

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Tayrone Araujo Machado

**Sistema de análise visual de objetos em vídeos
de vigilância**

Uberlândia, Brasil

2023

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Tayrone Araujo Machado

**Sistema de análise visual de objetos em vídeos de
vigilância**

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: José Gustavo de Souza Paiva

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2023

Tayrone Araujo Machado

Sistema de análise visual de objetos em vídeos de vigilância

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

José Gustavo de Souza Paiva
Orientador

Professor

Professor

Uberlândia, Brasil
2023

Dedico este trabalho primeiramente a Deus, pois foi ele que me deu toda a força para concluir este trabalho, sem ele nada seria possível. Aos meus pais, que são os meus pilares da minha formação como ser humano e a minha namorada por todo apoio.

Agradecimentos

Agradeço a Deus por me dar a vida e perseverança durante todo esse tempo, aos meus pais pelo apoio e o incentivo que serviram de alicerce para as minhas realizações, a minha namorada pelo seu amor e toda ajuda nesse período e ao meu professor orientador pelas valiosas contribuições dadas durante todo o processo.

*“Só se pode alcançar um grande êxito quando nos mantemos fiéis a nós mesmos.
Friedrich Nietzsche ”*

Resumo

Este trabalho visa apresentar e detalhar a construção de um sistema web que foi idealizado com o objetivo de acomodar diversos sistemas de visualização desenvolvidos na Facom-UFU, demonstrando sua aplicabilidade em diferentes contextos. O sistema foi construído de forma modular, permitindo que cada módulo funcione de maneira independente, sem comprometer o funcionamento do sistema como um todo. Para comprovar sua efetividade, foi adicionado um módulo que integra um sistema de análise visual desenvolvido no âmbito de um programa de mestrado na Facom UFU, proporcionando uma integração funcional e interativa no sistema web.

Palavras-chave: sistema web, sistemas de visualização, modularidade, análise visual, integração

Lista de ilustrações

Figura 1 – Processo de visualização da informação	15
Figura 2 – Bounding box	17
Figura 3 – Modulos do sistema web	20
Figura 4 – Diagrama de tecnologias	21
Figura 5 – Passo 1	24
Figura 6 – Passo 2	24
Figura 7 – Passo 3	25
Figura 8 – Passo 4	26
Figura 9 – Diagrama de trabalho do sistema externo	27
Figura 10 – Interface do sistema de análise visual de vídeos de vigilância	29
Figura 11 – Exemplo Appearance Bars View no sistema externo	29
Figura 12 – Exemplo de views no sistema externo	30
Figura 13 – Exemplo de entrada do arquivo de configuração	31
Figura 14 – Exemplo de retorno iFrame	32

Lista de tabelas

Tabela 1 – Exemplo de conteúdo do arquivo de configurações	30
--	----

Sumário

1	INTRODUÇÃO	10
1.1	Objetivos	11
1.2	Justificativa	11
1.3	Metodologia	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Conceitos básicos	14
2.1.1	Visão computacional	15
2.1.2	Deteccção e rastreamento de objetos	16
2.1.3	Critério de deteccção	17
2.2	Trabalhos Correlatos	18
2.2.1	Estratégia para análise visual de trajetórias em vídeos de vigilância aplicando-se técnicas de visualização de informação	18
2.2.2	Identificação de objetos em vídeos de câmera estática	18
2.2.3	Aplicação de Visão Computacional no Auxílio ao Levantamento de Defeitos em Pavimento Rodoviário	19
2.2.4	Deteccção e contagem de plantas utilizando técnicas de inteligência artificial e <i>machine learning</i>	19
3	DESENVOLVIMENTO	20
3.1	Visão geral	20
3.2	Tecnologias	21
3.3	Desenvolvimento do website	23
3.3.1	Arquivo de Configurações	23
3.3.2	Página principal	23
4	RESULTADOS	27
4.1	Sistema de análise visual de vídeos de vigilância	27
4.1.1	Descrição geral do sistema externo	27
4.2	Adequação do sistema externo para utilização no sistema proposto	28
4.2.1	Funcionalidades	29
4.3	Comunicação entre Sistema Web e Sistema Externo	30
5	CONCLUSÃO	33
	REFERÊNCIAS	34

1 Introdução

Há alguns anos, câmeras estão se tornando cada vez mais presentes na rotina das pessoas e estão por toda parte. Existem atualmente diversos sistemas de vigilância que as utilizam para monitorar determinados locais públicos ou privados. Segundo [Pitzer \(2009\)](#) “não há como saber quem está por trás das câmeras, não há como saber o momento em que o vigilante está te observando [...] Isso leva o vigiado a acreditar que está sendo vigiado o tempo todo”. Esta sensação de segurança a todo momento difundiu o uso das câmeras.

Estes sistemas capturam, informações a respeito de objetos presentes em uma determinada cena, permitindo a identificação de eventos de interesse, monitoramento do fluxo de pessoas, carros, objetos, prevenção de possíveis ataques terroristas, roubos e comportamentos incomuns, dentre outras tarefas. De acordo com [Lao, Han e With, de \(2009\)](#), a maioria dos sistemas de vigilância desenvolvidos focam em compreender eventos por meio do estudo de trajetórias e posições de pessoas nas cenas. Porém, segundo [Garcia \(2007, p.113\)](#), seres humanos não são bons em detectar eventos, devido ao fato de não conseguirem manter a concentração em uma atividade durante longos períodos de tempo, acarretando em um aumento de falhas no processamento de vigilância. Além disso para que um sistema de vigilância seja eficaz, é necessário que haja um esforço considerável por parte do vigilante, pois os vídeos de monitoramento podem ser muito extensos, dificultando cada vez mais a detecção de eventos em vídeo.

Desta forma, para reduzir as falhas presentes causadas por ineficiências de um agente humano, existe necessidade em automatizar o processo de monitoramento. Diversas áreas de pesquisas computacionais propõem estratégias para lidar com problemas relacionados a análise de vídeos de vigilância. Uma delas é a visão computacional, ciência que estuda e desenvolve métodos teóricos e algorítmicos para extração de informações oriundas de imagens e vídeos. De acordo com [Parker \(1993, p.1\)](#), o objetivo da visão computacional é desenvolver formas de um computador interpretar imagens para alguma aplicação útil.

Segundo [Samuel \(1959\)](#) outra estratégia utilizada é o aprendizado de máquina, responsável por métodos e algoritmos que possuem a capacidade de aprender com informações obtidas através de imagens ou uma base de dados. Aprendizado de máquina pode ser definido como o campo de estudo de algoritmos e modelos estatísticos que computadores utilizam para executar tarefas sem serem explicitamente programados, mas através do seguimento de padrões ou utilização de inferência. As técnicas de aprendizado de máquina empregam um princípio de inferência denominado indução, no qual obtém-se conclusões

genéricas a partir de um conjunto particular de exemplos. A indução é caracterizada pelo raciocínio originado em um conceito específico e generalizado, ou seja, da parte para o todo. Segundo [Rezende et al. \(2003\)](#) na indução, um conceito é aprendido efetuando-se inferência indutiva sobre os exemplos apresentados. Portanto, as hipóteses geradas através da inferência indutiva podem ou não preservar a verdade. O aprendizado de máquina pode ser de três modalidades: supervisionado, não-supervisionado e semi-supervisionado.

Levando em consideração as informações descritas, existe um trabalho desenvolvido no âmbito de um projeto de mestrado da Faculdade de Computação da Universidade Federal De Uberlândia, cujo objetivo principal é aplicar estratégias visuais para auxiliar na análise da movimentação de objetos que compõem uma cena monitorada por câmeras de vigilância em um determinado período de tempo. O projeto envolve a utilização de informações providas por técnicas de detecção e rastreamento de objetos em cena. Segundo [Li et al. \(2013\)](#) as tarefas relacionadas ao rastreamento de objetos visam identificar a localização de objetos presentes durante toda a execução do vídeo, com o objetivo de compreender comportamento dos objetos em cena e analisar aspectos como velocidade e direção do movimento. Esse processo precisa ser bem executado, sob o risco de que os resultados produzidos não reflitam o que realmente ocorreu em cena.

Existem desafios relacionados a este projeto pois quando capturamos imagens provenientes das câmeras há variáveis que devemos levar em conta como o foco, a luz, o ângulo em que se enquadra a cena, posição, ambiente, qualidade da imagem, quantidade de objetos em cena entre outros. Portanto o método é dependente de uma boa técnica de identificação e rastreamento.

