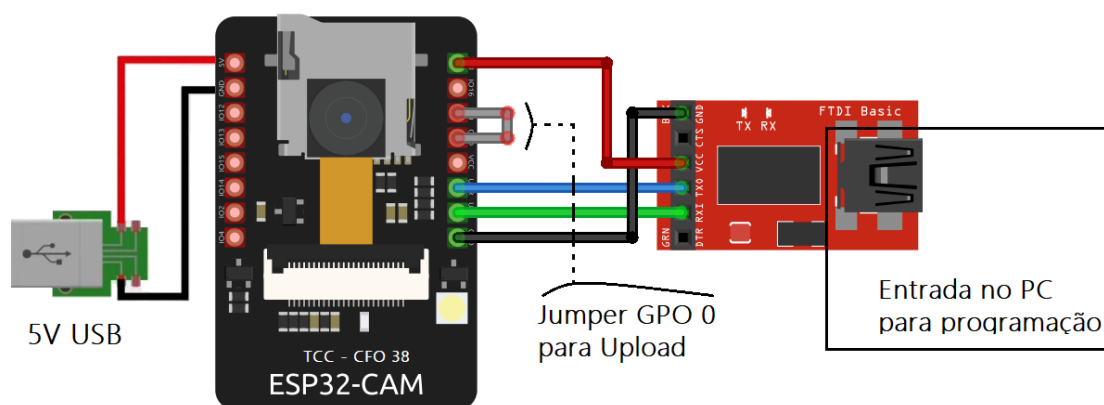
Fonte: DFROBOT (2021)

Pinos seriais

Os GPIO 1 (U0T) e GPIO 3 (U0R) são os pinos seriais de transmissão e recepção de dados (TX e RX, respectivamente). Como o ESP32-CAM não tem um programador embutido, foi necessário usar esses pinos para se comunicar com a placa e fazer o *upload* do código. A melhor maneira encontrada a partir de alguns testes de fazer *upload* do código para o módulo deste projeto é usando o módulo FTDI.

O GPIO 0 determina se o módulo está no modo intermitente ou não, em outras palavras se o *clock* do microcontrolador está trabalhando ou não. Ele determina quando o módulo está no modo programação ou em funcionamento normal. Quando o GPIO 0 é conectado ao GND, o ESP32 entra no modo programação e assim foi possível fazer o *upload* do código para a placa. Para fazer o ESP32 funcionar no modo normal de funcionamento, foi necessário desconectar o GPIO 0 do GND e deixar o *clock* trabalhar.

Figura 14 – Esquema Geral para fazer o *upload* da programação

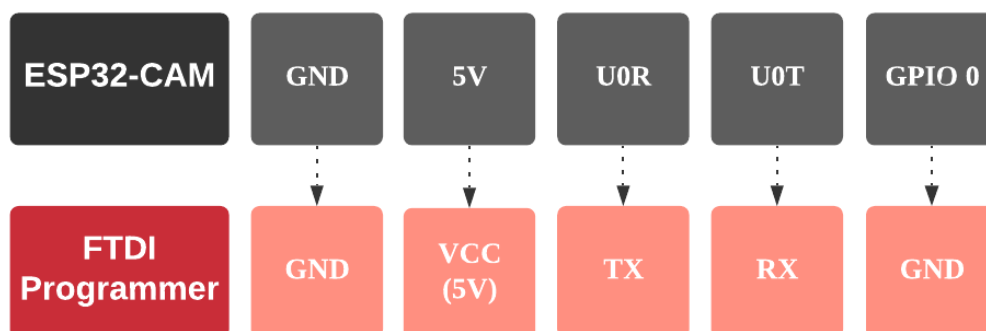


Fonte: o autor

A figura 14 demonstra as ligações que foram feitas da ESP32-CAM com o módulo de gravação FTDI. Percebe-se que existe uma ligação feita nos próprios pinos da ESP32 (cor cinza), que é comumente conhecido como “*jumper*”. Essa ligação nada mais é do que colocar o GPIO 0 no modo de baixa tensão, ou seja, conectar o GPIO 0 no GND.

