http://journal.unoeste.br/index.php/ce/index
DOI: 10.5747/ce.2024.v16.e430
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 29/06/2024 Revisado: 28/08/2024 Aceito: 23/11/2024

SEGURANÇA RESIDENCIAL INTELIGENTE: DETECÇÃO DE COMPORTAMENTO SUSPEITO POR MEIO DE ANÁLISE DE VÍDEO

Smart Home Security: Detecting Suspicious Behavior Through video Analytics

Ariel Araujo Oliveira Menezes; Leandro Luiz de Almeida; Mário Augusto Pazoti; Francisco Assis da Silva

Universidade do Oeste Paulista – UNOESTE; Presidente Prudente, SP
E-mail: arielmenezess@unoeste.edu.br, llalmeida@unoeste.br, mario@unoeste.br, chico@unoeste.br

RESUMO - Este trabalho apresenta o desenvolvimento de uma solução computacional para monitoramento inteligente de residências, projetado para detectar pessoas, analisar seus comportamentos e emitir alertas para atividades anormais. Foi utilizada a tecnologia YOLOv8-Pose para detecção de pontos chave das pessoas e uma rede neural MLP (*Multilayer Perceptron*) modificada com características de LSTM (*Long Short-Term Memory*) para classificação de comportamentos. O conjunto de dados de treinamento foi montado por meio da gravação de vídeos usando um smartphone estrategicamente posicionado para capturar a região frontal de uma residência, resultando em 154 recortes de vídeos na qual o ator, com comportamentos normais, caminha ao longo da cena, e 154 vídeos em que há comportamentos anormais, na qual o ator realiza uma série de ações que poderiam resultar na invasão da residência, como tentar escalar a grade ou arrombar o portão da garagem. Para realizar os experimentos, vídeos com múltiplos atores e diferentes comportamentos foram analisados para avaliar a eficácia da metodologia desenvolvida. Os resultados indicaram uma alta taxa de acerto na detecção de comportamentos normais, embora desafios permaneçam em cenários com oclusão parcial. A precisão do modelo na classificação de comportamentos foi de 91,6%, refletindo a eficácia em identificar corretamente as atividades normais e anormais.
Palavras-chave: Monitoramento inteligente, YOLOv8-Pose, MLP, LSTM, detecção de comportamento, segurança residencial.

ABSTRACT – This work presents the development of a computational solution for intelligent home monitoring, designed to detect people, analyze their behaviors and issue alerts for abnormal activities. YOLOv8-Pose technology was used to detect people's key points and a modified MLP (Multilayer Perceptron) neural network with LSTM (Long Short-Term Memory) characteristics was used to classify behavior. The training dataset was created by recording videos using a smartphone strategically positioned to capture the frontal area of a residence. This resulted in 154 video clips where the actor exhibited normal behaviors by walking along the scene and 154 videos where the actor performed actions that could lead to home invasion, such as attempting to climb the fence or break the garage gate. To carry out the experiments, videos with multiple actors and different behaviors were analyzed to evaluate the effectiveness of the developed methodology. The results indicated a high success rate in detecting normal behaviors, although challenges remain in scenarios with partial occlusion. The model's accuracy in classifying behaviors was 91.6%, reflecting its effectiveness in correctly identifying normal and abnormal activities.
Keywords: Intelligent monitoring, YOLOv8-Pose, MLP, LSTM, behavior detection, home security.

1. INTRODUÇÃO

A vigilância por vídeo é uma prática amplamente adotada para aumentar a segurança em áreas residenciais, especialmente em países como o Brasil, onde muitas casas possuem muros e portões. Tradicionalmente, empresas de segurança oferecem monitoramento 24 horas por dia, analisando transmissões ao vivo das câmeras para identificar atividades suspeitas. No entanto, essa abordagem pode ser onerosa e depende de operadores humanos, que estão sujeitos a falhas e cansaço (Shinde; Ashwin; Vikram, 2018).

Este trabalho propõe um sistema de segurança inteligente que utiliza câmeras de segurança externas para detectar comportamentos anormais de pessoas em tempo real, eliminando a necessidade de vigilância humana constante. O sistema emprega a tecnologia do YOLOv8-Pose para a detecção de pessoas e extração de seus pontos chaves corporais. Estes dados são então processados por uma rede neural Multi-Camadas (MLP - *Multilayer Perceptron*) com comportamento de LSTM (*Long Short-Term Memory*), utilizando uma janela deslizante de quadros para capturar a sequência temporal dos movimentos e entender padrões de comportamento.

Para definir comportamentos normais, foram realizados treinamentos com indivíduos andando normalmente em frente à uma residência, entrando e saindo do campo de visão da câmera. Já os comportamentos anormais foram definidos com base em ações suspeitas, tais como olhar insistentemente para dentro da residência, tentar escalar o muro, mexer na fechadura do portão da garagem, tentar erguer o portão da garagem e sentar-se em frente à residência. Estas ações foram escolhidas por representar potenciais tentativas de invasão ou outras atividades suspeitas, para resumir os tipos de comportamento anormais, já que no mundo real um cenário de invasão possui muitas variáveis.

A proposta central deste trabalho é a criação de um sistema de monitoramento que, além de ser leve e de fácil implementação, se destaca pela sua confiabilidade na identificação de comportamentos suspeitos em áreas residenciais. Utilizando inteligência artificial, busca-se um equilíbrio entre precisão e eficiência, permitindo a detecção em tempo real de ações potencialmente suspeitas, como tentativas de invasão ou comportamentos incomuns. Com isso, espera-se não apenas fornecer uma alternativa mais acessível e autônoma à vigilância tradicional, mas também contribuir para a redução de custos operacionais e a diminuição da dependência de operadores humanos.

Após esta seção introdutória, o trabalho está organizado da seguinte maneira. Na Seção 2 são apresentados os trabalhos relacionados, que deram base para o desenvolvimento deste. Na Seção 3 é detalhado todo o processo desenvolvido da metodologia proposta. Na Seção 4 são apresentados os resultados obtidos e, por fim, na Seção 5 encontram-se as considerações finais e propostas para trabalhos futuros.

