UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

JEFERSON ROSA DE SOUZA

RECONHECIMENTO E CLASSIFICAÇÃO DE CENAS DE RISCO EM VÍDEOS DE CÂMERAS DE SEGURANÇA

PATO BRANCO

2024

JEFERSON ROSA DE SOUZA

RECONHECIMENTO E CLASSIFICAÇÃO DE CENAS DE RISCO EM VÍDEOS DE CÂMERAS DE SEGURANÇA

Protocol-Driven Recognition and Classification of Risk Scenes in Security Cameras

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Dr. Ives Renê Venturini Pola

Coorientador: Luiz Henrique Birck Vicari

PATO BRANCO

2024

JEFERSON ROSA DE SOUZA

RECONHECIMENTO E CLASSIFICAÇÃO DE CENAS DE RISCO EM VÍDEOS DE CÂMERAS DE SEGURANÇA

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 21/junho/2024

Dr. Ives Renê Venturini Pola
Universidade Tecnológica Federal do Paraná

Bac. Luiz Henrique Birck Vicari
Universidade Tecnológica Federal do Paraná

Drª. Viviane Dal Molin
Universidade Tecnológica Federal do Paraná

Drª. Soelaine Rodrigues Ascari
Universidade Tecnológica Federal do Paraná

PATO BRANCO
2024

AGRADECIMENTOS

Agradeço primeiramente à Universidade Tecnológica Federal do Paraná pela oportunidade de crescimento acadêmico e pessoal proporcionada durante todo o curso.

Ao meu orientador Prof. Dr. Ives Renê Venturini Pola, pela orientação dedicada, paciência e valiosos *insights* que foram fundamentais para o desenvolvimento deste trabalho.

Ao meu coorientador Luiz Henrique Birck Vicari, pelo apoio constante, pelas contribuições essenciais e pela disponibilidade em auxiliar sempre que necessário. Acima de tudo, pela amizade adquirida ao longo do tempo.

À banca examinadora, composta por Profª. Drª. Viviane Dal Molin e Profª. Drª. Soelaine Rodrigues Ascari, pela disposição em avaliar este trabalho e pelas valiosas sugestões e críticas que contribuíram significativamente para a sua qualidade.

Aos meus pais, Jair e Bernadete, pelo amor incondicional, pelo apoio inestimável e por sempre acreditarem em mim, sustentando não apenas financeiramente, mas também emocionalmente, ao longo desta jornada acadêmica.

À minha esposa Patrícia, pelo companheirismo constante, pelo incentivo incansável e por estar ao meu lado em todos os momentos, sempre me motivando a seguir em frente e aprofundar minhas pesquisas.

Agradeço também à minha filha Melissa, por ser minha fonte de ânimo e motivação diária. É por você que busco sempre melhorar e progredir na minha carreira, com o objetivo de construir um futuro promissor e digno para nós.

Aos meus amigos que ganhei na faculdade, pelos momentos compartilhados, pelo apoio mútuo e pelas experiências enriquecedoras que vivemos juntos ao longo desses anos.

Cada um de vocês desempenhou um papel fundamental na realização deste trabalho e na minha jornada acadêmica como um todo. Sou profundamente grato por todo apoio, compreensão e encorajamento que recebi.

RESUMO

Este trabalho apresenta o projeto e desenvolvimento de um sistema para reconhecimento e classificação de cenas de risco em vídeos de câmeras de segurança, utilizando técnicas de visão computacional e aprendizado profundo. O objetivo principal foi permitir ao sistema detectar objetos e situações de risco, como armas de fogo e facas, em tempo real, e possibilitar alertas automáticos. A metodologia empregou o uso do modelo *You Only Look Once* (YOLO), que permite uma identificação eficiente de objetos em imagens, essencial para cenários de segurança dinâmicos. Os resultados obtidos demonstraram a precisão na detecção e classificação de riscos juntamente com testes práticos em ambientes reais. A pesquisa também mostrou a importância da escolha de bases de dados adequadas para o treinamento dos modelos e a necessidade de aperfeiçoamento contínuo dos algoritmos para lidar com a complexidade das cenas.

Palavras-chave: reconhecimento de cenas de risco; visão computacional; aprendizado profundo; yolo; segurança.

ABSTRACT

This work presents a system for recognizing and classifying risk scenes in security camera videos, using computer vision and deep learning techniques. The main objective is to develop a system capable of detecting objects and risk situations, such as firearms and knives, in real-time, and generating automatic alerts. The methodology employs the YOLO (You Only Look Once) model, which allows fast and accurate object identification in images, essential for dynamic security scenarios. The obtained results demonstrate high precision in risk detection and classification, with proven effectiveness in practical tests. The research also emphasizes the importance of choosing suitable databases for model training and the need for continuous algorithm improvement to handle scene complexity. It is concluded that the developed system offers an advanced tool for security, with potential for future enhancements.

Keywords: risk scene recognition; computer vision; deep learning; yolo; security.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

Abreviaturas

OpenCV	*Open Source Computer Vision Library*

Siglas

API	*Application Programming Interface*
CCTV	*Closed Circuit Television*
COCO	*Common Objects in Context*
DVR	*Digital Video Recorder*
FPS	*frames* por segundo
HTTP	*HyperText Transfer Protocol*
IMFDB	*Movie Firearm Database*
IOU	*Intersection Over Union*
IP	*Internet Protocol*
NMS	*Non-Maximum Suppression*
PTZ	*Pan, Tilt e Zoom*
ReLU	*Rectified Linear Unit*
RNA	Rede Neural Artificial
RNC	Rede Neural Convolucional
RTSP	*Real Time Streaming Protocol*
tanh	Tangente Hiperbólica
VGG Net	*Visual Geometry Group Network*
YOLO	*You Only Look Once*
ZF Net	*Zeiler and Fergus Network*

SUMÁRIO

1 INTRODUÇÃO

Prover segurança eficaz em propriedades por meio do monitoramento têm se tornado uma prioridade crescente em um mundo no qual prevenção de incidentes é crucial. Câmeras de segurança desempenham um papel essencial nesse contexto, oferecendo vigilância contínua e fornecendo uma visão detalhada das atividades ocorrendo nas proximidades (Lejmi; Khalifa; Mahjoub, 2019).

A tarefa de captura e análise contínua e ininterrupta das imagens por meio de câmeras é inviável de ser executada manualmente e ainda pode sobrecarregar as equipes de segurança. A necessidade de análise instantânea, a identificação precisa de indivíduos e objetos em diferentes cenários, bem como a demanda por intervenções ágeis são algumas das complexidades presentes nesse contexto (BRICALLI, 2015).

Tais desafios se tornam ainda mais evidentes em um cenário social em que a incidência de eventos de risco tem se tornado cada vez mais comum. Entre 2002 e 2019, foram registrados sete atentados em escolas, enquanto nos quatro anos seguintes esse número mais do que dobrou, atingindo 17 incidentes. Assim, é evidente que o aumento da violência motivados por inúmeros fatores, com destaque para os culturais, tem aumentado significativamente nos últimos anos (Bond, 2023).

A aplicação de tecnologias automatizadas, pelo reconhecimento de padrões em imagens em análise de vídeos, tem revolucionado a forma como se aborda a segurança. No entanto, a natureza dinâmica e complexa das cenas capturadas por câmeras de segurança torna a tarefa de detecção de riscos um desafio (Kang; Kim, 2023). Neste contexto, surge a necessidade de adotar uma abordagem para o reconhecimento de cenas de risco em sistemas de segurança, visando a implementação de ferramentas de visão computacional.

Dentre as ferramentas que têm se destacado, YOLO, representa um avanço significativo no reconhecimento de objetos e na identificação precisa de pessoas em tempo real. A abordagem utilizada é a detecção de objetos na qual a imagem é dividida em grades e, em uma única iteração, uma rede neural é capaz de identificar e delinear objetos de interesse. Essa metodologia permite uma resposta rápida e eficiente, sendo essencial para cenários dinâmicos de segurança (Redmon *et al.*, 2016; Kang; Kim, 2023).

Este trabalho aborda o uso de técnicas computacionais para identificar situações de risco em imagens provindas de sistemas de segurança por meio de um sistema de reconhecimento automatizado.

1.1 Objetivos

Nesta seção são descritos os objetivos gerais e específicos alinhados ao plano de trabalho que visa a automação de um sistema de segurança específico por vídeo.

1.1.1 Objetivo geral

O objetivo geral deste trabalho foi projetar e desenvolver um sistema de detecção de situações de potencial risco a partir de imagens de câmeras de segurança, sendo capaz de registrar e categorizar as ocorrências com base em regras pré-definidas.

1.1.2 Objetivos específicos

- Selecionar uma ou mais bases de dados públicas com imagens de objetos de risco como armas de fogo e facas;
- Aplicar a ferramenta YOLO para identificar objetos de risco;
- Desenvolver um sistema de classificação que avalie a natureza e a gravidade das detecções realizadas, gerando *logs* de alerta;
- Realizar experimentos e avaliações para verificar a eficácia e a precisão do sistema proposto.
- Integrar o reconhecimento com *Real Time Streaming Protocol* (RTSP) por meio do *Open Source Computer Vision Library* (OpenCV), permitindo a observação de equipamentos CCTV em tempo real.

1.2 Organização do trabalho

No Capítulo 2, são apresentados os fundamentos teóricos necessários para a compreensão e realização deste trabalho. No Capítulo 3 são apresentados trabalhos que contribuem para o estudo de forma referencial e comparativa. No Capítulo 4 são abordados os passos seguidos para atingir o objetivo, bem como os materiais necessários e cronograma proposto. O Capítulo 5 apresenta os resultados obtidos no trabalho realizado bem como discussões. Por fim, o Capítulo 6 apresenta a perspectiva sobre o que foi alcançado conforme o propósito do trabalho e aponta trabalhos futuros para este estudo de caso.

2 REFERENCIAL TEÓRICO

Neste capítulo, são apresentados conceitos técnicos relacionados a câmeras de vigilância bem como técnicas de processamento de imagem e aplicações de processamento de vídeo em tempo real. Em seguida uma breve introdução sobre redes neurais artificiais e redes neurais convolucionais. Por fim, a arquitetura YOLO e algumas ferramentas que serão fortemente utilizadas neste trabalho.

2.1 Conceitualização técnica em câmeras de vigilância

De acordo com Damjanovski (2013), dentro do contexto de um sistema de CCTV, as câmeras desempenham um papel vital como componentes sensores do processo de vigilância. Uma ampla variedade de modelos de câmeras de segurança está disponível, incluindo minicâmeras, câmeras profissionais, câmeras equipadas com tecnologia infravermelha, câmeras *Pan, Tilt e Zoom* (PTZ) e câmeras do tipo *Internet Protocol* (IP). Todos esses modos possibilitam acesso à internet e podem ser acessados remotamente para visualização das câmeras, permitindo assim um acompanhamento conveniente das imagens (BAESSO, 2016).

