

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

MONITORAMENTO DE EMISSÕES FUGITIVAS À PARTIR DE IMAGENS DE CÂMERAS DE SEGURANÇA

Trabalho Científico

Aluna: Paula Jeniffer dos Santos Viriato
Orientador: Jefersson Alex dos Santos

Belo Horizonte, 05 de dezembro de 2019

Sumário

1	INTRODUÇÃO	3
2	REFERENCIAL TEÓRICO	4
3	METODOLOGIA	6
3.1	Fluxo Óptico (<i>Optical Flow</i>)	6
3.2	Subtração de Fundo (<i>Background Subtraction</i>)	7
3.3	Processamento de Imagens	7
3.4	Extração de <i>Features</i>	7
3.5	Aprendizado de Máquina	8
3.6	Tecnologias Utilizadas	9
4	RESULTADOS	10
4.1	Fluxo Óptico (<i>Optical Flow</i>)	10
4.2	Subtração de Fundo (<i>Background Subtraction</i>)	11
4.3	Testes com os Dados da Usiminas	13
4.4	Testes com um <i>Dataset</i> Público	15
4.5	Comparação Entre os Histogramas	15
5	CONCLUSÕES	17
6	REFERÊNCIAS BIBLIOGRÁFICAS	18

1 Introdução

A Resolução 382/2006 do Conselho Nacional de Meio Ambiente define que emissões fugitivas são lançamentos difusos na atmosfera de qualquer forma de matéria sólida, líquida ou gasosa, efetuada por uma fonte que não possui dispositivo projetado para dirigir ou controlar seu fluxo. É importante salientar que emissões fugitivas resultam em perdas econômicas, além de ambientais, devido aos produtos contidos em tais emissões.

Mediante isso, um problema que foi apresentado pela Usiminas é de que a empresa não conseguia identificar as emissões fugitivas, de uma forma rápida e eficiente, uma vez que dependiam da análise de um funcionário que visualizava as imagens gravadas no dia anterior e classificava as emissões ocorridas dentro dos três níveis de classificação, sendo então este um processo lento, visto que alguma emissão poderia passar despercebida por esse funcionário. Ademais, os vídeos que eles utilizam para analisar os vazamentos, assim como os que foram cedidos, apresentam baixa resolução e pouca visibilidade, não sendo possível distinguir o tipo de emissão ao avaliar a imagem apenas por um ângulo, uma vez que parte da qualidade dos vídeos é perdida ao extrair do sistema da empresa.

A solução deste problema é fundamental para Usiminas, assim como para outras siderúrgicas, uma vez que há a necessidade de uma maior acurácia na distinção dos tipos de emissão, como por exemplo, se uma emissão é de algum gás nocivo ou se é de vapor.

As emissões são classificadas em níveis de intensidade visual de acordo com a experiência da equipe de gerenciamento de emissões atmosférica da Gerência Geral de Meio Ambiente e das áreas operacionais.

- GRAU 1: Pequena Intensidade
- GRAU 2: Média Intensidade
- GRAU 3: Elevada Intensidade

As metas basearam-se em torno da tentativa de sanar o problema. Sendo assim, elas eram mudadas a cada 15 dias, quando as reuniões de equipe ocorriam, juntamente com pessoas de diferentes áreas da empresa (meio ambiente, recursos humanos e inovação).

Alguns problemas enfrentados para solucionar o problema da empresa foram: conseguir analisar os vídeos com clareza, uma vez que as imagens não possuem muita nitidez; conseguir um bom *dataset* de imagens, para utilizar no treinamento do algoritmo de aprendizado. Entretanto, ir até a Usiminas, no polo de Ipatinga, ajudou na visualização do real problema que seria solucionado, esclarecendo dúvidas e dando maior embasamento para o que poderia ser feito.

2 Referencial Teórico

Tendo ciência completa do problema apresentado, sobre as emissões fugitivas que ocorrem na cadeia produtiva do aço, em Ipatinga (Minas Gerais), começou-se uma pesquisa de artigos científicos em busca de alguma tecnologia já implantada que pudesse ajudar a realizar essa detecção automatizada. Vários métodos foram achados, porém cada um com sua peculiaridade, o que deixava inviável a aplicação de tais métodos no projeto.

Uns dos métodos mais achados foi com a utilização espectrométrica de imagens, como por exemplo com radiação infravermelho, onde uma ou mais áreas específicas são monitoradas para detecção de materiais gasosos, particularmente poluentes por coleta de radiação de fundo (visível e infravermelha) presente na área selecionada. Analisar a radiação coletada com um espectrômetro e disponibilizar os resultados de tal análise são uma forma compreensível para a pessoa ou dispositivo monitorar a área em busca de materiais gasosos específicos, como é dito no trabalho de SANDRIDGE^[13].

Uma das tecnologias também encontrada foi a utilização de sensores, como é citado no trabalho de FINE^[5], onde é utilizado um sensor de gás oxidante e um semicondutor, e o gás alvo interage com a superfície do filme de óxido de metal (geralmente através de íons de oxigênio adsorvidos na superfície), resultando em uma mudança na concentração do material. Esta mudança na concentração da portadora de carga serve para alterar a condutividade (ou resistividade) do material. Um semicondutor do tipo n é aquele em que os portadores de carga majoritária são elétrons e após interação com um gás redutor, ocorre um aumento na condutividade. Por outro lado, um gás oxidante serve para esgotar a camada sensorial de elétrons de carga, resultando em uma diminuição na condutividade.