Assim este trabalho propõe que as técnicas de análise visual desenvolvidas no projeto de mestrado supracitado sejam inseridas em um sistema *web* que garanta ao vigilante acesso fácil e efetivo à todo o processo, desde o fornecimento do vídeo a ser analisado até a interação com os *layouts* produzidos.

1.1 Objetivos

Este trabalho tem como objetivo desenvolver um sistema *web* que acomode os mais diversos sistemas de análise visual de objetos em vídeos de vigilância, bem como suas configurações e dependências.

1.2 Justificativa

Segundo [Nguyen, Li e Ogunbona \(2016\)](#) os seres humanos tem uma capacidade considerada indiscutível em detectar e identificar objetos com apenas algumas pistas. Os sistemas de detecção de objetos, entre eles os sistemas de vídeo-vigilância inteligente

buscam, com certa precisão, realizar estas tarefas de forma autônoma, sem a necessidade de seres humanos para sua operação, como ocorre em sistemas convencionais. No entanto, os resultados atuais apresentados na literatura, mostram que os algoritmos automatizados estão longe de poderem ser comparados com as habilidades humanas.

Toda essa dificuldade em alcançar a eficácia humana para a tarefa de detecção de objetos, se deve a um conjunto de fatores que vão desde a representação computacional dos objetos, ao ambiente em que os objetos se encontram. Outro desafio para a vídeo-vigilância inteligente ocorre devido as configurações de câmera, desde as propriedades de *hardware* até as configurações de instalação como posição e foco; tudo isso pode prejudicar a qualidade em que os objetos serão capturados e descritos computacionalmente [Leite \(2018\)](#). Além disso há diversas dificuldades em detectar um objeto através de uma câmera como as variações na iluminação, diversas poses do corpo humano, vestimentas, cores, ângulos, campo de visão, multidoes. Existem muitos fatores que dificultam este trabalho.

Motivado por toda a aplicabilidade desta tarefa, esta pesquisa propõe uma ferramenta para lidar com todos os desafios destacados até aqui, no contexto da identificação de objetos em vídeos de câmera estática.

1.3 Metodologia

Este trabalho propõe a criação de um sistema *web* capaz de acomodar o projeto de mestrado de Cibele Marafonseca [Fonseca \(2018\)](#), de forma a auxiliar o processo de análise e visualização dos dados originados da análise dos vídeos. O desenvolvimento da pesquisa é composto pelas etapas descritas a seguir:

1. Pesquisa bibliográfica: Estudo de trabalhos relacionados, que serão utilizados como referência.
2. Definição de métodos, bibliotecas e tecnologias: Consiste em encontrar métodos consolidados que possam resultar em análises consistentes para a extração de objetos, além de definir a linguagem de programação que será utilizada para a criação do sistema *web*.
3. Implementação: Desenvolvimento do sistema *web* para acomodar as técnicas de identificação e rastreamento de objetos, bem como a comunicação dos resultados produzidos nesse processo com as técnicas do trabalho de mestrado de Cibele Marafonseca [Fonseca \(2018\)](#), de forma a integrar os módulos de carregamento e processamento de vídeos.

2 Fundamentação Teórica

Este capítulo busca trazer uma introdução ao leitor aos conceitos teóricos que compõem as estratégias de análise visual de objetos em vídeos de vigilância utilizadas neste trabalho.

2.1 Conceitos básicos

Serão apresentados alguns conceitos necessários para uma melhor compreensão relacionado a análise visual de trajetória em vídeos de vigilância. Entre estes conceitos, segundo [Fonseca \(2018\)](#) estão:

- **Câmera:** é um equipamento dotado de mecanismos que captura imagens em tempo real;
- **Vídeo:** técnica de reprodução eletrônica de imagens em movimento;
- **Frame:** unidade de imagem em um conjunto de imagens que formam um vídeo;
- **Cena:** campo de abrangência registrado pela câmera, onde geralmente os eventos de interesse ocorrem;
- **Foreground:** região do vídeo que contém objetos em movimento;
- **Background:** toda região que compõe o vídeo, com exceção do *Foreground*;
- **Trajetoária:** caminho percorrido por um objeto em cena;
- **Rastreamento de objetos:** envolve a correspondência de objetos no *Foreground* detectados entre quadros consecutivos, usando diferentes recursos do objeto, como movimento, velocidade, cor e textura. O rastreamento de objetos é o processo para rastrear o objeto ao longo do tempo, localizando sua posição em todos os *frames* do vídeo no sistema de vigilância;
- **Evento:** ocorrência de uma atividade em um local específico durante um intervalo de tempo específico;
- **Layout:** mapeamento gerado a partir de técnicas de visualização de informação;
- **Vídeo de vigilância:** são vídeos usados com o objetivo de observar uma área;

2.1.1 Visão computacional

Segundo [Oliveira \(2018\)](#) A visão computacional tem como objetivo treinar e ensinar computadores a enxergar como humanos, fornecendo ao computador informações precisas através de imagens ou vídeos, de forma que seja possível executar tarefas inteligentes, simulando a inteligência humana.

Segundo [Conci e Azevedo \(2008\)](#) o reconhecimento da importância da visão computacional se deu com a ideia de que os computadores poderiam ser usados para simular processamentos complexos de percepção visual. Segundo [Pedrini e Schwartz \(2008\)](#), uma das tarefas mais importantes de aplicações que envolvem algum tipo de visão computacional é a análise ou interpretação de imagens, obtendo como resultado uma descrição de imagem que contenha os objetos de interesse nela encontrados, com um mínimo de intervenção humana e com razoável grau de confiabilidade. De acordo com [Filho e Neto \(1999\)](#) a Visão Computacional pode ser estruturada com seis principais etapas, como demonstrado na Figura 1.

As demais etapas podem ser definidas da seguinte forma:

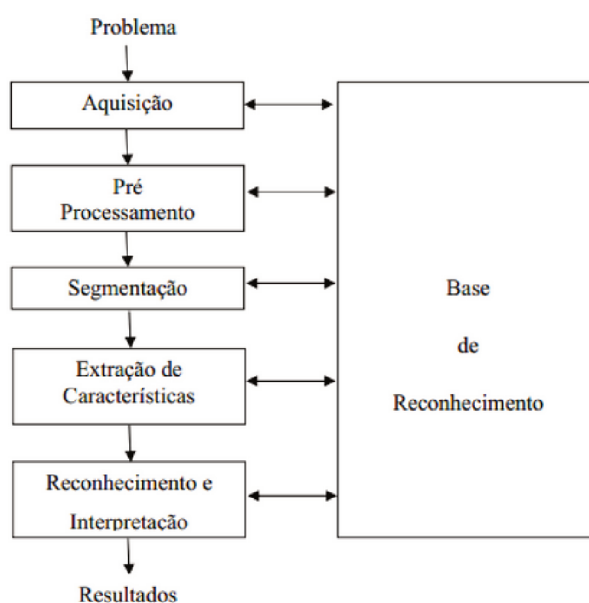


Figura 1 – Processo de visualização da informação [Filho e Neto \(1999\)](#)

- **Aquisição:** é a etapa na qual os dados brutos são adquiridos e estruturados de tal maneira que possam ser utilizados em análise. Pode ser realizada a coletas dos dados através de câmeras ou sensores.
- **Pré-Processamento:** utiliza-se dos dados estruturados capturados no passo anterior e aplica-se operações com o intuito de remover ou reduzir falhas e imperfeições da imagem ([FILHO; NETO, 1999](#)).

- Segmentação: consiste em dividir sub-imagens da imagem inicial de acordo com sua significância, para facilitar a identificação de padrões ou objetos na imagem (FILHO; NETO, 1999).
- Extração de características: esta etapa procura extrair características das imagens resultantes da segmentação através de descritores que permitam caracterizar com precisão Conci e Azevedo (2008) cada objeto de interesse.
- Reconhecimento e interpretação: nesta última etapa o processo de atribuição de um rótulo a um objeto baseado em suas características, traduzidas por seus descritores. A tarefa de interpretação, por outro lado, consiste em atribuir significado a um conjunto de objetos reconhecidos Conci e Azevedo (2008).