As câmeras PTZ são dispositivos que podem ser controlados remotamente para ajustar a cena conforme a necessidade momentânea. Elas oferecem a capacidade de movimento vertical e horizontal, além de recursos de zoom para aproximar a cena e revelar detalhes com maior clareza. É fundamental notar que, além dos componentes eletrônicos, as câmeras PTZ incorporam um sistema mecânico complexo para garantir a precisão na execução dos comandos de movimento (Thirthala; Sinha; Pollefeys, 2005).

Por outro lado, as câmeras IP são dispositivos que transmitem áudio e vídeo para um computador ou diretamente para a Internet. Utilizando softwares específicos, é possível visualizar várias imagens ao vivo simultaneamente na tela ou receber notificações por aplicativos que detectem atividades fora do comum conforme destacado por (Pinheiro, 2006).

Além disso, as câmeras de vídeo podem ser classificadas de acordo com a imagem que produzem, podendo ser coloridas ou em preto e branco. Câmeras coloridas proporcionam uma riqueza de detalhes na cena, exibindo as cores do ambiente de forma adequada. Em contraste, câmeras em preto e branco oferecem imagens com maior nitidez, especialmente em ambientes com pouca luminosidade.

Os sistemas de CCTV, ou sistemas de câmeras de segurança, podem ser implementados de várias maneiras. Isso inclui câmeras individuais com armazenamento próprio, câmeras individuais que se conectam a um servidor na nuvem para armazenamento, ou ainda, um sistema com um DVR. Empresas de segurança geralmente utilizam o DVR, que capturam as imagens de uma câmera, registra as informações e mantém uma conexão com internet, conforme ilustrado na Figura 1.

Figura 1 – Circuito de um Sistema DVR Comum

Fonte: Autoria própria (2024).

Portanto, tanto as câmeras PTZ quanto as câmeras IP oferecem benefícios distintos para sistemas de CCTV, proporcionando versatilidade e aprimorando a capacidade de vigilância e monitoramento. A escolha entre câmeras coloridas e em preto e branco permite adaptar o sistema conforme as necessidades específicas de cada ambiente, ampliando sua eficácia.

2.2 Aquisição e processamento de imagens

O processamento de imagens envolve uma série de transformações contínuas, visando atingir o resultado desejado. Essas transformações são realizadas por meio de técnicas que filtram objetos e eliminam dados irrelevantes, conhecidos como ruídos. Este procedimento tem dois principais objetivos, conforme explica o autor: aprimorar a informação visual para a interpretação humana e processar dados de imagens para a automação por máquinas.(Baskar *et al.*, 2023)

Na área da visão computacional, cujo objetivo é realizar tarefas semelhantes às do olho humano (Pedrine; Schwartz, 2008), a aplicação de técnicas de inteligência computacional não apenas facilitam, mas também contribuem significativamente para a melhoria e otimização das tarefas, proporcionando resultados mais precisos e eficientes (Szeliski, 2021). Essa abordagem visa extrair informações significativas para resolver problemas específicos, proporcionando análises que podem ser aplicadas à resolução de uma problemática em questão. Para atingir tais objetivos, é essencial considerar as ferramentas e métodos disponíveis para a aquisição eficiente e precisa de imagens. Nesse contexto, o OpenCV surge como uma escolha natural, oferecendo uma ampla gama de funcionalidades para processamento de imagem e visão computacional.

O OpenCV é uma biblioteca de código aberto que se tornou uma referência na comunidade de visão computacional devido à sua versatilidade e eficácia. Sua aplicação neste trabalho está fundamentada na necessidade de manipular e processar imagens provenientes de câmeras de segurança. A biblioteca oferece ferramentas para a leitura, manipulação e análise de vídeos, sendo essencial para a implementação dos algoritmos de reconhecimento propostos. De acordo com Zelinsky (2009), o OpenCV proporciona uma infraestrutura eficaz para desen-

volver aplicativos de visão computacional em tempo real, tornando-se uma escolha sólida para projetos que demandam processamento de imagens de forma rápida.

Além disso, é relevante considerar as estatísticas de uso do OpenCV para destacar a sua popularidade e aceitação na comunidade científica e industrial. Conforme mencionado por Baggio *et al.* (2016), o OpenCV é amplamente adotado em projetos de visão computacional em diversos setores, evidenciando sua confiabilidade e eficiência. As estatísticas de uso são indicativas da confiança depositada nesta biblioteca e fortalecem a escolha do OpenCV como uma ferramenta fundamental para a aquisição e processamento de imagens no contexto deste trabalho.

Uma das técnicas essenciais disponíveis no OpenCV são as transformações geométricas, que incluem escala, rotação e translação. Essas transformações ajudam a normalizar as imagens, assegurando que os objetos de interesse sejam identificados consistentemente, independentemente de sua posição ou orientação na cena (Szeliski, 2021).

Além disso, o ajuste de contraste e brilho também é uma etapa crítica no pré-processamento de imagens. O OpenCV oferece funcionalidades para manipular esses aspectos, permitindo adaptar as imagens às variáveis condições de iluminação típicas em vídeos de vigilância. Essa capacidade de ajuste é crucial para garantir a detecção precisa de objetos de risco em diferentes ambientes de iluminação (Baskar *et al*., 2023).

2.2.1 Métodos de Aquisição de Imagem em Vídeo

No que diz respeito aos métodos de aquisição de imagem em vídeo, é necessário considerar as diferentes abordagens disponíveis para capturar informações visuais de câmeras de segurança. A aquisição de imagens em vídeo desempenha um papel central na detecção de cenas de risco, pois fornece sequências contínuas de *frames* para análise. Segundo Baskar *et al.* (2023), a captura de vídeo pode ser realizada por meio de câmeras analógicas ou digitais, cada uma apresentando características específicas.

Câmeras analógicas, por exemplo, convertem a luz em sinais elétricos analógicos, enquanto as câmeras digitais convertem a luz em sinais digitais. A escolha entre esses métodos de aquisição pode influenciar a qualidade das imagens e, consequentemente, a precisão do reconhecimento de cenas de risco. Ao considerar a importância da aquisição de imagens em vídeo para o sistema proposto, é crucial avaliar a eficácia de cada método em atender aos requisitos de vigilância e segurança.

Para integrar a manipulação de imagens e os métodos de aquisição de imagem em vídeo é indicado o uso de OpenCV para conexão a câmeras de DVR ferramenta que desempenha um papel fundamental na integração do reconhecimento de cenas de risco em câmeras de segurança em tempo real (ITSEEZ, 2014). O RTSP é um protocolo de comunicação utilizado para a transmissão contínua de dados de vídeo e áudio, possibilitando a observação em tempo real de equipamentos de CCTV. Nesse contexto, a utilização do RTSP em conjunto com o

OpenCV, conforme sua documentação, oferece uma solução tanto para a visualização dinâmica de cenas quanto para a obtenção de imagens, por meio de RTSP, que serviram ou servirão de informações para o sistema.

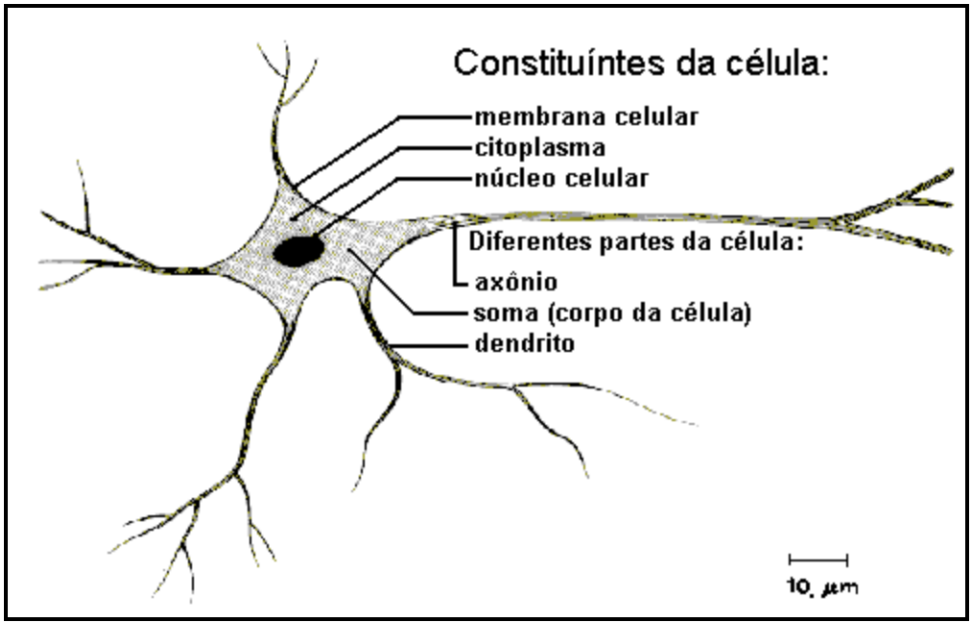
Ao adotar o RTSP, o sistema é capaz de estabelecer uma conexão eficiente com câmeras de segurança e receber fluxos de vídeo em tempo real. Essa abordagem proporciona uma resposta imediata a eventos de risco, permitindo a análise contínua e a tomada de decisões em tempo hábil. Segundo (Firmansyah; Suharto; Prasetyo, 2022), o RTSP oferece uma arquitetura robusta para a transmissão de dados multimídia, sendo especialmente adequado para aplicativos que demandam observação em tempo real, como é o caso de sistemas de segurança.

2.3 Redes neurais artificiais

Uma Rede Neural Artificial (RNA) é inspirada pelo funcionamento do cérebro humano. Essas redes empregam uma estrutura de interconexões entre unidades de processamento, conhecidas como neurônios, para extrair informações a partir de conjuntos de dados (Haykin, 2008).

Uma RNA emula os neurônios biológicos, que compreendem quatro componentes fundamentais: dendritos, corpo celular (soma), axônios e sinapses. Os dendritos são responsáveis por receber informações de outros neurônios e encaminhá-las ao corpo celular. Este último processa essas informações, gerando novas saídas que, por sua vez, são transmitidas pelos axônios a outros neurônios por meio de conexões sinápticas, conforme Figura 2 (Carvalho, 2020).

