http://journal.unoeste.br/index.php/ce/index
DOI: 10.5747/ce.2025.v17.e438
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 29/06/2024 Revisado: 25/08/2025 Aceito: 19/09/2025

MONITORAMENTO AUTOMÁTICO DE VÍDEOS A PARTIR DE CÂMERAS DE SEGURANÇA UTILIZANDO REDES NEURAIS CONVOLUCIONAIS E VISÃO COMPUTACIONAL

Automatic Video Monitoring from Security Cameras Using Convolutional Neural Networks and Computer Vision

Sérgio Everton Rigolin Duarte[1], Francisco Assis da Silva[1], Leandro Luiz de Almeida[1], Almir Olivette Artero[2], Danillo Roberto Pereira[2]

[1]Universidade do Oeste Paulista – UNOESTE, Faculdade de Informática de Presidente Prudente – FIPP, Presidente Prudente, SP. [2]Faculdade de Ciências e Tecnologia, UNESP – Universidade Estadual Paulista, Departamento de Matemática e Computação, Presidente Prudente.
Email: sergiioduart@gmail.com, chico@unoeste.br, llalmeida@unoeste.br, almir.artero@unesp.br, danillo.pereira@unesp.br

RESUMO – A maioria dos sistemas de segurança residenciais ou comerciais baseia-se em monitoramento onde a análise é feita pela interpretação humana. Para aprimorar esse processo, têm sido propostos modelos que realizam a análise e o reconhecimento de características de pessoas, como altura, cor da pele e reconhecimento facial, entre outros, com o objetivo de tornar a identificação e o monitoramento mais eficientes. Os algoritmos e técnicas de visão computacional e de redes neurais convolucionais desempenham um papel crucial na busca por processos automatizados para fornecer mais segurança para as pessoas, sendo amplamente utilizados e constantemente aprimorados. Neste trabalho, foi utilizada a rede neural convolucional YOLO (*You Only Look Once*) para a identificação de objetos (pessoas e veículos) e YOLO-Pose para a identificação dos pontos chave das articulações do corpo. Além disso, os espaços de cores HSV (*Hue*, *Saturation*, *Value*) e YCbCr (brilho (luma), (B-Y) diferença de crominância azul, (R-Y) diferença de crominância vermelha) foram combinados e aplicados para a identificação e extração de características, como o tipo e a cor da roupa de pessoas, para possibilitar a contagem de vezes que uma pessoa esteve no foco da câmera. A partir dos experimentos realizados pelos autores, utilizando diferentes combinações de roupas, o algoritmo foi capaz de detectar com precisão de 90%, as peças de roupa utilizadas pelas pessoas na cena.
Palavras-chave: Sistemas de monitoramento, Câmeras de Segurança, YOLO; Redes Neurais Convolucionais, Visão Computacional.

ABSTRACT – Most residential or commercial security systems are based on monitoring where the analysis is done through human interpretation. To improve this process, models have been proposed that perform the analysis and recognition of people's characteristics, such as height, skin color, and facial recognition, among others, with the aim of making identification and monitoring more efficient. Computer vision and convolutional neural network algorithms and techniques play a crucial role in the search for automated processes to provide more safety for people, and are widely used and constantly being improved. In this work, the YOLO (You Only Look Once) convolutional neural network was used to identify objects (people and vehicles) and YOLO-Pose to identify the key points of the body's joints. In addition, the HSV (Hue, Saturation, Value)

and YCbCr (brightness (luma), (B-Y) blue-difference chrominance, (R-Y) red-difference chrominance) color spaces were combined and applied for the identification and extraction of characteristics, such as the type and color of people's clothing, to enable the counting of times a person has been in the camera's focus. Based on the experiments carried out by the authors, using different combinations of clothing, the algorithm was able to detect the items of clothing worn by the people in the scene with 90% accuracy.

Keywords: Monitoring Systems, Security Cameras, YOLO; Convolutional Neural Networks, Computer Vision.

1. INTRODUÇÃO

O aumento da criminalidade e da violência intensificou a sensação de insegurança, levando à adoção de novas formas de monitoramento e serviços de segurança patrimonial. Esse cenário impulsionou o uso de tecnologias em empresas de vigilância eletrônica, com destaque para sistemas de câmeras — desde circuitos fechados até webcams simples (Nunes, 2023). A frase “Sorria, você está sendo filmado”, comum em locais monitorados, não busca provocar um sorriso, mas inibir comportamentos inadequados (Rodrigues, 2005). Apenas a presença de câmeras já pode influenciar a conduta das pessoas.

Em 2024, o setor de segurança eletrônica no Brasil cresceu 16,1% em relação a 2023, segundo a Abese. Parte desse aumento ocorreu pela migração de pessoas de apartamentos para casas, que passaram a investir em monitoramento (Nunes, 2023). Ainda assim, esses sistemas não impedem crimes, apenas registram ou permitem acompanhamento em tempo real. Como dependem da análise humana, mesmo em monitoramento 24h, estão sujeitos a falhas (Souza, 2023).

A visão computacional surge como alternativa para automatizar esse processo. Ela busca replicar a capacidade humana de reconhecer objetos e pessoas a partir de imagens (OJHA; SAKHARE, 2015). Redes neurais convolucionais (CNN) são amplamente usadas nessa tarefa, aplicando filtros em imagens digitais para extrair características e classificá-las (Brandizzi, 2020).

Entre os modelos mais eficientes está o YOLO (*You Only Look Once*), capaz de identificar múltiplos objetos em tempo real, atribuindo rótulos e delimitando suas posições (Ultralytics, 2024). Essa tecnologia demonstra como o aprendizado profundo aprimorou a análise de padrões em imagens e vídeos.

Aplicada à segurança, a YOLO possibilita vigilância inteligente em residências, identificando situações suspeitas, como pessoas circulando repetidamente ou veículos estacionados em áreas estratégicas (portões, entradas). Dessa forma, o morador pode ser alertado em tempo real e agir imediatamente diante do ocorrido.

