

USO DE REDE NEURAL IMPLEMENTADA SOBRE DISPOSITIVOS IOT PARA RECONHECIMENTO DE INDIVÍDUOS ARMADOS

NEURAL NETWORKS IMPLEMENTED ON IOT DEVICES FOR ACTIVE SHOOTER RECOGNITION

Rafael Santana Coelho¹
Lucas Arruda Ramalho²
Armando Da Silva Filho³
Alexandre Berndt⁴

RESUMO

Este artigo apresenta a utilização de métodos de detecção de objetos implementados sobre dispositivos embarcados para a área da segurança. Os sistemas de vigilância com câmeras inteligentes e conexões IP, fazem parte da maioria dos sistemas de segurança tradicionais. Este artigo propõe um modelo de câmera implementado em um dispositivo ESP32, que é conectado a um servidor Raspberry Pi o qual é responsável pela análise de imagens através de uma rede neural para reconhecimento de armas. Foram utilizadas técnicas de pré-treinamento em nuvem através da plataforma Google Collab na rede MobileNet como forma de eliminar problemas com relação ao tempo de treinamento da rede neural. Além disso, também foi realizada a implementação de um serviço TCP/IP multithread para comunicação entre os dispositivos IoT anteriormente citados, o que permitiu diversas conexões no servidor simultaneamente. Os testes realizados buscaram utilizar a rede neural MobileNet em mais de 400 cenários com diferentes objetos. Através de uma matriz confusão, o sistema proposto apresentou resultados promissores de 57% de acurácia e 90% de precisão. A implementação provou ser funcional, capaz de detectar o objeto procurado, e enviar os alertas utilizando a conectividade WIFI, sem apresentar altas taxas de falso negativo.

Palavras-chave: Redes Neurais Convolucionais. IoT. Detecção de objetos. Sistema Embarcados.

ABSTRACT

This paper introduces the use of object detection methods implemented on embedded devices for the image monitoring systems. Those surveillance systems with intelligent cameras and IP connections are part of most traditional security systems. This paper proposes a camera model implemented in an ESP32 device, which is connected to a Raspberry Pi server that is responsible for image analysis through a neural network for active shooter recognition. Cloud pre-training techniques were used through the Google Collab platform on the MobileNet network. In addition, also the implementation of a multithread TCP/IP service for connection between the IoT devices cited above, to handle multiple connections on the server

¹ Universidade do Estado de Mato Grosso – Unemat.

² Universidade do Estado de Mato Grosso – Unemat.

³ Universidade do Estado de Mato Grosso – Unemat.

⁴ Universidade do Estado de Mato Grosso – Unemat.

simultaneously. The tests performed have aimed to test the MobileNet neural network in more than 400 scenarios with a person carrying different objects. Using a fuzzy matrix, the proposed system showed promising results of 57% accuracy and 90% precision. The implementation proved to be functional, able to detect the sought object, and send the alerts using WIFI connectivity.

Keywords: Convolutional Neural Networks. IoT. Object Detection. Embedded Systems.

1 INTRODUÇÃO

Com o crescimento da criminalidade, lugares com grande aglomeração de pessoas estão propensos a ataques, como assaltos e arrastões, que põe em riscos vidas, pois, em sua maioria, são praticados com armas de fogo. De acordo com Blair e Scheweit (2014), lugares como estes têm ainda tendência a atos como *active shooter*, em que um indivíduo está ativamente determinado a matar ou tentar matar pessoas seja em lugares abertos ou fechados. Incidentes recentes envolvendo armas de fogo têm apontado a necessidade de mais medidas de segurança para salvar vidas. De acordo com o G1 (2019), só no Brasil ocorreram oito ataques em ambientes escolares, entre 2002 e 2018. Um deles foi o Massacre de Realengo, em 2011, no qual um indivíduo entrou armado com dois revólveres em uma escola no bairro de Realengo, na Zona Oeste do Rio, e começou a disparar contra vários alunos, deixando 12 mortos e 22 feridos. Segundo Madeiro (2019), a taxa de assassinato por arma de fogo no Brasil registra a marca de 72% dos homicídios do país em 2017, somando 45.510 mortes.