Figura 2 – Estrutura celular de um neurônio biológico

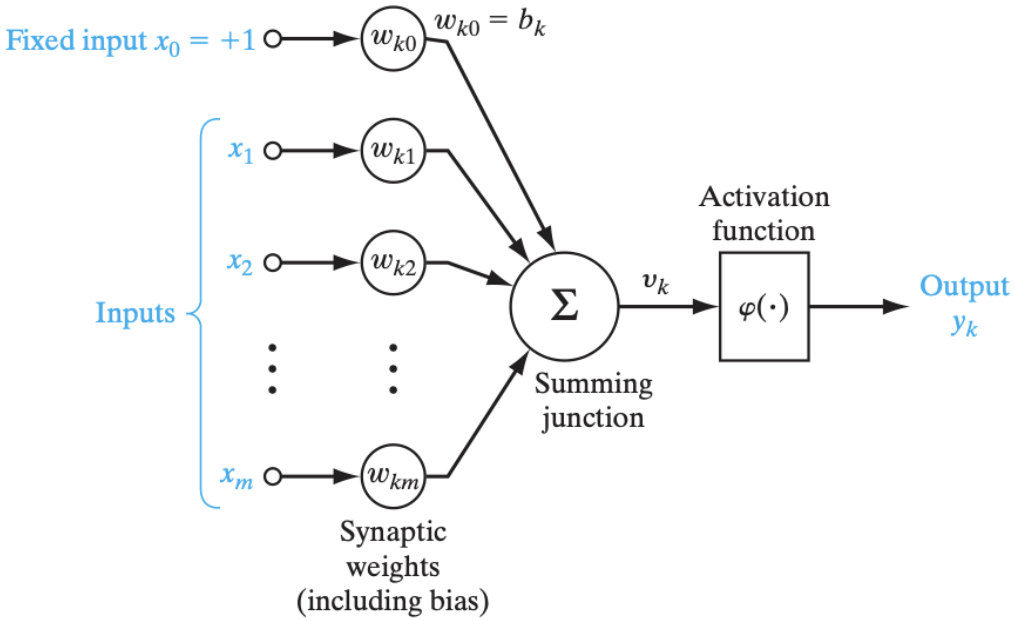


Fonte: (Carvalho, 2020).

> Os neurônios se comunicam através de sinapses. Sinapse é a região onde dois neurônios entram em contato e através da qual os impulsos nervosos são transmitidos entre eles. Os impulsos recebidos por um neurônio A, em um determinado momento, são processados, e atingindo um dado limiar de ação, o neurônio A dispara, produzindo uma substância neurotransmissora que flui do corpo celular para o axônio, que pode estar conectado a um dendrito de um outro neurônio B. O neurotransmissor pode diminuir ou aumentar a polaridade da membrana pós-sináptica, inibindo ou excitando a geração dos pulsos no neurônio B. Este processo depende de vários fatores, como a geometria da sinapse e o tipo de neurotransmissor. Em média, cada neurônio forma entre mil e dez mil sinapses. O cérebro humano possui cerca de 10^{11} neurônios, e o número de sinapses é de mais de 10^{14}, possibilitando a formação de redes muito complexas (Carvalho, 2020).

Analogamente, um neurônio artificial, também conhecido como perceptron, é representado por Haykin (2008), como uma sequência de processos, como retratado na Figura 3. Esse processo tem início no recebimento de informações por meio de sinais de entrada, os quais são representados pela variável x, combinados aos seus respectivos pesos sinápticos, representados pela variável w. Em seguida, o sinal resultante da combinação entre os sinais de entrada

Figura 3 – Estrutura neurônio artificial



Fonte: (Haykin, 2008).

e seus pesos sinápticos passa por uma função somatória. Essa função soma os produtos dos sinais de entrada pelos seus pesos sinápticos, gerando assim uma nova informação.

A última etapa, a função de ativação, determina a saída do neurônio, transformando a informação resultante da função somatória em um conjunto limitado de valores de ativação.

Isso é inspirado na natureza não-linear das respostas dos neurônios biológicos. Diversas funções de ativação são utilizadas em redes neurais, como a *sigmoid*, Tangente Hiperbólica (tanh) e *Rectified Linear Unit* (ReLU), cada uma trazendo propriedades específicas ao processo de aprendizado. Essa abordagem permite que as redes neurais representem relações complexas e não-lineares em dados, proporcionando flexibilidade na modelagem de problemas do mundo real (Haykin, 2008).

As RNAs são compostas por uma arquitetura de rede e seu algoritmo de treinamento. O algoritmo de treinamento de uma RNA baseia-se em ajustar os parâmetros adaptáveis da rede, conhecidos como pesos, para que a saída da rede se aproxime o máximo possível dos rótulos ou valores desejados associados aos exemplos de entrada. Durante o treinamento, a rede recebe exemplos de dados de entrada e produz saídas correspondentes (Rauber, 2005).

2.3.1 Redes neurais convolucionais

Uma Rede Neural Convolucional (RNC) é um tipo especializado de arquitetura de RNA, desenvolvida especialmente para lidar com dados que possuem uma topologia de grade, como imagens. Ela é particularmente eficaz em tarefas relacionadas à visão computacional, destacando-se em reconhecimento de padrões em imagens e vídeos (He *et al.*, 2016)

A estrutura básica de uma RNC consiste em camadas convolucionais, de *pooling* e completamente conectadas. A camada convolucional é a peça central, onde ocorre a operação de convolução, essa operação envolve um filtro (também chamado de *kernel*) que percorre a imagem de entrada, multiplicando seus valores pelos valores correspondentes na região da imagem. O resultado é uma soma ponderada desses produtos, criando um mapa de características. Esse processo é repetido em diferentes regiões da imagem, gerando mapas de características que destacam padrões específicos (Boureau *et al.*, 2010).

O princípio da convolução é fundamental em uma RNC, visto que permite a detecção de características locais, como bordas, texturas e padrões simples, em diferentes partes da imagem. Isso é crucial para o aprendizado de hierarquias de características complexas à medida que as camadas mais profundas da rede processam características mais abstratas e globalmente significativas (Goodfellow; Bengio; Courville, 2016).

A causa fundamental do princípio de convolução reside na capacidade de capturar a dependência espacial e a invariância de translacional em dados visuais. A convolução permite que a rede aprenda padrões independentemente de sua posição exata na imagem, tornando a arquitetura robusta a variações e deslocamentos. Isso é crucial para reconhecimento de padrões em imagens, onde a posição específica dos elementos pode variar (Baskar *et al.*, 2023).

Em termos de relação com redes neurais artificiais, as RNC são uma evolução que busca superar as limitações das redes neurais tradicionais na manipulação de dados espaciais, como imagens. A aplicação do princípio de convolução permite que as RNC aprendam e generalizem padrões visuais de maneira eficiente, tornando-as especialmente eficazes em

tarefas como classificação de imagens, detecção de objetos e segmentação semântica. Essa abordagem tornou-se fundamental em muitas aplicações de aprendizado profundo em visão computacional (Baskar *et al.*, 2023).

2.3.2 Avaliação de modelo de classificação

Para avaliar a precisão de um modelo treinado é possível usar a matriz de confusão. Essa forma de avaliação consiste em uma tabela que resume o desempenho de um algoritmo de classificação, cada linha da matriz representa as instâncias reais de uma classe, enquanto cada coluna representa as instâncias previstas de uma classe. Isso permite ver não apenas os acertos do modelo (verdadeiros positivos e verdadeiros negativos), mas também os erros (falsos positivos e falsos negativos) (Géron, 2019).

De acordo com Géron (2019) uma matriz de confusão com duas classes (positivo e negativo) tem a seguinte forma:

	Previsto negativo	Previsto positivo
Atual negativo	TN	FP
Atual positivo	FN	TP

Na matriz, TN (*True Negatives*) representa o total de negativos verdadeiros, FP (*False Positives*) de falsos positivos, FN (*False Negatives*) de falsos negativos e TP (*True Positives*) de positivos verdadeiros. Com esses valores, é possível calcular várias métricas de avaliação, como a precisão e *recall*.

A precisão (*precision*) é definida como a proporção de previsões positivas corretas em relação ao total de previsões positivas feitas pelo modelo. Matematicamente, é expressa como:

$$\text{Precisão} = \frac{TP}{TP + FP} \tag{1}$$

Onde TP é o número de verdadeiros positivos e FP é o número de falsos positivos. A precisão é uma medida útil quando o custo de falsos positivos é alto, pois indica a exatidão das previsões positivas do modelo. Por exemplo, em um sistema de detecção de fraudes, uma alta precisão significa que poucas transações legítimas são incorretamente classificadas como fraudulentas.

No entanto, ainda segundo Géron (2019), a precisão sozinha não é suficiente para avaliar o desempenho do modelo, especialmente em conjuntos de dados desbalanceados. Por isso, ela é frequentemente utilizada juntamente com outra métrica chamada *recall* (ou sensibilidade), que é a proporção de instâncias positivas corretamente identificadas pelo classificador. O *recall* é calculado como:

$$\textit{Recall} = \frac{TP}{TP + FN} \tag{2}$$

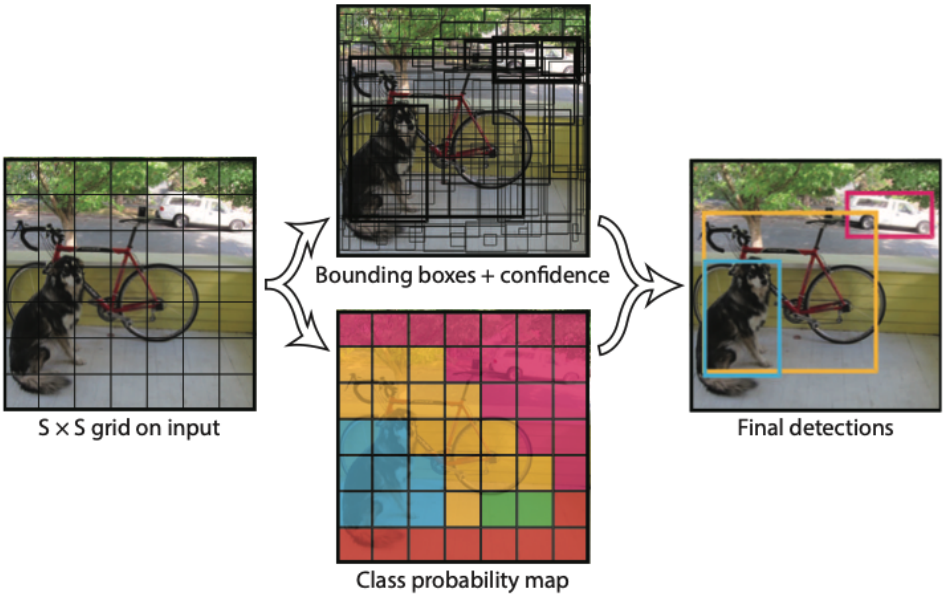
Onde FN é o número de falsos negativos. Uma combinação equilibrada de precisão e *recall* é muitas vezes desejada, e pode ser avaliada pela métrica *F1-score*, que é a média harmônica da precisão e do *recall*.

2.4 Extração de características em imagens via arquitetura YOLO

O YOLO é um *framework* de detecção de objetos baseado em RNC. A arquitetura do YOLO difere das abordagens tradicionais ao tratar a detecção de objetos como um problema de regressão de caixas delimitados (*bounding boxes*), permitindo que a rede realize previsões de objetos em uma única passagem pela imagem (Redmon *et al.*, 2016).

A eficácia do YOLO em tempo real se deve à sua capacidade de realizar detecções precisas com uma única passagem pela imagem, tornando-o um candidato ideal para a detecção de objetos de risco em cenas de vigilância. Segundo Redmon *et al.* (2016), a rede YOLO opera a 45 *frames* por segundo (FPS) sem processamento de *batches* em uma Titan X GPU (Graphics Processing Unit), com versões mais rápidas atingindo até 150 FPS. Essa abordagem eficiente e rápida é fundamental para atender aos requisitos de sistemas em tempo real.