2. TRABALHOS RELACIONADOS

Shinde, Ashwin e Vikram (2018) apresentaram um estudo no qual propõem um modelo em tempo real para reconhecimento e localização de ações humanas em vídeos de vigilância. Utilizando o modelo YOLO, os autores demonstraram que é possível reconhecer e localizar ações de interesse em quase tempo real a partir de quadros isolados de vídeo, sem a necessidade de analisar o fluxo óptico de múltiplos quadros. A abordagem proposta processa quadros periódicos em vez de vídeos completos, atribuindo rótulos de ação e escores de confiança a cada quadro individualmente. Os escores de confiança representam a probabilidade atribuída pelo modelo a cada detecção ou classificação de ação, indicando o nível de certeza da rede neural na sua predição. Os resultados experimentais, utilizando o conjunto de dados LIRIS Human Activities, mostraram que o método é eficaz e significativamente mais rápido em comparação com abordagens anteriores, atingindo uma precisão de 88,372%. Esse estudo destaca a capacidade da YOLO em detectar e reconhecer ações humanas com precisão e eficiência, utilizando um número reduzido de quadros.

Sultani, Chen e Shah (2018) é evidenciada uma abordagem para a detecção de anomalias em vídeos de vigilância. O método utiliza aprendizado por instâncias múltiplas (MIL) para explorar vídeos normais e anômalos sem a necessidade de anotações temporais detalhadas, que são usualmente trabalhosas. O modelo aprende a localizar automaticamente as anomalias, tratando vídeos como segmentos de vídeo e instâncias. Os autores introduzem um novo conjunto de dados de larga escala com 1900 vídeos de vigilância não editados, cobrindo 13 tipos de anomalias realistas e atividades normais. Os

resultados experimentais mostram que o método proposto supera significativamente as abordagens anteriores na detecção de anomalias, oferecendo um novo benchmark desafiador para essa tarefa.

Ko e Sim (2018) propuseram uma estrutura unificada baseada em uma rede profunda convolucional para detectar comportamentos humanos anormais em imagens RGB padrão. O objetivo principal era melhorar a velocidade de detecção mantendo a precisão do reconhecimento. A estrutura convolucional profunda consiste em três módulos principais: um módulo de detecção e discriminação de sujeitos humanos, um módulo de classificação de posturas para extrair características espaciais de comportamentos anormais e um módulo de detecção de comportamentos anormais baseado em LSTM. Experimentos conduzidos em um conjunto de dados de benchmark mostraram que o método proposto fornece um desempenho satisfatório na detecção de comportamentos anormais em cenários do mundo real. A abordagem foi testada no conjunto de dados UT-Interaction-Data, contendo atividades como: aperto de mãos, abraço, chute, apontar, soco e empurrão, com resultados indicando alta precisão na detecção dessas atividades.

Anala, Makker e Ashok (2019) propuseram um trabalho que aborda a detecção de anomalias em vídeos de vigilância, um desafio crucial devido à vasta quantidade de dados gerados por câmeras de CFTV (Circuito Fechado de TeleVisão). A proposta é um sistema inteligente que realiza a classificação de quadros utilizando um modelo CNN-LSTM (Convolutional Neural Network - Long Short-Term Memory), alcançando uma precisão de 85%. A abordagem envolve a extração de características espaciais de cada quadro (frame de vídeo) através da VGG16 e a passagem dessas características para uma rede LSTM para aprendizado de sequência. O sistema é projetado para classificar quadros em categorias como explosão, luta, acidente de trânsito ou normal. Os resultados indicam que, embora nem todos os quadros sejam classificados corretamente, o modelo é eficaz em identificar eventos anômalos, garantindo uma baixa taxa de falsos negativos. Este estudo destaca a importância da combinação de características espaciais e temporais na detecção de anomalias e sugere melhorias futuras para aumentar a robustez do sistema, incluindo a adição de mais classes de anomalias.

Nasaruddin *et al.* (2020) apresentam um método para detecção de anomalias em vídeos de vigilância, utilizando uma abordagem de atenção visual para focar nas regiões de interesse, em vez de analisar o quadro completo. O método proposto emprega a subtração robusta de fundo para extrair movimentos, que indicam as regiões de atenção, e essas regiões são então processadas por uma Rede Neural Convolucional tridimensional (3D CNN). Utilizando a base de dados UCF-Crime, que contém 1900 vídeos reais de vigilância, o sistema foi testado e demonstrou uma precisão de 99,25%. Durante os experimentos, o modelo foi treinado e testado em um computador equipado com processador Intel i7-7700HQ, 16GB de memória e uma GPU NVIDIA GeForce GTX 1050 Ti. Os resultados mostraram que a metodologia proposta, que se concentra nas regiões de atenção, não só melhorou a precisão da detecção de anomalias em diversos tipos de eventos (roubo, briga e acidentes de trânsito), mas também reduziu significativamente os falsos positivos, tornando o sistema mais eficiente para aplicações industriais e de segurança.

Ji *et al.* (2020) desenvolveram um método de detecção de comportamento anômalo em vídeos de vigilância utilizando o modelo de rede T-Tiny-YOLO (You Only Look Once). A pesquisa define vários comportamentos anômalos dentro de um escopo restrito e, em seguida, propõe uma melhoria na arquitetura da rede YOLO para aumentar o desempenho em tempo real. A implementação do método na plataforma embutida NVIDIA Jetson TX2 demonstrou resultados promissores, alcançando uma velocidade de detecção de 12 quadros por segundo e uma taxa de recall de 80,87%. Esse trabalho se destaca por sua aplicabilidade em cenários de monitoramento externo, com foco na detecção rápida e eficiente de comportamentos humanos anômalos.

Wu e Cheng (2020) propuseram um método de detecção para comportamentos humanos anormais em ambientes internos. No trabalho, os autores abordam a lacuna deixada pelos algoritmos que são mais eficazes em ambientes externos. A metodologia inclui a modelagem de fundo baseada em um modelo de mistura gaussiana, processamento de blocos espaço-temporais e a utilização do algoritmo de clustering fuzzy C-means (FCM) para detectar anomalias. Por meio de experimentos com conjuntos de dados públicos e criados manualmente, os autores demonstraram que seu método não só oferece um desempenho de detecção superior em ambientes internos, mas também é simples de implementar e tem baixa complexidade temporal. Além disso, a precisão da detecção foi melhorada em comparação com outros

métodos existentes, mostrando a viabilidade prática de sua aplicação para a segurança pública em ambientes internos.