As ligações feitas entre os módulos podem ser vistas com mais detalhes na figura 15. Um ponto também importante que deve ser ressaltado é que os pinos U0R e U0T foram ligados com seus contrapinos, U0R→TX e U0T→RX

Figura 15 – Diagrama de conexões entre o ESP32-CAM e o FTDI



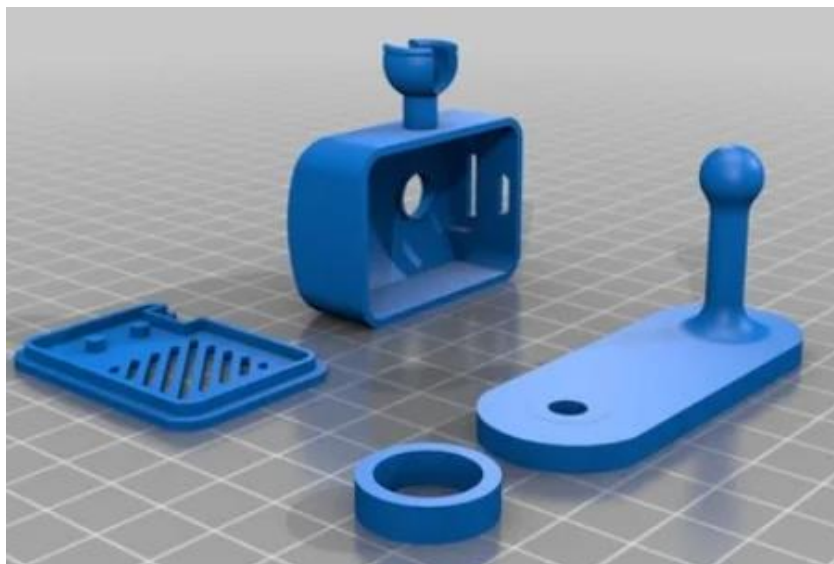
Fonte: o autor

Esta configuração é a utilizada quando se necessita atualizar a programação do módulo. O adaptador FTDI não precisa ser conectado permanentemente, pois só é usado durante a fase de programação do projeto.

Modelo 3D

Para criação do protótipo foi utilizado o modelo 3D da figura abaixo. Esse modelo foi enviado ao software da impressora 3D, definindo dimensões e detalhes de texturas. O desenho em 3D desse objeto foi fatiado em várias camadas de impressão e produzido uma a uma até que se forme por completo. Ele possui aberturas para alimentação, transmissão de dados e antena.

Figura 16 – Modelo 3D utilizado na criação do produto.



Fonte: MAKERBOT (2021)

3.2. Desenvolvimento do Código

A placa de desenvolvimento ESP32-CAM foi programada usando o Arduino IDE na linguagem de programação *Javacript*. O primeiro passo foi ter o Arduino IDE instalado, bem como a configuração e montagem do módulo ESP32.

Para obtenção das imagens utilizando o hardware de captura foram implementados algoritmos de controles responsáveis pelo ajuste de parâmetros

de captura de imagens, decodificação de dados, envolvendo a detecção usando a visão computacional que fazem parte do programa principal.

O programa consiste em duas partes:

1. o programa de servidor que roda na câmera ESP32
2. o programa cliente que é executado no navegador de internet

O programa foi dividido em dois arquivos: o arquivo `Principal_OpenCV.ino` contendo o programa do servidor e o arquivo de cabeçalho `indice_OpenCV.h` contendo o programa cliente (HTML, CSS e JavaScript com `OpenCV.js`), responsável pela parte visual.

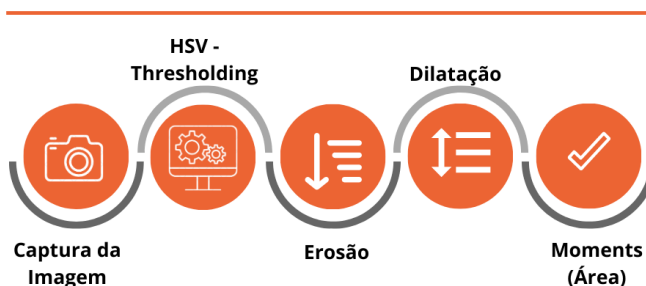
O ajuste dos parâmetros foram feitos na parte do processamento de imagens no arquivo `Principal_OpenCV.ino`, fase onde a imagem capturada é tratada. Para o tratamento da imagem é necessário a captura da imagem ou, especificamente, o quadro que contém a imagem (*frame*). O sistema padrão é fornecido em cores RGB, este é obtido através das cores básicas: *Red* (vermelho), *Green* (verde) e *Blue* (azul). Estas cores são representadas pixel por pixel em um vetor tridimensional, por exemplo, a cor vermelha é representada com valores de (0, 255, 0), respectivamente, representadas para cada canal. Isto é, cada pixel tem o seu valor RGB representado por três *bytes* (verde e azul vermelho).