Este trabalho apresenta um método para monitoramento inteligente em câmeras de segurança, com foco em detectar pessoas e veículos e analisar comportamentos em áreas de interesse. O objetivo é oferecer ao morador informações em tempo real sobre situações suspeitas, como pessoas paradas por muito tempo próximas ao imóvel ou veículos estacionados em locais estratégicos

A metodologia foi organizada em três etapas: detecção de objetos com a rede YOLO na versão 8 (YOLOv8), rastreamento de pessoas por meio do ponto central de cada detecção para reduzir processamento e extração de características de roupas para facilitar reidentificação em caso de falhas. O processo também considera a validação em regiões de interesse e situações de oclusão.

Nos testes realizados, com nove vídeos (18 minutos no total) e 18 pessoas analisadas, o método alcançou 90% de precisão na identificação de roupas: 88% para peças superiores e 94% para peças inferiores. Os resultados mostram eficiência mesmo em cenários com sobreposição parcial ou variações sutis de cor.

Como contribuição, o trabalho entrega uma solução prática para tornar sistemas de vigilância residencial mais inteligentes e menos dependentes da análise humana, aumentando a segurança e a confiabilidade do monitoramento.

2. CONCEITOS FUNDAMENTAIS

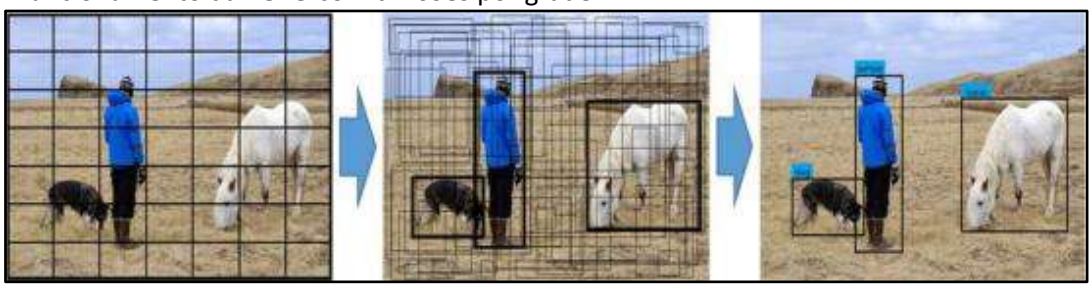
Nesta seção são abordados os conceitos fundamentais para o entendimento da rede neural YOLO utilizada para detecção de objetos, bem como o esclarecimento das abordagens utilizadas para a

implementação no trabalho. Também são abordados os espaços de cores RGB, HSV e YCbCr necessários no processo de segmentação de pele para a identificação dos tipos de roupas.

2.1 YOLO

Segundo Shinde, Kothari e Gupta (2018), YOLO (*You Only Look Once*) é uma abordagem de rede neural avançada para a detecção de objetos, que aplica uma única CNN (*Convolutional Neural Network*) à imagem inteira e divide a imagem em grades. A previsão das caixas delimitadoras e a respectiva pontuação de confiança são calculadas para cada grade. Essas caixas delimitadoras são analisadas pela pontuação de confiança prevista (Figura 1).

Figura 1. Funcionamento da YOLO com divisões por grade.



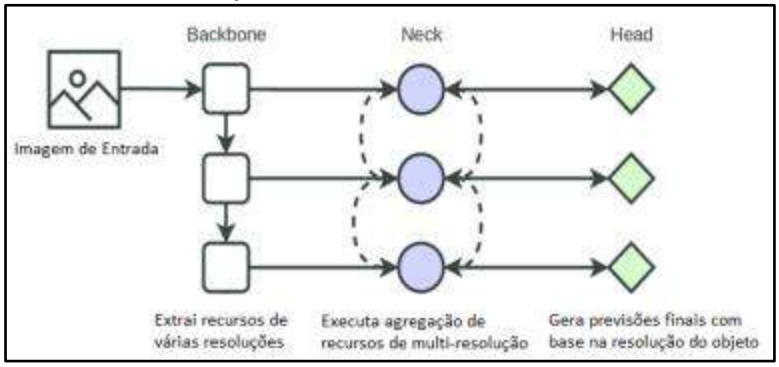
Fonte: Deci Team (2023).

Diferente dos algoritmos que percorrem várias vezes a mesma imagem, em busca dos objetos treinados, a YOLO apenas precisa percorrer a imagem uma única vez (PIEMONTEZ, 2022).

Os detectores de objetos podem ser divididos em duas categorias: Detectores de Um Estágio (*One-Stage Detector*) e Detectores de Dois Estágios (*Two-Stage Detector*). Os detectores de dois estágios separam a tarefa de localização e classificação de objetos para cada caixa delimitadora. Detectores de um estágio fazem previsões para localização e classificação de objetos ao mesmo tempo. A YOLO é considerada um detector de um estágio (Solawetz, 2020).

Um sistema geral de detecção de ponta a ponta consiste em três segmentos principais: (a) *backbone*, (b) *neck*, (c) *head*. O *backbone* de um mecanismo de detecção realiza a extração de características. As características extraídas são encaminhadas para o *neck* do detector de objetos, que realiza uma fusão das características coletadas das diferentes camadas do modelo *backbone*. Finalmente, o *neck* submete os mapas de características para o *head* do modelo detector. O *head* finalmente prevê as classes e as regiões das caixas delimitadoras, que são a saída final produzida pelo modelo de detecção de objetos. O *head* pode produzir uma pilha de saídas, principalmente projetadas para reconhecer diferentes tamanhos de objetos em uma imagem (Kateb *et al.*, 2021). Na Figura 2 é apresentada a estrutura de um detector de objetos.

Figura 2. Estrutura de um detector de objetos.



Fonte: Adaptado de Kateb *et al.* (2021).

2.1.1. Backbone

Neste trabalho foi utilizada a rede neural YOLOv8 devido à sua alta velocidade de processamento quando comparado com outros modelos com o mesmo propósito, como a R-CNN (*Regions with*

Convolutional Neural Networks), e SSD (*Single Shot MultiBox Detector*), apesar de a R-CNN trabalhar melhor com conjuntos de dados onde existem objetos muito pequenos, os cenários apresentados não justificaram a troca de velocidade de processamento para obter maior precisão.

A rede de *backbone* para um detector de objetos normalmente é pré-treinada na classificação ImageNet. O pré-treinamento significa que os pesos da rede já foram adaptados para identificar características relevantes em uma imagem, embora sejam ajustados na nova tarefa de detecção de objetos (Solawetz, 2020).