Zhang *et al.* (2020) desenvolveram um método para lidar com o desafio de detectar e localizar comportamentos anormais em vídeos de vigilância de cenas com muitas pessoas. O método proposto associa os fluxos ópticos entre múltiplos quadros para capturar trajetórias de curto prazo, introduzindo um descritor de forma baseado em histograma para descrever tais trajetórias. Este descritor reflete fielmente a tendência de movimento e detalhes em patches locais. Além disso, os autores propõem um método para detectar anomalias ao longo do tempo e do espaço, julgando se as semelhanças entre a amostra de teste e as amostras K-NN (K-Nearest Neighbors) recuperadas seguem a distribuição padrão de semelhanças intraclasse homogêneas. Esse é um aprendizado não supervisionado de uma classe que não requer agrupamento nem suposição prévia. As experiências conduzidas em vídeos de vigilância do mundo real demonstram que o método proposto pode detectar e localizar de forma confiável os eventos anormais em sequências de vídeo.

Heidari e Iosfidis (2021) propuseram um método inovador para o reconhecimento de ação humana baseada em esqueleto. Eles apresentaram a PST-GCN (Progressive Spatio-Temporal Graph Convolutional Network), que otimiza a topologia da rede em termos de largura e profundidade. A PST-GCN ajusta progressivamente a topologia do modelo ST-GCN com base no desempenho durante o treinamento. Experimentos realizados em dois conjuntos de dados padrão para reconhecimento de ação baseado em esqueleto demonstram que o método proposto alcança ou supera o desempenho de métodos de última geração, enquanto constrói um modelo compacto com até 15 vezes menos parâmetros, resultando em complexidade computacional significativamente reduzida.

3. MÉTODO PROPOSTO

Nesta seção é apresentada a metodologia empregada no desenvolvimento da solução computacional de videomonitoramento inteligente. O objetivo principal é detectar a presença de humanos no vídeo e analisar seu comportamento para identificar anomalias, acionando alertas quando necessário. Foram utilizadas técnicas de visão computacional e aprendizado de máquina para alcançar esse objetivo.

O funcionamento do sistema começa com a detecção de um humano entrando no quadro do vídeo. A partir desse momento, cada quadro é processado pela rede neural YOLOv8-Pose, que gera um vetor de vetores contendo 17 pontos chave do corpo da pessoa, dispostos no formato [[X,Y], [X,Y], ... [X,Y]]. Após acumular 10 leituras consecutivas (10 quadros), esses dados são consolidados em um único vetor e enviados para a Inteligência Artificial (IA) de detecção de comportamento. Esta IA utiliza uma rede MLP (*Multi-Layer Perceptron*) que simula o comportamento de uma LSTM (*Long Short-Term Memory*) por meio de uma janela deslizante, permitindo uma análise temporal dos dados. Optou-se por uma MLP em vez de uma LSTM devido à natureza binária da saída desejada, que representa 0 (normal) ou 1 (anormal). A MLP é reconhecida por sua eficácia em tarefas de classificação simples como essa.

O vetor de entrada é continuamente atualizado usando uma técnica de janela deslizante (*Slide Window*). Após acumular 10 leituras consecutivas, o primeiro elemento do vetor é removido e uma nova leitura é adicionada no final. Por exemplo, considere uma janela deslizante v, onde uma pessoa entrou no vídeo e permaneceu nele por 10 quadros, resultando em $v[0:9]$. Após processar esse vetor pela MLP, o primeiro elemento $v[0]$ é descartado e uma nova leitura é esperada para ser inserida no final. Assim, após a primeira detecção, a janela deslizante se torna $v[1:10]$, pronta para processar a próxima sequência de quadros.

A metodologia proposta está dividida em cinco fases principais, detalhadas nas subseções a seguir.

3.1. Configuração do Ambiente

Para garantir que o YOLOv8-Pose e o modelo de rede neural MLP possam ser executados de maneira eficiente utilizando uma GPU, foi necessário configurar um ambiente adequado. Embora o trabalho tenha sido realizado em um *laptop* Acer Nitro 5 com um processador Intel i7-7 e GPU GTX 1050Ti, que não é a máquina ideal para o projeto, conseguiu-se um desempenho razoável.

Inicialmente, foi instalada a biblioteca CUDA no Windows 10 para habilitar o uso da GPU nos processos de computação, permitindo uma execução mais rápida e eficiente do YOLOv8-Pose. Em seguida, foi necessário configurar o TensorFlow para rodar o modelo MLP também na GPU. No entanto, devido à

falta de suporte nativo do TensorFlow para o Windows nas versões mais recentes, foi preciso ativar o Windows Subsystem for Linux 2 (WLS2).

Com o WSL2 ativado, que por padrão utiliza o Ubuntu, instalou-se o Python 3.11.5, pois o GTX 1050Ti não suporta a versão mais recente do CUDA, necessitando da versão CUDA 11.8 para garantir a compatibilidade e a correta execução do TensorFlow nesta configuração.

Posteriormente, foi instalado o ambiente Anaconda dentro do WSL2 para facilitar a gestão dos pacotes e a configuração do projeto e foi criado um Virtual Environment (VENV) específico para o trabalho, onde foram instaladas todas as bibliotecas necessárias, incluindo o TensorFlow, para garantir que o modelo MLP pudesse ser executado utilizando os Cuda Cores da NVIDIA.