Porém, no projeto não havia como ser aplicado os estudos acima, já que são necessários equipamentos específicos, como câmera térmica e sensores de gases, e a proposta inicial do projeto era utilizar os acessórios já existentes na empresa.

Com isso, vendo que a literatura não possui algo implementado que possa assim também ser utilizado no projeto, começamos a procurar uma solução para detecção de movimentos diversos, para que ao final, possamos fazer alterações e colocar em prática. E assim encontramos uma solução que usa a técnica de fluxo óptico (*Optical Flow*) para detecção de vazamentos de gases fugitivos. Dois dos trabalhos citados são por CHUNYU^[3] (utilizado para detecção de fumaça), e RINSURONGKAWONG^[4] (utilizado para detecção de fogo), que falam que o resultado do fluxo óptico é uma aproximação do campo de movimento. O método proposto começa com movimentações de quadros de pixels em regiões do vídeo, que são determinados

por um método de estimativa de fundo. Então, uma implementação piramidal *Lucas–Kanade* de recurso *tracker* é proposto para calcular o fluxo óptico de regiões determinadas. A média e a variância da velocidade óptica dos pontos dos cantos é calculada, o que chamamos de recursos de fluxo óptico, e são usados para diferenciar fumaça de alguns outros objetos em movimento. Após a explicação do funcionamento, são mostrados exemplos em execução, onde um deles é em uma estrada, e em que o algoritmo apresentado no artigo é implementado usando Visual C++ e biblioteca OpenCV, e todos os vídeos são padronizados para 320X240 pixels. Assim o algoritmo pode extrair o vetor de velocidade de movimento em cada ponto do recurso e indicar a direção e coordenadas do movimento. Isso sendo aplicado também em movimento de pessoas e fumaças.

Um outro trabalho que similarmente fala do fluxo óptico é citado no artigo de HSU^[2], que atualmente colabora com uma comunidade local no monitoramento e documentação de emissões fugitivas de uma refinaria de coque. O projeto apresenta uma ferramenta de visão computacional para detectar emissões de fumaça industrial e gerar imagens animadas automaticamente, o que reduz significativamente a carga de trabalho dos funcionários. A tarefa é detectar quadros de uma câmera estática contendo fumaça, excluir os que tenham vapor e sombra, identificar os quadros iniciais e finais das emissões, e produzir animações, imagens que podem incluir fumaça. Nos trabalhos de GUBBI^[6] e CHEN^[7], é desenvolvido um algoritmo para detectar fogo no estado inicial, para evitar maiores danos na área. Já no trabalho de FAVORSKAYA^[8], são utilizadas sequências de vídeos para detectar movimentações anômalas de fumaça. No trabalho de TÖREYIN^[9], são utilizadas imagens de câmeras paradas para que haja uma maior nitidez.

Para tentar diminuir os erros da detecção, o trabalho de FRIZZI^[11] propõe reduzir tais erros com a utilização de algoritmo de aprendizado para extrair os recursos úteis, com as redes neurais convolucionais (CNN). Outra maneira achada para diminuir os fatores de erro foi o trabalho de XU^[12], onde sinteticamente são produzidas imagens adequadas de fumaça sintéticas com uma ampla variação na forma da fumaça, além das condições de fundo e iluminação, assim aumentando o banco de dados do algoritmo, e diminuindo as chances de erro.

3 Metodologia

Após uma análise sucinta do problema e dos dados disponíveis, foi verificado que problemas referentes à compactação dos vídeos poderiam ser um empecilho no processo de detecção e classificação das emissões fugitivas. Devido a tais problemas, foram realizados experimentos utilizando dois métodos existentes na literatura, com o objetivo de verificar qual é mais adequado aos dados disponíveis: Fluxo Óptico (*Optical Flow*) e Subtração de Fundo (*Background Subtraction*).

3.1 Fluxo Óptico (*Optical Flow*)

Um dos métodos experimentados para detecção de emissões fugitivas é o *Optical Flow* (fluxo óptico). De acordo com MOTA^[1], estimar o movimento presente em uma imagem ou em um vídeo consiste em medir a projeção 2D no plano da imagem de um movimento real 3D. O fluxo óptico, movimento 2D, pode ser definido como o campo de velocidades descrevendo o movimento aparente das intensidades dos pixels da imagem sob a hipótese de conservação de luminância, ou seja, que a luminância de um ponto físico da sequência de imagens não varia durante o tempo. Para que seja determinado o movimento de um objeto, o mesmo deve ser observado através dos frames, de onde são obtidos os gradientes de movimento. Se o gradiente de intensidade no frame é orientado em uma única direção, o movimento é percebido como normal ao contorno do objeto, porém a combinação de informações de diferentes orientações dos gradientes permite encontrar o movimento real do objeto.