Figura 4 – Demonstração do Modelo YOLO



Fonte: (Redmon *et al.*, 2016).

O sistema divide a imagem de entrada em uma grade $S \times S$, se o centro de um objeto estiver dentro de uma célula da grade, essa célula é responsável por detectar o objeto, o que pode-se observar nas primeiras etapas da Figura 4. Cada célula da grade faz previsões para B *bounding boxes* e seu respectivo nível de pontuações de confiança. As pontuações de confiança

refletem a confiança do modelo de que a caixa contém um objeto e quão precisa ela acredita que a caixa predita. As *bounding boxes* incluem previsões para coordenadas (x, y), largura (w), altura (h) e confiança, sendo esta última calculada como a *Intersection Over Union* (IOU) entre a caixa predita e a verdade real. Além disso, cada célula da grade faz previsões de probabilidades condicionais de classe, $Pr(Classe_i|Objeto)$, que estão condicionadas à presença de um objeto na célula (Redmon *et al.*, 2016).

$$Pr(Classe_i|Objeto) \times Pr(Objeto) \times IOU^{real}_{previsto} = Pr(Classe_i) \times IOU^{real}_{previsto} \quad (3)$$

Ainda conforme Redmon *et al.* (2016), durante o teste, as probabilidades condicionais de classe i dado que um objeto está presente na célula e as previsões individuais de confiança da caixa são multiplicadas, conforme Equação 3, para obter pontuações de confiança específicas da classe para cada caixa. Essas pontuações codificam tanto a probabilidade de que a classe apareça na caixa quanto quão bem a caixa predita se ajusta ao objeto, proporcionando uma abordagem abrangente para a detecção de objetos no sistema.

Após a análise do *framework* YOLO, é importante destacar as evoluções específicas dentro deste modelo, Kang e Kim (2023) detalha todas as versões atuais do YOLO, dentre elas as versões YOLOv5 e YOLOv8, versões de interesse para esse trabalho.

A versão YOLOv5, lançada pela Ultralytics em 2021, representa uma evolução significativa em relação às versões anteriores do YOLO. Ela se destaca por utilizar a CSPNet, que melhora a eficiência computacional ao remover gargalos na utilização de camadas de redes neurais convolucionais. Esta versão apresenta uma estrutura de *backbone* que pode ser ajustada em profundidade e largura, oferecendo quatro variações de tamanho: YOLOv5s (pequeno), YOLOv5m (médio), YOLOv5l (grande) e YOLOv5x (extra grande). Cada variação oferece um equilíbrio diferente entre velocidade e precisão, com a YOLOv5s sendo a mais rápida, porém menos precisa, e a YOLOv5x sendo a mais precisa, porém mais lenta.

Já a YOLOv8, lançada em 2023 também pela Ultralytics, é um avanço ainda mais significativo. Esta versão é apresentada como um framework integrado para detecção de objetos, subdivisão de instâncias e treinamento de modelos de classificação de imagens. O YOLOv8 inclui variações como YOLOv8n, YOLOv8x e YOLOv8m, todas baseadas na estrutura CSPDarkNet53, que oferece um desempenho aprimorado. Entre as mudanças significativas da YOLOv8 em relação à versão anterior, destaca-se a substituição do módulo C3 pelo C2f e ajustes nas camadas de convolução. Além disso, o modelo sem ancoras prevê diretamente o centro do objeto, o que melhora a velocidade do processo de *Non-Maximum Suppression* (NMS).

3 TRABALHOS RELACIONADOS

Todos os trabalhos aqui apresentados foram selecionados, através do Google Acadêmico, por similaridade ao tema proposto por meio de pesquisa e espera-se que o conteúdo apresentado não sirva somente para comparação de resultados mas sim de ajuda para o desenvolvimento.

3.1 Utilização de aprendizado profundo para detecção de armas de fogo em imagens

O estudo realizado por Cardoso, Ciarelli e Vassallo (2019), focou no desenvolvimento de um sistema automatizado para a detecção de armas de fogo em imagens. A pesquisa tinha como objetivo elevar a eficácia da vigilância e segurança pública, utilizando para isso a tecnologia de redes neurais convolucionais e o detector de objetos YOLO, integrado ao *framework* de redes neurais convolucionais *Darknet*. Especificamente, eles utilizaram a versão 2 do YOLO, conhecida como YOLOv2, que se baseia no modelo de classificação *darknet*-19.

Para a realização dos experimentos, a equipe utilizou bases de dados construídas por Olmos, Tabik e Herrera (2018), disponíveis publicamente, que incluíam imagens com e sem armas. Essas bases foram nomeadas como DB-1, DB-2 e DB-T. A metodologia envolveu a implementação em uma máquina com configuração específica, incluindo o sistema operacional Linux Ubuntu 18.04 LTS, processador Intel i7-8700K, 32GB de RAM DDR4 e a placa de vídeo Nvidia Titan V.

Os resultados foram avaliados com base no banco de testes DB-T, tratando o problema como uma classificação binária e utilizando métricas como Precisão, Sensibilidade e F1 *Score*. A primeira abordagem do estudo, que utilizou somente o banco DB-2, atingiu uma precisão de 95,10%. Já a segunda abordagem, que combinou os bancos DB-1 e DB-2 formando o DB-3, mostrou-se mais eficaz, alcançando uma sensibilidade de 100,00% e uma precisão de 89,15%. Esses resultados superaram os anteriores apresentados na literatura, destacando-se especialmente pela segunda abordagem, que equilibrou melhor as classes e apresentou um desempenho geral mais satisfatório.

Um ponto crucial do estudo foi a análise dos falsos positivos, como ilustrado na Figura 5. A presença de uma única classe (“pistol”) no banco DB-2 contribuiu negativamente nos treinamentos, impedindo uma generalização mais efetiva. A segunda abordagem, ao contrário, proporcionou a detecção de todas as armas no conjunto de teste e uma elevada precisão do modelo, mantendo a quantidade de verdadeiros positivos e reduzindo a quantidade de falsos positivos.

Figura 5 – Exemplos de verdadeiros positivos e falsos positivos do segundo experimento.

Fonte: (Cardoso; Ciarelli; Vassallo, 2019).

3.2 Modelos convolutivos para a detecção de armas de fogo em vídeos de vigilância

O estudo realizado por Romero e Salamea (2019), focou no desenvolvimento de um sistema de detecção de armas de fogo em vídeos de vigilância. Este sistema é direcionado para ambientes como varejo, bancos, estações de trem e paradas de ônibus, onde a detecção automática de armas de fogo é crucial.

A pesquisa desenvolveu um sistema composto por duas partes: o *Front End*, utilizando o sistema de detecção e localização de objetos YOLO, e o *Back End*, que consiste no modelo de detecção de armas desenvolvido no trabalho. A motivação para este desenvolvimento foi a inexistência de uma grande base de dados de imagens de pessoas com armas de fogo na internet. Assim, os pesquisadores criaram uma base de dados com 17.684 imagens, obtidas de vídeos de vigilância de roubos reais e práticas de tiro, e provenientes de fontes como YouTube, Instagram e Google.

Para o desenvolvimento do sistema, eles utilizaram redes neurais convolucionais, experimentando com arquiteturas baseadas em *Visual Geometry Group Network* (VGG Net) e *Zeiler*

and Fergus Network (ZF Net). O sistema foi desenvolvido usando Python e TensorFlow. Antes de usar o banco de dados, foi necessário transformá-lo em um formato binário que facilita a importação dos dados para o TensorFlow.

A arquitetura do sistema foi projetada para simplificar ambientes complexos capturados por câmeras de vigilância, concentrando-se nas áreas da imagem onde há pessoas, pois é onde as armas de fogo são mais prováveis de ser encontradas. Esta estratégia tinha o objetivo de reduzir os falsos positivos.

Durante a fase de treinamento, 70% da base de dados foi usada para treinamento, 15% para avaliação e 15% para testes. Os melhores resultados foram obtidos com a arquitetura C2, usando imagens em tons de cinza. Esta configuração mostrou uma melhoria de 21,4% na perda e 3,33% na precisão, em comparação com uma arquitetura baseada em ZF Net. Na fase de teste, com um conjunto de teste de 2.723 imagens, o sistema alcançou uma precisão e um *recall* de 86%.

A interface do sistema foi desenvolvida para exibir o vídeo de vigilância e, quando uma arma de fogo é detectada, a segmentação da imagem onde a arma foi detectada é mostrada, juntamente com uma mensagem de alerta.

Os pesquisadores concluíram que a configuração do sistema permitiu reduzir a complexidade dos ambientes reais de roubo, focando nas áreas mais importantes das imagens para detecção. Para pesquisas futuras, sugerem o uso da mesma arquitetura com modelos convolucionais para focar apenas nas partes importantes da imagem, usando um novo modelo para detectar armas de fogo, seja treinando o modelo em um banco de dados maior ou usando outros modelos e adaptando-os ao problema com aprendizado por transferência.

3.3 Detecção de armas em vídeos em tempo real de CCTV utilizando aprendizado profundo

O estudo de Bhatti *et al.* (2021) apresenta um sistema automático de detecção de armas em tempo real para vídeos de CCTV, visando melhorar a segurança e a ordem pública. Este trabalho é crucial para países que sofrem com atividades violentas, impactando positivamente a economia ao atrair investidores e turistas.

A pesquisa utilizou algoritmos de aprendizado profundo e arquiteturas CNN para a detecção e classificação de armas, optando pela abordagem de transferência de aprendizado e utilizando modelos pré-treinados do ImageNet e *Common Objects in Context* (COCO). Diferentes conjuntos de dados foram criados para classificação e detecção, e um novo conjunto de dados foi compilado para propósitos em tempo real. Esse conjunto incluiu fotos de armas tiradas com câmeras próprias, imagens coletadas manualmente de vídeos de vigilância de roubos, dados da Internet *Movie Firearm Database* (IMFDB) e da Universidade de Granada.

Para a classificação de armas, foram utilizados algoritmos como VGG16, Inception-V3 e Inception-Resnet V2, enquanto para a detecção em tempo real, foram empregados SSD Mobile Net V1, YOLOv3, Faster RCNN-Inception ResNetV2 e YOLOv4. O YOLOv4 se destacou, alcançando uma precisão média de 91,73% e uma pontuação F1 de 91%. O conjunto de dados foi dividido em duas classes: "Pistola"e "Não-Pistola", com a inclusão de objetos de confusão relevantes para reduzir falsos positivos e negativos.

O pré-processamento dos dados foi uma etapa crucial, envolvendo a limpeza, padronização e seleção de recursos, além da aplicação de diferentes filtros OpenCV para melhorar a detecção em condições de baixa luminosidade e resolução. Foram utilizadas técnicas como escalonamento de imagem, aumento de dados, rotulagem de imagem e filtragem usando OpenCV.