Em geral, sistemas de visão computacional tentam imitar a visão humana, no sentido de extrair informações relevantes a partir de imagens capturadas. Nas últimas décadas, estes sistemas têm sido usados com os mais diferentes propósitos, como por exemplo, em vídeos de vigilância. (MONTANARI, 2016; LEITE, 2018; FONSECA, 2018).

2.1.2 Detecção e rastreamento de objetos

A detecção e rastreamento de objetos a partir de imagens é uma tarefa bastante estudada na área de Visão Computacional, onde consiste em basicamente estabelecer relações temporais entre objetos alvo de quadros consecutivos de vídeos, ou seja, identificar a localização dos objetos alvo detectados na sequência de *frames* de um vídeo (COSTA, 2008).

Segundo Silva et al. (2018b) um sistema de detecção de objetos de uma classe específica pode ser definido como uma função f que mapeia uma imagem digital I no conjunto de localizações B , dado o modelo W . Assim, tem-se:

$$\mathbf{B} = f(I, W)$$

A localização de um objeto pode ser definida em termos de uma *bounding box*, um retângulo alinhado aos eixos x e y da imagem que descreve o objeto. Uma *bounding box* é definida por um vetor $\mathbf{b} \in \mathbb{R}^4$ que indica as coordenadas esquerda, direita, superior e inferior do retângulo, conforme demonstrado na Figura abaixo:

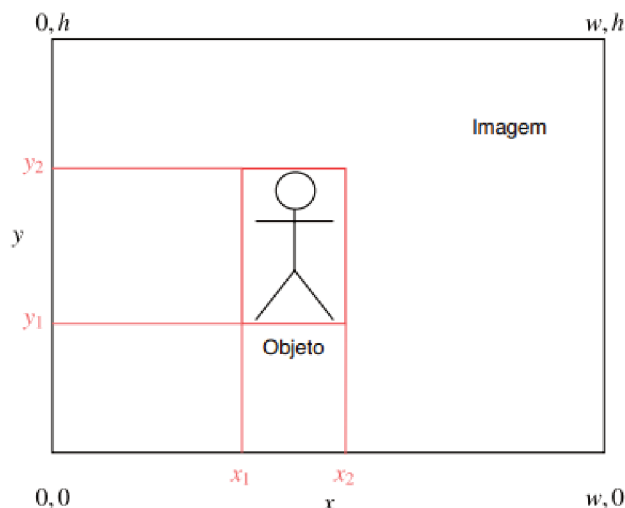


Figura 2 – Exemplo de uma bounding box descrevendo um objeto em uma imagem [Silva et al. \(2018b\)](#)

Após encontrado o objeto, ainda são feitos ajustes para prosseguimento da análise dos próximos *frames*. É construído um *bounding box* para o objeto encontrado e esse *bounding box* é expandido por uma borda. A análise para o *frame* seguinte é feita inicialmente na região correspondente à borda. Isso poupa tempo de processamento para casos em que não há mudanças bruscas de posicionamento do objeto alvo entre *frames* ([OLIVEIRA, 2018](#)).

Técnicas para detecção de objetos utilizam apenas a informação de um único *frame* para extrair características do objeto e realizar a estimação de sua localização, sem conhecimento prévio sobre a cena de entrada. No caso do rastreamento, a posição inicial do objeto é conhecida previamente e a informação estimada é recuperada *frame a frame*, registrando o movimento do objeto durante toda a cena. Em geral, as técnicas de detecção e rastreamento são utilizadas em conjunto, utilizando a detecção para recuperar a pose inicial do objeto e inicializar o algoritmo de rastreamento. A detecção também pode ser utilizada para recuperar a posição do objeto nos casos de perda durante o rastreamento ([TJADEN; SCHWANECKE; SCHOMER, 2017](#)).

2.1.3 Critério de detecção

Durante o treinamento de um sistema de detecção de objetos, um critério deve ser definido para identificar se uma localização detectada pelo sistema corresponde a um objeto na imagem processada. Se o conjunto de exemplos for anotado com as *bounding boxes* dos objetos contidos em suas imagens, as localizações detectadas são comparadas a cada uma das anotações de acordo com uma métrica de similaridade. Se uma detecção for comparada às localizações de referência e não obter um valor mínimo de similaridade, considera-se que a localização não corresponde a um objeto da classe detectada, ou seja,

ocorreu falsa detecção. Se mais de uma detecção obter uma similaridade acima do valor mínimo em relação a uma mesma referência, considera-se que ocorreram detecções repetidas. E se nenhuma detecção obtiver similaridade mínima em relação a uma referência específica, diz-se que ocorreu falta na detecção (SILVA et al., 2018b).

2.2 Trabalhos Correlatos

Nesta seção, apresentaremos alguns trabalhos encontrados na literatura que abordam a detecção de objetos e o rastreamento. Em seguida, destacaremos o projeto de mestrado mencionado anteriormente. Esses trabalhos fornecem uma visão abrangente das abordagens e pesquisas realizadas nessa área

2.2.1 Estratégia para análise visual de trajetórias em vídeos de vigilância aplicando-se técnicas de visualização de informação

O trabalho desenvolvido por (FONSECA, 2018) tem como objetivo o desenvolvimento de uma metodologia para análise visual de trajetórias presentes em vídeos de vigilância aplicando técnicas de visualização de informação. A metodologia desenvolvida propõe suprir a difícil tarefa de um agente humano analisar vídeos de segurança com extensões muito longas que torna o processo muito difícil e demorado através de técnicas de visualização de informação aliadas a ferramentas interativas.

Foi utilizado para detecção e identificação de objetos em vídeos o *YOLO (You Only Look Once)* um sistema de detecção de objetos que detecta e identifica objetos presentes nos *frames* de uma imagem. A etapa anterior pode ser modificada de acordo com resultados das análises de extratores que serão testados neste trabalho.

Outro objetivo é representar trajetórias de objetos detectados em cada *frame* em *layouts* interativos.

2.2.2 Identificação de objetos em vídeos de câmera estática

O trabalho desenvolvido por (LEITE, 2018) propõe a construção de um *framework* para identificação de objetos em movimento em vídeos de câmera estática. Este trabalho pode ser utilizado para identificar pedestres, carros, ciclistas ou nenhum deles.

Para obtenção de resultados foram utilizadas técnicas para a detecção de objetos em movimento. Foi empregada o método de subtração de plano de fundo denominado Mistura de Gaussianas, em conjunto com operações morfológicas visando maior precisão de detecção dos objetos.

A caracterização dos objetos extraídos foi composta por três métodos, sendo eles, HOG, para características de forma, Filtros de Gabor e GLDM, ambos para extração de características de textura. Devido a grande dimensionalidade do descritor de objetos resultante da composição de múltiplos extratores de característica, foi considerado o uso de técnicas para redução de dimensionalidade através dos métodos LDA e ICA. Por fim, para a tarefa de classificação, foram empregados os métodos SVM, árvore de decisão, *Naive Bayes* e o kNN.

2.2.3 Aplicação de Visão Computacional no Auxílio ao Levantamento de Defeitos em Pavimento Rodoviário

O trabalho desenvolvido por (HOLLERWEGER et al., 2019) é focado no estudo da viabilidade e eficácia de estratégias de visão computacional aplicadas a sistemas de auxílio ao levantamento de defeitos em pavimento rodoviário. A ideia central é aumentar o nível de automatização e assertividade de tais sistemas.

As imagens para estudo são coletadas através da câmera em um veículo em movimento, o principal defeito a ser detectado são buracos na rodovia. Para obtenção de resultados são utilizadas redes neurais para classificação e detecção de objetos utilizando ResNet (Residual *Neural Network*) e *YOLO (You Only Look Once)*

Segundo Hollerweger et al. (2019) ResNet é uma topologia residual em redes neurais, e consiste na criação de “atalhos” dentro da rede, de forma que os valores na saída de uma camada “pulam” uma ou mais camadas seguintes, sendo posteriormente combinados aos valores de saída das camadas que foram puladas. O *YOLO (You Only Look Once)* pode ser definido como uma estratégia na tarefa de detecção de objeto que utiliza um método focado apenas em detecção.

2.2.4 Detecção e contagem de plantas utilizando técnicas de inteligência artificial e *machine learning*

O trabalho desenvolvido por (SILVA et al., 2018a) tem como objetivo fazer a contagem de todos eucaliptos de uma plantação utilizando técnicas de aprendizado de máquina e visão computacional. A expectativa de precisão é de 95% de acerto e 5% de erro.