Depois que a imagem foi capturada, a sequência é a conversão do sistema de cores RGB para a cor HSV (matiz, saturação e valor), uma vez que este modelo facilita a determinação dos limiares para segmentar uma determinada cor ou outra característica de forma robusta. O sistema RGB (vermelho, verde e azul) tem as cores baseadas em combinações de cores primárias (vermelho, verde e azul) e o sistema HSV define as cores como (*Hue*, *Saturation*, *Value*).

Com a imagem no formato HSV, foi possível encontrar os valores de correção de HSV mínimo e máximo. Por fim, para encontrar a região final a ser delimitada como incêndio, foram seguidos os procedimentos representados na Figura 17.

Figura 17 – Fluxo do Processamento de Imagem

ETAPAS DO PROCESSAMENTO DE IMAGEM



Fonte: o autor

Efeitos da Erosão:

- Diminuir partículas
- Eliminar componentes menores que o elemento estruturante
- Aumentar buracos
- Permitir a separação de componentes conectados

Efeitos da dilatação:

- Aumentar partículas
- Preencher buracos
- Conectar componentes próximos

Depois do processamento de imagem demonstrado na Figura 17, foi possível detectar a existência de incêndios florestais.

3.3. Testes

Os testes desempenham o papel mais importante no desenvolvimento do trabalho, pois a partir dele foi possível encontrar os valores de correção de HSV mínimo e máximo. Para a realização dos testes foi utilizado o módulo para estudo do comportamento do fogo e fenômenos do fogo, a fim de validar e variar os valores de HSV para encontrar sua faixa de trabalho na identificação do incêndio florestal e calibrar o algoritmo de detecção.

Foram utilizadas as duas imagens da Figura 18 para realização dos testes. A primeira imagem retrata as características da chama de um ambiente interno e através de um vidro temperado. Já a segunda foi extraída para representar uma chama mais viva e externa.

Figura 18 – Imagens capturadas para Testes do módulo de estudo do comportamento do fogo



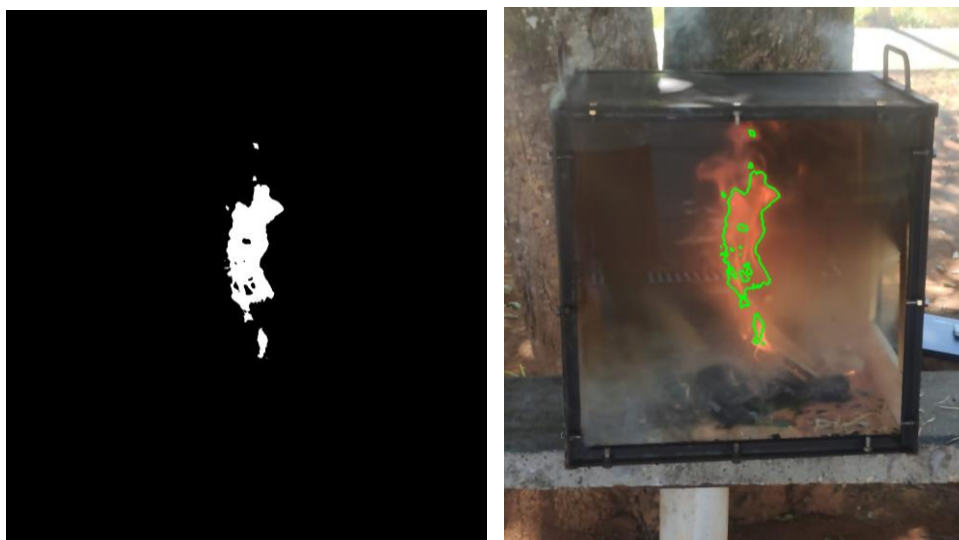
Fonte: o autor

Os experimentos foram divididos em dois momentos: O primeiro representando a detecção do incêndio, ou seja, os parâmetros necessários para o algoritmo encontrar as chamas. O segundo para entender a interferência das características do ambiente externo e como as técnicas de erosão e dilatação podem auxiliar em uma melhor identificação.

3.3.1. Detecção

O primeiro teste executado foi o de detecção, neste caso, a primeira imagem da figura 18 foi escolhida por apresentar características de um foco de incêndio. O teste tinha como objetivo reconhecer as cores que determinam um incêndio.

Figura 19 – Resultado do Primeiro Teste de detecção



Fonte: o autor

A primeira imagem da figura 19 demonstra o resultado do algoritmo de detecção. A parte branca representa a percepção do microcontrolador, do que ele reconhece como chama. No resultado da figura 19 percebe-se que o limiar HSV escolhido não foi o mais eficiente para representar as características da chama.