Como forma de combater esses tipos de ataques, os sistemas de segurança estão cada vez mais populares. Segundo Machado (1991), os sistemas eletrônicos estão se multiplicando por todas as partes. Regiões, como aeroportos, supermercados, fábricas, lojas, já estão sendo submetidos a esses olhares técnicos de câmeras de observação, porém, para uma possível tomada de decisão e monitoramento em tempo real, necessita-se ainda que essas observações sejam feitas com a supervisão humana, crescendo assim espaços para falhas (DIVS, 2015).

Contudo, pesquisas, como as de Ferreira (2021) e de Talayviya *et al* (2020), buscam a implementação de tecnologias de alertas automáticos para o monitoramento na área da agricultura, e outras pesquisas, como a de Maksymenko (2020), com aplicação no campo da saúde. Este trabalho, entretanto, segue por outra área, a da segurança, tendo como objetivo propor uma alternativa de baixo custo para intensificar o monitoramento, colaborando com as unidades competentes e, dessa forma, auxiliando no salvamento de vidas. Este artigo descreve

a implementação de uma inteligência artificial em câmeras, com sistema de alerta automático embutido, para detectar indivíduos portando armas de fogo em locais muito povoados.

2 VISÃO COMPUTACIONAL SOBRE DISPOSITIVOS EMBARCADOS

De acordo com Honorato e Milano (2014), a visão computacional é responsável pela maneira como o computador percebe o mundo em sua volta. Sendo assim, a máquina extrai informações de vídeos e imagens, realiza reconhecimento, manipulação e detecção de objetos. Para um sistema de visão computacional algumas etapas são estipuladas:

- Aquisição - Recebimento de imagens;
- Pré-Processamento - Identificação das informações contidas na imagem;
- Extração - Extração das características da imagem;
- Detecção - Destacamento das regiões relevantes da imagem;
- Processamento de Alto nível - Validação dos resultados obtidos, estimativa dos parâmetros da imagem e classificação dos objetos encontrados

Os sistemas de visão computacional, tem se colocado bastante necessário dentro do desenvolvimento de produtos, tal como automação robótica, automação de tarefas, detecção de padrões e agilidade nos processos de produção, podendo agilizar tarefas complexas dentro de uma indústria (RUDEK; COELHO; CANGIOLIERI JR, 2001).

Em um sistema embarcado, o uso da detecção de objetos se dificulta devido a etapas de treinamento em que memória e processamento são vitais para algoritmos de detecção e reconhecimento. Essa limitação de memória pode ser contornada por meio da implementação de redes neurais pré-treinadas⁵.

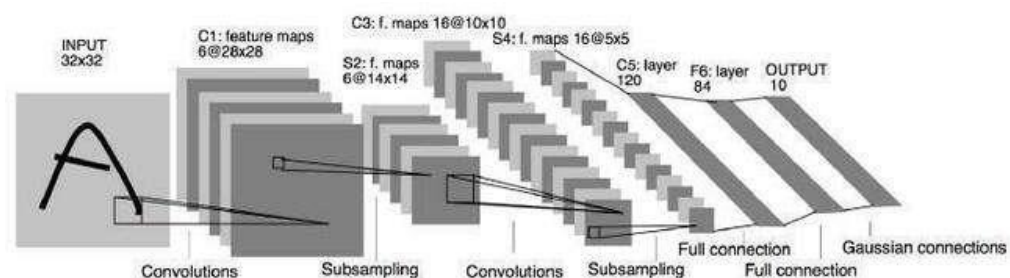
O uso de RNAs em tarefas de processamento de imagens se estende em diversos tipos de tarefas como: reconhecimento de impressão digital, reconhecimento de assinaturas de cheques e documentos, reconhecimento de caracteres de texto em imagens, classificação de imagens com propriedades similares, filtros inteligentes e visão artificial. A visão artificial é uma forma de reconhecimento aplicada à detecção de expressões faciais, pessoas e objetos (OSÓRIO; BITTENCORT, 2000).

⁵ Redes Neurais Pré-Treinadas: o dispositivo hospedeiro da rede, não será submetido ao custo computacional de treinamento, apenas pelo reconhecimento das imagens.

O uso de redes neurais em processamentos de imagens para a classificação de padrões, tem se destacado diversas redes, como as redes neurais convolucionais (CNN) que utilizam camadas ocultas para extrair características de um conjunto de dados (PACHECO, 2016).