Os resultados foram analisados após treinar e testar modelos em conjuntos de dados variados, usando métricas padrão de precisão, *recall* e pontuação F1. O trabalho também incluiu comparações de desempenho entre modelos de classificação e detecção de objetos, com os modelos de detecção de objetos mostrando melhores resultados em termos de precisão e *recall*.

O documento inclui várias figuras ilustrativas, como gráficos de desempenho dos modelos de detecção de objetos e comparações entre os melhores modelos em termos de precisão e pontuação F1. Além disso, são apresentados resultados de detecção para a classe "Pistola"com e sem fundo, em vídeos e em transmissões de CCTV em tempo real.

Em conclusão, este estudo representa um avanço significativo na detecção automática de armas em tempo real, contribuindo para a segurança e bem-estar da sociedade. A pesquisa enfatiza a importância de ter um conjunto de dados em tempo real para treinamento, a fim de alcançar resultados eficazes em situações reais de vigilância.

3.4 Detecção de objetos e contagem de objetos em uma imagem usando TensorFlow

O trabalho proposto por Sai e Sasikala (2019) foca na detecção e contagem de objetos ameaçadores em imagens, utilizando a *Application Programming Interface* (API) de Detecção de Objetos do TensorFlow e o algoritmo Faster R-CNN. Este estudo insere-se no campo da visão computacional, uma área da ciência que lida com o entendimento e manipulação de vídeos e imagens digitais. Com o desenvolvimento do aprendizado profundo, a visão computacional tem sido aplicada em diversas áreas, como reconhecimento facial, recuperação de imagem, inspeção industrial e realidade aumentada.

O trabalho abordou a detecção de objetos, que não se limita apenas à classificação e reconhecimento de objetos em uma imagem, mas também inclui a localização desses objetos e a criação de caixas delimitadoras ao seu redor. O enfoque foi dado especialmente à detecção de objetos ameaçadores, como armas, utilizando tecnologias avançadas como o TensorFlow e o algoritmo Faster R-CNN.

O modelo desenvolvido foi treinado em aproximadamente 4500 etapas para alcançar uma perda inferior a 0,1, processo que levou cerca de doze horas. Após o treinamento, o mo-

delo foi testado com imagens, apresentando resultados promissores na detecção de objetos ameaçadores. Para trabalhos futuros, os autores propõem melhorar ainda mais a eficiência do modelo, aumentando o número de imagens no treinamento e ampliando o número de etapas de treinamento para obter resultados melhores.

Este estudo é significativo para o campo da segurança pública e sistemas de vigilância, onde a rápida e precisa detecção de objetos ameaçadores em imagens é crucial. O uso de tecnologias avançadas como o TensorFlow e o Faster R-CNN destaca como o aprendizado profundo pode ser efetivamente aplicado para aprimorar a detecção de objetos em vários contextos.

3.5 Discussão

De modo geral, todos os trabalhos contribuem para este estudo, especialmente nos aspectos de tecnologias e bases de dados utilizadas. O Quadro 1 apresenta de forma sumarizada as informações focais para análise deste trabalho.

Quadro 1 – Comparação entre Trabalhos Relacionados

Autor	**Métodos Utilizados**	**Bases de Dados**
Cardoso, Ciarelli e Vassallo (2019)	YOLOv2, Darknet	DB-1, DB-2, DB-T Construídas por Olmos, Tabik e Herrera (2018)
Romero e Salamea (2019)	YOLO, VGG Net, ZF Net	17.684 imagens de vídeos de vigilância Criadas a partir de vídeos de vigilância reais
Bhatti *et al.* (2021)	VGG16, YOLOv4	Criadas a partir do site Internet Movie Firearms Database (IMFDB)
Sai e Sasikala (2019)	TensorFlow, Faster R-CNN	Não especificado

Fonte: Autoria própria (2024).

4 MATERIAIS E MÉTODOS

Esta seção descreve a abordagem metodológica que será adotada para atingir os objetivos propostos, assim como os materiais e fluxo de pesquisa para seu desenvolvimento.

4.1 Materiais

Nesta seção são apresentados os materiais necessários para o desenvolvimento deste trabalho.

Quadro 2 – Materiais utilizados no desenvolvimento do sistema

Material	Versão/Modelo	Descrição
Google Colab	Versão gratuita	Plataforma de desenvolvimento em nuvem que permite escrever e executar código Python
Python	Versão 3.11	Linguagem de programação utilizada para o desenvolvimento do sistema
PyTorch	Versão 1.8	Biblioteca de aprendizado profundo utilizada para criar e treinar modelos de redes neurais
OpenCV	Versão 4.8	Biblioteca de processamento de imagens e visão computacional
YOLO	YOLOv8	Algoritmo de detecção de objetos em tempo real
Base de Dados de Objetos de Risco	-	Conjunto de dados específico criado para identificar objetos considerados de risco
Câmeras de Segurança	Modelos compatíveis com RTSP e OpenCV	Dispositivos de captura de vídeo usados para monitoramento e coleta de dados

Fonte: Autoria própria (2024).

4.2 Métodos

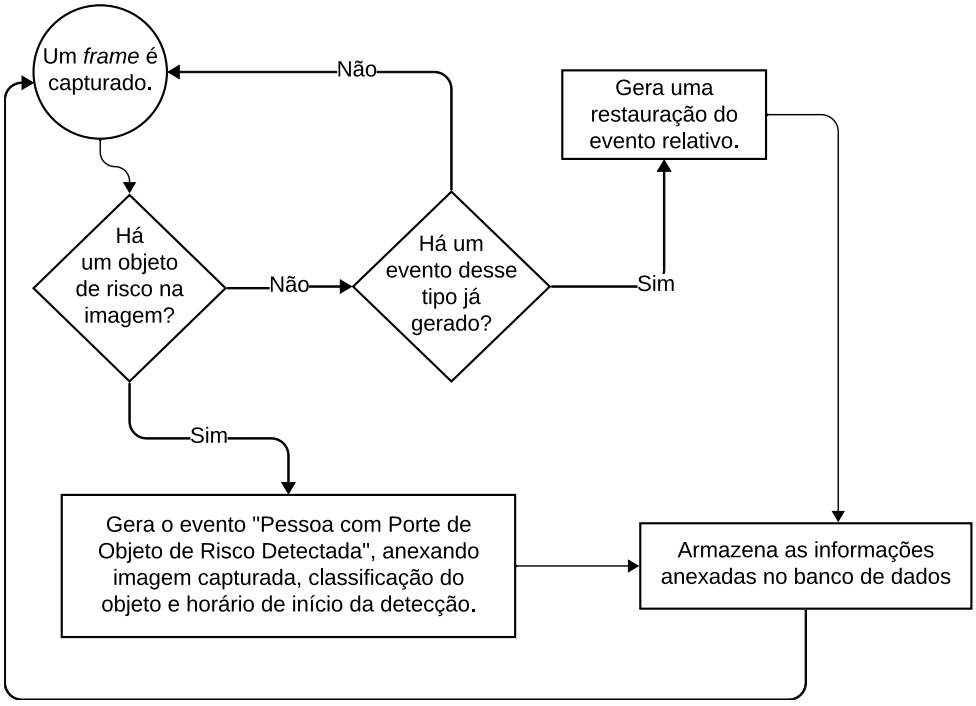
Para abordar a ausência de imagens de armas de fogo na base COCO, foi necessário selecionar uma base específica para essa categoria. Essa etapa incluiu a coleta e anotação de imagens relevantes contendo armas de fogo, facas e também objetos comumente carregadas nas mãos, que não representam risco considerável. A avaliação da qualidade da base de dados empregada se dará junto a avaliação do treinamento e nos experimentos realizados com base na quantidade de imagens.

A configuração do YOLO (ver seção 2.4), foi ajustada para incorporar as particularidades da base complementar de armas de fogo, determinando o número adequado de classes, o tamanho dos lotes para o treinamento e o número de épocas necessário para um aprendizado eficaz.

Para avaliar a eficiência do treinamento, foi analisada a matriz de confusão e *recall* das classes individuais, conforme descrito na subseção 2.3.2, e taxas de perdas de classificação com base nos resultados gerados pela ferramenta YOLO após seu treinamento.

A etapa seguinte foi a implementação do mecanismo para geração de eventos de alerta, registrando eventos como detecção de objetos de risco. Este mecanismo é demonstrado pela Figura 6. No fluxograma, o processo inicia a partir de um *frame* capturado e em seguida é verificado as condições para geração de evento de detecção ou de restauração, um evento de restauração é comum em sistemas de alerta e representa que o evento parou de ser capturado naquele momento. Gerando assim um um *log* contendo horário de geração, a classe do objeto e a precisão detectada.

Figura 6 – Esquema de geração de eventos

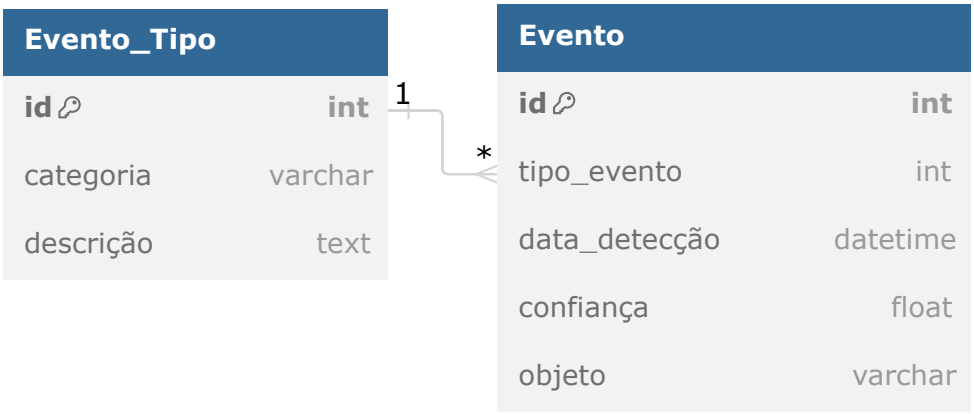


Fonte: Autoria própria (2024).

Posteriormente, os *logs* de eventos são armazenados conforme a Figura 7 para facilitar consultas e análises futuras. A utilização eficiente desses *logs* é crucial para a melhoria contínua do sistema de detecção e para fornecer relações úteis durante a avaliação do sistema.

Após o treinamento concluído para o reconhecimento, bem como o algoritmo para classificar os eventos, a próxima etapa foi a integração do sistema com RTSP que, conforme subseção 2.2.1, permite a observação em tempo real de equipamentos CCTV, utilizando a biblioteca

Figura 7 – Diagrama do Banco de Dados



Fonte: Autoria própria (2024).

OpenCV para a captura e processamento dos fluxos de vídeo provenientes das câmeras de segurança.

Na fase de experimentação e avaliação, primeiramente foi verificado a geração de eventos verdadeiros positivos, em um segundo experimento, foi verificada a quantidade de eventos gerados falsos positivos em um estabelecimento comercial não que tenha a presença de objetos como pistolas e facas, mas sim objetos como cartão de crédito e celulares, com a prospecção de manter uma câmera em funcionamento por pelo menos um dia para verificar as variações de luminosidades.

