Queimadas no Brasil

UNIP – 2024

Marcos Vinicius Carvalho de Oliveira

F310353

CC6P04

Indice

Resumo

Visto as recentes queimadas acontecendo no Brasil inteiro foi realizado uma aplicação computacional utilizando técnicas de Visão Computacional para a identificação dessas queimadas. O objetivo agora é transformar em vídeo essa solução e, assim, identificar queimadas de maneira mais rapida, reduzindo o tempo para a chegada de bombeiros nos locais e ajudar a preservar a fauna e flora de nosso país. Foi usado a biblioteca cv2, numpy e matplot para conseguir concluir essa tarefa e mostrado todos os filtros necessários para a conclusão.

Abstract

Since the recent fires happening throughout Brazil, a computational application was carried out using Computer Vision techniques to identify these fires. The goal now is to video this solution and thus identify fires faster, reducing the time for the arrival of firefighters at the sites and help preserve the fauna and flora of our country. The cv2, numpy and matplot library was used to complete this task and all the necessary filters were shown for completion.

1. Descrição do Problema

O problema das queimadas não é algo que possa ser resolvido de forma simples, principalmente com o aumento significante nas últimas semanas. As queimadas e incêndios criminosos são os principais problemas em nosso país no ano de 2024. Esse problema traz diversos malefícios não só as cidades dos interiores, onde estão acontecendo as principais queimadas, mas também aos moradores de grande São Paulo, como a dificuldade de respirar o ar e os céus mais nebulosos.

Listei abaixo os principais malefícios que pode causar ao nosso país e planeta:

Destruição de Ecossistemas

- Perda de Biodiversidade: Queimadas, especialmente em áreas como a Amazônia e o Cerrado, destruem habitats naturais e ameaçam a vida de inúmeras espécies de flora e fauna. Muitas dessas espécies são endêmicas e não conseguem se adaptar rapidamente a mudanças tão drásticas no seu ambiente.

- Alteração dos Ecossistemas: As queimadas podem mudar a estrutura dos ecossistemas, afetando a composição das espécies vegetais e animais. Isso pode levar à degradação do solo e mudanças no ciclo da água.

Mudanças Climáticas

- Emissão de Gases de Efeito Estufa: As queimadas liberam grandes quantidades de dióxido de carbono (CO_2) e outros gases de efeito estufa na atmosfera, contribuindo para o aquecimento global e as mudanças climáticas.

- Redução da Capacidade de Armazenamento de Carbono: Florestas e vegetação queimadas perdem sua capacidade de absorver CO_2, o que agrava ainda mais o problema das mudanças climáticas.

Impactos na Saúde Pública

- Poluição do Ar: A fumaça das queimadas contém partículas finas e compostos tóxicos, como monóxido de carbono e óxidos de nitrogênio, que podem causar problemas respiratórios e cardiovasculares. A poluição do ar pode afetar a saúde de milhões de pessoas, especialmente em áreas urbanas e rurais próximas às queimadas.

- Risco de Doenças: O aumento da poluição do ar pode exacerbar doenças respiratórias crônicas, como asma e bronquite, e aumentar a incidência de infecções respiratórias agudas.

2. Referencial Teórico

2.1. Processamento de Imagem

O processamento de imagem é uma técnica fundamental em visão computacional que permite a manipulação de imagens digitais para a extração de informações ou melhoria de qualidade. Ele consiste em operações que facilitam a análise, destacando características ou padrões visuais específicos, como áreas de fogo em florestas.

2.2. Espaços de Cor

As imagens digitais são representadas em diferentes espaços de cor, e a escolha do espaço é essencial para tarefas específicas de análise e segmentação.

- RGB (Red, Green, Blue): O espaço de cor RGB é o padrão para imagens digitais, mas pode dificultar a segmentação quando as cores são próximas entre si.
- HSV (Hue, Saturation, Value): Mais utilizado em segmentação de cores, o espaço HSV separa a cor (matiz), intensidade (saturação) e brilho (valor). Esse espaço facilita a criação de máscaras para isolar áreas com cores específicas, como vermelho, laranja e amarelo, características de incêndios.

2.3. Segmentação de Cores

A segmentação de cores é o processo de separar e destacar partes da imagem de acordo com suas propriedades de cor. No contexto da detecção de incêndios, a segmentação de cores no espaço HSV ajuda a identificar áreas onde predominam tons de vermelho e laranja, associados às chamas. Com o uso de filtros e máscaras, é possível realçar regiões com essas características.

2.4. Máscara e Sobreposição

A técnica de sobreposição utiliza máscaras para realçar partes da imagem que se enquadram em uma faixa de cor específica. No OpenCV, a função cv2.inRange é

aplicada para criar uma máscara binária que destaca regiões de interesse, como áreas com cores indicativas de fogo.