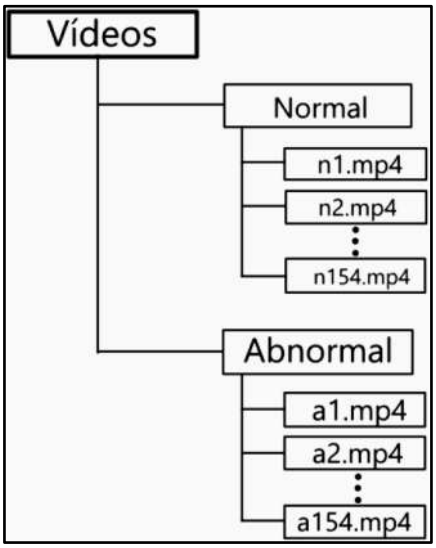
3.2. Gravação da Base de Dados de Vídeo

A gravação da base de dados de treino da rede MLP (*Multilayer Perceptron*) foi realizada utilizando um suporte improvisado para celular, fixado no muro de uma residência, no canto mais à esquerda de quem olha de frente para a casa. Esse posicionamento permitiu capturar toda a extensão da frente da casa, facilitando a coleta de dados a diferentes distâncias da câmera e garantindo uma visão abrangente do cenário.

A base de dados foi composta de dois tipos de comportamentos: normais e anormais. Para os comportamentos normais, foram gravados vídeos de uma pessoa (o ator) entrando e saindo do quadro, simulando o trânsito comum de pedestres na calçada ou rua. Já os comportamentos anormais foram definidos em seis categorias: ficar parado em frente a residência encarando para dentro dela, tentar subir na grade, tentar acesso a fechadura do portão da garagem, tentar erguer o portão da garagem, sentar em frente ao portão da garagem e ficar encostado em alguma parte do muro ou grade.

A gravação dos vídeos foi realizada de forma contínua, com o celular permanecendo no muro por um longo período para capturar todas as cenas necessárias. Posteriormente, o vídeo produzido foi segmentado em clipes menores com duração de 10 segundos. Esse processo resultou em um total de 154 vídeos contendo comportamentos normais e 154 contendo comportamentos anormais, todos com aproximadamente 10 segundos de duração. A organização das pastas contendo os vídeos segmentados pode ser observada na Figura 1.

Figura 1. Disposição das pastas.



Fonte: Os autores.

Os arquivos de vídeo foram numerados com *flags* indicando seu tipo, por exemplo, os vídeos contendo comportamentos normais foram nomeados com “n1.mp4”, “n2.mp4”, …, “n154.mp4”**,** e os vídeos anormais seguiram a mesma lógica, mas com a letra “a” iniciando os nomes.

3.3. Conversão da Base de Dados de Vídeos para informações numéricas

Para converter os dados de vídeo para um formato numérico, foi utilizada a biblioteca OpenCV para ler cada vídeo das pastas e processar *frame* a *frame* usando para isso a YOLOv8-Pose, cujos resultados

contendo os pontos chaves lidos em cada *frame* foram salvos em arquivos JSON (*JavaScript Object Notation*). O formato JSON é um formato de dados leve e de fácil leitura e escrita, que se assemelha a um dicionário em Python. Esse formato foi utilizado porque sua estrutura facilita significativamente a leitura e escrita dos dados e, por ser muito parecido com um dicionário em Python, torna-se mais fácil de trabalhar no contexto do código, permitindo manipulações eficientes e intuitivas.

O processo foi realizado da seguinte maneira: primeiramente, cada vídeo da pasta "Vídeos/Normal/" foi lido. Por exemplo, ao encontrar o arquivo "n1.mp4", o código executou a função "cv2.VideoCapture()" do OpenCV responsável por ler o vídeo e permitir tratá-lo quadro a quadro. A partir de cada quadro (*frame*) lido, a YOLOv8-Pose retorna um vetor de vetores contendo os pontos chave (*keypoints*) da pessoa presente na cena.

Os resultados foram salvos na pasta "Dataset/Normal/" seguindo uma estrutura específica. Para o vídeo anteriormente citado, foi criada uma pasta chamada "n1" dentro de "Dataset/Normal", e dentro desta pasta foi criada uma outra chamada "frames". Ao ser processado pela YOLOv8-Pose, cada quadro de vídeo gera um vetor de pontos chave. Esses vetores são salvos em arquivos JSON na pasta "frames/", junto com uma *flag* que indica se o quadro pertence a um vídeo de comportamento normal ou anormal. Por exemplo, ao processar o terceiro quadro do vídeo "n1.mp4", a pasta "frames/" contêm os arquivos "frame_1.json", "frame_2.json" e "frame_3.json", cada um contendo os pontos chave daquele quadro específico e a respectiva *flag* de comportamento ('flag': 'normal' ou 'flag': 'abnormal'). Este processo também foi aplicado aos vídeos anormais localizados na pasta "Videos/Abnormal/".

Após essa etapa inicial, os dados foram limpos para remover quadros vazios nos conjuntos de dados. Percorrendo todas as pastas no diretório "Dataset/", quadros que continham dados vazios (representados por [ ]) foram excluídos. Por exemplo, se o arquivo "frame_10.json" na pasta "Dataset/Normal/n23/frames/" estivesse vazio, ele era removido. Após a exclusão dos quadros vazios, todos os quadros restantes foram renomeados para manter uma sequência contínua, evitando lacunas nos nomes dos arquivos.

Esse processo assegurou que os pontos chave de cada quadro de vídeo fossem armazenados em arquivos JSON, juntamente com a *flag* que indica o tipo de comportamento (normal ou anormal), resultando em um conjunto de dados limpo e bem estruturado ideal para o treinamento do modelo.

3.4. Desenvolvimento da Rede Neural (MLP)

Para a detecção de comportamentos anormais, foi modelada uma rede neural do tipo MLP (*Multi-Layer Perceptron*) com comportamento de LSTM (*Long Short-Term Memory*). A arquitetura da MLP foi projetada com várias camadas, começando com uma camada de entrada de 512 unidades, seguida por camadas ocultas com unidades reduzidas em um formato de funil para melhor compressão das informações.

A primeira camada oculta possui 256 unidades, seguida por uma camada de normalização em lote (*Batch Normalization*) para estabilizar o treinamento, e um *dropout* para regularização. A segunda camada oculta possui 128 unidades, também seguida por normalização em lote e *dropout*. A camada de saída é composta por uma única unidade com ativação sigmoide, adequada para classificação binária.