Foram utilizados os modelos mais precisos de redes neurais convolucionais atualmente existentes através da plataforma *TensorFlow*. O treinamento da rede foi realizado com um dataset obtido através do sobrevoo de uma região de plantação de eucaliptos. Os métodos utilizados foram a SSD *Inception V2*, R-CNN *Resnet 101* e a R-CNN *Resnet Inception*

3 Desenvolvimento

Neste capítulo, as etapas do desenvolvimento do trabalho são especificadas, incluindo o detalhamento do funcionamento do sistema e sua estrutura, no qual está descrito por módulos computacionais e telas.

3.1 Visão geral

O sistema foi construído de forma modular, com a finalidade de definir um fluxo de trabalho baseado em responsabilidades e tarefas a serem executadas para atingir os objetivos propostos. A Figura 3 representa o diagrama de blocos do funcionamento do sistema. Cada um dos módulos ilustrados são detalhados a seguir.

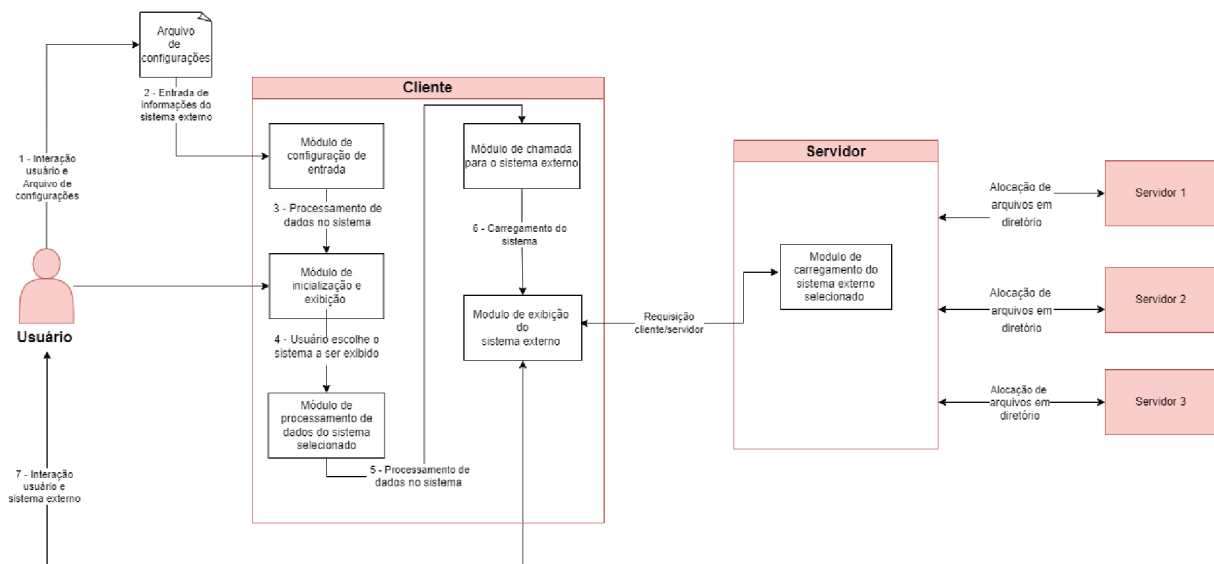


Figura 3 – Diagrama de funcionamento dos módulos do sistema

- **Módulo de configuração de entrada:** Responsável por coletar dados de utilização e exibição dos sistemas externos e dispor as informações necessárias para executá-los. Essas informações são recuperadas de um arquivo de configurações que será utilizado nos demais módulos;
- **Módulo de inicialização e exibição:** Responsável por exibir as informações necessárias para a execução dos sistemas disponíveis, além de receber do usuário as informações para essa execução;

- Módulo de processamento de dados do sistema selecionado: Responsável por identificar qual sistema foi selecionado pelo usuário e carregar as informações configuradas;
- Módulo de chamada para o sistema externo: Responsável por acessar o endereço *web* construído utilizando as configurações correspondentes;
- Módulo de carregamento para o sistema externo selecionado: Responsável por carregar o sistema externo para utilização do usuário;
- Módulo de exibição do sistema externo: Responsável por exibir o sistema externo e possibilitar o usuário interagir com o sistema exibido.

3.2 Tecnologias

Nesta seção serão abordadas e descritas todas as tecnologias utilizadas para o desenvolvimento do sistema *web*. A Figura 4 apresenta um diagrama que ilustra quais tecnologias atuam para o funcionamento do sistema, além de exibir como essas tecnologias interagem entre si para que o sistema execute suas funções.

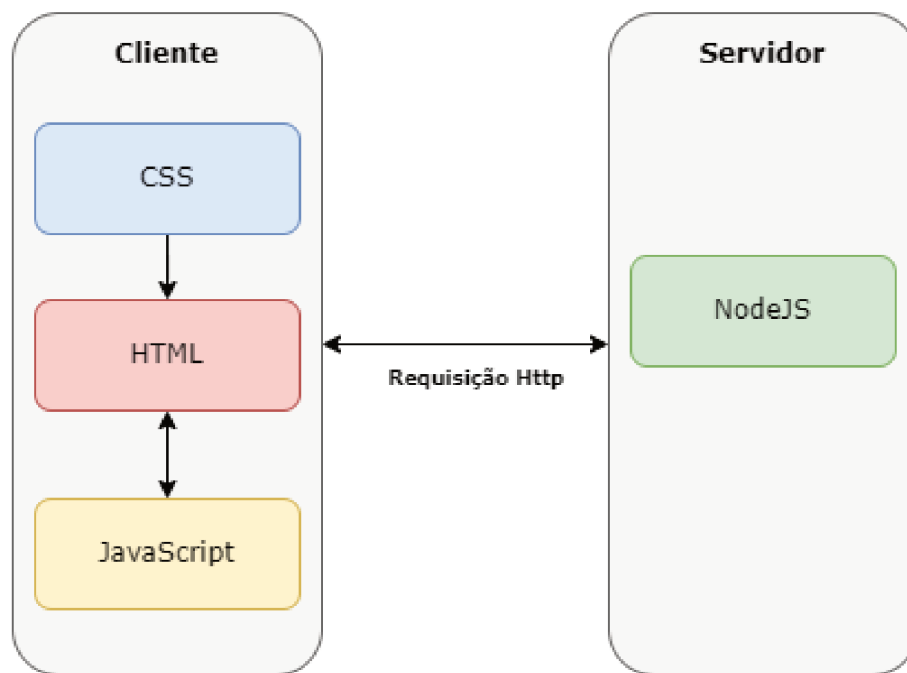


Figura 4 – Diagrama de tecnologias

- **HTML:** *HyperText Markup Language* ou Linguagem de Marcação de Hipertexto foi empregado na estruturação de todas as telas do sistema *web* desenvolvido. O HTML foi utilizado para criar os documentos que servem para descrever, dentre outras coisas, quais são os elementos gráficos presentes na página, e onde estes estão localizados em relação uns aos outros. A versão utilizada foi HTML 5, que oferece recursos para construção e estruturação de páginas dinâmicas no sistema *web*.
- **CSS:** *Cascading Style Sheets* ou Folhas de Estilo em Cascata, é uma linguagem de formatação visual/funcional de elementos codificados em uma linguagem de marcação. Neste projeto o CSS foi utilizado para definir o estilo visual dos elementos presentes nos documentos HTML criados. Essa linguagem de folhas de estilo foi escolhida por possibilitar a criação das telas responsivas e modernas.
- **JavaScript:** É uma linguagem de *script* interpretada pelos navegadores *web*, caracterizada por ser uma linguagem dinâmica multiparadigma fundamentada em protótipo que oferece suporte a estilos orientados a objetos imperativos e declarativos (como programação funcional), além de ser uma forma dos usuários interagirem melhor com páginas da *Web*. (PRESCOTT, 2016). Neste projeto o *JavaScript* foi utilizado na criação dos elementos HTML de forma dinâmica de acordo com as escolhas do usuário na página *web* desenvolvida. Essa linguagem de *script* foi escolhida pois possibilita criar, gerenciar e atualizar os elementos HTML, multimídias, imagens e entre outros de forma dinâmica, além de oferecer diversas bibliotecas que facilitam muito o desenvolvimento.