YOLO utiliza CPSDarknet53, uma versão modificada da arquitetura Darknet, como seu *backbone*. O CSPDarknet53 (*Cross Stage Partial* Darknet53) é uma variação da arquitetura Darknet53, projetada para otimizar a eficiência e a precisão em redes neurais profundas. Essa arquitetura adota a técnica *Cross Stage Partial* (CSP), que divide o fluxo de características em duas partes: uma processada pelo bloco convolucional denso (DenseBlock) e outra que contorna esse bloco, sendo posteriormente combinada à saída da primeira. Essa estratégia reduz a redundância de gradientes, melhora a utilização de memória, acelera a inferência e, ao mesmo tempo, mantém alta capacidade de extração de características. Dessa forma, o CSPDarknet53 fornece um *backbone* mais leve e eficiente em comparação ao Darknet53 original, tornando-se uma escolha recorrente em arquiteturas de detecção modernas, como a YOLOv8, sobretudo em aplicações de tempo real (Wang *et al.*, 2020), aumentando a capacidade de aprendizado e eficiência.

2.1.2. Neck

Segundo Bouraya e Belangour (2021), a arquitetura inclui uma nova estrutura de '*neck*', que é responsável pela característica de fusão, seu papel é coletar mapas de características de diferentes estágios. Os modelos de *neck* são compostos por vários caminhos de cima para baixo e vários caminhos de baixo para cima. A ideia por trás dessa agregação de características existente neste modelo é permitir que as características de baixo nível interajam mais diretamente com as características de alto nível. Essa é uma combinação crucial para informações multi-escala e aprimora a habilidade do modelo de detectar objetos de tamanhos variados.

YOLOv8 introduz a PANet (*Path Aggregation Network*), uma característica de rede em pirâmide que facilita o fluxo de informação através de diferentes escalas. PANet aumenta a habilidade do modelo em lidar com objetos em diversas escalas de uma maneira mais efetiva (Torres, 2024).

2.1.3. Head

De acordo com Bouraya e Belangour (2021), *head* é a última parte do modelo de detecção de objetos, nela é feita a previsão das caixas delimitadoras e das classes dos objetos. Pelo fato da YOLO ser um detector de um estágio, ela possui altas velocidades de inferência, onde o modelo prevê as caixas delimitadoras em uma única etapa.

A YOLO *head* prediz coordenadas das caixas delimitadoras, a pontuação dos objetos e as probabilidades de classes para cada ponto de caixa associado a grade. A arquitetura utiliza as caixas para eficientemente predizer objetos de diferentes tamanhos e formas (Torres, 2024).

2.2 YOLO POSE

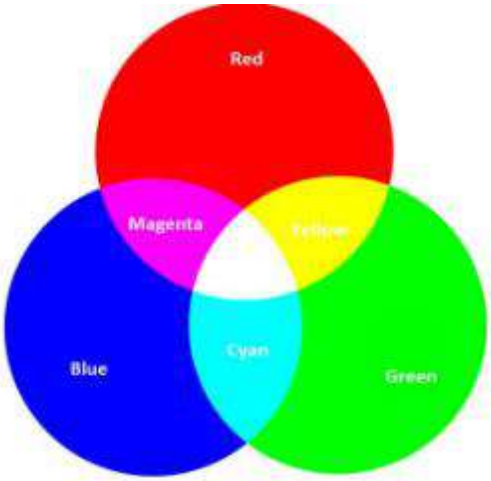
YOLO-Pose é uma extensão da arquitetura de rede neural YOLO (*You Only Look Once*) projetada para a detecção e estimação de posturas humanas. Enquanto o modelo YOLO original é amplamente utilizado para a detecção de objetos em imagens, o modelo YOLO-Pose adapta essa capacidade para identificar e localizar pontos chave do corpo humano, como articulações, em uma imagem (Ultralytics, 2024).

A estimativa de postura é uma tarefa que envolve identificar a localização de pontos específicos em uma imagem, geralmente chamados de pontos chave. Os pontos chave podem representar várias partes do objeto, como articulações, marcos ou outras características distintivas. As localizações dos pontos chave são geralmente representadas como um conjunto de coordenadas 2D [*x*, *y*] ou 3D [*x*, *y*, *visible*]. A saída de um modelo de estimativa de postura é um conjunto de pontos que representam os pontos chave em um objeto na imagem, geralmente acompanhados das pontuações de confiança para cada ponto (Ultralytics, 2024).

2.3 Espaço de cor RGB, HSV e YCbCR

O espaço de cor RGB (Figura 3) é composto por três cores primárias: *Red* (R), *Green* (G) e *Blue* (B). Quando combinadas, essas cores permitem a visualização de uma vasta gama de cores (Ennehar; Brahim; Hicham, 2010). Segundo Rewar e Lenka (2013), o RGB é um modelo de cores básico, do qual todos os outros modelos derivam. É amplamente utilizado para o processamento e armazenamento de imagens digitais. No entanto, por ser um modelo sensível à luz, ele dificulta a distinção das regiões coloridas.

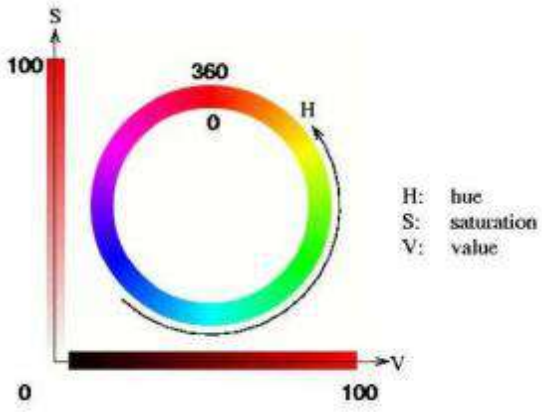
Figura 3. Modelo de cor RGB.



Fonte: Rewar e Lenka (2013).

O espaço de cor HSV contém três canais de cores: *Hue* (H), que define a percepção de cor; *Saturation* (S), que indica o quão pura é a cor; e *Value* (V), que descreve o quanto de brilho as imagens contêm (Al-Tairi *et al.*, 2014; Rewar; Lenka, 2013). Na Figura 4 é apresentado o modelo de cor HSV.