Neste projeto foi utilizada a linguagem de programação Python, juntamente da biblioteca multiplataforma OpenCV, muito utilizada em problemas de Visão Computacional e Processamento Digital de Imagens. Os resultados de fluxo óptico dos vídeos disponibilizados foram gerados através da função `calOpticalFlowFarneck`, que calcula o fluxo óptico de cada pixel utilizando o algoritmo de FARNEBÄCK^[14], algoritmo este que realiza estimativas de movimento baseadas em expansão polinomial, sendo que a ideia da expansão polinomial é aproximar a vizinhança de cada pixel a um polinômio. Também foram realizados testes para verificar quais parâmetros da função `calOpticalFlowFarneck` são mais adequados para os vídeos trabalhados.

3.2 Subtração de Fundo (*Background Subtraction*)

O segundo experimento realizado utilizou o método de Subtração de Fundo (*Background Subtraction*). Este método baseia-se na remoção dos objetos que são estáticos, em um dado período de tempo, do restante da cena, sobrando apenas os itens que estão se movendo no vídeo. Existem diversos algoritmos para subtração de fundo, e o escolhido utiliza um modelo de mistura de gaussianas (*BackgroundSubtractorMOG2*), por ser o mais robusto dentro da biblioteca OpenCV. O modelo de mistura de gaussianas para subtração de fundo é proposto por ZIVKOVIC^{[15][16]}, e utiliza a ideia de que um frame de um vídeo influencia uma quantidade limitada dos frames seguintes, e à medida que o tempo passa essa influência diminui exponencialmente, até que não exista. No GMM (*Gaussian Mixture Model* - Modelo de Mistura de Gaussianas), uma quantidade pré-determinada de frames, anteriores ao frame corrente, recebe pesos que determinam sua influência no frame atual, e utilizando estes pesos, as gaussianas dos frames anteriores são misturadas e o resultado subtraído do frame corrente, restando apenas o movimento dos objetos em cena.

O método de Subtração de Fundo utilizando o Modelo de Mistura de Gaussianas mostrou-se interessante para o problema tratado por dois motivos: é temporalmente adaptativo (leva em consideração uma quantidade limitada de frames anteriores, sendo robusto às mudanças de iluminação do dia), e é muito utilizado em aplicações de tempo real.

3.3 Processamento de Imagens

Após a tarefa de isolar as áreas em movimento para cada frame dos vídeos cedidos pela empresa, foi realizado um trabalho de processamento das imagens, de forma que as áreas em que foram identificados os movimentos fossem marcadas com o valor binário 1, representando a presença de movimento, e as áreas restantes foram marcadas com o valor binário 0, indicando a ausência de movimento. Depois, cada frame dos vídeos teve seu histograma de cores equalizado, a fim de que as variações de iluminação não interferissem na predição do algoritmo de aprendizado utilizado, e a máscara de movimentos (imagem binarizada de movimentos) foi aplicada em seu respectivo frame equalizado, com a finalidade de isolar as áreas com a possível presença de fumaça.

3.4 Extração de *Features*

Pelas características dos aspectos documentados pela Usiminas, é conveniente que as *features* utilizadas sejam referentes às cores nas imagens, e os dois descritores mais citados

na literatura neste caso são o *Color Coherence Vector (CCV)*^[19] e o *Border/Interior Color (BIC)*^[20], porém nenhum dos dois descritores são implementados nas bibliotecas de Computação Visual mais amplamente utilizadas. A alternativa, portanto, seria a utilização de Histogramas de Cores Equalizados. Existem duas maneiras de se utilizar um histograma de cores como *feature*: com as bandas de cores sendo tratadas independentemente (um histograma para cada banda de cor); e com as bandas de cor correlacionadas, de maneira que a interação entre as cores em um pixel também seja relevante.

As bibliotecas de Computação Visual utilizadas disponibilizam algoritmos para obtenção do histograma de cores por banda, porém não disponibilizam algoritmos para obtenção do histograma de cores correlacionado. Portanto, foi necessária a implementação de um algoritmo para obtenção de tal histograma. O histograma de cores correlacionado é importante pois as cores que notamos na natureza não são formadas pelas bandas de cores agindo independentemente, e sim pela relação entre elas. Este tipo de histograma cresce exponencialmente, mas pode oferecer informações de cor preciosas para as classificações necessárias.

3.5 Aprendizado de Máquina

O algoritmo de aprendizado de máquina utilizado foi o *Decision Tree* (Árvore de Decisão), por resultar em um modelo leve e que se adaptou bem aos dados gerados. O modelo de árvores de decisão é um dos mais simples em *Machine Learn*, porém foi suficiente para a demanda deste projeto. Uma das maiores vantagens é que quando colocado na aplicação final, consome muito pouca memória, em geral menos que um *megabyte*. Em sistemas de monitoramento, um algoritmo leve pode ser crucial, já que as aplicações em si já são pesadas e o tempo de processamento já é um gargalo.

Foi definido que 70% do *dataset* para treino do algoritmo e 30% do *dataset* para avaliação (ou teste). Para separar o *dataset* em treino e teste foram considerados todos os *frames*, e foram contabilizados quantos exemplos de cada classe existiam, para todos os modelos propostos. Por fim, foram escolhidos os itens aleatoriamente dentro do *dataset* para entrarem em teste e treino, respeitando que 70% dos itens de cada classe ficasse no treino e 30% ficasse no teste. Para isto foi utilizada uma variável aleatória de valor entre 0.0 e 1.0, sendo que quando o valor estivesse abaixo de 0.7 significaria que o item deveria ser colocado no conjunto de treino, e caso contrário o item deveria ser colocado no conjunto de teste.