$HSVmin([10,150,160])$

$HSVmax([35,250,255])$



Limiar HSV definido para o primeiro teste de detecção. Fig. 19

Figura 20 – Resultado do Segundo Teste de detecção



Fonte: o autor

Já a figura 20 demonstrou um resultado melhor do algoritmo de detecção. No resultado figura 20 percebe-se que o limiar HSV escolhido foi mais eficiente para representar as características da chama do que o da figura 19.

$HSVmin([0,130,150])$
 $HSVmax([35,255,255])$

✓
Limiar HSV definido para o segundo teste de detecção. Fig. 20

3.3.2. Erosão e Dilatação

O segundo teste executado foi o de erosão e dilatação, neste caso, o objetivo foi retirar ruídos ou outras formas de interferência no resultado da percepção do algoritmo.

Figura 21 – Resultado do Primeiro Teste de erosão e dilatação



Fonte: o autor

Analisando o resultado do teste de erosão e dilatação da figura 21

percebe-se que existem muitos ruídos na imagem, fazendo com o que o resultado para detecção de incêndio não seja o adequado.

Figura 22 – Resultado do Segundo Teste de erosão e dilatação



Fonte: o autor

O resultado da figura 22 demonstra uma melhora significativa da detecção do incêndio. Os limiares HSV foram os mesmos utilizados nas figuras 21 e 22, porém a figura 22 teve o tratamento das técnicas de processamento de imagens: erosão e dilatação.

4. CARACTERÍSTICAS DO PRODUTO

Constituído por uma câmera, interface de conexão e placa controladora ESP32 conforme apresentado na figura abaixo, é o elemento principal de recepção de imagens. Possui estrutura para encaixe e desencaixe de fácil manuseio e estrutura acoplada que permite a regulação do posicionamento angular da câmera.

Figura 23 – Partes Gerais do Produto Final



Fonte: o autor

A câmera utilizada foi a OV2640 conforme apresentado na Figura 7. Conforme dados técnicos do dispositivo, esta câmera CMOS com total funcionalidade em UXGA 1632x1232 de resolução possui dispositivo *Serial Camera Control Bus* (SCCB) com processamento em único chip fornecendo imagens *full-frame*, *sub-sample* ou dimensionadas com 8 a 10 bits em diversos formatos. Opera com taxa de 15/30 FPS, com controle completo sobre a qualidade da imagem, formato de saída e transferência de dados. Todas as funções de processamento de imagem, incluindo controle de exposição, gama, equilíbrio de branco, saturação controle de matiz, cancelamento de pixel branco, cancelamento de ruído e outras funções, são programáveis através da interface SCCB, justificativa principal para a utilização desta câmera no hardware de captura.

Características Físicas Gerais

1. O produto ficou com peso de aproximadamente 30g e com dimensões de 3 cm de altura, 1,5 cm de profundidade e 4,5 de largura;
2. A face lateral direita ficou com saídas para MiniUSB e cartão MicroSD ;
3. A face frontal ficou com aberturas para a câmera e o flash;
4. A face traseira ficou com aberturas de ventilação e os pinos de comunicação com o FTDI;
5. A face inferior possui um encaixe para rotação de 360°;
6. O material da estrutura é PLA e foi impresso em um uma impressora 3D.

Figura 24 – Demonstração do Módulo ESP32-CAM encapsulado



Fonte: o autor

O material **PLA** ou poliácido láctico é um polímero termoplástico mais usual na impressão 3D. Refere-se a uma substância biodegradável de origem natural e de fontes renováveis que vem substituindo os plásticos convencionais em diversas aplicações.

Características

O processador do ESP32-CAM é o tradicional ESP32 com 4M de PSRAM – uma memória RAM SPI externa.

Suporta as câmeras OV2640 e OV7670, intercambiáveis. Ao lado da câmera tem um LED maior para servir de flash. Ela possui também um encaixe para antena externa, com propósito de aumentar o alcance.

Possui um slot de cartão micro-SD, ao que parece, de tamanho máximo de 4GB. A câmera suporta 3 resoluções nativas, sendo UXGA (1600×900), SVGA (800×600) e CIF (formato que varia em CCTV: 252×288 em PAL e 252×240 em NTSC).

O formato **jpg** só é suportado na câmera OV2640, mas suporta também o formato **bmp** e a imagem pode ser passada para *grayscale*.

Consumo de energia

A alimentação se dá pelo *header*, no pino 5V. É importante considerar esses fatores para definir a fonte de alimentação para o ESP32-CAM, a fim de evitar os mais diversos tipos de problemas, às vezes sem vestígios.

Por padrão, o *flash* está desligado, mas se for necessário utilizar o *flash* da placa, os primeiros 180mA pertencerão a ele em um brilho moderado. Se for na intensidade máxima, 310mA. Em qualquer circunstância, à 5V.