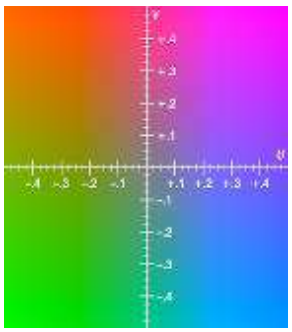
Figura 4. Modelo de cor HSV.



Fonte: Ennehar, Brahim e Hicham (2010).

O espaço de cor YCbCr é composto pelo canal Y (*luma* – brilho) responsável pela especificação da luminância, e pelos canais Cb (B-Y – diferença de crominância azul) e Cr (R-Y – diferença de crominância vermelha). O YCbCr é um modelo de cor independente da luminância, em comparação ao modelo RGB (Rewar; Lenka, 2013). Ennehar, Brahim e Hichan (2010) mostram que o YCbCr é amplamente utilizado em vídeos digitais, tanto para a transmissão de cores quanto para a compressão de imagens, ele separa a informação de cor (crominância) da informação de luminância (brilho), o que permite uma compressão mais eficiente, preservando a qualidade visual. Na Figura 5 é apresentado o modelo de cor YCbCr.

Figura 5. Modelo de cor YCbCr.

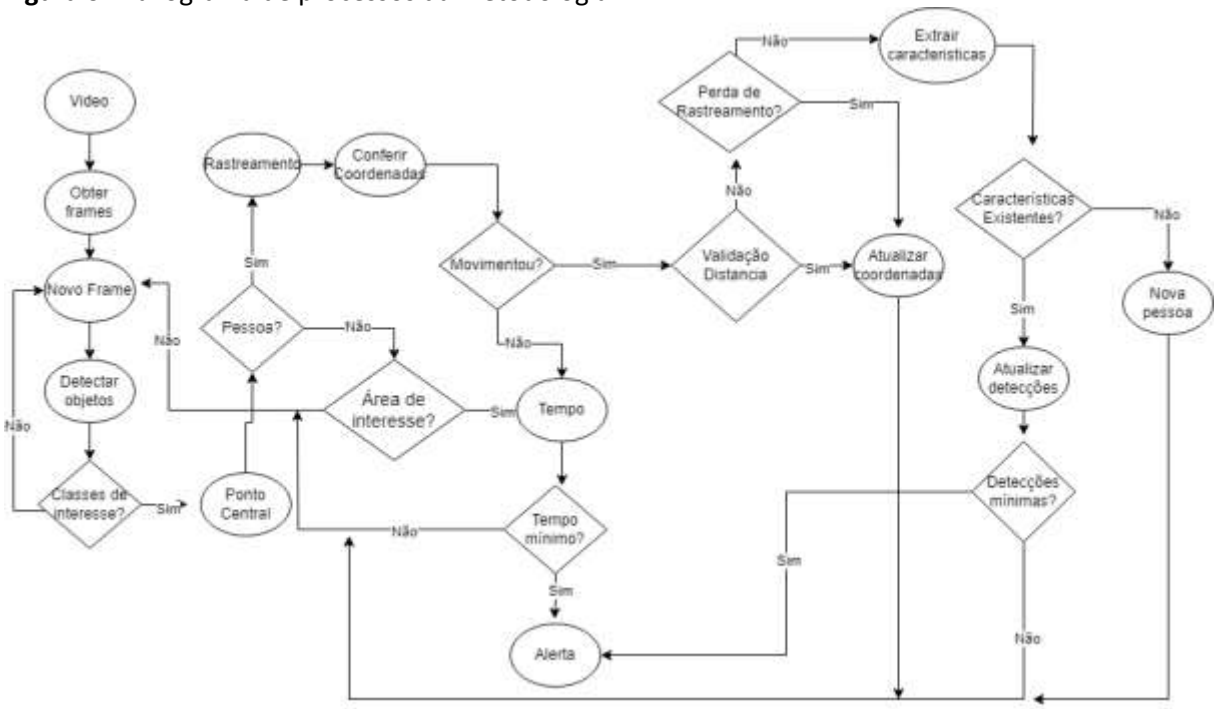


Fonte: Ennehar, Brahim e Hicham (2010).

3. MÉTODO PROPOSTO

O método proposto possui uma sequência de etapas que são descritas sucintamente e exploradas nas subseções na sequência. A partir dos frames de vídeo, é aplicada a rede neural convolucional YOLOv8 para detecção dos objetos de interesse que são pessoas e veículos (motos, carros, vans, furgões entre outros). Com as classes dos objetos de interesse detectadas, é extraído o ponto central de cada objeto (Subseção 3.1). O centro do objeto é utilizado para diminuir o número de comparações necessárias entre os objetos. Se o objeto for uma pessoa, é aplicado o rastreamento enquanto ela estiver em cena (Subseção 3.3), atualizando sempre suas coordenadas. A movimentação da pessoa é acompanhada e, ao não se movimentar, é analisado o tempo em que ficou parada (Subseção 3.2.3). Esse tempo é extraído e utilizado para análises, como verificar se a pessoa está dentro de uma área de interesse (Subseção 3.2) sem se movimentar durante um período de tempo determinado. A validação de estar dentro da região de interesse (ROI - *Region of Interest*) é feita utilizando a distância entre o centro de cada objeto (Subseção 3.2.2). Durante o processo de rastreamento, podem ocorrer falhas, e é preciso compensar essas possíveis falhas devido à oclusão, detecção ou falha na captura das imagens (Subseção 3.3.1). No caso de não ocorrer uma nova detecção da pessoa dentro de um tempo preestabelecido, é feito o processo para obter suas características (Subseção 3.4), que então são analisadas e comparadas caso já existem outras pessoas que passaram por esse processo, aplicando a re-detecção (Subseção 3.4.2). Todos os processos do método proposto são mostrados no Fluxograma na Figura 6.

Figura 6. Fluxograma de processos da metodologia.



Fonte: Os autores.

3.1 Obtenção do Ponto Central

O ponto central de cada objeto é determinado por meio das coordenadas das caixas delimitadoras fornecidas pela YOLO, com as coordenadas sendo representadas como x_1, y_1 para o canto superior esquerdo e x_2, y_2 para o canto inferior direito.