Para concluir, a otimização do sistema foi realizada possíveis pontos de melhoria. Pode-se citar ajustes nos parâmetros e arquitetura da rede YOLO conforme necessário, garantindo a capacidade do sistema de lidar com variações de iluminação, qualidade de vídeo e diferentes ambientes. Esses métodos constituíram uma abordagem integrada para o desenvolvimento do sistema automatizado de reconhecimento e classificação de cenas de risco em vídeos de câmeras de segurança.

5 RESULTADOS

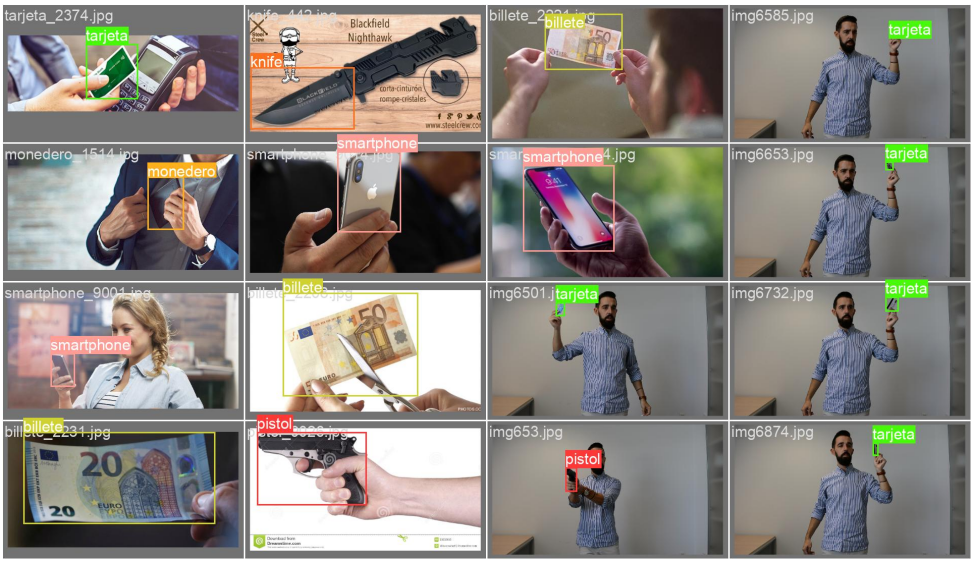
Este capítulo apresenta os resultados obtidos no trabalho, especificamente os resultados sobre o treinamento, criação do algoritmo de detecção e também experimentos com o projeto. Em cada subseção é apresentada uma discussão pontual dos detalhes de cada resultado atingido.

5.1 Treinamento da rede

Nesta etapa, dois processos de treinamento foram executados inicialmente um com uma base de cenas de filmes com armas, obtidas por meio do site *Internet Movie Firearms Database (2024)*. Entretanto, no momento da criação e classificação da base de treino foi percebido que as imagens destoavam muito umas das outras, tanto em questão de qualidade de imagem quanto na variedade de armas e, assim, foi necessário delimitar a base somente utilizando imagens de armas que representam pistolas e complementar com outra base de dados já criada, para isso foi utilizada a base de dados criada por Hernandez (2022) que também foi utilizado no trabalho de Bhatti *et al.* (2021). Este *dataset* possui tanto imagens de pistolas e facas, quanto exemplos de outros objetos que podem ser carregados nas mãos, como dinheiro, cartão de crédito e bolsa de mão que podem ser classificados como objetos que não oferecem risco.

Posteriormente, foi necessário definir as etiquetas das imagens por meio de suas coordenadas. Um exemplo dessa representação pode ser visto na Figura 8. Inclusive, foi a partir dessa figura que foi verificada a qualidade do treinamento do modelo, como o conjunto de imagens nela faz parte do conjunto de validação. Após a predição do treinamento torna-se possível verificar se a rede neural acertou ou não cada classe.

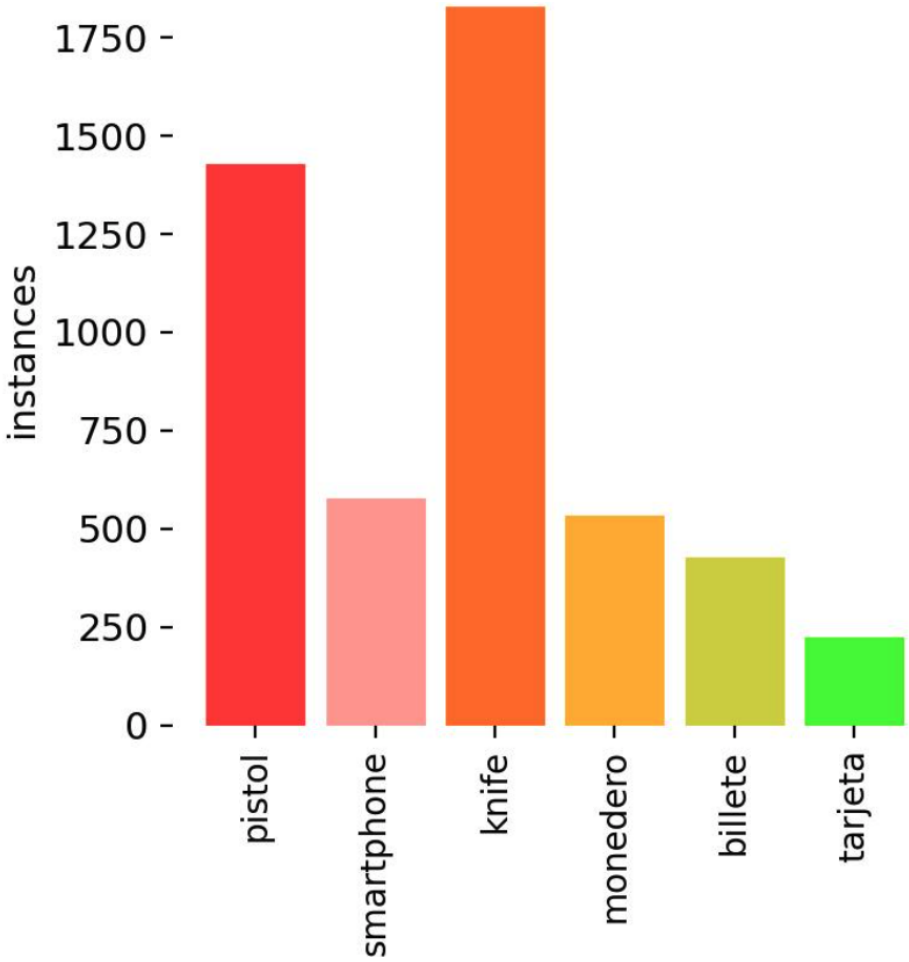
Figura 8 – Conjunto de imagens classificadas para validação



Fonte: Autoria própria (2024).

As imagens e suas respectivas etiquetas foram então empregadas no treinamento da rede neural. Neste processo, a rede foi treinada com um conjunto de aproximadamente 5000 imagens e etiquetas de acordo com a Figura 9.

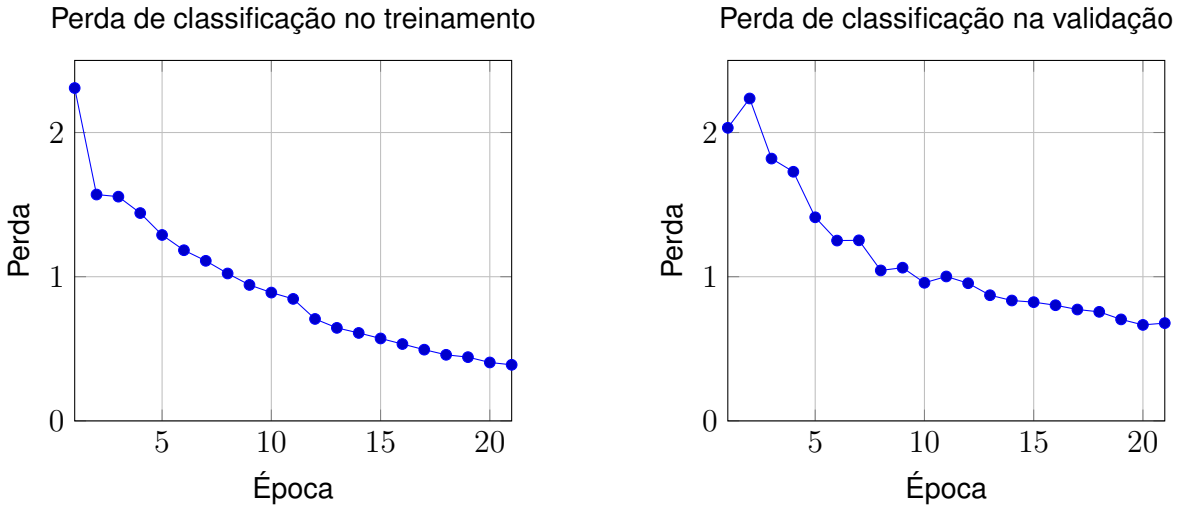
Figura 9 – Divisão de classes do *dataset*



Fonte: Autoria própria (2024).

O processo de decaimento da perda pode ser analisada com base na Figura 10, na qual é possível observar que o valor de perdas de treinamento (*train loss*) e validação (*val loss*) ao longo de 21 épocas, sofrem uma redução consistente nas perdas para todas as métricas de classificação. Isso significa acertar tanto a região de confiança quanto a classe correta, durante o treinamento indicando que o modelo está aprendendo de forma eficaz. As perdas de validação também diminuem ao longo das épocas. A quantia de épocas foram definidas por meio desse resultado, o processo foi feito incrementando a quantidade de épocas até encontrar um estabilidade nos resultados, e isso deve ser levado em conta para os resultados que seguem.