No mode *deep-sleep* o mais econômico, o consumo é de 6mA. *Modem-sleep* o mais ativo, o consumo é de 20mA e no *light-sleep*, 6.7mA.

Figura 25 – Produto Final



Fonte: o autor

Formas Gerais de Uso

1. O protótipo necessita de acesso à internet
2. O fornecimento de energia depende de um cabo usb conectado ao computador ou uma fonte de energia externa de 5V.
3. Para a melhor qualidade do sinal de internet é aconselhado utilizar a antena
4. Para atualizar o algoritmo deve-se acoplar o módulo FTDI para upload de uma nova programação

5. ANÁLISE DO RESULTADO

Esse capítulo traz como primeiro resultado o levantamento dos dados coletados junto ao Grupamento de Proteção Ambiental do CBMDF (GPRAM) a fim de elucidar o primeiro objetivo específico e justificar o estudo do tópico de visão computacional.

De acordo com os dados coletados nesta pesquisa, pode-se perceber que são empregados muitos recursos na Operação Verde Vivo, dentre militares, materiais, viaturas e aeronaves. No ano de 2020 as aeronaves desempenharam um papel fundamental tanto no monitoramento quanto no combate. A seguir, consta o demonstrativo estatístico da utilização das aeronaves cedidos pelo GAVOP/CBMDF e presente no relatório final da Operação Verde Vivo (OpVV).

Tabela 1 – Acionamentos de Aeronaves – Asa Rotativa

Helicóptero (Asa Rotativa)	
Quantidade de Acionamentos	48
Tempo de voo	40,7 horas
Quantidade de Lançamentos	46
Quantidade de Água	18400 litros

Fonte: Seção de Operações/GPRAM.

Tabela 2 – Acionamentos de Aeronaves – Asa Fixa

Air Tractor (Asa Fixa)	
Combustível (QAV)	25170 litros
Quantidade de Acionamentos	61
Tempo de voo	112,5 horas
Quantidade de Lançamentos	274
Quantidade de Água	432122 litros

Fonte: Seção de Operações/GPRAM.

As tabelas 1 e 2 trazem as informações gerais de acionamento das aeronaves. Não especificam se o acionamento foi para combate ou monitoramento. Porém cabe ressaltar que o emprego das aeronaves em

ocorrências florestais foi bastante solicitado e que o custo no geral para o seu uso é alto.

Os Bombeiros militares empregados na Operação Verde Vivo utilizaram viaturas de pequeno e grande porte para o atendimento das ocorrências florestais sendo elas: ARF – Auto Rápido Florestal, ATT's – Auto Transporte de Tropa e ABTF's Auto Bomba Tanque Florestal. Outras viaturas tais como: AC's – Auto Caminhões, e AO's – Auto Ônibus foram utilizadas em incêndios de grande porte, onde havia a necessidade de emprego de um número atípico de militares. Contabilizam também a esta lista duas Aeronaves de Asa Fixa (Air Tractor) e duas Aeronaves de Asa Rotativa do Grupamento de Aviação Operacional - GAVOP/CBMDF. Os quantitativos destes recursos estão descritos abaixo no Gráfico 1, sendo que, está incluído neste somatório viaturas alocadas como reservas, para a manutenção do poder operacional.