Três modelos preditivos foram propostos para o problema da Usiminas: um modelo para identificação de fumaça, um modelo para classificação quanto ao aspecto da fumaça e um modelo para classificação da intensidade da fumaça, todos seguindo os padrões utilizados pela Usiminas. Para os modelos de classificação de aspecto e intensidade haviam duas situações:

classificar depois de o modelo para identificação de fumaça ter predito presença de fumaça e classificar independente do modelo de identificação. No segundo caso, haveria uma outra classe a ser levada em conta nos modelos, que é a classe nula, supondo não existência de fumaça. Assim haveriam duas categorias de modelos para aspecto e intensidade: pós-deteção e pré-deteção.

Os algoritmos de aprendizado foram treinados e testados utilizando tanto o histograma descorrelacionado quanto o histograma correlacionado como features. Os resultados de acurácia dos modelos utilizando estes dois tipos de histogramas serão apresentados mais adiante.

3.6 Tecnologias Utilizadas

- Linguagem de programação Python
- Biblioteca Multiplataforma OpenCV
- Biblioteca Scikit-Image
- Biblioteca Scikit-Learn
- Google Colab Research

4 Resultados

Foi obtido um algoritmo capaz de informar a existência de emissões fugitivas nas cenas de um vídeo, o horário em que tais emissões ocorreram e suas classificações quanto ao aspecto e a intensidade da fumaça (seguindo os dados de avaliação disponibilizados pela Usiminas). O algoritmo desenvolvido é robusto aos ruídos provenientes da compactação inerente, gerando resultados condizentes com o que é visualizado nos vídeos disponibilizados.

A seguir serão mostrados os resultados obtidos para cada uma das partes que compõe o algoritmo completo, e os resultados dos respectivos testes aplicados.

4.1 Fluxo Óptico (*Optical Flow*)

Foram realizados testes para verificar quais parâmetros da função *calOpticalFlowFarneck* (utilizado na extração do Fluxo Óptico) são mais adequados para os vídeos trabalhados, e na tabela 1 estão sendo mostrados os parâmetros modificados juntamente com sua utilidade e a faixa de valores testados. Após o teste de cada conjunto de parâmetros, o vídeo resultante foi binarizado (imagem em preto e branco) para melhor visualização dos resultados (*pipeline* experimental na Figura 1).

Tabela 1 - Conjunto de parâmetros experimentados

Parâmetro	Descrição	Valores
<code>pyr_scale</code>	Escalamento dos frames	0.1 - 1.0
<code>levels</code>	Número de reescalamentos no processo	1 - 10
<code>winsize</code>	Tamanho da janela de detecção	1 - 30
<code>iterations</code>	Número de repetições do algoritmo	1 - 5
<code>poly_n</code>	Número de pixels de vizinhança	5 ou 7
<code>poly_sigma</code>	Desvio padrão da gaussiana usada como base para a expansão polinomial	1.1 - 1.5

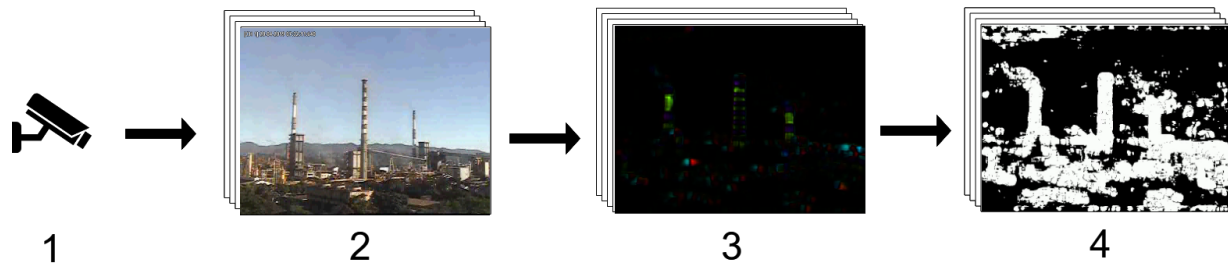


Figura 1 – Pipeline Experimental para o *Optical Flow*:

1 - Aquisição dos vídeos; 2 - Entrada dos vídeos no algoritmo; 3 - Identificação dos fluxos ópticos; 4 - Binarização do resultado.

Os resultados dos testes com fluxo óptico não foram satisfatórios, uma vez que os vídeos resultantes possuíam muito ruído, consequência do tipo de compactação utilizado. Não foi possível a separação de fumaça dentro da cena em nenhum dos conjuntos de parâmetros testados, concluindo que o método não é adequado ao problema.

4.2 Subtração de Fundo (*Background Subtraction*)

Os experimentos com o *Background Subtractor* foram executados utilizando uma temporalidade de 500 *frames*, correspondendo a aproximadamente 17 segundos de vídeo (o valor de 500 *frames* foi escolhido após um teste com valores entre 50 e 1000, e a quantidade de 500 *frames* mostrou-se o melhor valor). Sobre o vídeo resultante foi realizada uma suavização, utilizando um filtro de mediana, e outra binarização, para que o ruído ainda existente nos vídeos fosse excluído (ruído proveniente da compactação inicial dos vídeos).