2.5. Detecção de Formas

A detecção de contornos é útil para isolar e delinear áreas identificadas por cor ou textura. Funções como cv2.findContours e cv2.drawContours do OpenCV permitem localizar e desenhar essas regiões, facilitando a análise de áreas específicas de incêndio. Além disso, as operações morfológicas de erosão e dilatação ajudam a refinar as regiões detectadas, removendo ruídos ou preenchendo lacunas na máscara.

2.6. Aprendizado de Máquina e Redes Neurais Convolucionais

A inteligência artificial e o aprendizado de máquina oferecem métodos avançados para a detecção de padrões em imagens, indo além dos filtros tradicionais. Redes Neurais Convolucionais (CNNs) são amplamente usadas em visão computacional devido à sua capacidade de identificar padrões complexos, como texturas e bordas, presentes em incêndios.

2.7. Novas Técnicas de Aprendizado de Máquina

Recentemente, abordagens mais sofisticadas, como redes neurais profundas e modelos baseados em aprendizado transferido, têm sido aplicadas com sucesso na detecção de desmatamento e incêndios florestais.

- Redes Profundas e Transferência de Aprendizado: Modelos pré-treinados em grandes conjuntos de dados são ajustados para detectar sinais de incêndio. Isso acelera o desenvolvimento e melhora a precisão ao trabalhar com conjuntos de dados limitados de imagens de desastres ambientais.
- Segmentação Semântica: Esse método permite a classificação de pixels da imagem, identificando áreas específicas de interesse. Aplicado à detecção de desmatamento, a segmentação semântica ajuda a identificar e delimitar regiões afetadas de maneira mais precisa.
- Redes Baseadas em Atenção: As redes de atenção conseguem focar em regiões relevantes da imagem, como áreas com padrões e cores relacionadas ao fogo, proporcionando uma detecção mais eficiente.

3. Proposta

Neste trabalho, propomos duas abordagens complementares para a detecção de desmatamento em imagens e vídeos: uma abordagem baseada em filtros de segmentação de cor para detectar alterações específicas no ambiente e uma abordagem de aprendizado profundo, com o uso de redes neurais convolucionais, para identificar padrões de desmatamento de forma mais precisa e robusta.

3.1 Filtros para Detecção de Desmatamento

O uso de filtros e segmentação de cor possibilita a identificação rápida e direta de áreas afetadas pelo desmatamento. A técnica baseia-se na aplicação de filtros de cor específicos para isolar elementos visuais relacionados ao desmatamento, como áreas de solo exposto, vegetação seca ou queimadas recentes, que geralmente apresentam colorações características (tons de marrom, amarelo ou cinza, por exemplo). Essa abordagem é especialmente útil em situações de monitoramento em tempo real ou em imagens de baixa resolução, onde é possível identificar alterações básicas nas áreas de vegetação de forma ágil.

A proposta é criar máscaras binárias que destaquem áreas suspeitas, facilitando a segmentação e classificação de regiões afetadas. Esta abordagem requer menos recursos computacionais e pode ser aplicada de forma eficiente em imagens de satélite ou em vídeos, gerando alertas preliminares para análises mais detalhadas.

3.2 Aprendizado Profundo para Detecção de Desmatamento

O aprendizado profundo, particularmente com redes neurais convolucionais (CNNs), oferece uma abordagem mais avançada e precisa para detectar desmatamento. As CNNs são treinadas para identificar padrões complexos que ocorrem em áreas de desmatamento, como fragmentação de vegetação, presença de áreas queimadas e mudanças drásticas na densidade verde. A rede neural é capaz de aprender características detalhadas e sutis, que podem escapar à detecção por filtros simples, como os contornos específicos da vegetação ou o tipo de solo exposto.

A proposta com o uso de aprendizado profundo é criar um modelo que, uma vez treinado, possa identificar automaticamente o desmatamento em novas imagens, superando limitações de variabilidade de ambiente e qualidade de imagem. Essa abordagem permite maior acurácia e é ideal para análises de médio e longo prazo, em que a detecção de mudanças progressivas é essencial para monitoramento ambiental.

A combinação dessas duas técnicas possibilita um sistema de monitoramento versátil e eficaz, integrando detecções rápidas e análises aprofundadas.

4. Desenvolvimento

Essas técnicas, quando integradas a métodos de segmentação de cor e processamento de imagem, fornecem um sistema robusto para a detecção de desmatamento e incêndios, elevando a precisão e reduzindo falsos positivos ao identificar áreas de risco ambiental.