O *JavaScript* possui uma biblioteca denominada D3.js (*Data-Driven Documents*) segundo (BOSTOCK, 2021), a mesma deve ser utilizada para manipulação de documentos com base em dados, permitindo vincular os dados arbitrários a um DOM (*Document Object Model*) e posteriormente aplicando transformações orientadas por dados ao documento. O atual sistema não depende dessa ferramenta, porém ela é utilizada na exibição dos *layouts* presentes nos sistemas externos existentes.

- **NodeJS:** É uma plataforma desenvolvida em *JavaScript* responsável por compartilhar o código entre o servidor e o cliente, e gerenciar entrada e saída de dados. (NODE. . . ,). Neste projeto a plataforma foi aplicada na construção de uma API que é responsável por armazenar os dados recebidos do usuário em um diretório no servidor para posterior transmissão para os sistemas externos. O seu funcionamento consiste em fazer uma solicitação pelo *front-end* e retornar uma resposta por meio da estrutura de dados, para ser recebido pelo cliente, que processa o resultado. O *nodeJs* foi empregado por conseguir trabalhar bem com requisições assíncronas. Sendo assim, traz simplicidade e flexibilidade para o desenvolvimento.

3.3 Desenvolvimento do website

O desenvolvimento do site foi realizado utilizando as linguagens de programação citadas na seção 3.2. A seguir são apresentados o arquivo de configurações e as telas do sistema, juntamente com o detalhamento de sua utilização.

3.3.1 Arquivo de Configurações

O sistema desenvolvido utiliza um arquivo de configurações para viabilizar a integração de múltiplos sistemas externos. Esse arquivo é estruturado em formato JSON, que descreve uma lista na qual cada elemento é um conjunto de configurações específicas de um sistema externo. Todas as configurações de cada sistema são definidas em campos, detalhados a seguir. O objetivo é garantir um padrão de configurações que permita ao correta integração dos sistemas de análise visual;

- **Name:** Este campo corresponde ao nome do sistema externo que deseja ser incluído. É importante fornecer o nome correto e preciso do sistema para garantir sua identificação e integração adequadas;
- **ServerDirectory:** Neste campo, é especificado o endereço do sistema no servidor. Essa informação é essencial para estabelecer a conexão e acesso ao sistema externo;
- **DefaultParameters:** Aqui, são listadas as dependências do sistema externo. Essas dependências incluem bibliotecas, arquivos ou quaisquer outros componentes necessários para o funcionamento adequado do sistema. É fundamental fornecer as informações corretas sobre essas dependências para garantir a compatibilidade e execução sem problemas;
- **DestinationApiFiles:** Este campo indica o endereço do diretório de *upload* no servidor. Ele define o local onde os arquivos são enviados ou armazenados quando ocorre a interação com o sistema externo.

3.3.2 Página principal

A página é construída de forma dinâmica, de acordo com as informações recuperadas do arquivo de configuração. Durante a fase de desenvolvimento foram adicionadas listas, botões e ícones para deixar o *website* mais atrativo e organizado.

Ao abrir o sistema pelo navegador a interface da página principal é mostrada, inicialmente contendo: um título(A) e um componente de seleção Figura (B) como mostra a Figura 5.

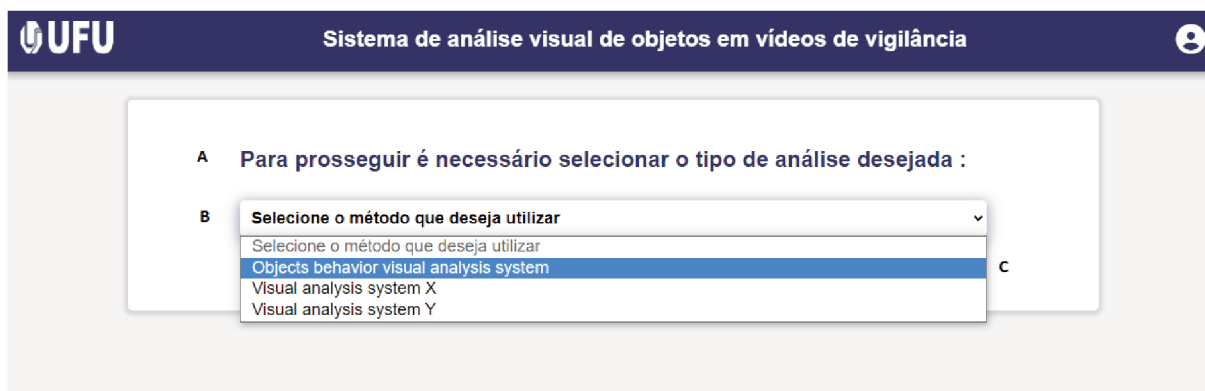


Figura 5 – Seleção de sistema externo

O componente de seleção ao ser clicado exibirá todos os sistemas externos disponíveis para serem utilizados. Ao selecionar um sistema externo (C), são chamadas as funções *JavaScript* encarregadas de criar os demais elementos no documento *HTML*, como mostra a Figura 6.

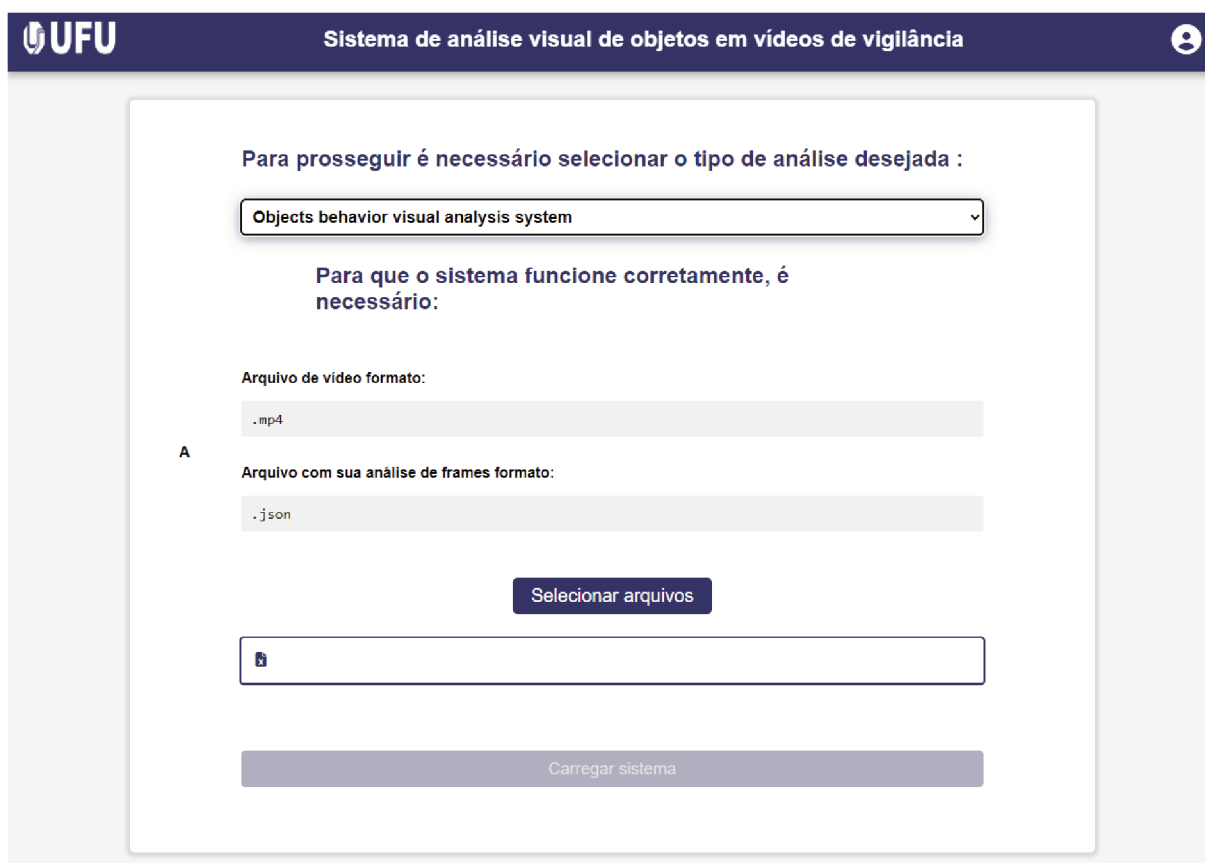


Figura 6 – Carregamento informações sistemas selecionado

Os parâmetros/arquivos necessários são destacados através do seu formato. O usuário então fornece todas as informações necessárias ao sistema, que aparecem listadas na parte inferior da tela (B), como mostra a Figura 7.