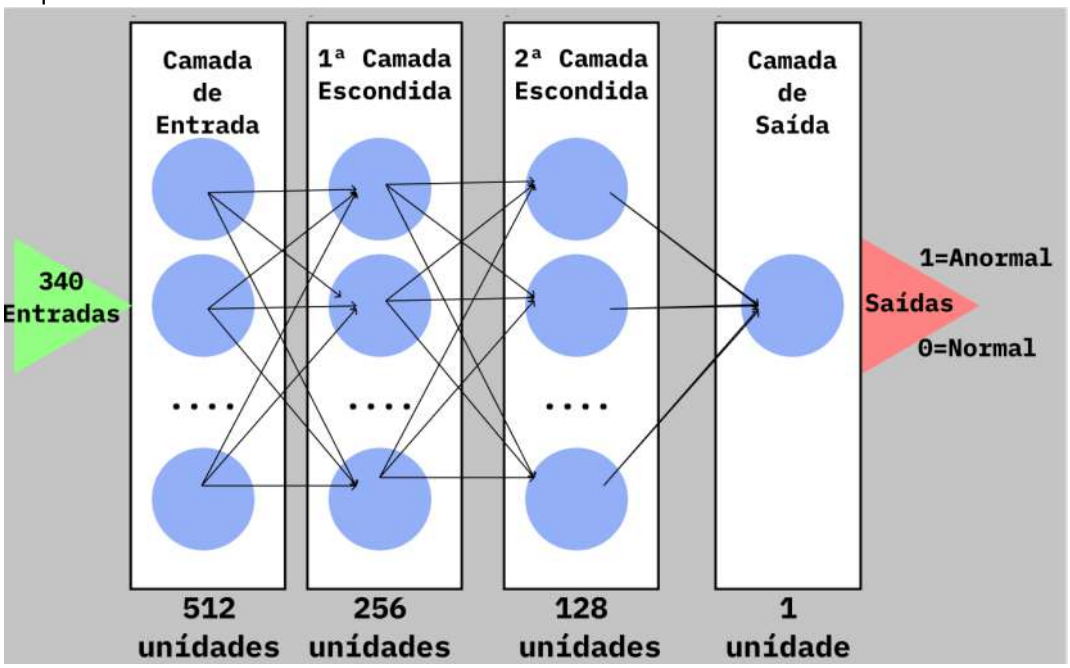
A MLP foi configurada para uma entrada de 340 valores. Este número é resultado do uso de uma janela deslizante (*slide Window*) que lê 10 quadros consecutivos, cada um contendo 17 pontos chaves (*keypoints*), com cada ponto chave representado por dois valores (*x* e *y*), resultando em um vetor de 340 pontos.

Para preparar os dados, todos os quadros de vídeos com comportamentos normais e anormais foram processados utilizando uma janela deslizante de 10 quadros. Os resultados foram salvos em um vetor de vetores, onde cada vetor interno representa uma sequência contínua de 10 quadros e inclui uma *flag* indicando se o comportamento era normal (0) ou anormal (1). Posteriormente, essa estrutura foi embaralhada para garantir uma distribuição aleatória dos dados durante o treinamento.

Finalmente, o modelo foi treinado por 50 épocas com um *batch size* de 64. A função de perda utilizada foi a entropia cruzada binária e o otimizador foi o Adam, com a métrica de precisão.

A arquitetura da rede em formato de funil, que vai comprimindo as informações camada a camada, foi escolhida com base nos trabalhos relacionados, indicando que essa abordagem é eficaz para a finalidade de detecção, representada pela Figura 2. A saída binária da rede é ideal para indicar se o comportamento observado é normal ou anormal.

Figura 2. Arquitetura do modelo MLP.



Fonte: Os autores.

3.4. *Pipeline* Final

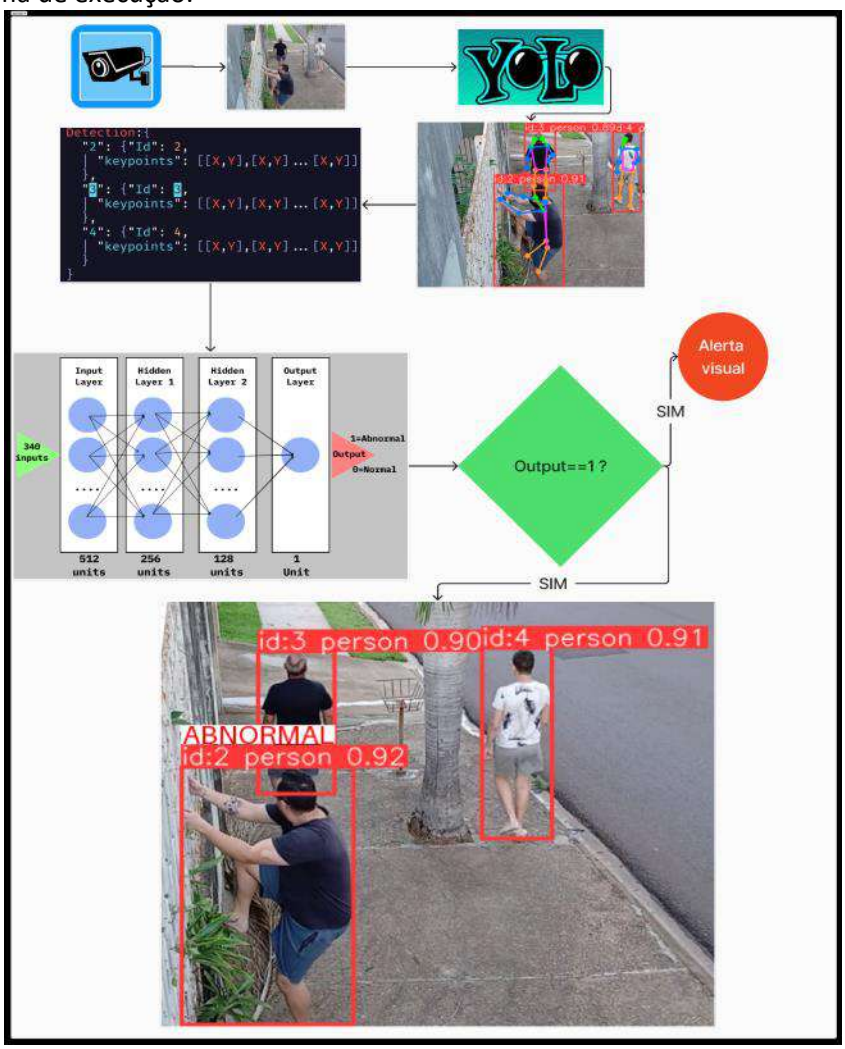
No *pipeline* final, a intenção era usar transmissões ao vivo das câmeras de segurança. No entanto, devido à escassez de recursos financeiros, não foi viável. Como alternativa, foram usadas gravações capturadas com celular, seguindo o mesmo método descrito anteriormente para criar os vídeos de treino. Nos vídeos de teste, três atores foram empregados, cada um representando um comportamento distinto.