Gráfico 1 – Viaturas e aeronaves empregadas na Operação Verde Vivo

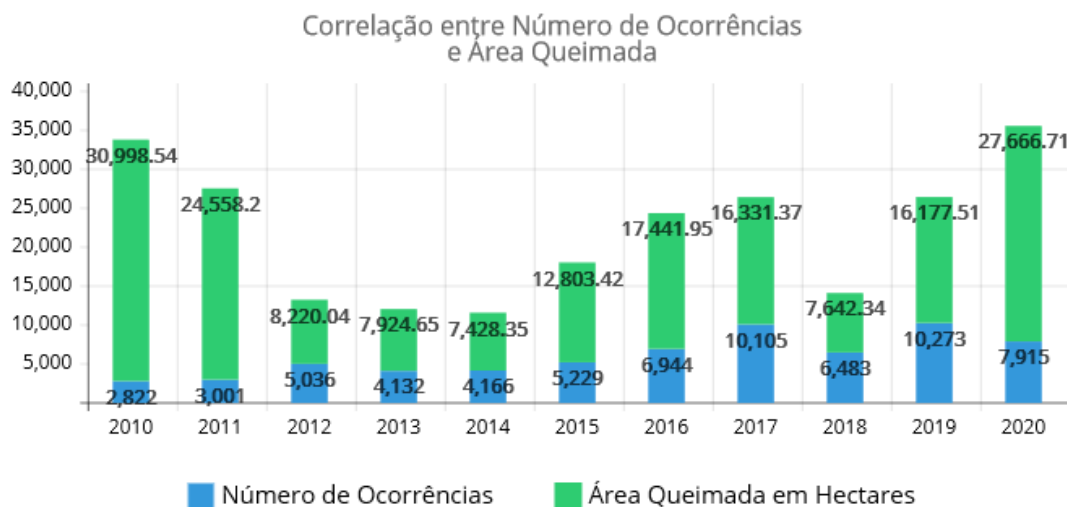


Fonte: o autor.

Percebe-se analisando o gráfico 1 que a viatura do tipo ATT (Auto Transporte de Tropa) também foi muitas vezes empregadas, o que corrobora quando comparado com o gráfico 3 quanto a quantidade de militares que foram utilizados na OpVV.

Gráfico 2 – Número de Ocorrências e Área Queimada em Hectares

REGISTRO HISTÓRICO DAS OPERAÇÕES VERDE VIVO



Fonte: o autor

Gráfico 3 – Militares empregados

RECURSOS HUMANOS OPVV 2020

Militares empregados na operação Verde Vivo de 2020



Fonte: o autor.

Dando sequência na análise dos resultados e com o intuito de responder o objetivo principal do trabalho que é identificar focos de incêndios florestais usando visão computacional. Seguem-se os resultados encontrados na seção de processo de construção:

Resultados dos Testes

A utilização dos algoritmos de pré-processamento como detecção, erosão e dilatação descritos nas figuras 20 e 22 levaram a uma melhora significativa dos resultados. Contudo, deve ser utilizada com um limiar específico para obter melhores informações e detalhes em um espaço de cores HSV.

No espaço de cores HSV, é mais provável que o componente de matiz de todos os ambientes seja semelhante: a sombra influenciará principalmente o valor, ou talvez o componente de saturação, enquanto o matiz, indicando a “cor” primária (sem brilho e diluição do branco) não deve mudar muito, pois o incêndio possui um limiar pequeno de amarelo ao laranja.

Resultados em Imagens Florestais

Aplicou-se o também algoritmo implementado na base de fotos que os RPAs obtêm em ocorrências de incêndios florestais. E chegou-se aos resultados mostrados nas figuras 26 e 27.

Figura 26 – Resultado da primeira foto do apêndice



Figura 27 – Máscara da primeira foto do apêndice

Fonte: o autor

Figura 27 – Resultado da segunda foto do apêndice



Fonte: o autor

Ao aplicar o algoritmo em imagens florestais constatou-se que os resultados foram semelhantes aos encontrados com os testes.

Resultado do Produto

O resultado da figura 28 foi a implementação do algoritmo desenvolvido sob um vídeo com as mesmas resoluções da câmera que foi desenvolvida neste projeto. Assim pode-se verificar como é o desempenho sob um formato de vídeo com a mesma resolução.

Figura 28 – Resultado sob o *frame* de um video



Fonte: o autor

A figura 28 apresenta os resultados obtidos da detecção de incêndio florestal do conjunto de dados em duas classes, ou seja, fogo e fumaça e os resultados da mostram que o algoritmo teve dificuldade de diferenciar as duas classes devido aos diferentes padrões de fumaça, ou seja, fina, densa, branca e assim por diante. Porém mesmo assim o resultado foi satisfatório no reconhecimento de incêndio florestal usando o algoritmo proposto.

Os resultados mostram que é possível utilizar a solução proposta para detecção de focos de incêndios florestais. Para diminuir o problema de ruídos, é importante a aplicação das técnicas de segmentação inclusive o processo de erosão e dilatação. Durante a realização de testes foi perceptível a diferença no resultado final.

Considerações Finais e Recomendações

O projeto proposto teve como principal objetivo analisar a efetividade do uso da visão computacional no processo de detecção de incêndios florestais. Para isso, teve como objetivos específicos: levantar os recursos empregados no combate e os históricos de incêndios florestais atendidos no CBMDF, apresentar o conceito de visão computacional, implementar o algoritmo de detecção de incêndios florestais usando a visão computacional e por último avaliar o funcionamento do algoritmo implementado através de testes e comparações no produto final.

Assim, foram obtidos dados quantitativos da Operação Verde Vivo, evidenciando a importância de criação de medidas de melhor aproveitamento desses recursos, por conseguinte, atender melhor as ocorrências de incêndios florestais.