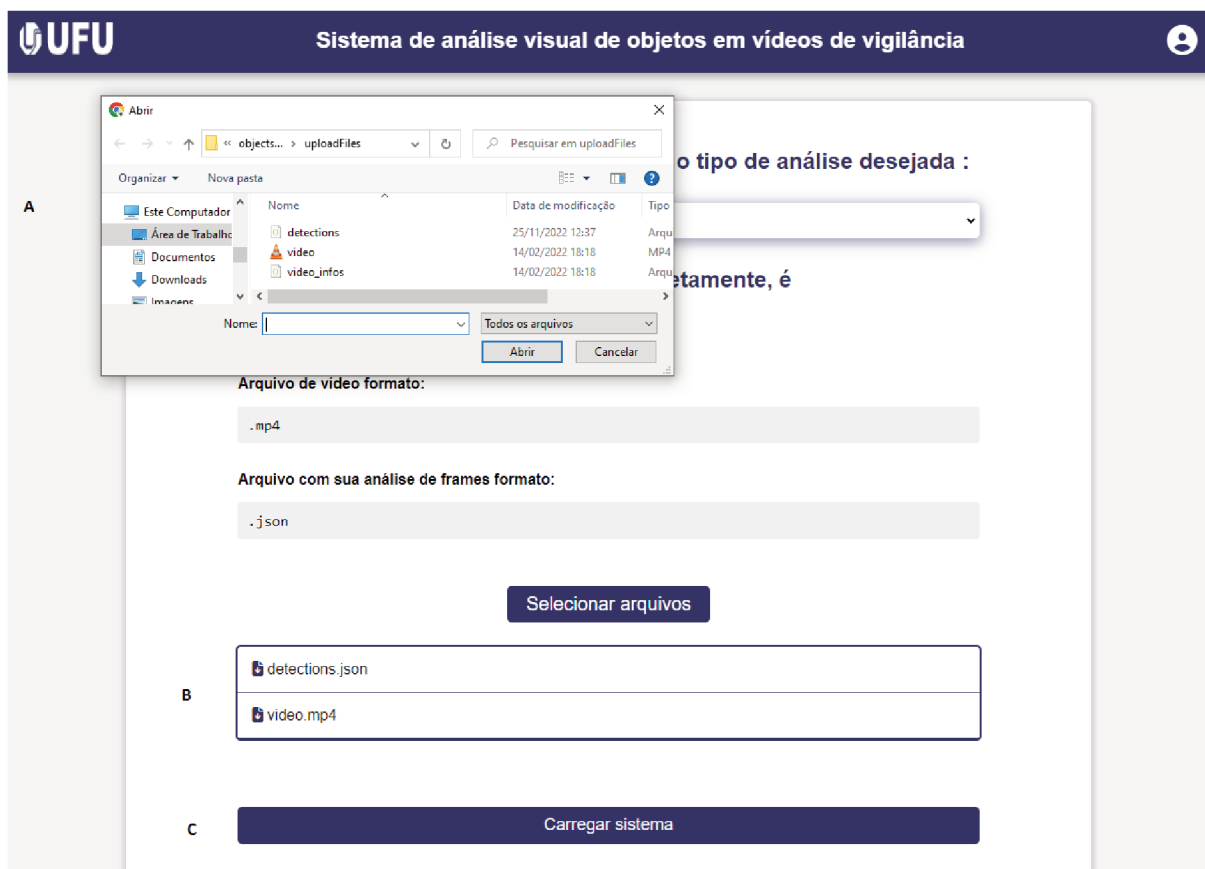


Figura 7 – Seleção de arquivos e upload

Em seguida, o botão “Carregar sistema” é habilitado (C) e ao clique do usuário é chamada uma ação responsável por fazer *upload* dos arquivos através de uma API desenvolvida com *NodeJS*, que serão armazenados em uma pasta no servidor. Então o sistema web monta uma URL que invoca o sistema externo escolhido, passando através dela os parâmetros necessários para que o sistema externo possa buscar os arquivos que foram armazenados em diretório anteriormente no servidor. Finalmente, o sistema externo é executado e posicionado em um *iframe* que contém o sistema selecionado anteriormente pronto para utilização, como mostra a Figura 8.



Figura 8 – visualização sistema externo

4 Resultados

Este capítulo apresenta os resultados da aplicação do sistema desenvolvido. As seções a seguir apresentam o detalhamento de um exemplo de sistema externo, bem como os detalhes de como o sistema desenvolvido faz a conexão com ele para execução e análise pelo usuário.

4.1 Sistema de análise visual de vídeos de vigilância

Esta seção apresenta e detalha o sistema construído por (FONSECA, 2018) em seu trabalho de mestrado.

4.1.1 Descrição geral do sistema externo

O sistema foi desenvolvido com objetivo de analisar visualmente vídeos de vigilância, com foco no comportamento e relacionamento entre objetos nesses vídeos, além de auxiliar na análise de vídeos de vigilância previamente gerados.

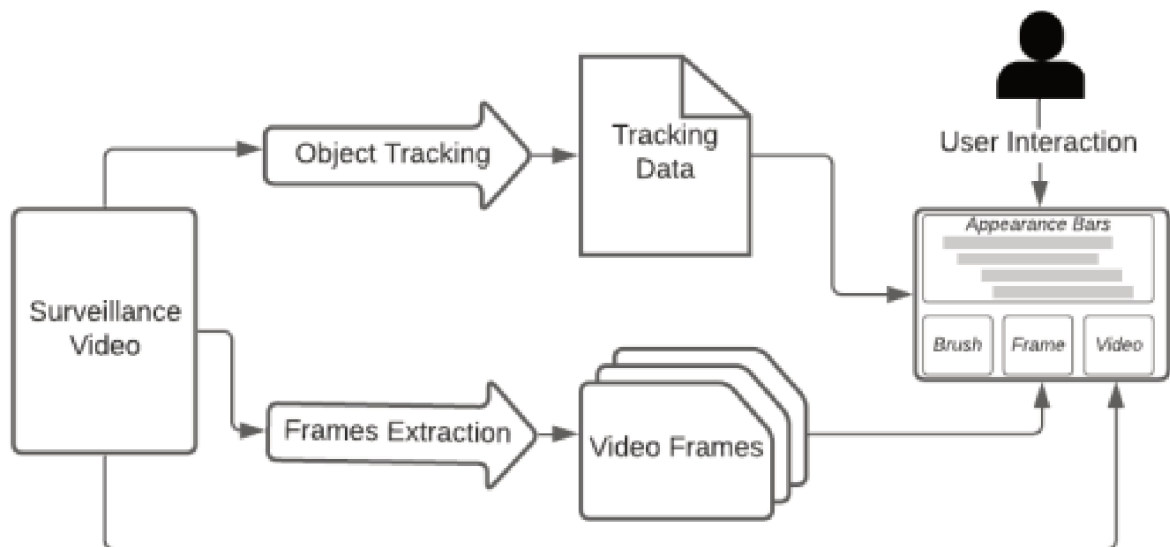


Figura 9 – Diagrama de funcionamento do sistema externo (FONSECA, 2018)

A Figura 9 ilustra o fluxo de trabalho de análise proposto. A primeira etapa consiste em empregar uma estratégia para identificar e rastrear objetos de um vídeo previamente gerado. As informações de rastreamento produzidos por essas estratégias são então utilizadas para gerar o *layout*. O sistema também extrai do vídeo todos os *frames* utilizados no processo de rastreamento, que serão utilizados para auxiliar na interação do usuário. Por fim, o usuário pode interagir com o *layout*, executando uma variedade de tarefas básicas de exploração com todos os objetos detectados, como zoom/panorâmica da linha do tempo e seleção de objetos. Para obter informações adicionais sobre o sistema, é possível consultar em (FONSECA; PAIVA, 2021)

4.2 Adequação do sistema externo para utilização no sistema proposto

Esta seção tem como objetivo detalhar as adaptações realizadas no sistema externo para aplicação no sistema desenvolvido, de forma a tornar possível a troca de informações entre ele e o sistema desenvolvido neste trabalho.