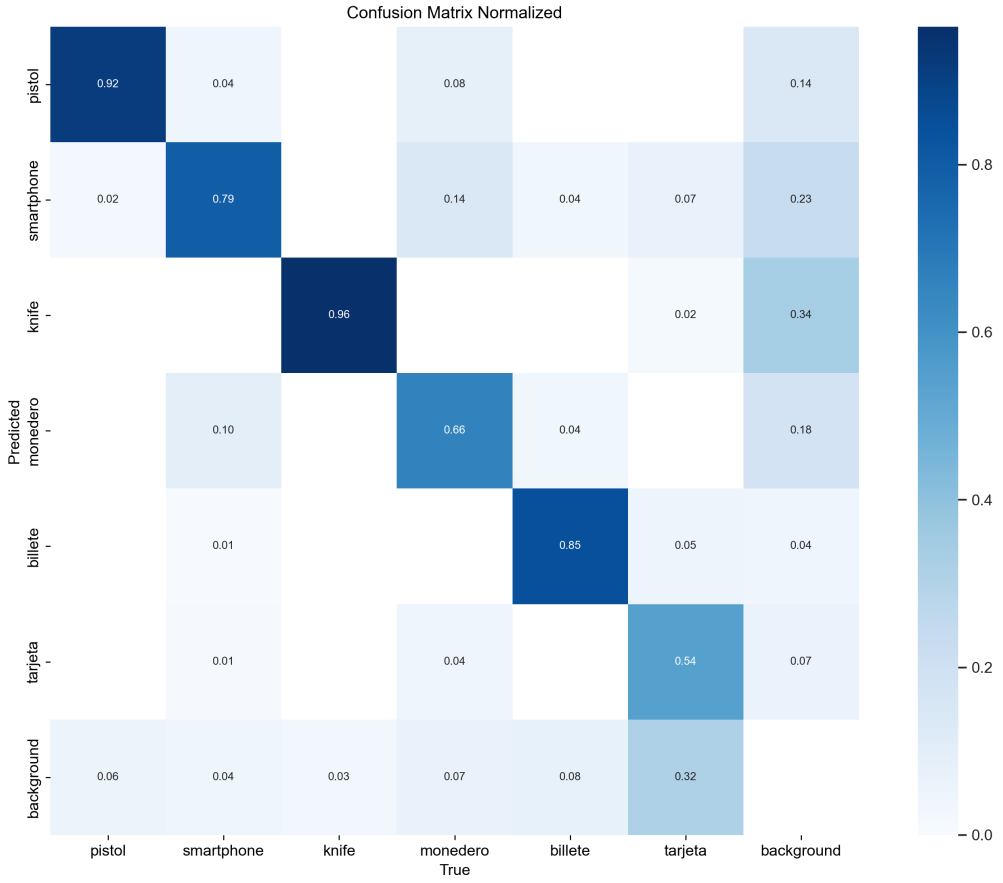
Figura 10 – Evolução do *classification Loss*



Fonte: Autoria própria (2024).

Além da perda em função da época, o YOLO também provê dados para a matriz de confusão conforme a Figura 11 que neste caso está normalizada para se adequar as diferentes quantidades de classes vistas na Figura 9. No eixo y do gráfico tem-se as predições, ou seja, o que a rede neural tenta acertar com base no seu treinamento, já no eixo x é disposto as classes verdadeiras. Vele ressaltar que as classes de interesse são *pistol* e *knife* mas as demais classes também podem serem avaliadas nos casos de falsos positivos, indicando por exemplo a detecção de uma *pistol* quando deveria ser detectado um *smartphone*.

Figura 11 – Matriz de confusão para a classificação das imagens

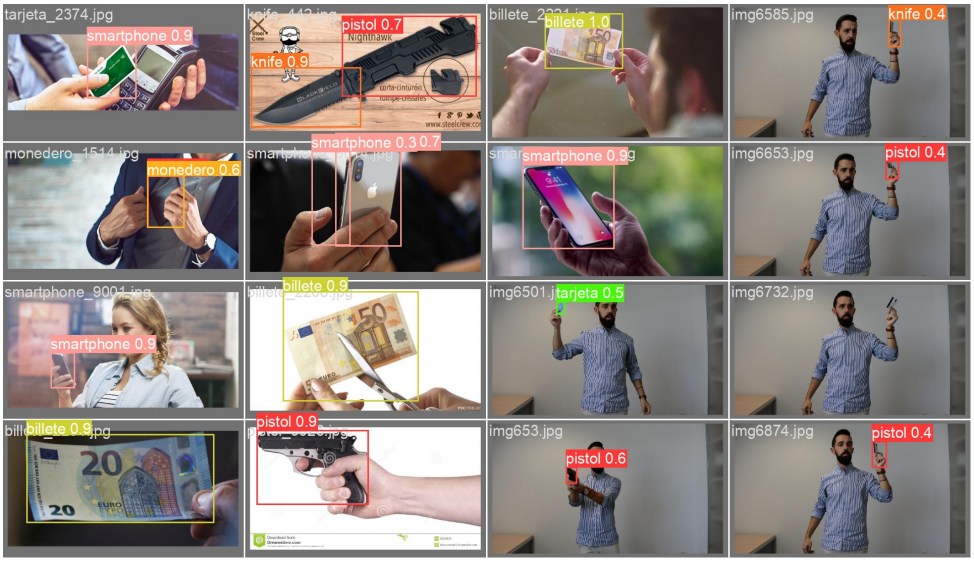


Fonte: Autoria própria (2024).

A partir da matriz de confusão, o *recall* alcançado na última época de treinamento foi de 0.73, com o maior valor obtido durante o processo sendo 0.77. Esses resultados indicam que a rede neural identifica a maioria das instâncias corretamente, embora não todas. Apesar disso, esse desempenho é considerado satisfatório, pois um algoritmo treinado com esse nível de *recall* já contribui significativamente para a detecção de situações de risco. Espera-se que o *recall* melhore com dados mais robustos e uma base de dados mais ampla.

Por fim, após a conclusão do treinamento, é possível observar a acurácia do modelo em um conjunto de imagens. Esse conjunto é o mesmo apresentado na Figura 8, no qual as classes foram definidas para conferência. Os resultados obtidos após o processo de validação do modelo são apresentados na Figura 12. Pode-se observar que ainda ocorreram algumas classificações incorretas nesse processo, contudo, a confiança associada a essas predições é baixa ou vem associado a uma predição com maior confiança, o que pode ser ajustado no algoritmo de detecção de eventos alterando a confiança mínima para iniciar a detecção.

Figura 12 – Conjunto de imagens classificadas para predição



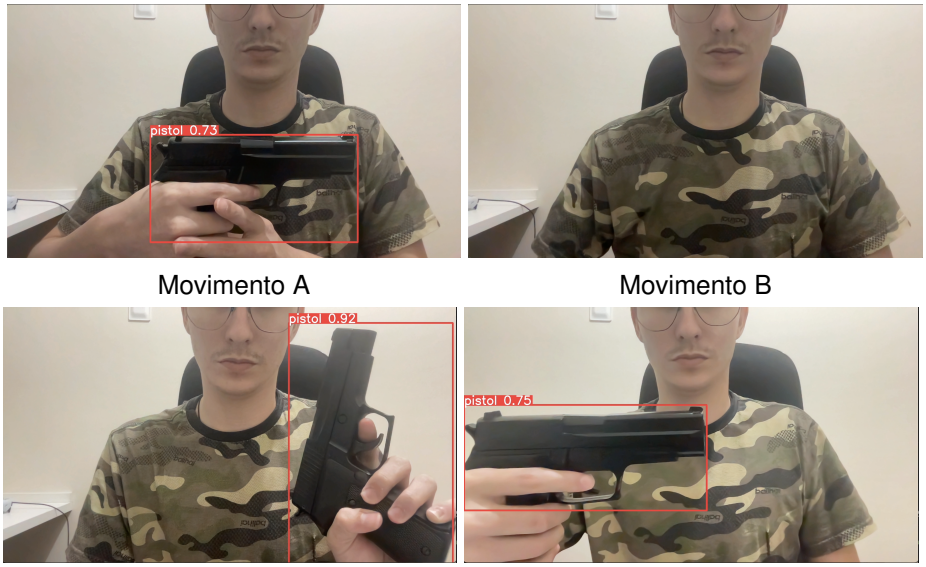
Fonte: Autoria própria (2024).

5.2 Detecção e geração de eventos

Nesta etapa, foi aplicado o algoritmo de detecção com base no modelo treinado. O processo teve início com a importação das bibliotecas necessárias e a especificação do modelo treinado que é utilizado pelo YOLO. Em seguida, foi implementado o processo de geração de *logs*. Conforme representado no esquema da Figura 6, um evento é gerado apenas no caso de detecção de um objeto, enquanto um novo evento de restauração é criado quando o objeto deixa de ser detectado.

Para a validação do algoritmo implementado, foram utilizadas novamente uma *webcam* e uma pistola cinematográfica para avaliar se a geração de eventos seria afetada e se haveria impacto na reprodução em tempo real. Na Figura 13, o protagonista inicia com o objeto na posição do Movimento A e o desloca até o Movimento B, o mesmo ocorre entre os movimentos C e D, movimentos realizados com a arma para simular a atividade no ambiente em tempo real e verificar se, durante esse processo, seria perdido a detecção do objeto de risco, resultando na geração de um evento de restauração e, com uma nova captura, um novo registro de detecção.

Figura 13 – Simulação da movimentação de objeto de risco em tempo real



Movimento A | Movimento B

Movimento C | Movimento D

Fonte: Autoria própria (2024).

Durante os movimentos A e B da Figura 13, foi detectado apenas um evento de detecção e um de restauração, conforme esperado e como pode ser visto na Listagem 1. Este movimento foi executado para verificar se o registro de restauração seria disparado corretamente quando o objeto sumisse da cena.

Listagem 1 – *Logs* dos eventos gerados nos movimentos A e B

```
1 2024-05-23 15:53:21,764 - INFO - Detectado: pistol. Confidence ---> 0.82
2 2024-05-23 15:53:27,867 - INFO - Restauração: pistol
```

Fonte: Autoria própria (2024).

Entretanto, entre os movimentos C e D, a detecção ocorreu conforme mostrado na Listagem 2, resultando em três eventos de detecção e dois de restauração. Esta segunda situação não é ideal, pois, em um ambiente real, o disparo de eventos seria excessivo e deveria propagar o registro da cena somente no momento da detecção inicial e ao final do rastreamento do objeto.

Listagem 2 – *Logs* dos eventos gerados nos movimentos C e D

```
1 2024-05-23 15:54:06,336 - INFO - Detectado: pistol. Confidence ---> 0.89
2 2024-05-23 15:54:10,066 - INFO - Restauração: pistol
3 2024-05-23 15:54:10,505 - INFO - Detectado: pistol. Confidence ---> 0.81
4 2024-05-23 15:54:20,138 - INFO - Restauração: pistol
5 2024-05-23 15:54:21,422 - INFO - Detectado: pistol. Confidence ---> 0.77
```

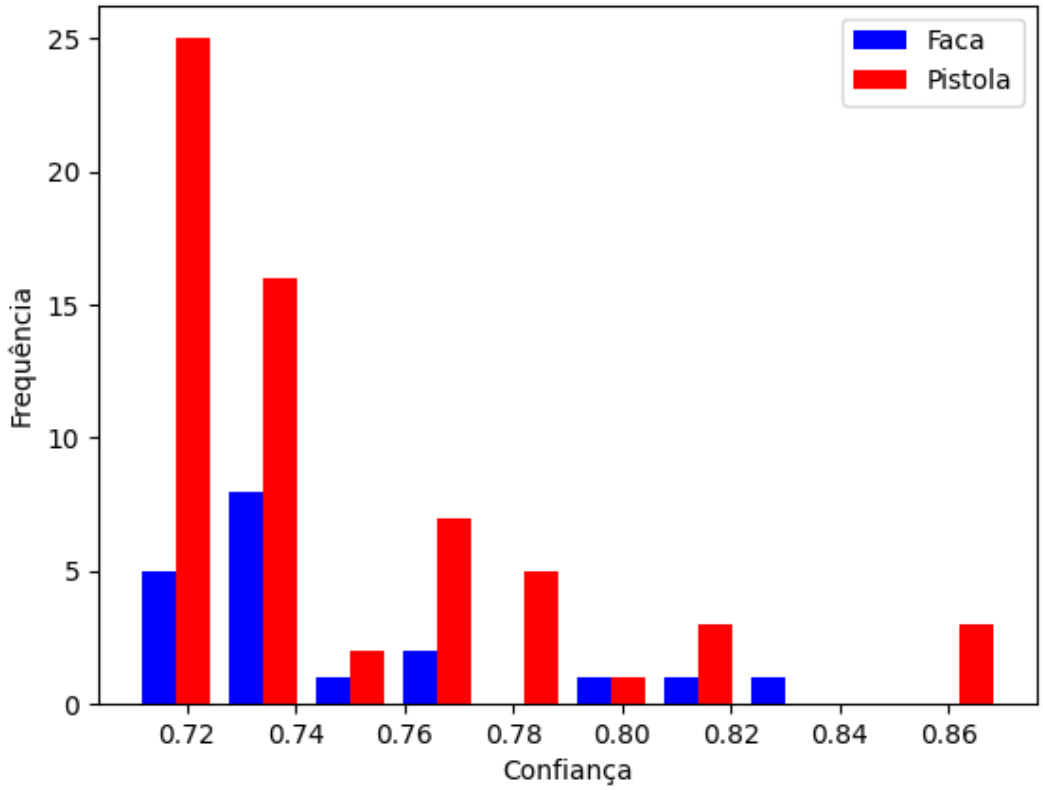
Fonte: Autoria própria (2024).