Após a execução da subtração de fundo, é equalizado o histograma de cores do *frame* correspondente. Equalizar o histograma de cores significa obter a máxima variância do histograma de uma imagem, obtendo assim uma imagem com o melhor contraste, sendo que o contraste é uma medida qualitativa e que está relacionada com a distribuição dos tons em uma imagem^[21]. O processo de equalização aumenta a variabilidade de cores na imagem, destacando diferenças de coloração. A equalização do histograma também é interessante pela variação de luminosidade que ocorre no ambiente durante o dia, sendo que a imagem por vezes fica muito clara ou muito escura devido à luz do sol no horário, e a equalização permite que tais mudanças relativas à iluminação sejam atenuadas.

Por fim, é realizada uma multiplicação entre o *frame* cujo histograma foi equalizado e o resultado do processo de Subtração de Fundo. Com esta multiplicação, as cores internas à fumaça são isoladas do restante da imagem, podendo ser utilizadas para a predição do aspecto da emissão. Neste processo, o resultado da Subtração de Fundo é chamado de máscara, já que

a multiplicação da imagem equalizada com tal máscara descarta as áreas que não possuem movimento.

Com o resultado da multiplicação do *frame* equalizado pela máscara de movimento, já pode ser realizada a extração de *features*, utilizando o algoritmo para obtenção de histogramas por banda ou o algoritmo para obtenção do histograma correlacionado. A seguir é apresentado o *pipeline* experimental para o *Background Subtraction* (Figura 2) e um exemplo de aplicação do algoritmo em um vídeo (um *frame* em específico) (Figura 3).

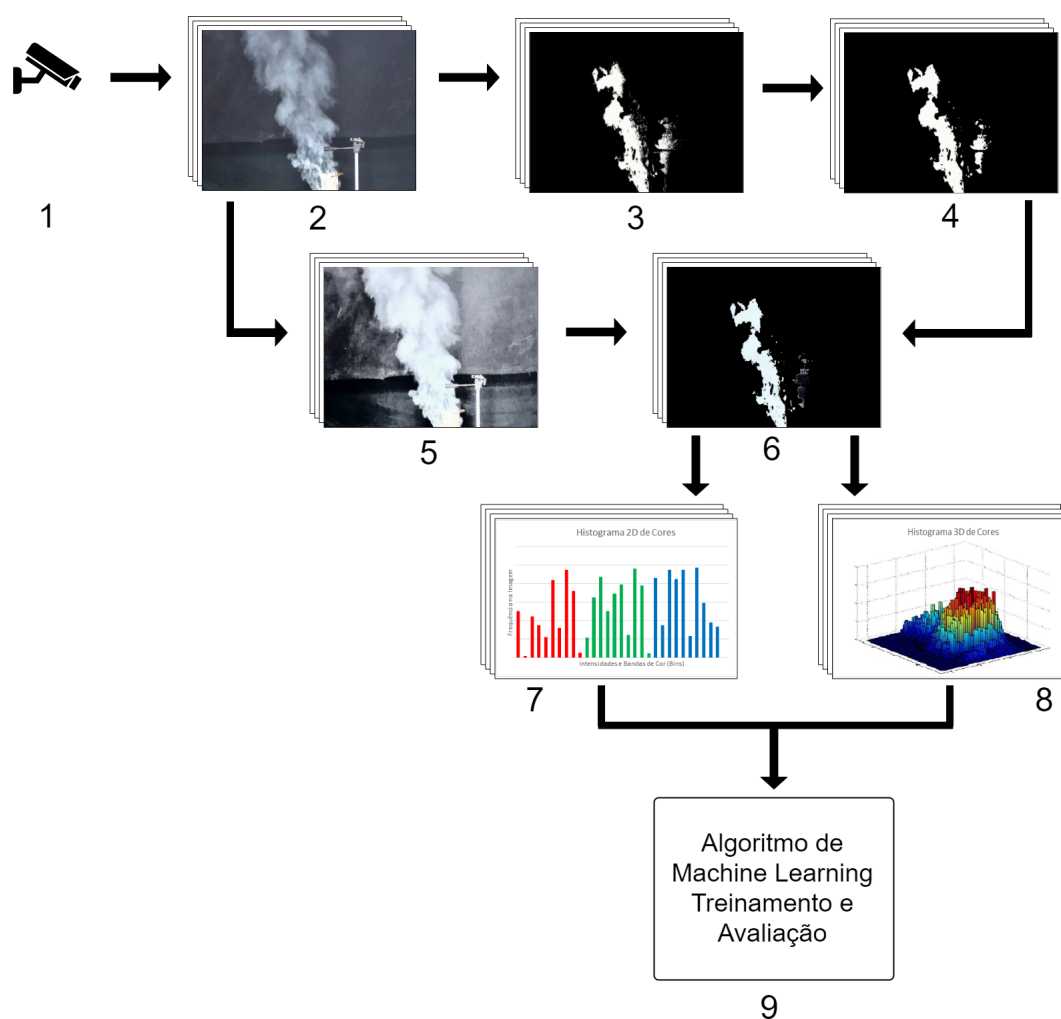


Figura 2 – Pipeline Experimental para o *Background Subtractor*:

1 - Aquisição dos vídeos; 2 - Entrada dos vídeos no algoritmo; 3 - Processo de Subtração de Fundo; 4 - Suavização do Resultado (Geração de Máscara); 5 - Equalização do Histograma de Cor dos frames; 6 - Multiplicação do frame equalizado pela máscara; 7 - Extração do Histograma por Banda; 8 - Extração do Histograma de Cores Correlacionado; 9 - Treinamento do algoritmo de aprendizagem.