O *pipeline* final foi implementado utilizando a função de rastreamento (*track*) do YOLOv8-Pose. Essa função gera um ID para cada pessoa identificada, permitindo o rastreamento simultâneo de múltiplos indivíduos. Esses IDs foram utilizados como chaves em um dicionário Python, onde cada chave possui um vetor que registra as detecções de pontos chave para aquela pessoa, *frame* a *frame*.

À medida que o *pipeline* avança, o vetor associado a cada ID é atualizado com os pontos chave detectados em cada novo *frame*. Quando o vetor atinge 10 valores, isso significa que 10 quadros foram processados para aquela pessoa, capturando seus 17 pontos chave em cada frame. Nesse ponto, uma instância da MLP para leitura de comportamento é inicializada, utilizando-se esse vetor para determinar o comportamento da pessoa associada àquele ID.

Essa abordagem baseada em dicionário permite manter o rastreamento de todas as pessoas que aparecem no vídeo, tratando cada uma individualmente. Após o vetor de comportamento ser processado pela MLP, o primeiro elemento do vetor é removido, aguardando uma nova leitura para que um novo valor seja adicionado ao final do vetor. Isso implementa a janela deslizante (*slide window*), permitindo a leitura contínua e atualização dos comportamentos detectados. A Figura 3 ilustra o funcionamento do *pipeline* final.

Figura 3. Fluxograma de execução.



Fonte: Os autores.

Como pode ser observado, a Figura 3 mostra a sequência de etapas da metodologia desenvolvida, desde a detecção de pessoas no vídeo até a classificação do comportamento (normal ou anormal) pela MLP.

4. RESULTADOS OBTIDOS

Para avaliar o desempenho do sistema desenvolvido, foi realizada uma série de experimentos utilizando gravações de vídeo, cada vídeo com três atores realizando uma série de comportamentos. Cada teste foi nomeado de acordo com a direção dos atores em relação à câmera, a quantidade de atores e a presença de comportamentos normais ou anormais.

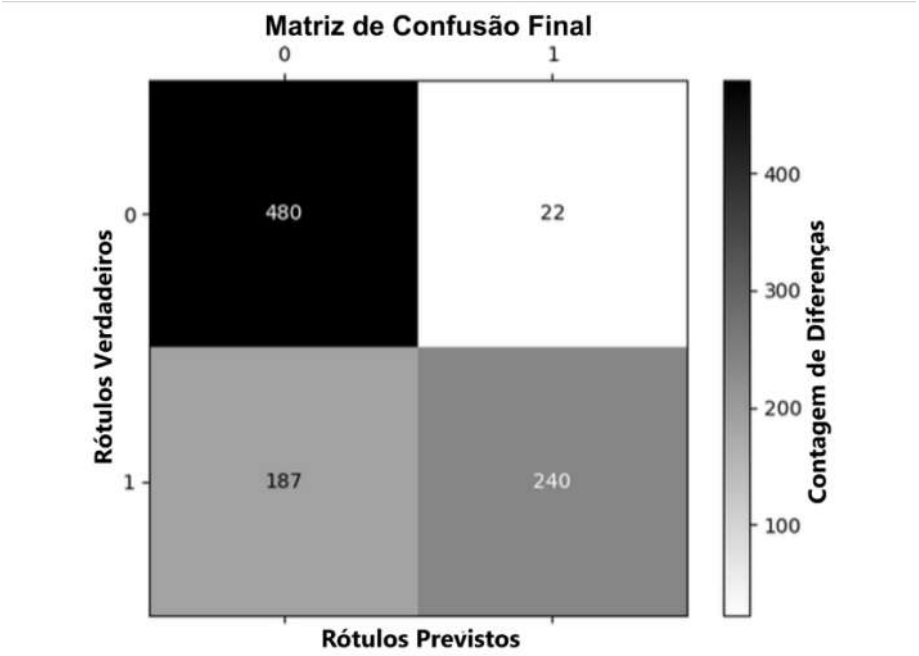
- test1_down_3act_3n.mp4
- test2_up_3act_3n.mp4
- test3_down_3act_1a_fence.mp4
- test4_up_3act_1a_gate.mp4
- test5_down_3act_2a_fence.mp4
- test6_up_3act_2a_gate.mp4
- test7_down_3act_3a_fence.mp4
- test8_up_3act_3a_gate.mp4
- test9_up_3act_1a_fence.mp4
- test10_up_3act_2a_fence.mp4
- test11_up_3act_3a_fence.mp4

A matriz de confusão, mostrada na Figura 4, é uma ferramenta fundamental para avaliar o desempenho do modelo após a execução dos testes. Ela apresenta uma visão detalhada das previsões

feitas pelo modelo em relação aos resultados reais. No eixo Y da matriz estão os rótulos verdadeiros (0 para normal e 1 para anormal), enquanto no eixo X estão os rótulos previstos pelo modelo. Os quadrantes da matriz são identificados como segue: o quadrante superior esquerdo contém os verdadeiros negativos, o superior direito contém os falsos positivos, o inferior esquerdo contém os falsos negativos, e o inferior direito contém os verdadeiros positivos.