Para cada objeto detectado, essas coordenadas delimitadoras são utilizadas para calcular um ponto central, que serve como uma referência precisa para o objeto no plano bidimensional. Esse ponto central é obtido ao calcular a média das coordenadas dos eixos *X* e *Y*, resultando em uma localização exata que facilita o processo de análise de coordenadas. Esse cálculo é essencial para o rastreamento eficiente e a validação da posição do objeto dentro da cena, permitindo uma análise mais detalhada e precisa. O ponto central é obtido calculando a média das coordenadas *x* e *y* dos dois pontos fornecidos x_1 e x_2, y_1 e y_2.

Figura 7. Ponto central através das caixas delimitadoras.

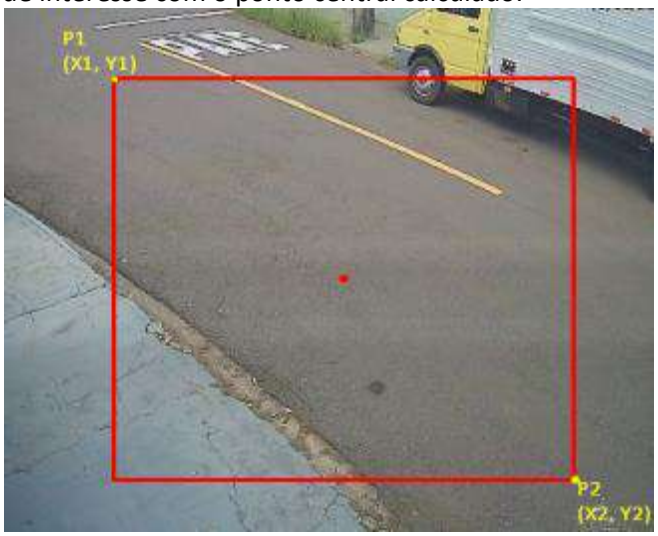


Fonte: Os autores.

3.2 Regiões de Interesse (ROI)

As regiões de interesse (ROI - *Region of Interest*) são áreas específicas em uma imagem ou em um conjunto de dados que são identificadas e selecionadas para análise ou processamento adicional. Essas áreas são delimitadas manualmente através de suas coordenadas fornecidas por dois pontos, *P*1 e *P*2. Como apresentado na Figura 8, essas coordenadas definem os limites exatos da ROI. Além disso, o ponto central da ROI também é extraído para facilitar a detecção de objetos que estejam dentro dessa área específica. Esse ponto central é crucial, pois simplifica a análise e o rastreamento de objetos em repouso dentro da ROI (detalhado na Subseção 3.2.1).

Figura 8. Exemplo de área de interesse com o ponto central calculado.



Fonte: Os autores.

3.2.1 Objetos em Áreas de Interesse

Um objeto é considerado dentro de uma área de interesse quando os pontos do objeto estão dentro de um limite estabelecido, conforme ilustrado na Figura 9. Esse limite é verificado durante um período de tempo, conforme descrito na Subseção 3.2.3. A determinação da permanência do objeto dentro dessa área baseia-se no cálculo da distância euclidiana entre dois pontos, como apresentado na Subseção 3.2.2. Essa condição é essencial para validar a posição do objeto em relação à região de interesse, garantindo que ele permaneça dentro dos limites especificados por um tempo determinado, permitindo uma análise mais precisa e detalhada de sua localização e movimento.

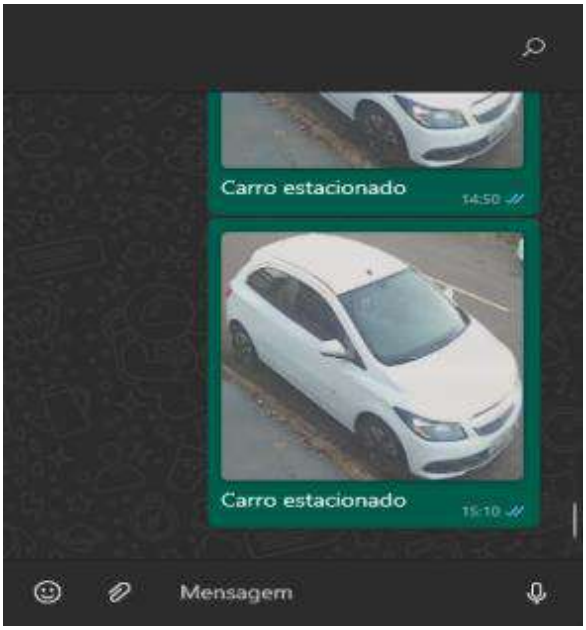
Figura 9. Objeto dentro da área de interesse.



Fonte: Os autores.

Se for constatado que o objeto está dentro da região de interesse no tempo estabelecido, o objeto é segmentado e a imagem resultante enviada em uma mensagem para o smartphone do proprietário da residência. Para realizar essa operação de envio de mensagem, foi utilizada a biblioteca “PyWhatKit” (Pywhatkit, 2022) (exemplo de mensagem enviada na Figura 10). A biblioteca utiliza o WhatsApp Web para o envio, juntamente com a biblioteca de automação “PyAutoGUI” (Pyautogui, 2019) para enviar a mensagem. Esse processo automatizado garante que as notificações sobre a presença do objeto na área de interesse sejam entregues no momento do ocorrido.

Figura 10. Exemplo de mensagem enviada contendo um veículo parado em uma região de interesse estabelecida.



Fonte: Os autores.

3.2.2 Validação de distância de objetos

O Algoritmo 1 determina se os pontos de interesse estão localizados a uma determinada distância em relação ao ponto central de outro objeto, retornando de acordo com esse critério.

Algoritmo 1 - Validação de distância de objetos
Entrada: Coordenadas (x_1, y_1) e (x_2, y_2) dos objetos.
Saída: Indicador verdadeiro ou falso para a distância.
1: Defina o ponto central de verificação (RoiX, RoiY).
2: Para cada par de coordenadas (x_1, y_1) e (x_2, y_2) dos objetos de interesse:
3: Calcule o ponto central (midX, midY).
4: Calcule a distância entre (RoiX, RoiY) e (midX, midY).
5: **Se** a distância for menor que x:
    Retorne Verdadeiro.
6: **Caso contrário**:
    Retorne Falso.