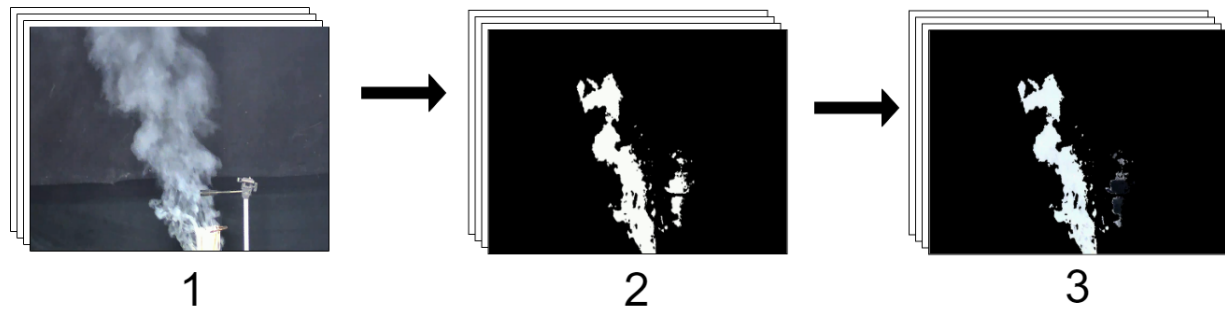


Figura 3 – Exemplo de Aplicação do Algoritmo:

- 1 - Imagem Original;
- 2 - Máscara de Movimento;
- 3 - Regiões com Movimento Isoladas.

Como é possível visualizar, este método consegue separar bem o que está movimentando-se do restante da imagem. Nesta fase ainda não é possível dizer que o que se movimenta é fumaça, mas já é uma indicação de em qual região a fumaça pode estar ocorrendo. Verificamos aqui que este método é bem adequado aos dados utilizados.

4.3 Testes com os Dados da Usiminas

Foram geradas *features* utilizando o histograma de cores descorrelacionado e o histograma de cores correlacionado. O tamanho escolhido para o histograma de cores descorrelacionado foi de 192 valores (64 valores para cada banda de cor), e para o histograma de cores correlacionado foi de 64 valores (4 faixas de valores para cada banda de cor, tornando-se $4^3 = 64$). Os valores foram escolhidos por um padrão de custo-benefício para cada algoritmo, sendo que valores maiores não geravam resultados tão melhores, enquanto que o tempo gasto na geração era razoavelmente maior.

Durante os treinos, para os modelos de classificação quanto ao aspecto e à intensidade, foram treinados também modelos supondo que a detecção da fumaça no modelo de identificação já tenha sido realizada (modelos pós-deteção). Além dos modelos pós-deteção, foram treinados os modelos que lidam com uma classe nula, que corresponde à ausência de fumaça (pré-deteção).

Assim, temos na tabela 3 os resultados de acurácia para os modelos que utilizaram o histograma de cores descorrelacionado como *feature*. Já na tabela 4 temos os resultados de acurácia para os modelos que utilizaram o histograma de cores correlacionado como *feature*. A coluna *Max Árvore* corresponde ao tamanho máximo da árvore de decisão que gerou o modelo.

Tabela 3 - Resultados de Acurácia Utilizando o Histograma Descorrelacionado

Modelo	Acurácia de Treino	Acurácia de Teste	Max Árvore
Identificação	90,3431957%	81,5932941%	5
Aspecto Pré-deteccção	90,2644581%	79,9498584%	4
Intensidade Pré-deteccção	89,0104900%	79,0054687%	5
Aspecto Pós-deteccção	99,5671371%	96,3577635%	4
Intensidade Pós-deteccção	97,8356854%	95,0333138%	7

Tabela 4 - Resultados de Acurácia Utilizando o Histograma Correlacionado

Modelo	Acurácia de Treino	Acurácia de Teste	Max Árvore
Identificação	97,7499916%	99,9641800%	61
Aspecto Pré-deteccção	97,7449430%	99,9673053%	78
Intensidade Pré-deteccção	97,6400303%	99,9649013%	56
Aspecto Pós-deteccção	99,9624366%	100,000000%	77
Intensidade Pós-deteccção	99,8379198%	100,000000%	90

Como já podemos perceber, o histograma de cores correlacionado obteve resultados superiores em relação ao histograma de cores descorrelacionado. Na tabela 5 temos a comparação entre as acurácias de teste obtidas com os histogramas correlacionado e descorrelacionado, para todos os tipos de modelos gerados.