Para o exemplo específico da matriz mencionada, os valores são [[480, 22], [187, 240]], o que significa que o modelo classificou corretamente 480 instâncias como normais (verdadeiros negativos), errou ao classificar 22 instâncias normais como anormais (falsos positivos), errou ao classificar 187 instâncias anormais como normais (falsos negativos) e acertou ao classificar 240 instâncias como anormais (verdadeiros positivos). Essa análise detalhada é crucial para entender como está o desempenho do modelo na detecção de comportamentos normais e anormais.

Figura 4. Matriz de confusão dos testes, contendo o quadrante superior esquerdo com os verdadeiros negativos, o superior direito com os falsos positivos, o inferior esquerdo com os falsos negativos, e o inferior direito com os verdadeiros positivos.



;

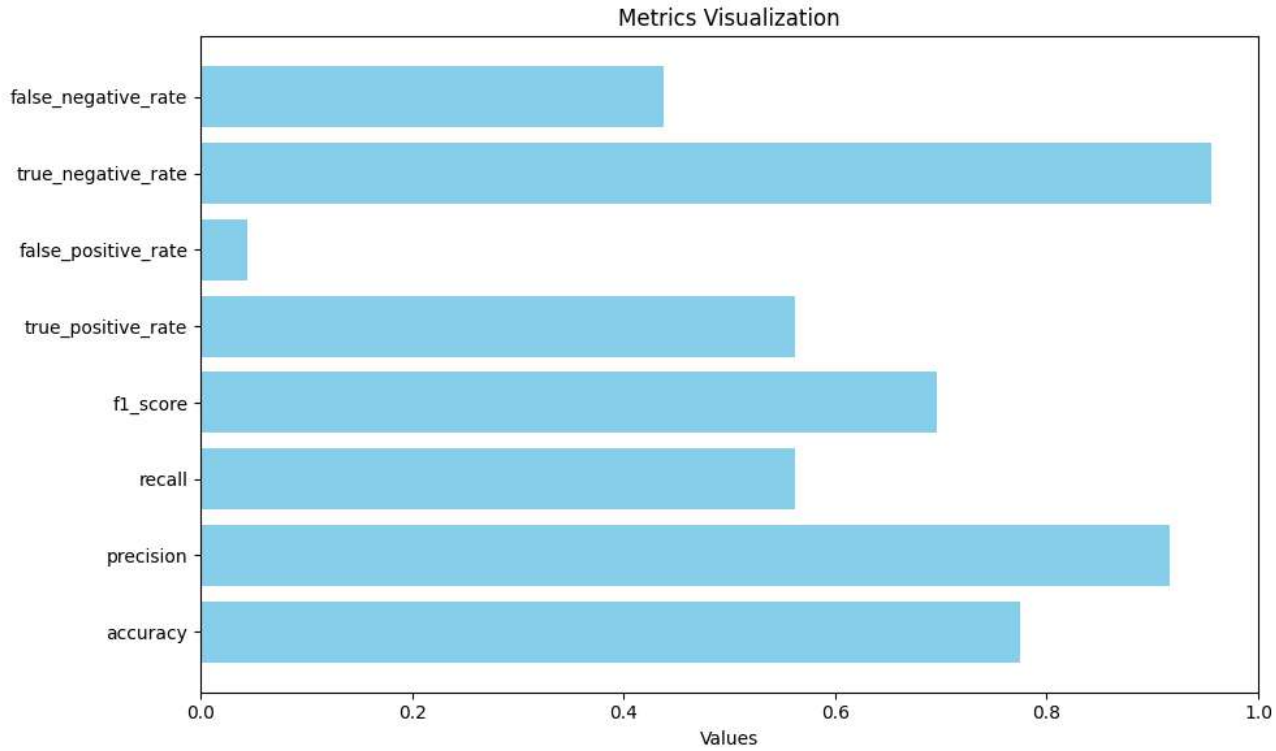
Fonte: Os autores.

A partir da matriz de confusão, obteve-se uma acurácia de 77,5%, indicando que, em média, o modelo classificou corretamente aproximadamente 77,5% das instâncias testadas. A precisão alcançada foi de 91,6%, o que significa que, das predições feitas como anormais, 91,6% estavam corretas, refletindo a eficácia do modelo em evitar falsos positivos. O *recall* foi de 56,2%, demonstrando que o modelo identificou 56,2% dos comportamentos anormais presentes no conjunto de testes, revelando uma limitação em captar todos os comportamentos anormais. O F1-Score, que é a média harmônica entre precisão e o *recall*, foi de 69,7%, apresentando um balanço entre esses dois aspectos e destacando áreas para melhorias.

Além disso, a taxa de verdadeiros positivos (*True Positive Rate*) foi de 56,2%, enquanto a taxa de falsos positivos (*False Positive Rate*) foi de apenas 4,4%, mostrando que o modelo possui uma baixa taxa de falsos alarmes. A taxa de verdadeiros negativos (*True Negative Rate*) foi de 95,6%, indicando que a maioria das instâncias normais foi corretamente identificada. Por fim, a taxa de falsos negativos (*False Negative Rate*) foi de 43,8%, sugerindo que há uma proporção significativa de comportamentos anormais que o modelo não conseguiu detectar.

Essas métricas fornecem uma visão detalhada e abrangente do desempenho do modelo, permitindo identificar tanto seus pontos fortes quanto suas limitações, tudo isso pode ser observado no gráfico da Figura 5.

Figura 5. Gráfico das métricas retiradas da matriz de confusão.



Fonte: Os autores.

Nos testes onde os atores estão descendo a rua, observou-se que quando os corpos não estão 100% expostos ao vídeo, a detecção frequentemente indica comportamento anormal como ilustrado na Figura 6. Isso se deve à fase de treinamento, onde comportamentos anormais realizados no portão da garagem resultaram em partes do corpo dos atores oclusas pela parede. Esses pontos chave, na leitura do YOLOv8-Pose, ficam com valores $x = 0.0$, $y = 0.0$. Consequentemente, o modelo da rede MLP aprendeu que quando um corpo tem partes dos pontos chave como 0.0 (oclusão), isso indica um comportamento anormal como ilustrado na Figura 7.

Figura 6. Corpos parcialmente oclusos.

Fonte: Os autores.

Figura 7. Oclusão de pontos chave pelo muro.