3.2.3 Tempo de análise

Para que se possa considerar que um objeto esteja realmente dentro de uma área de interesse, além de utilizar os pontos do objeto quando estão dentro de um limite estabelecido, também é verificado o tempo de permanência. Entretanto, o tempo não pode ser considerado exato, pois pode variar conforme o processamento de cada computador e a taxa de quadros do vídeo (*frames*). Dessa forma necessita-se obter, inicialmente, a taxa de quadros do vídeo analisado e, a cada detecção, quando o objeto está dentro da região de interesse, o quadro em questão é armazenado. Esse quadro é então comparado com as próximas detecções, utilizando o quadro atual e a taxa de quadros FPS (*Frames per Second*) do vídeo como base. Dessa forma, garante-se que a análise e o monitoramento do objeto dentro da região de interesse sejam mais precisos, independentemente das variações no processamento e na taxa de quadros.

3.3 Rastreamento de pessoas

O processo de rastreamento de pessoas atua desde o primeiro momento em que uma pessoa aparece na cena do vídeo até sua saída, tendo como base seu ponto central. Uma vez que uma pessoa é detectada em cena, atribui-se a ela um ponto central, e as coordenadas desse ponto são armazenadas, assim como algumas informações, como o último frame em que foi detectado, a imagem e um identificador.

Ao detectar uma nova pessoa, verifica-se se existe alguma que esteja sendo rastreada. Então, a nova coordenada do ponto central é utilizada e comparada com as demais rastreadas. Essa diferença de distância leva em conta dois fatores: o tamanho horizontal obtido por meio das caixas delimitadoras e o último frame em que foi capturado, considerando que a detecção de objetos possa falhar ou que ocorra sobreposição por algum elemento na cena. Quanto maior essa diferença de frames, maior será a possível área de deslocamento, com uma perda máxima de três segundos (definido empiricamente), sendo superior a esse valor, é iniciado o processo de extração de características para determinar o tipo de roupa que cada pessoa está utilizando (regata, camisa, manga longa, shorts ou calça) e as cores de predominância. Caso esteja dentro da faixa de valor, as coordenadas do ponto central da região da pessoa na imagem, assim como o último frame detectado e seu status de rastreamento são atualizados. O status é atualizado para evitar que objetos muito próximos interfiram uns com os outros. Não encontrando nenhum correspondente, esse novo objeto é adicionado à lista de rastreados. Na Figura 11 é mostrado um frame do processo de rastreamento, em que é aplicada uma identificação única para cada pessoa na cena, cuja identificação é representada na imagem pelos números 0 e 1 em vermelho.

Figura 11. Exemplo de rastreamento com identificação única.



Fonte: Os autores.

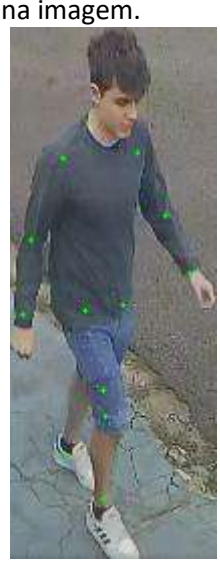
3.3.1 Perda de rastreamento ou oclusão

O rastreamento pode apresentar erros devido a falhas na detecção do objeto, oclusão e falha na captura de imagem. Considerando isso, foi necessário encontrar maneiras de contornar esses cenários, realizando o registro do último frame em que o objeto foi detectado. Foi considerada também a taxa de quadros do vídeo (FPS – *Frames Per Second*) durante o processo de rastreamento de pessoas (Subseção 3.3). A distância utilizada para validar a proximidade entre os objetos (Subseção 3.2.2) pode variar de acordo com o frame atual do vídeo e a taxa de quadros, aumentando conforme aumenta a diferença de tempo entre o frame atual e o último detectado do objeto.

3.5 Extração de características

Para este processo é necessário fazer algumas preparações na imagem para que possam ser detectados os tipos de roupa da pessoa na imagem. A partir da seleção de caixas delimitadoras (*bounding boxes*) contendo as pessoas, o primeiro passo é a extração das coordenadas das articulações de interesse utilizando a YOLOv8-Pose, que é uma extensão avançada do modelo de detecção de objetos YOLO, projetada especificamente para a estimativa de pose humana. A estimativa de pose é a tarefa de detectar e localizar diversas partes do corpo humano (articulações) como apresentado na Figura 12, em uma imagem ou vídeo. Essas coordenadas são armazenadas para utilizar em combinação com o próximo passo, que é a detecção da pele humana.

Figura 12. Pontos de articulação detectados na imagem.



Fonte: Os autores.

Para identificar a pele da pessoa na imagem, foi utilizado o método proposto por Rahman *et al.* (2006), cujo método realiza uma varredura em cada pixel da imagem utilizando três filtros baseados nos espaços de cores (RGB, HSV e YCbCr) para indicar intervalos válidos que representam a pele humana. O pixel que passar pela verificação de todos filtros é mantido na imagem, caso contrário então é atribuído um pixel preto em substituição a ele.

No trabalho proposto, o espaço de cor RGB não foi utilizado, pois nos testes realizados, esse espaço de cor indicou muitos falsos positivos no processo de segmentar a pele. Neste caso, foram utilizados apenas os espaços de cores HSV e YCbCr do método original. As verificações dos filtros adotados dos dois espaços de cores utilizados são as seguintes:

- HSV
  H > 230 ou H < 25
- YCrCb
  Cr <= 1.5862 x Cb + 20 E
  Cr >= 0.3448 x Cb + 76.2069 E
  Cr >= -4.5652 x Cb + 234.5652 E
  Cr <= -1.15 x Cb + 301.75 E
  Cr <= -2.2857 x Cb + 432.85

Ao final da varredura, apenas os pixels que representam a cor de pele permaneceram na imagem assim como apresentado na Figura 13.

Figura 13. Segmentação de pele. Em (a) tem-se o resultado após aplicar o filtro baseado no espaço de cor HSV. Em (b) o resultado após aplicar o filtro baseado no espaço de cor YCrCb. Em (c) a junção de ambos resultados.