As redes neurais convolucionais, se comportam de maneiras diferentes das redes neurais regulares. Isso se dá devido a sua organização de camadas, as entradas que consistem em imagens, e a possibilidade de codificação de certas características na arquitetura da rede.

Figura 1 - Redes Neurais Convolucionais



Fonte: Alves (2018)

MobileNet é uma rede neural convolucional trabalhada para ter eficiência e ao mesmo tempo tamanho reduzido para ser suportada em dispositivos com recursos limitados (FERREIRA, 2021). Esse tipo de rede utiliza convoluções separáveis em profundidade, técnica utilizada para a redução da quantidade de parâmetros da rede, a qual utiliza 8 a 9 vezes menos poder computacional (HAWARD, 2017).

Devido ao seu tamanho reduzido em ocupação de memória, a rede MobileNet foi utilizada neste artigo com o objetivo de possibilitar sua implementação em dispositivos embarcados com Raspberry Pi⁶, que normalmente apresentam capacidade de armazenamento reduzida, tanto em memória primária ou secundária.

3 IMPLEMENTAÇÃO DO SISTEMA INTELIGENTE.

Para a implementação do sistema, algumas estratégias foram estipuladas. O funcionamento do sistema dá-se por uma sequência de etapas no Raspberry e no Esp32:

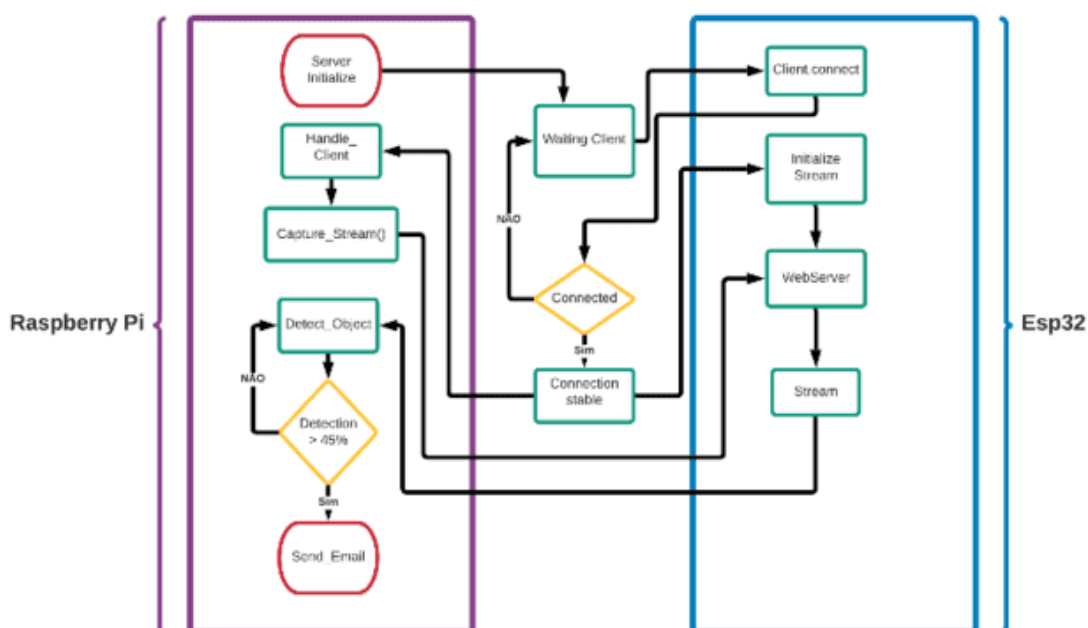
- Comunicação - Raspberry inicia um servidor para uma conexão cliente/servidor;

⁶ Raspberry Pi: Mini computador de baixo custo integrado em uma placa única, baseado em sistemas System on Chips (Tudo em um único chip).

- Aquisição - Esp32 se conecta ao servidor, iniciando um WebServer para disponibilizar as imagens;
- Detecção - Raspberry recebe as imagens do WebServer por meio de uma função Capture_Stream(), e inicia o processo de detecção de objetos nas imagens;
- Alerta - Envio dos alertas via email.

Essas etapas, permitem que os dois dispositivos se comuniquem um com o outro e realizem a detecção de objetos nas imagens disponibilizadas. Exemplificada na Figura 2.