A integração entre os dois sistemas é um processo que permite a comunicação e a troca de informações entre eles, possibilitando que trabalhem juntos de forma harmoniosa. Para viabilizar essa integração, foi desenvolvido um módulo no sistema externo responsável por receber uma chamada contendo a URL montada no sistema web. Esse módulo processa a URL e extrai os caminhos em diretório de cada arquivo, buscando os arquivos nesses diretórios e fornecendo os dados necessários aos módulos já existentes no sistema externo. Além disso, foi criada uma API (*Application Programming Interface*) para que os sistemas se conectem e compartilhem dados de forma estruturada e automatizada. Essa API também é responsável por alocar os arquivos no diretório obtido pelo módulo anterior.

4.2.1 Funcionalidades

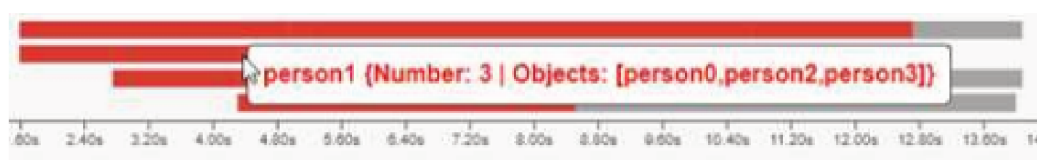
Esta seção apresenta brevemente as ferramentas contidas no sistema externo utilizado neste trabalho.

De forma geral a interface do sistema é apresentada na Figura 10 e descrito em seguida.



Figura 10 – Interface do sistema

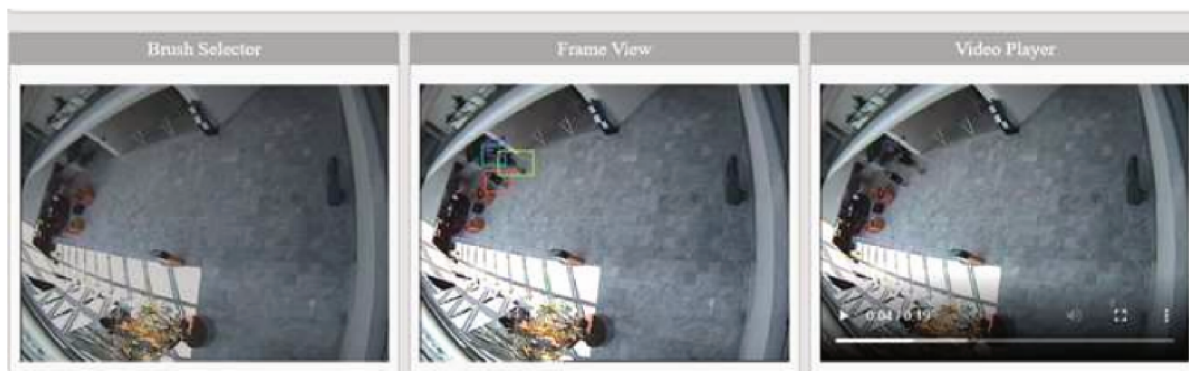
A *Appearance Bars View* (Figura 11) apresenta uma visão geral de todos os objetos detectados, mostrando seus momentos de presença na cena ao longo de toda a duração do vídeo e destaca seu comportamento durante essa presença, incluindo seus padrões de movimento e a relação entre eles, também demonstra este funcionamento destacando quais objetos participam da cena ao passar o mouse na barra de permanência e mostrando a seleção de um instante.



(a)

Figura 11 – Exemplo Appearance Bars View no sistema externo (FONSECA, 2018)

A *Brush View* (Figura 12) permite ao usuário selecionar uma região de interesse do fundo do vídeo para filtrar os objetos que cruzaram a região (FONSECA, 2018). A *Frame View* (Figura 12) permite ao usuário observar detalhes de um determinado instante do vídeo selecionado nas barras de aparência ou exibido na Visualização do *Player* de Vídeo, destacando as caixas delimitadoras dos objetos identificados nos respectivos quadros (FONSECA, 2018). O *Video Player View* (Figura 12) implementa uma ferramenta de interação tradicional utilizada para assistir e navegar em vídeos, permitindo ao usuário reproduzir/pausar a execução do vídeo, avançar/rebobinar *frames*, ligar/desligar o som e colocar o vídeo em tela cheia ou em imagem a imagem (FONSECA, 2018).



(b)

Figura 12 – Exemplo de views no sistema externo (FONSECA, 2018)

4.3 Comunicação entre Sistema Web e Sistema Externo

Esta seção tem como objetivo detalhar o funcionamento do sistema desenvolvido no contexto do processo de carregamento do sistema externo previamente mencionado.

Primeiramente, é necessário adicionar um elemento na lista de configurações contida no arquivo de configurações previamente explicado. A Tabela 1 apresenta um exemplo de conteúdo que deve ser inserido no formato JSON.

Tabela 1 – Exemplo de conteúdo do arquivo de configurações

Campo	Conteúdo
<i>Name</i>	Objects behavior visual analysis system
<i>ServerDirectory</i>	http://127.0.0.1:5501/index.html
<i>DefaultParameters</i>	"Arquivo de vídeo formato", ".mp4", true
<i>DestinationApiFiles</i>	http://localhost:8080/diretorioBackend/

Ao recuperar essas informações do arquivo de configurações, o sistema web é capaz de construir a página web com os campos necessários para a execução do sistema externo especificado, além de adicionar uma entrada na lista de seleções da página principal, como mostra a Figura 13.

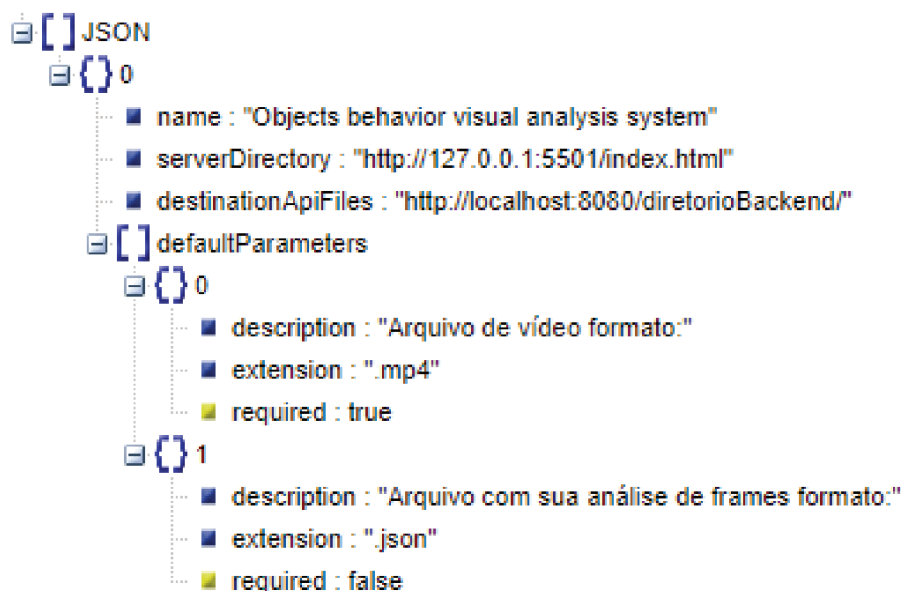


Figura 13 – Exemplo de entrada do arquivo de configuração

Em seguida, após a seleção do sistema externo desejado, a página do sistema web será construída de acordo com as informações presentes no arquivo de configuração.

Quando o botão "Carregar sistema" é acionado, o sistema web invoca um método responsável por enviar uma requisição para a *API NODE*. Através dessa requisição, os arquivos selecionados são enviados para o servidor de destino. Antes de carregar o sistema externo, o sistema web monta uma URL contendo o caminho dos arquivos alocados no servidor e, em seguida, abre um *iframe* com o sistema externo carregado. Dessa forma, o sistema externo é carregado no *iframe* criado, permitindo a correta comunicação e troca de informações entre os sistemas envolvidos na integração. conforme mostra a Figura 14.