Além disso, esse trabalho permitiu a implementação de um algoritmo que detectasse incêndios florestais. Algumas limitações foram observadas ao adquirir os resultados, como a dificuldade de detecção quando o ambiente apresenta ruído e o incêndio florestal possui bastante fumaça. Porém os resultados encontrados no desenvolvimento foram bastante satisfatórios.

A importância do trabalho se dá pelo grave problema ambiental que constituem os incêndios florestais, cujas consequências comprometem o equilíbrio dos fatores que dão suporte à vida. As consequências da ação do fogo sem controle são numerosas, comprometendo a vegetação, a fauna, o solo, os cursos d'água, o ar atmosférico, a população e a biodiversidade.

Um impacto muito visível dos incêndios florestais está relacionado à poluição e à qualidade do ar. A fumaça e as cinzas produzidas durante as queimadas contêm partículas muito pequenas, causando graves problemas sociais, econômicos, ambientais e problemas de saúde como: a inflamação nas vias aéreas superiores, asma e até câncer.

Ademais sugere-se, para trabalhos futuros, com base nas investigações da atual pesquisa:

- Como forma de melhorar o monitoramento de incêndios florestais, verificar a aplicabilidade do módulo em RPAs, a corporação adquiriu recentemente alguns RPAs (drones), e em torres de vigilância.
- Como em toda ocorrência de incêndio florestal deve ser feito o cálculo de área queimada, propor a implementação de um algoritmo para detecção e cálculo de área queimada.
- A fim de melhorar a precisão da detecção de incêndios florestais, implementar um algoritmo para detecção de fumaça e aprimorar o processo de reconhecimento de incêndios florestais;
- Como as pesquisas no incêndio urbano estão em crescimento, verificar a utilização do módulo de detecção de incêndios florestais em residências e fazer um comparativo e viabilidade da sua aplicabilidade como instrumento de segurança contra incêndio e pânico.

REFERÊNCIAS

ALVES, João Miguel Ribeiro. **Deteção de incêndios florestais com recurso a Deep Learning e Visão Computacional. Dissertação.** (Mestrado em Engenharia Informática) - UFP, Porto, 2018. 70p.

ARAÚJO, L.O. **Dinâmica Espaço-Temporal dos Focos de Calor na Microrregião de Paragominas-Pará**, Trabalho de Conclusão de Curso Bacharelado em Engenharia Florestal da UFRA, Paragominas, 2019. 50p.

ARDUINO. **Arduino Software (IDE)**. Disponível em: <<https://www.arduino.cc/en/Guide/Environment>>. Acesso em: 06 abr. 2021.

Bradski, G. and Kaehler, A. **Learning OpenCV Computer Vision with the OpenCV**. Library. “O’Reilly Media, Inc.”, California, 2008.

CHI YUAN. **Automatic Fire Detection Using Computer Vision Techniques for UAV-based Forest Fire Surveillance**. Concordia University, 2017.

CORPO DE BOMBEIROS MILITAR DO DISTRITO FEDERAL – CBMDF. **Plano estratégico do Corpo de Bombeiros Militar do Distrito Federal. 2017 - 2024**. [S. l.], 15 dez. 2016. Disponível em: <https://www.cbm.df.gov.br/downloads/edocman/estrategico//Plano%20Estrategico%202017-2024.pdf>. Acesso em: 2 set. 2020

COUTO, L. N. **Sistema para localização robótica de veículos autônomos baseado em visão computacional por pontos de referência**. 2012. 82 f. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) –Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2012.

CUNHA, A. L. **Introdução à Visão Computacional**, USP 2016. Disponível em: https://www.stt.eesc.usp.br/andre/palestras/CEFET-MG/01_Intro.html. Acesso em: 04 julho 2021

DFROBOT. **ESP32-CAM Development Board** . Shangai, China, 2021.

Disponível em: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf. Acesso em: 20 junho. 2021.

EDWARD. A. K; R. H. Blodgett, **Riesgos naturales: procesos de la Tierra como riesgos, desastres y catástrofes**. Madrid, Pearson Prentice Hall, 2007, p. 320.

ESTEVAM, T. S. **Análise da susceptibilidade a incêndios florestais no Parque Estadual do Itacolomi como ferramenta auxiliar de prevenção e combate às queimadas**. Trabalho de Conclusão de Curso de Engenharia Ambiental da UFOP, Ouro Preto, 2020. Disponível em:

https://www.monografias.ufop.br/bitstream/35400000/2916/9/MONOGRRAFIA_AnáliseSusceptibilidadedeIncendios.pdf. Acesso em: 20 fevereiro 2021.