Figura 2 – Diagrama de Bloco do Sistema



Fonte: Autor (2022)

3.1 Preparação dos Dados

A aprendizagem artificial de uma rede é realizada por meio do processamento de uma alta quantidade de dados de amostragem, e ao avaliar esses dados, a rede é capaz de tomar decisões (APPEN, 2020). Para áreas com visão computacional, onde os dados são imagens, bancos de imagens são utilizados para as análises de características.

No presente trabalho utilizamos os bancos de imagens *OpenImageDataset* e *Roboflow*, em que os objetos nas imagens já vêm rotulados, redimensionados e preparados para serem passados para o treinamento da rede. No total foram obtidas 1536 imagens, 1229 para o treinamento e

307 para a validação e teste da rede, a escolha dos bancos foi feita por terem as imagens rotuladas com as classes *Shotgun*⁷ e *Handgun*⁸.

3.2 Treinamento

Ao trabalharmos com imagens, a quantidade inserida para treinamento pode ser grande. Por esse motivo é utilizado um formato disponibilizado pelo *tensorflow* chamado de *TFRecords*. Esse formato armazena os binários de todos os dados eliminando problemas de memória. Geralmente o conjunto de dados de treinamento comporta 60 - 90 % dos dados da rede e o conjunto de dados para teste 10 - 40 % das amostras (SILVA *et al*, 2010).

O Treinamento da rede foi realizado no *Google Collaboratory* também conhecido como *Google Collab*, uma plataforma em nuvem que fornece uma Unidade de processamento gráfico(GPU) para pesquisas envolvendo *Machine Learning*, favorecendo na agilidade ou na velocidade do treinamento e na praticidade de uso, se tornando bastante utilizado hoje em dia em diversas aplicações (KANAMI; PADOLE, 2019; KATARIA, 2020).

3.3 Implementação Raspberry

A utilização do Raspberry como um servidor do projeto foi devido a sua capacidade de processamento, memória e o fato de ser um mini computador com sistema operacional baseado no Linux. A implementação no Raspberry foi dividida em partes:

- Comunicação - A conexão socket com os clientes;
- Aquisição - Criação de uma função para obtenção das imagens do Servidor HTTP;
- Multithreading - Threads para várias conexões simultaneamente;
- Detecção - Detecção de objetos;
- Alerta - Envio dos alertas via email.

A comunicação é dada pela biblioteca *Socket* da linguagem *python*, que permite o levantamento de um servidor *socket* em um *IP* e porta *TCP* específica (SARKER; WASHINGTON, 2015).

Com os dispositivos conectados na mesma rede e as imagens sendo disponibilizadas no Servidor *HTTP* no ESP32, foi implementado a função de aquisição das imagens, cujo o servidor Raspberry Pi captura as imagens do servidor *HTTP* ESP32.

⁷ Shotgun: Espingarda

⁸ Handgun: Pistola

O uso de *MultiThreads* no sistema foi necessário para permitir que o servidor Raspberry Pi pudesse se conectar com diversos ESP32 em ângulos diferentes, podendo executar as conexões separadamente (MOREIRA; 33 CARRIEL; GUTIERREZ, [entre 2010 e 2020]).

O acesso dos dispositivos no servidor separadamente e simultâneo, faz com que todos os dispositivos recebem uma identificação baseado no número da *thread* que ele se encontra, assim ao acessarem a função *Detect_image*, os valores de sua detecção são retornados e inserido à uma lista junto à sua identificação. A emissão do alerta é feito caso a análise feita na lista das detecção for superior a 45% (valor arbitrário), dada pela função *email-alert* enviando um email. O código de implementação do sistema em Raspberry Pi está disponível no github (URL: <https://github.com/RafaelSCoelho/Object-Detection-Raspberry-Pi-Esp32>).

3.4 Implementação TTGO-Esp32-Camera

O uso de componentes como esp32 estão sendo utilizados em diversos dispositivos populares hoje em dia. Sua utilização com câmeras gerou um desenvolvimento de sistemas de segurança baseados em câmeras inteligentes, capazes de monitorar e tomar decisões através de seus resultados (LUKETINA *et al*, 2019).