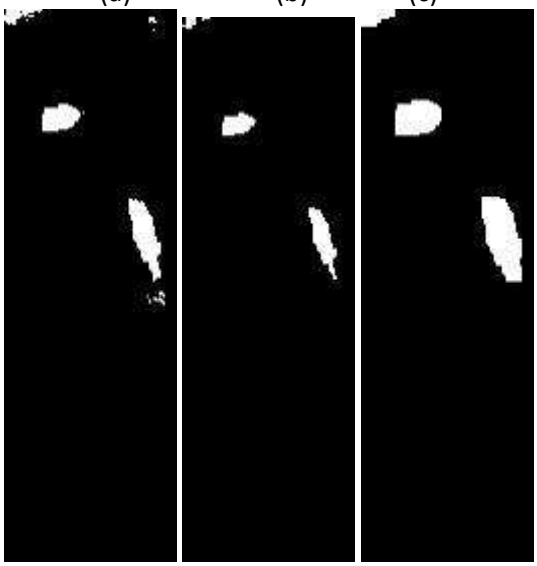


Fonte: Os autores.

Com a imagem contendo somente os pixels próximos da tonalidade da pele humana, inicia-se uma varredura para trocar esses pixels para a cor branca. Em seguida, algumas operações morfológicas são aplicadas, com um *kernel* (elemento estruturante) 3x3 de valores 1. Para remover alguns casos que foram considerados como pele, porém, contém uma área pequena na imagem, é aplicada uma operação de erosão para eliminar esses pequenos ruídos da imagem. Com a imagem resultante, são aplicadas três operações de dilatação para expandir as áreas brancas, conforme mostrado na Figura 14.

Figura 14. Resultado da aplicação de todo o processo para segmentar as regiões de pele humana. Em (a) tem-se a substituição dos pixels de pele por pixels brancos. Em (b) a aplicação da operação morfológica de erosão. Em (c) a aplicação de três operações morfológicas da dilatação.

(a) (b) (c)



Fonte: Os autores.

As coordenadas de alguns pontos chave obtidas com o uso da YOLOv8-Pose são utilizadas em conjunto com a imagem binária contendo pele e não pele (Figura 14 (b)). Alguns pontos são comparados para checar se naquela coordenada específica está a cor branca (representando a pele). Como apresentado na Figura 15, é possível observar que a coordenada representando o cotovelo direito está branca, porém, a coordenada da esquerda não está. Sendo assim, é levada em consideração a informação referente ao maior grau de confiança da detecção.

Figura 15. Verificação de tipo de roupa.

(a) (b)



Fonte: Os autores.

O método utilizado para detecção das cores das roupas foi o espaço de cor HSV. Uma região da peça de roupa é recortada da imagem, a partir do ombro mais visível até a região da cintura, conforme mostrado na Figura 16.

Figura 16. Exemplo de recorte da peça de roupa.



Fonte: Os autores.

Em seguida, é feita a conversão dessa região segmentada do espaço de cor RGB para o HSV, e calculado o histograma para cada canal. Com isso, o valor estatisticamente dominante presente em cada canal é extraído e armazenado, para futuras comparações.

3.4.2 Re-detecção

A partir das características extraídas, como o tipo de peça de roupa e a cor predominante de cada item os dados são comparados com os previamente armazenados para verificar a presença recorrente de um mesmo indivíduo, configurando um processo de re-identificação. Para que essa correspondência ocorra, o tipo da peça de roupa deve coincidir exatamente com os registros anteriores, enquanto as cores podem apresentar pequenas variações, sendo aceita uma margem de erro de até 5% na correspondência das tonalidades.

Caso ambas as condições sejam satisfeitas, o sistema considera que houve uma re-identificação e realiza uma nova detecção associada ao mesmo indivíduo. Quando a quantidade mínima de re-detecções é atingida, conforme parametrizado, uma notificação de alerta é automaticamente enviada via WhatsApp, indicando a recorrência daquela pessoa no ambiente monitorado.

4. RESULTADOS

Neste trabalho, foram utilizados nove vídeos de durações variadas, totalizando 18 minutos, com o objetivo de validar a precisão do algoritmo na identificação das roupas utilizadas. No total, foram analisadas 18 pessoas, cada uma utilizando diferentes tipos de vestimentas.

Durante o experimento, cada indivíduo apareceu nas cenas dos vídeos em múltiplas situações, permitindo ao algoritmo identificar e catalogar as roupas utilizadas. Os dados foram coletados por meio de análise visual manual, sendo comparados com os resultados produzidos pelo algoritmo para cada situação, conforme apresentado na Tabela 1 e na Tabela 2. Os resultados mostraram uma precisão média de 90% na detecção das vestimentas, confirmando a robustez do modelo em reconhecer diferentes tipos de roupas. As vestimentas foram classificadas com base nos tipos. Os experimentos indicaram que o algoritmo consegue diferenciar eficazmente entre várias combinações de roupas.

Em relação às roupas superiores, o algoritmo atingiu uma precisão de 88% na identificação, incluindo regatas, camisas e roupas de manga longa. Já em relação às roupas inferiores como calças e shorts a precisão foi de 94%.

Os vídeos utilizados apresentaram algumas situações de oclusão parcial. Com o ponto de captura fixo, variou-se a distância em que essas situações ocorriam, além de haver casos em que a roupa utilizada diferia apenas na cor, mantendo o mesmo tipo. Esses fatores permitiram testar a capacidade do algoritmo em condições mais próximas à realidade, incluindo desafios como sobreposição parcial e variações visuais sutis.

Tabela 1. Resultados da detecção de roupas superiores.

	Regata	Camisa	Manga Longa
Regata	2		
Camisa		13	
Manga Longa			1
Sem Resposta		2	

Fonte: Os autores.

Tabela 2. Resultados da detecção de roupas inferiores.

	Shorts	Calça
Shorts	7	
Calça		10
Sem Resposta		1

Fonte: Os autores.

5. CONSIDERAÇÕES FINAIS

Neste trabalho, foi proposto um método para monitoramento de câmeras de segurança, combinando detecção de pessoas e veículos com YOLO, rastreamento baseado no ponto central e extração de características das vestimentas para re-identificação. A metodologia apresentou bom desempenho, com detecção precisa e rastreamento eficiente em cenários de baixo fluxo, além de identificação correta das roupas em imagens sem oclusão.