Tabela 5 - Comparação dos Resultados de Acurácia de Teste para os Histogramas Correlacionado e Descorrelacionado

	Modelo	Histograma	
		Descorrelacionado	Correlacionado
Acurácia de Teste	Identificação	90,3431957%	97,7499916%
	Aspecto Pré-deteccção	90,2644581%	97,7449430%
	Intensidade Pré-deteccção	89,0104900%	97,6400303%
	Aspecto Pós-deteccção	99,5671371%	99,9624366%
	Intensidade Pós-deteccção	97,8356854%	99,8379198%

Os resultados foram excelentes, principalmente quando observamos que estes resultados foram obtidos para a predição de cada frame em separado de um vídeo. Precisa ser levado em conta também que todos os vídeos do *dataset* privado da Usiminas são de câmeras paradas, porém na próxima seção será tratado o caso de um *dataset* mais diversificado.

4.4 Testes com um *Dataset* Público

Além de ser testada no *dataset* privado de vídeos disponibilizado pela Usiminas, a metodologia desenvolvida foi testada em um *dataset* público de vídeos, cujos *frames* são categorizados como contendo ou não fogo e fumaça (*FiSmo Dataset*^{[22][23]}). Foram utilizados 79 vídeos do *FiSmo Dataset*, sendo selecionados os vídeos dos *subdatasets* *FireVid* e *RESCUER Video Dataset*, por estes possuírem *labels* para cada um dos *frames* de cada um dos vídeos. Este *dataset* categoriza os *frames* como "fire"(quando há fogo e fumaça) e "nofire"(quando não há fogo e fumaça). Também foi colocada uma terceira categoria, a "ignore", que se refere àqueles *frames* que são difíceis de determinar.

A principal diferença deste *dataset* em relação ao *dataset* privado da Usiminas é a existência de vídeos sem câmera parada, o que poderia ser um desafio à mais. A grande parte dos vídeos são de câmera em movimento, havendo vídeos registrados através de helicópteros e vídeos de câmeras movimentando-se muito rapidamente, além de que os vídeos foram registrados em diferentes ambientes, diferentes tipos de câmeras e em diferentes resoluções.

O método foi testado no *FiSmo Dataset* sem que nenhuma modificação fosse realizada, e os resultados foram os apresentados na tabela 5:

Tabela 6 - Resultados de Acurácia do Método no *FiSmo Dataset*

Features	Acurácia de Teste	Acurácia de Treino	Histórico
Histograma Descorrelacionado	83,3949900%	87,6613670%	500
Histograma Correlacionado	94,2606490%	98,6750917%	450
Descorrelacionado + Correlacionado	94,7415707%	98,7770077%	500

A coluna *Histórico* refere-se ao número do histórico de frames que é utilizado no algoritmo de *Background Subtraction* do melhor modelo encontrado. Como é possível observar, mesmo em um *dataset* não regular como o da Usiminas, com câmeras em movimento e em diferentes ambientes, o método apresentou ótimos resultados. Isto comprova que o método é utilizável em diversas situações, e não apenas para identificar emissões fugitivas, mas para identificar fogo também.

4.5 Comparação Entre os Histogramas

O histograma de cores correlacionado, apesar de descrever melhor o que ocorre na natureza (as cores são formadas pela correlação entre os canais RGB, e não com tais canais agindo separadamente), a quantidade de itens do histograma cresce exponencialmente de acordo com o número de divisões de níveis que será utilizado para cada banda. Por exemplo, se os níveis

de cor para cada banda forem resumidos em 4, então o histograma final terá 64 valores ($4 \times 4 \times 4 = 64$), mas se a quantidade de níveis for resumida em 6, o histograma final passará a ter 216 valores.

Além disto, o cálculo de um histograma deste tipo é caro, tendo custo computacional $O(n^3)$, enquanto que o histograma não correlacionado possui custo $O(n)$, sendo o n a quantidade de níveis de cor para o qual os canais serão reduzidos. Ou seja, apesar de o histograma correlacionado obter ótimos resultados, é necessária uma implementação otimizada e encontrar um valor de níveis de cor para cada banda adequado, com a finalidade de não tornar o algoritmo desenvolvido neste projeto, que por si só já é muito caro, ainda mais custoso.

As primeiras implementações do algoritmo para histogramas de cores correlacionados chegavam a levar um minuto para processar uma imagem de tamanho 1280×720 , porém, depois de algumas otimizações, os seguintes resultados foram obtidos (para uma imagem de tamanho 1280×720 em uma máquina com GPU):

Tabela 7 - Tempo de Geração para os Diferentes Tipos de Histogramas

Tipo de Histograma	Tamanho	Tempo (segundos)
Descorrelacionado	192	0.13689804
Correlacionado	8	0.11524177
Correlacionado	27	0.12232065
Correlacionado	64	0.12565374
Correlacionado	125	0.14761591
Correlacionado	216	0.15279388
Correlacionado	343	0.15948081
Correlacionado	512	0.17972422
Correlacionado	729	0.19347262

Desta forma, o histograma correlacionado de tamanho 64 ($4 \times 4 \times 4$) substitui bem o histograma não correlacionado, já que ambos conseguem ser obtidos com um tempo muito próximo. Assim, um histograma correlacionado de tamanho 64 mostra-se mais adequado ao processo desenvolvido neste projeto.