O esp32 contém em seu *firmware* nativo um *HTTP Server* que disponibiliza funcionalidades de manipulação da câmera. As imagens obtidas pela câmera são disponibilizadas e enviadas ao servidor.

No dispositivo ESP32 foi utilizado o método *WifiClient* da biblioteca *WiFi.h*, que permite a conexão de um dispositivo em um determinado IP do servidor representado por *SERVER*, e uma porta indicada pela variável *PORT*. O IP e a porta TCP são utilizados para conectar ao servidor e enviar dados do dispositivo.

O código de implementação do sistema em ESP32 está disponível no github (URL:<https://github.com/RafaelSCoelho/Object-Detection-Raspberry-Pi-Esp32>).

4 ANÁLISE DOS TESTES

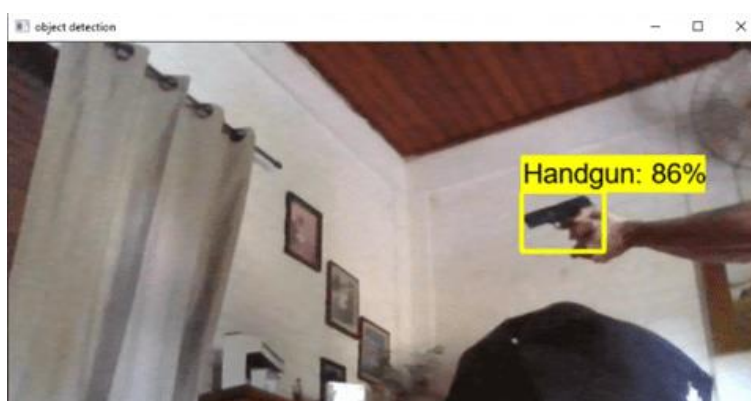
Para a realização dos experimentos executados na aplicação de detecção de objetos, foram estipulados 10 tipos de objetos diferentes. Para o uso dos objetos nos testes, definimos as seguintes estratégias: para casos de verdadeiros positivos e falsos negativos, foi utilizado uma arma de brinquedo e uma espingarda de pressão, cada objeto foi passado por 80 cenários de validação. Já para os casos de falsos positivos e verdadeiros negativos, foram utilizados os

objetos: furadeira, parafusadeira, pistola de cola, controle remoto, celular, guarda-chuva, pistola de solda e uma faca, tendo cada um desses objetos, passado por 40 testes de validação.

4.1 Resultados

Os resultados mostraram que, ao posicionar a arma de brinquedo em frente a câmera do dispositivo, em que o resultado esperado seria a identificação da classe *Handgun*, a rede foi capaz de identificar o objeto com valores acima de 75% com a denominação de classe correta. A detecção foi realizada em aproximadamente 1 segundo, em uma distância estimada de 1 metro. Os casos de verdadeiro positivos podem ser observados nas Figuras 3 e 4.

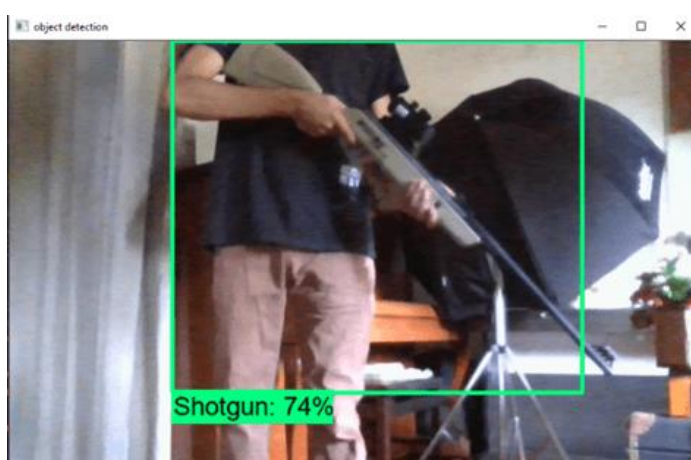
Figuras 3 - Caso de verdadeiro positivo para identificação da classe Handgun



Fonte: Autor (2022)

Com os testes realizados para a classe *shotgun*, os resultados apresentaram valores superiores a 70% com a denominação da classe correta e a detecção foi obtida com sucesso em aproximadamente 1 segundo com distância superior aos testes realizados com classes *handgun*.