Limitações foram observadas em casos de oclusão parcial e aparições parciais de indivíduos, apontando a necessidade de aprimoramento para múltiplos objetos em movimento e adaptação a diferentes condições de iluminação e câmeras.

Trabalhos futuros incluem testes em cenários com maior fluxo de pessoas e veículos, ampliação das classes de objetos de interesse, processamento em tempo real em hardware de baixo custo e integração em sistemas residenciais ou comerciais. Na prática, o método pode fornecer monitoramento inteligente, permitindo identificar indivíduos e veículos em áreas críticas, analisar tempos de permanência e emitir alertas em tempo real, aumentando a confiabilidade da vigilância e reduzindo a dependência da análise humana.

REFERÊNCIAS

AL-TAIRI, Z. G. *et al.* Skin segmentation using YUV and RGB color spaces. **Journal Of Information Processing Systems**, v. 10, n. 2, p. 283-299, 2014. DOI: https://doi.org/10.3745/JIPS.02.0002.

BOURAYA, S.; BELANGOUR, A. Deep Learning based Neck Models for Object Detection: A Review and a Benchmarking Study. **International Journal of Advanced Computer Science and Applications**. 2021. DOI: https://doi.org/10.14569/ijacsa.2021.0121119.

BRANDIZZI, L. **Visão computacional: O que é? Como funciona?:** Uma breve explicação sobre os sistemas artificiais que permitem que computadores "enxerguem". [S.l.]: Serpro, 7 abr. 2020. Disponível em: https://www.serpro.gov.br/menu/noticias/noticias-2020/o-que-eh-visao-computacional. Acesso em: 15 ago. 2023.

DECI Team. **The History of YOLO Object Detection Models from YOLOv1 to YOLOv8**. 2023. Disponível em: https://deci.ai/blog/history-yolo-object-detection-models-from-yolov1-yolov8/. Acesso em 22 maio 2024.

ENNEHAR, B. C.; BRAHIM, O.; HICHAM, T. An appropriate color space to improve human skin detection. Laboratory of Automatics and Informatics of Guelma (LAIG 2010). **Proceedings [...].** Guelma/Algeria: Eight May 1945 University of Guelma, 2010.

KATEB, F. A.; MONOWAR, M. M.; HAMID, M. A.; OHI, A. Q.; MRIDHA, M. F. FruitDet: Attentive Feature Aggregation for Real-Time Fruit Detection in Orchards. **Agronomy**, v. 11, n. 12, 2021. DOI: https://doi.org/10.3390/agronomy11122440.

NUNES, F. Setor de segurança eletrônica registra aumento nas vendas durante a pandemia. **Diario da Região**, 17 de março de 2021. Disponível em: https://www.diariodaregiao.com.br/economia/setor-de-seguranca-eletronica-registra-aumento-nas-vendas-durante-a-pandemia-1.10005. Acesso em: 15 ago. 2023.

OJHA, S.; SAKHARE, S. Image processing techniques for object tracking in video surveillance - A survey. *In*: ICPC: INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING, 2015, p. 1-6. DOI: https://doi.org/10.1109/PERVASIVE.2015.7087180.

PIEMONTEZ. **YOLO para Detecção de Objetos – Visão Geral**. 2022. Disponível em: https://visaocomputacional.com.br/yolo-para-deteccao-de-objetos-visao-geral /. Acesso em: 15 maio. 2024.

PYAUTOGUI. **Welcome to PyAutoGUI's documentation** 2019. Disponível em: https://pyautogui.readthedocs.io/en/latest/. Acesso em: 27 junho 2024.

PYWHATKIT. 2022. Disponível em: https://github.com/Ankit404butfound/PyWhatKit/wiki. Acesso em: 27 junho 2024.

RODRIGUES, C. Z. Discurso da Vigilância na escola: o caso do "sorria você está sendo filmado". *In*: CONGRESSO INTERNACIONAL LINGUAGEM E INTERAÇÃO, 2005, São Leopoldo. Anais [...]. São Leopoldo, 2005.

RAHMAN, N. A. B. A.; WEI, K. C.; SEE, J. RGB-H-CbCr skin model for human face detection. *In*: MMU INTERNATIONAL SYMPOSIUM ON INFORMATION AND COMMUNICATION TECHNOLOGIES (M2USIC 2006). **Proceedings [...]** Malaysi, 2006.

REWAR, E.; LENKA, S. K. Comparative analysis of skin color based models for face detection. **Signal & Image Processing: An International Journal**, v. 4, n. 2, Apr. 2013. DOI: https://doi.org/10.5121/sipij.2013.4206.

SHINDE, S.; KOTHARI, A.; GUPTA, V. YOLO based Human Action Recognition and Localization. **Procedia Computer Science**, v. 133, p. 831–838, 2018. DOI: https://doi.org/10.1016/j.procs.2018.07.112.

SOLAWETZ, J. **YOLOv4 - An explanation of how it works**. 2020. Disponível em: https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/#yolov4-backbone-network-feature-formation. Acesso em: 15 maio. 2024.

SOUZA, D. A.; SOUZA, C. G.; OLIVEIRA, S. J.; FAJAN, F. D.; NABARRO, C. B. M. **Câmeras de segurança e seus sistemas tecnológicos: Percepções sobre os motivos da utilização**. Disponível em: https://www.aedb.br/seget/arquivos/artigos17/12425136.pdf. Acesso em: 15 ago. 2023.

TORRES, J. **YOLOv8 Architecture Explained:** Exploring the YOLOv8 Architecture. 2024. Disponível em: https://yolov8.org/yolov8-architecture-explained/. Acesso em: 22 maio 2024.

ULTRALYTICS. **Ultralytics YOLO Docs** 2024. Disponível em: https://docs.ultralytics.com/models/yolov8/. Acesso em: 22 maio 2024.

WANG, C.-Y.; LIAO, H.-Y. M.; WU, Y.-H.; CHEN, P.-Y.; HSIEH, J.-W.; YEH, I.-H. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. *In*: IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS (CVPRW 2020). **Proceedings** [...] p. 390–391, 2020. DOI: https://doi.org/10.1109/CVPRW50498.2020.00203.