Neste projeto, o desenvolvimento é dividido em duas abordagens para a detecção de incêndios em imagens e vídeos: uma abordagem baseada em filtros para segmentação de cor e outra utilizando redes neurais convolucionais (CNN) para classificação de imagens. A seguir, detalhamos o código e as funcionalidades de cada parte, desde a segmentação e classificação até a execução do modelo em vídeos sequenciais.

4.1. Segmentação de Cor

Nesta parte, utilizamos o espaço de cores HSV (Hue, Saturation, Value) para identificar áreas da imagem que possuem cores indicativas de fogo, como laranja e vermelho claro. Para realizar essa segmentação, definimos intervalos de cor em HSV e criamos máscaras binárias para isolar essas áreas.

```
lower_orange = (5, 150, 150)
upper_orange = (15, 255, 255)

lower_red_light = (0, 150, 150)
upper_red_light = (10, 255, 255)
```

4.2 Classificação com CNN

Usamos uma rede neural convolucional (CNN) personalizada para classificar frames com fogo. A CNN foi treinada previamente e seus pesos carregados para realizar previsões.

A função check_fire_color utiliza esses limites para detectar cores relacionadas a incêndios em cada frame do vídeo:

```
def check_fire_color(frame):
    hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    mask_orange = cv2.inRange(hsv_frame, lower_orange, upper_orange)
    mask_red_light = cv2.inRange(hsv_frame, lower_red_light, upper_red_light)

    area_orange = cv2.countNonZero(mask_orange)
    area_red_light = cv2.countNonZero(mask_red_light)

    if area_orange > 300 or area_red_light > 300:
        return True
    return False
```

Se a área total estiver maior que 300 pixels no vídeo, então é identificado o fogo.

A função detect_fire utiliza o modelo para determinar se há fogo em um frame:

```
def detect_fire_in_video(frame):
    if check_fire_color(frame):
        if detect_fire(frame):
            return True
    return False
```

4.3 Dupla filtragem

Para evitar falsos positivos, foi feita uma dupla filtragem com o CNN e as filtragens de cores.

```
def detect_fire_in_video(frame):
    if check_fire_color(frame):
        if detect_fire(frame):
            return True
    return False
```

4.4. Reprodução de Vídeos Sequenciais

A função play_video_sequentially permite a reprodução de múltiplos vídeos em sequência. Ela chama a função detect_fire_in_video para verificar a presença de fogo e exibe uma notificação na tela.

```python
def play_video_sequentially(video_files):
    for video_file in video_files:
        cap = cv2.VideoCapture(video_file)
        if not cap.isOpened():
            print(f"Erro ao abrir o vídeo {video_file}!")
            continue

        print(f"Reproduzindo vídeo: {video_file}")

        while cap.isOpened():
            ret, frame = cap.read()
            if not ret:
                break

            if detect_fire_in_video(frame):
                cv2.putText(frame, "Fogo Detectado", (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)

            cv2.imshow('Q para Sair', frame)

            if cv2.waitKey(1) & 0xFF == ord('q'):
                break

        cap.release()

    cv2.destroyAllWindows()

video_files = ['Queimadas.mp4', 'NATUREZA.mp4']
play_video_sequentially(video_files)
```

4.5. Transformações de Dados e Treinamento do Modelo

O treinamento da CNN foi feito previamente com um conjunto de dados específico e preprocessamento para normalização e data augmentation.

```
    def __init__(self):
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1)
        self.fc1 = nn.Linear(64 * 56 * 56, 128)
        self.fc2 = nn.Linear(128, 1)

    def forward(self, x):
        x = self.pool(torch.relu(self.conv1(x)))
        x = self.pool(torch.relu(self.conv2(x)))
        x = x.view(-1, 64 * 56 * 56)
        x = torch.relu(self.fc1(x))
        x = torch.sigmoid(self.fc2(x))
        return x

model = FireDetectionCNN().to(device)
criterion = nn.BCELoss()
optimizer = optim.Adam(model.parameters(), lr=learning_rate)

def train_model(model, train_loader, criterion, optimizer, num_epochs):
    for epoch in range(num_epochs):
        running_loss = 0.0
        for inputs, labels in train_loader:
            inputs, labels = inputs.to(device), labels.to(device).float().unsqueeze(1)

            optimizer.zero_grad()
            outputs = model(inputs)
            loss = criterion(outputs, labels)
            loss.backward()
            optimizer.step()

            running_loss += loss.item()

        print(f'[{epoch+1}/{num_epochs}], {running_loss/len(train_loader):.4f}')
    print("Treinamento concluído!")

train_model(model, train_loader, criterion, optimizer, num_epochs)

torch.save(model.state_dict(), 'modelo_fogo.pth')
```