Fonte: Os autores.

Observou-se que o modelo realiza uma leitura mais assertiva dos atores, quando estes estão realizando o comportamento anormal na parte superior da residência, na grade, onde estão mais próximos da visão da câmera, essa proximidade com a câmera diminui a chance de oclusão dos atores permitindo assim uma melhor leitura, como é verificado na Figura 8.

Figura 8. Melhor detecção na parte superior da residência.

Fonte: Os autores.

Os resultados mostraram que o sistema tem uma acurácia satisfatória em detectar comportamentos anormais em situações onde não há oclusão significativa dos pontos chave dos atores. No entanto, a presença de oclusões, especialmente nas áreas do portão da garagem, impacta negativamente a precisão do modelo ao dificultar a leitura dos pontos chave dos atores. A análise mais detalhada com a moda (o valor mais frequente) dos comportamentos a cada 10 janelas deslizantes melhorou a estabilidade da detecção, mas desafios permanecem em cenários com oclusão.

Esses resultados indicam que, apesar de o sistema não apresentar uma acurácia extremamente alta, ele cumpre seu papel como sistema de segurança. Ele detecta bem os verdadeiros negativos e, em um contexto de segurança, é mais aceitável que o sistema alerte sobre uma invasão falsa do que não notifique sobre uma invasão verdadeira. Portanto, mesmo com a detecção de alguns falsos positivos, o sistema oferece uma camada de proteção valiosa ao alertar sobre potenciais comportamentos anormais.

5. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

A solução computacional desenvolvida foca na detecção de pessoas, leitura de comportamentos e emissão de alertas para comportamentos anormais. Esta solução demonstrou ser capaz de executar essas tarefas em tempo real quando conectado a um sistema de videomonitoramento. Por meio da utilização do YOLOv8-Pose para a detecção de pontos chave e de uma rede MLP com comportamento de LSTM para a classificação de comportamentos, foi possível alcançar um nível considerável de precisão na identificação de atividades suspeitas.

Apesar dos resultados promissores, ainda há espaço para melhorias significativas. Primeiramente, a acurácia do sistema pode ser aprimorada. Uma abordagem futura inclui a adição de uma nova camada de segurança ao modelo, com uma terceira opção de saída que, quando a solução desenvolvida não consegue identificar o comportamento observado, emite um alerta. Esta estratégia pode gerar alguns falsos positivos, mas, considerando que se trata de um sistema de segurança residencial, é preferível alertas falsos a evitar comportamentos realmente suspeitos.

Outro ponto crucial para o aprimoramento da solução é a utilização de múltiplas câmeras. Com mais câmeras, seria possível obter um ângulo de visão mais abrangente dos eventos monitorados, reduzindo significativamente o problema de oclusão, que foi identificado como um dos principais desafios. A visão a partir de diferentes pontos permitiria ao sistema identificar e acompanhar pessoas de maneira mais eficaz, melhorando a precisão na detecção de comportamentos anormais.

Em suma, a solução computacional desenvolvida oferece uma ferramenta eficaz para o monitoramento inteligente de residências, mas futuras melhorias na precisão, na capacidade de lidar com oclusões e na integração com múltiplas fontes de vídeo são passos importantes para aumentar sua confiabilidade e eficácia.

REFERÊNCIAS

ANALA, M. R.; MAKKER, M.; ASHOK, A. **Anomaly Detection in Surveillance Videos**. *In*: INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE COMPUTING, DATA AND ANALYTICS WORKSHOP (HIPCW), 26., 2019, Hyderabad, India. Anais [...]. Hyderabad, India 2019. DOI: https://doi.org/10.1109/HiPCW.2019.00031

HEIDARI, N.; IOSIFIDIS, A. Progressive Spatio-Temporal Graph Convolutional Network for Skeleton-based Human Action Recognition. *In:* IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP), 2021. Toronto, CA. **Processing** [...]. Toronto, CA: IEEE, 2021. DOI: https://doi.org/10.1109/ICASSP39728.2021.9413860

JI, H.; ZENG, X.; LI, H.; DING, W. Human Abnormal Behavior Detection Method based on T-TINY-YOLO. *In:* INTERNATIONAL CONFERENCE ON MULTIMEDIA AND IMAGE PROCESSING (ICMIP). 5., 2020. New York. **Proceedings** [...]. New York: ACM, 2020. DOI: https://doi.org/10.1145/3381271.3381273.

KO, K-E.; SIM, K-B. Deep convolutional framework for abnormal behavior detection in a smart surveillance system. **Engineering Applications of Artificial Intelligence**, v. 67 p. 226-234, 2018. DOI: https://doi.org/10.1016/j.engappai.2017.10.001

NASARUDDIN, N.; MUCHTAR, K.; AFDHAL, A.; DWIYANTORO, A. P. J. Deep Anomaly Detection through Visual Attention in Surveillance Videos. **Journal of Big Data**, Switzerland, v. 7, n. 87, 2020. DOI: https://doi.org/10.1186/s40537-020-00365-y

SHINDE, S.; ASHWIN, K.; VIKRAM, G. YOLO based Human Action Recognition and Localization. **Procedia Computer Science**, Elsevier, v. 133, p. 831-838, 2018. DOI: https://doi.org/10.1016/j.procs.2018.07.112.

SULTANI, W.; CHEN, C.; SHAH, M. Real-world Anomaly Detection in Surveillance Videos. *In*: IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2018. Salt Lake City, UT. **Anais** [...].Salt Lake City, UT: IEEE, 2018, p. 6479-6488. DOI: https://doi.org/10.1109/CVPR.2018.00678.

WU, C.; CHENG, Z. A novel detection framework for detecting abnormal human behavior**. Mathematical Problems in Engineering**, v. 2020, p. 1-9, 2020. DOI: https://doi.org/10.1155/2020/6625695.

ZHANG, X.; YANG, S.; ZHANG, J.; ZHANG, W. Video anomaly detection and localization using motion-field shape description and homogeneity testing. **Pattern Recognition,** Elsevier, v. 105, p. 1-13, 2020. DOI: https://doi.org/10.1016/j.patcog.2020.107394.