FERNANDES, Euripedes Purcinio. **Visão computacional para identificação de cores em tempo real com Opencv e Python**. 2020. Monografia (Graduação em Engenharia da Computação) - Faculdade de Tecnologia e Ciências Sociais Aplicadas, Centro Universitário de Brasília, Brasília, 2020.

FILIPEFLOP. **Placa FTDI FT232RL Conversor USB Serial**. Florianópolis, 2021. Disponível em: <https://www.filipeflop.com/produto/placa-ftdi-ft232rl-conversor-usb-serial/> Acesso em: 20 junho. 2021.

GARCÍA, G. B. et al. **Learning Image Processing withOpenCV**. [S.l.]: Packt Publishing Ltd, 2015.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4 ed. Editora Atlas, 2002.

JOSEMAR, A. **O poderoso FTDI FT232R**. Embarcados, 2015. Disponível em: <https://www.embarcados.com.br/ftdi-ft232r/> Acesso em: 20 fevereiro. 2021.

KIM, C. et al. **Securing Real-Time Microcontroller Systems through Customized Memory View Switching**. NDSS (2018).

LAGARES, Robson de Oliveira. **Análise da efetividade e eficácia do plano de prevenção e combate a incêndios florestais no Distrito Federal**. 181 f., il. Dissertação (Mestrado em Desenvolvimento Sustentável) -Universidade de Brasília, Brasília, 2007.

LIMA, Aline Marcimiano de. **Zoneamento de risco de incêndios no Distrito Federal**. 2017. xi, 47 f., il. Trabalho de conclusão de curso (Bacharelado em Engenharia Florestal) —Universidade de Brasília, Brasília, 2017.

MAKERBOT. **Camera mount 360° (Arduino ESP32-Cam) V3**. Thingiverse. Disponível em: <https://www.thingiverse.com/thing:3652452>. Acesso em: 20 fevereiro 2021

MORRIS T. **Computer Vision and Image Processing**. Palgrave Macmillan. 2004.

NUNES, José Renato Soares. **FMA+ - Um novo índice de perigo de incêndios florestais para o estado do Paraná - Brasil**. 2005. 150 p. Tese (Doutorado em Ciências Florestais) - Curso de Ciências Florestais, Curitiba, Ufpr, 2005.

OMNIVISION. **TechnologiesPreliminary DatasheetOV2640 Color CMOS UXGA (2.0 MegaPixel) CAMERACHIPTM**. Disponível em: https://www.uctronics.com/download/OV2640_DS.pdf. Acesso em: 20 fevereiro 2021

OPENCV. **The best computer vision library in the world got even better.** Disponível em: <https://opencv.org/> Acesso em: 20 fevereiro 2021

PETERS, A. **STUDIO PETERS, ESP32 CAM – Troubleshooting.** Holanda, 2021. Disponível em: <https://www.studiopieters.nl/esp32-cam-troubleshooting/>. Acesso em: 20 junho. 2021.

PORTAL G1. **Em 13 horas, bombeiros combatem 83 focos de incêndios florestais no DF.** [Reportagem] 2021. Disponível em: <https://g1.globo.com/df/distrito-federal/noticia/2020/09/06/em-13-horas-bombeiros-combatem-83-focos-de-incendios-florestais-no-df.ghtml> Acesso em: 05 julho 2021

RADSKI, G.; KAEHLER, A. **Learning OpenCV.** Sebastopol: [s.n.], 2008.

SANTOS, R. **ESP32-CAM AI-Thinker Pinout Guide: GPIOs Usage Explained.** Portugal, 2021. Disponível em: <https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/>. Acesso em: 20 Maior. 2021.

SEPIR. **SISTEMA ELETRÔNICO DE PREVENÇÃO DE INCÊNDIOS EM RODOVIAS.** Brasil, 2015. Disponível em: <http://conic-semesp.org.br/anais/files/2015/1000020246.pdf> Acesso em: 10 fevereiro 2021.

SILVA, G.D. **Desenvolvimento de um Sistema de Visão Computacional para o Estudo do Comportamento de Animais Experimentais:** Belo Horizonte 2008. Disponível em: https://repositorio.ufmg.br/bitstream/1843/BUOS-8CUFBY/1/guilherne_damasceno_silva.pdf. Acesso em: 05 de fevereiro de 2020.

SZELISKI. R. **Computer Vision: Algorithms and Applications.** Springer, 2010.

APÊNDICES

Apêndice A – GitHub Código

Este conteúdo não está disponível para visualização.

ANEXOS

Este conteúdo não está disponível para visualização.

Este conteúdo não está disponível para visualização.

Este conteúdo não está disponível para visualização.