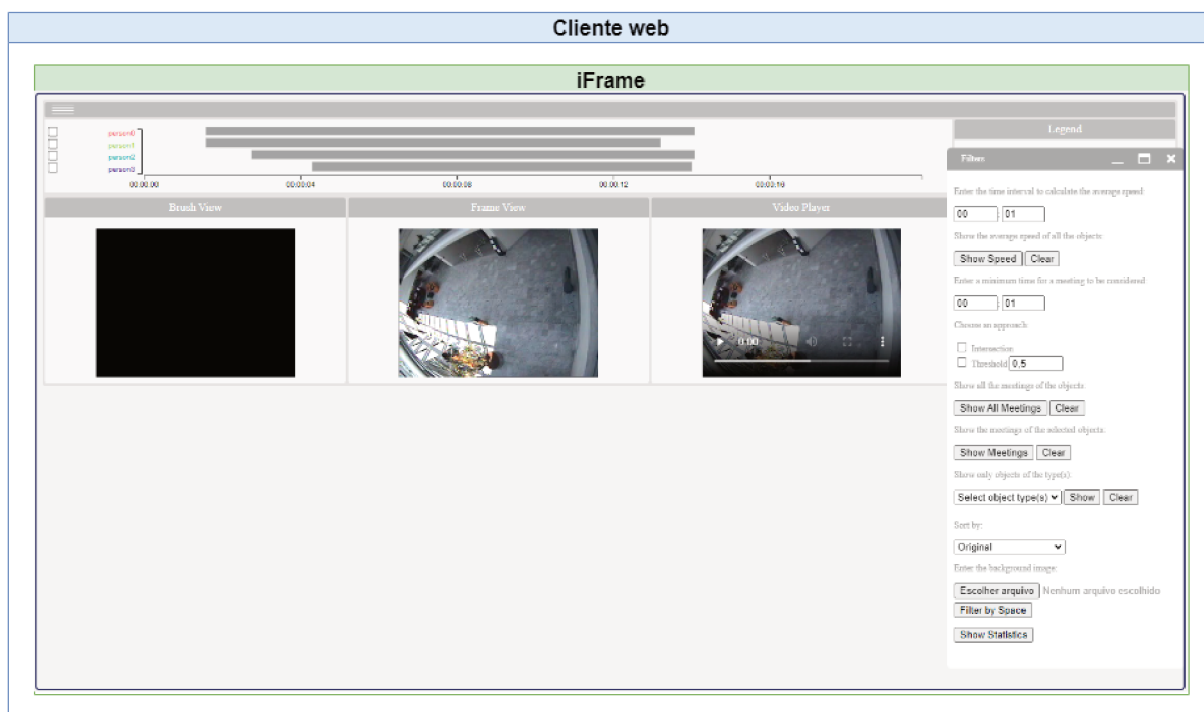


Figura 14 – Exemplo de retorno IFrame

5 Conclusão

Este trabalho apresentou um sistema web que acomoda e gerencia a execução de diversos sistemas de análise visual, com ênfase na análise de vídeos de vigilância. O principal objetivo alcançado foi fornecer uma plataforma flexível capaz de suportar diferentes sistemas de análise visual. O segundo foi construir o sistema de forma modular, permitindo a inclusão fácil e sem interferência de novos módulos. Ele foi idealizado para acomodar diversos sistemas de visualização desenvolvidos na Facom-UFU, mostrando assim sua capacidade de aplicação em diferentes contextos.

Para ilustrar a aplicação do sistema, foi utilizado um sistema externo desenvolvido em um projeto de mestrado da Facom-UFU, que permitiu destacar sua versatilidade e aplicabilidade em diferentes cenários. A utilização do sistema nesse caso possibilitou a fácil utilização do sistema externo, e foi possível comprovar que ele pode ser utilizado para incorporar diversas técnicas de análise visual com foco em materiais de vigilância.

O sistema apresenta potencial de melhoria em diversos aspectos, incluindo a criação de um novo módulo para a chamada de dois sistemas de análise visual de forma simultânea. Isso traria um ganho significativo ao considerar a possibilidade de cruzar os dados obtidos entre diferentes tipos de abordagens.

Referências

- BOSTOCK, M. *D3.js*. 2021. Disponível em: <<https://d3js.org/>>. Citado na página 22.
- CONCI, A.; AZEVEDO. volume 2 (processamento e análise de imagens digitais). *Campus/Elsevier*, 2008. Citado 2 vezes nas páginas 15 e 16.
- COSTA, B. A. D. Segmentação, rastreamento de objetos e detecção de eventos primitivos com aplicação no monitoramento automático de ações humanas em vídeo. 2008. Citado na página 16.
- FILHO, O. M.; NETO, H. V. *Processamento digital de Imagens*. [S.l.]: BRASPORT, 1999. Citado 2 vezes nas páginas 15 e 16.
- FONSECA, C. M. Estratégia para análise visual de trajetórias em vídeos de vigilância aplicando-se técnicas de visualização de informação. 2018. Citado 7 vezes nas páginas 13, 14, 16, 18, 27, 29 e 30.
- FONSECA, C. M.; PAIVA, J. G. S. A system for visual analysis of objects behavior in surveillance videos. In: *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.: s.n.], 2021. p. 176–183. Citado na página 28.
- GARCIA, M. L. *Design and Evaluation of Physical Protection Systems, Second Edition*. 2nd. ed. USA: Butterworth-Heinemann, 2007. ISBN 075068352X. Citado na página 10.
- HOLLERWEGER, M. M. et al. Aplicacao de visao computacional no auxilio ao levantamento de defeitos em pavimento rodoviario. Florianópolis, SC, 2019. Citado na página 19.
- LAO, W.; HAN, J.; With, de, P. Automatic video-based human motion analyzer for consumer surveillance system. *IEEE Transactions on Consumer Electronics*, Institute of Electrical and Electronics Engineers, v. 55, n. 2, p. 591–598, 2009. ISSN 0098-3063. Citado na página 10.
- LEITE, T. F. de S. Identificação de objetos em vídeos de câmera estática. 2018. Citado 3 vezes nas páginas 12, 16 e 18.
- LI, X. et al. *A Survey of Appearance Models in Visual Object Tracking*. 2013. Citado na página 11.
- MONTANARI, R. *Detecção e classificação de objetos em imagens para rastreamento de veículos*. Tese (Doutorado) — Universidade de São Paulo, 2016. Citado na página 16.
- NGUYEN, D.; LI, W.; OGUNBONA, P. Human detection from images and videos: A survey. *Pattern Recognit.*, v. 51, p. 148–175, 2016. Citado na página 11.
- NODE.JS. <<https://nodejs.org/en>>. (Accessed on 05/22/2023). Citado na página 22.
- OLIVEIRA, W. R. de. Sistema de visão computacional para reconhecimento de modelos cad complexos. 2018. Citado 2 vezes nas páginas 15 e 17.

- PARKER, J. R. *Practical Computer Vision Using C*. USA: John Wiley Sons, Inc., 1993. ISBN 0471592595. Citado na página 10.
- PEDRINI, H.; SCHWARTZ, W. R. *Análise de imagens digitais: princípios, algoritmos e aplicações*. [S.l.]: Cengage Learning, 2008. Citado na página 15.
- PITZER, R. M. Z. 2009. Disponível em: <<http://www.publicadireito.com.br/artigos/?cod=048e2f1447691907>>. Citado na página 10.
- PRESCOTT, P. *Programação em JavaScript*. [S.l.]: Babelcube Inc., 2016. Citado na página 22.
- REZENDE, S. O. et al. Mineração de dados. In: *Sistemas Inteligentes – Fundamentos e Aplicações*. 1. ed. Barueri, SP: Editora Manole Ltda, 2003. p. 307–336. ISBN 8520416837. Citado na página 11.
- Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, v. 3, n. 3, p. 210–229, 1959. Citado na página 10.
- SILVA, G. C. d. et al. Detecção e contagem de plantas utilizando técnicas de inteligência artificial e machine learning. Florianópolis, SC, 2018. Citado na página 19.
- SILVA, G. R. et al. Detecção de objetos em imagens utilizando técnicas de aprendizagem profunda. Florianópolis, SC, 2018. Citado 3 vezes nas páginas 16, 17 e 18.
- TJADEN, H.; SCHWANECKE, U.; SCHOMER, E. Real-time monocular pose estimation of 3d objects using temporally consistent local color histograms. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, 2017. p. 124–132. ISSN 2380-7504. Disponível em: <<https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.23>>. Citado na página 17.