5.3 Experimentação em caso real

Durante um período de 24 horas, o sistema foi integrado a uma câmera IP utilizando o protocolo RTSP em um estabelecimento comercial com acesso restrito ao autor deste trabalho para averiguar falsos positivos. Antes da integração, foi necessário ajustar o algoritmo. Em vez de registrar *logs*, o algoritmo foi modificado para enviar requisições *HyperText Transfer Protocol* (HTTP) a um serviço dedicado que armazena todos os eventos em um banco de dados PostgreSQL, conforme o diagrama apresentado na Figura 7. É importante destacar que o serviço intermediário foi criado para capturar os eventos, assegurando que o tempo de inserção no banco de dados não impactasse o processamento dos *frames* de vídeo.

O histograma mostrado na Figura 14 ilustra a distribuição dos eventos gerados, como falsos positivos, para cada classe de acordo com a confiança daquela detecção, o que significa que 25 pistolas foram detectadas com confiança de 0.72 dentro do período de 24 horas. Durante esse processo, nenhum objeto considerado de risco, como pistolas ou facas, deveria ser reconhecido, dado que o ambiente não contempla tais itens. Algumas das ocorrências foram analisados para identificar objetos erroneamente classificados; a maioria consistia em equipamentos eletrônicos como telefones e roteadores. É comum que sistemas de detecção apresentem falsos positivos, especialmente em contextos práticos, onde vários objetos que não estavam contemplados no treinamento do modelo. Esse tipo de erro é esperado, uma vez que o modelo foi treinado com pouca variedade de objetos.

Figura 14 – Histograma de confiança da geração de eventos do experimento



Fonte: Autoria própria (2024).

No entanto, o objetivo principal era que o algoritmo detectasse corretamente objetos de risco quando presentes. Em um evento simulado fora do período registrado no histograma (Figura 14), foi introduzida uma arma fictícia no ambiente, resultando em eventos detectados com confiança variando de 0.8 a 0.9. Isso sugere que é possível ajustar o algoritmo para capturar objetos de risco com uma confiabilidade acima de 0.75, reduzindo assim a quantidade de falsos positivos.

6 CONCLUSÃO

Este trabalho apresentou um sistema para o reconhecimento e a classificação de cenas de risco em vídeos de câmeras de segurança, utilizando técnicas avançadas de visão computacional e aprendizado profundo. Através da implementação do modelo YOLO, foi possível detectar, em tempo real, objetos e situações de risco, como a presença de armas de fogo e facas, contribuindo de forma inicial para a automação e integração dos sistemas de vigilância.

Os resultados obtidos durante os experimentos confirmaram a eficácia do sistema proposto, demonstrando uma alta precisão na identificação de objetos de interesse e na geração de alertas relevantes para a segurança. A aplicação prática em cenários reais mostrou-se promissora, indicando que o uso de redes neurais convolucionais pode aprimorar a capacidade de resposta a incidentes críticos, proporcionando um monitoramento mais proativo e preventivo.

O trabalho também destacou a importância da seleção adequada de bases de dados para o treinamento dos modelos, bem como a necessidade de um constante refinamento dos algoritmos para lidar com a diversidade e complexidade das cenas capturadas. Dessa forma, é possível garantir uma maior robustez e adaptabilidade do sistema às diferentes condições de operação encontradas no dia a dia.

Em resumo, o desenvolvimento deste sistema representa um avanço na área de segurança, oferecendo uma ferramenta para a proteção de propriedades e pessoas. Futuras pesquisas poderão explorar a integração de novas tecnologias e a expansão das capacidades do sistema, visando aprimorar ainda mais a detecção de riscos e a tomada de decisões em tempo real. Além disso, como trabalhos futuros, espera-se aprimorar o treinamento do modelo adequando melhor sua base de dados, focando em imagens que representem cenários reais. Para isso, pode-se considerar a utilização exclusiva de objetos de risco em porte manual, ou seja, a pessoa segurando a arma, o que resultaria em uma base de dados mais homogênea.

REFERÊNCIAS

BAESSO, P. H. Estudo técnico e econômico de diferentes sistemas de cftv. 2016.

BAGGIO, G. *et al.* The opencv library. **Journal of Software: Evolution and Process**, v. 28, n. 1, p. 5–18, 2016.

BASKAR, A. *et al.* **Digital Image Processing**. [*S.l.*: *s.n.*], 2023. ISSN 1432-1084.

BHATTI, M. T. *et al.* Weapon Detection in Real-Time CCTV Videos Using Deep Learning. **IEEE Access**, v. 9, 2021. ISSN 21693536.

BOND, L. **Brasil teve 24 ataques a escolas; mais da metade nos últimos 4 anos**. 2023. Agnência Brasil. Disponível em: https://agenciabrasil.ebc.com.br/geral/noticia/2023-05/brasil-teve-23-ataques-escolas-mais-da-metade-nos-ultimos-4-anos. Acesso em: 24 ago. 2023.

BOUREAU, Y.-L. *et al.* A theoretical analysis of feature pooling in visual recognition. *In*: **Proceedings of the 27th International Conference on Machine Learning (ICML-10)**. [*S.l.*: *s.n.*], 2010.

BRICALLI, I. L. O paradoxo da cidade monitorada: Vigilância limitada e espaços públicos fragilizados a partir do estudo do sistema das câmeras do município de vila velha-es. 2015.

CARDOSO, G. V. S.; CIARELLI, P. M.; VASSALLO, R. F. Use of Deep Learning for Firearms Detection in Images. *In*: . [*S.l.*: *s.n.*], 2019.

CARVALHO, A. C. P. d. L. F. d. **Redes neurais artificiais**. 2020. Acesso em 28 out. 2023. Disponível em: https://sites.icmc.usp.br/andre/research/neural.

DAMJANOVSKI, V. **CCTV: From Light to Pixels, Third Edition**. [*S.l.*: *s.n.*], 2013.

FIRMANSYAH, M. I. M.; SUHARTO, N.; PRASETYO, Y. H. RTSP and HTTP Protocol Analysis for Streaming Services on Manet Networks in State Polytechnic of Malang. **Jartel**, 2022. ISSN 2407-0807.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition**. [*S.l.*: *s.n.*], 2019.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [*S.l.*]: MIT Press, 2016. http://www.deeplearningbook.org.

HAYKIN, S. **Neural Networks and Learning Machines**. [*S.l.*: *s.n.*], 2008. v. 3. ISSN 14337851.

HE, K. *et al.* Deep residual learning for image recognition. *In*: **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. [*S.l.*: *s.n.*], 2016. v. 2016-December. ISSN 10636919.

HERNANDEZ, F. P. **Weapon detection datasets**. 2022. Disponível em: https://github.com/ari-dasci/OD-WeaponDetection.

Internet Movie Firearms Database. **Internet Movie Firearms Database**. 2024. Disponível em: http://imfdb.org.

ITSEEZ. **The OpenCV Reference Manual**. 2.4.9.0. ed. [*S.l.*], 2014.

KANG, C. H.; KIM, S. Y. Real-time object detection and segmentation technology: an analysis of the YOLO algorithm. **JMST Advances**, v. 5, n. 2-3, 2023. ISSN 2524-7905.

LEJMI, W.; KHALIFA, A. B.; MAHJOUB, M. A. Challenges and methods of violence detection in surveillance video: A survey. *In*: VENTO, M.; PERCANNELLA, G. (Ed.). **Computer Analysis of Images and Patterns**. Cham: Springer International Publishing, 2019. p. 62–73. ISBN 978-3-030-29891-3.

OLMOS, R.; TABIK, S.; HERRERA, F. Automatic handgun detection alarm in videos using deep learning. **Neurocomputing**, v. 275, 2018. ISSN 18728286.

PEDRINE, H.; SCHWARTZ, W. **Análise De Imagens Digitais: Princípios, Algoritmos E Aplicações**. 1. ed. [*S.l.*]: engage Learning, 2008.

PINHEIRO, D. **Câmeras IP permitem vigiar a casa pela Internet**. 2006. Acesso em: 6 ago. 2010. Disponível em: http://tecnologia.uol.com.br/ultnot/2006/12/13/ult2870u215.jhtm.

RAUBER, T. W. Redes neurais artificiais. **Universidade Federal do Espírito Santo**, v. 29, 2005.

REDMON, J. *et al.* You only look once: Unified, real-time object detection. *In*: **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. [*S.l.*: *s.n.*], 2016. v. 2016-December. ISSN 10636919.

ROMERO, D.; SALAMEA, C. Convolutional models for the detection of firearms in surveillance videos. **Applied Sciences (Switzerland)**, v. 9, n. 15, 2019. ISSN 20763417.

SAI, B. N. K.; SASIKALA, T. Object Detection and Count of Objects in Image using Tensor Flow Object Detection API. *In*: **Proceedings of the 2nd International Conference on Smart Systems and Inventive Technology, ICSSIT 2019**. [*S.l.*: *s.n.*], 2019.

SZELISKI, R. Computer Vision : Algorithms and Applications 2nd Edition. **Springer**, 2021.

THIRTHALA, S.; SINHA, S. N.; POLLEFEYS, M. Calibration of pan-tilt-zoom (ptz) cameras and omni-directional cameras. *In*: IEEE. **2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)**. [*S.l.*], 2005. v. 2, p. 1198–vol.

ZELINSKY, A. **Learning OpenCV—Computer Vision with the OpenCV Library**. 2009.