5 Conclusões

O método desenvolvido neste projeto, para identificar e classificar emissões fugitivas, obteve excelentes resultados, mostrou-se muito robusto, rápido e eficaz. Além disto, os testes demonstraram que o método pode ser aplicado não apenas no ambiente da Usiminas, mas também em outras situações que envolvam o monitoramento de fumaça ou fogo. Este projeto pode ter boas consequências além do escopo acadêmico, podendo a metodologia desenvolvida ser integrada em sistemas de monitoramento já instalados.

O próximo objetivo é publicar os resultados aqui apresentados.

6 Referências Bibliográficas

1. MOTA, Virgínia Fernandes et al. Tensor baseado em fluxo óptico para descrição global de movimento em vídeos. 2011.
2. HSU, Yen-Chia et al. Industrial smoke detection and visualization. arXiv preprint arXiv:1809.06263, 2018.
3. CHUNYU, Yu et al. Video smoke recognition based on optical flow. In: 2010 2nd International Conference on Advanced Computer Control. IEEE, 2010. p. 16-21.
4. RINSURONGKAWONG, Suchet; EKPANYAPONG, Mongkol; DAILEY, Matthew N. Fire detection for early fire alarm based on optical flow video processing. In: 2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. IEEE, 2012. p. 1-4.
5. FINE, George F. et al. Metal oxide semi-conductor gas sensors in environmental monitoring. *sensors*, v. 10, n. 6, p. 5469-5502, 2010.
6. GUBBI, Jayavardhana; MARUSIC, Slaven; PALANISWAMI, Marimuthu. Smoke detection in video using wavelets and support vector machines. *Fire Safety Journal*, v. 44, n. 8, p. 1110-1115, 2009.
7. CHEN, Thou-Ho et al. The smoke detection for early fire-alarming system base on video processing. In: 2006 International Conference on Intelligent Information Hiding and Multimedia. IEEE, 2006. p. 427-430.
8. FAVORSKAYA, Margarita; PYATAEVA, Anna; POPOV, Aleksei. Verification of smoke detection in video sequences based on spatio-temporal local binary patterns. *Procedia Computer Science*, v. 60, p. 671-680, 2015.
9. TÖREYİN, B. Uğur; DEDEOĞLU, Yiğithan; CETİN, A. Enis. Wavelet based real-time smoke detection in video. In: 2005 13th European Signal Processing Conference. IEEE, 2005. p. 1-4.
10. YUAN, Feiniu. A fast accumulative motion orientation model based on integral image for video smoke detection. *Pattern Recognition Letters*, v. 29, n. 7, p. 925-932, 2008.

11. FRIZZI, Sebastien et al. Convolutional neural network for video fire and smoke detection. In: IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2016. p. 877-882.
12. XU, Gao et al. Deep domain adaptation based video smoke detection using synthetic smoke images. *Fire safety journal*, v. 93, p. 53-59, 2017.
13. SANDRIDGE, Robert L.; HUNT, Robert N. Remote sensing gas analyzer. U.S. Patent n. 4,795,253, 3 jan. 1989.
14. FARNEBÄCK, Gunnar. Two-frame motion estimation based on polynomial expansion. In: *Scandinavian conference on Image analysis*. Springer, Berlin, Heidelberg, 2003. p. 363-370.
15. ZIVKOVIC, Zoran et al. Improved adaptive Gaussian mixture model for background subtraction. In: *ICPR (2)*. 2004. p. 28-31.
16. ZIVKOVIC, Zoran; VAN DER HEIJDEN, Ferdinand. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, v. 27, n. 7, p. 773-780, 2006.
17. GOLDBERGER, Jacob et al. Neighbourhood components analysis. In: *Advances in neural information processing systems*. 2005. p. 513-520.
18. BREIMAN, Leo. Random forests. *Machine learning*, v. 45, n. 1, p. 5-32, 2001.
19. PASS, Greg; ZABIH, Ramin; MILLER, Justin. Comparing Images Using Color Coherence Vectors. In: *ACM multimedia*. 1996. p. 65-73.
20. STEHLING, Renato O.; NASCIMENTO, Mario A.; FALCÃO, Alexandre X. A compact and efficient image retrieval approach based on border/interior pixel classification. In: *Proceedings of the eleventh international conference on Information and knowledge management*. ACM, 2002. p. 102-109.
21. RERY, Alejandro C.; LIMA, Manoel Eusebio de. *Equalização do Histograma*, 1999. Disponível em: https://www.di.ufpe.br/if143/projetos/99_2/equali/Equalizacao.html. Acesso em: 21 set. 2019.
22. AVALHAIS, L. P. S., RODRIGUES, J., and TRAINA, A. J. M. (2016). Fire detection on unconstrained videos using color-aware spatial modeling and motion flow. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 913–920.

23. CAZZOLATO, M. T., AVALHAIS, L. P. S., Chino, D. Y. T., Ramos, J. S., Souza, J. A., Rodrigues Jr., J. F., and Traina, A. J. M. (2017). FiSmo: A Compilation of Datasets from Emergency Situations for Fire and Smoke Analysis.