Figuras 4 - Caso de verdadeiro positivo para identificação da Shotgun



Fonte: Autor (2022)

Nos cenários com a furadeira e a parafusadeira em frente a câmera com distâncias diferenciadas, a rede apresentou casos de verdadeiros negativos. Cenário de verdadeiro negativo apresentado na Figura 5.

Figuras 5 - Caso de verdadeiro negativo



Fonte: Autor (2022)

Os casos com maiores índices de detecção de falsos positivos foram os testes realizados com a pistola de cola quente. O maior caso de detecção foi de 55% apresentado na Figura 6.

Figura 6 - Caso de Falso positivo.



Fonte: Autor (2022)

Devido a configuração e baixa resolução da câmera do dispositivo esp32, a rede mostrou alguns casos de falsos negativos, como demonstrado na Figura 7. Entretanto, esses casos não eram constantes, apresentando apenas em momentos de oscilação da rede, causada por um retardamento da detecção.

Figura 7 - Caso de Falso negativo.

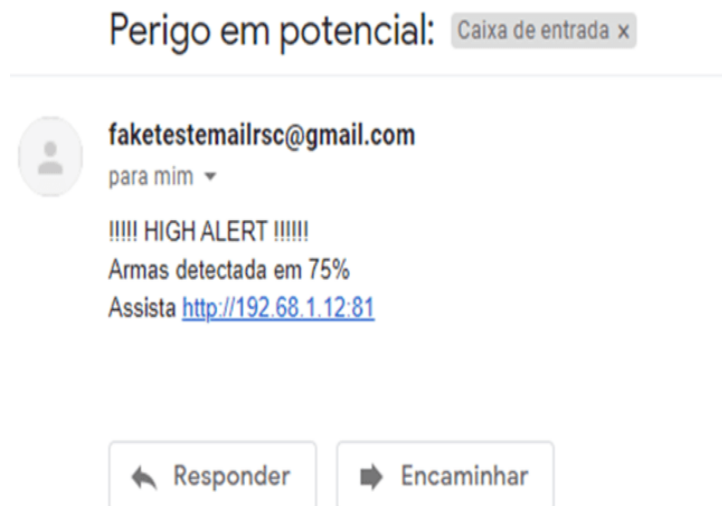


Fonte: Autor (2022)

Com exceção dos casos dos verdadeiros negativos e os casos com detecção abaixo de 45%, em todos os outros casos, foram enviados alertas aos E-mails cadastrados, para que a acurácia pudesse ser reanalisada pelo usuário final ilustrado na Figura 8.

Como demonstrado na Tabela 1, as análises da performance da rede para cada objeto testado nos 480 cenários avaliados, obtiveram resultados que tornaram possível criação de uma matriz confusão da rede, com objetivo de saber as métricas de avaliação do modelo.

Figura 8 - Alerta enviado ao e-mail.



Fonte: Autor (2022)

Tabela 1 – Matriz Confusão do modelo.

		Classe Prevista	
		Positivos	Negativos
Classe Real	Verdadeiro	145	15
	Falso	41	279

Fonte: Autor

As métricas utilizadas para medir a taxa de acurácia (1) e a precisão (2) do modelo, foram calculados utilizando a seguinte operação, dada por Nogare (2020):

- TP – Verdadeiro Positivos
- FP – Falsos Positivos
- TN – Verdadeiro Negativo
- FN – Falso Negativo

1. Acurácia - $TP + TN \div (TP + TN + FP + FN)$

2. Precisão - $TP \div (TP + FP)$

A rede neural *MobileNet* implementada, como demonstrado na matriz confusão, apresentou a acurácia de 57% dos casos investigados, e 90% de precisão, com a perda menor que nos 8% e com 200.000 iterações nos treinos. O nível de acurácia medido consiste em um valor impreciso, em consequência de que os números de objetos negativos, são maiores que os objetos verdadeiros positivos.

5 CONSIDERAÇÕES FINAIS

Ao longo deste projeto, buscamos o desenvolvimento na área de segurança de um sistema de visão computacional implementado sobre sistemas embarcados. Tendo em vista que os sistemas

tradicionais de vigilâncias por imagem, com a utilização de câmeras inteligentes, vêm se popularizando no mercado (BRAMBERGER; BRUNNER; RINNER, 2004).

Para atingir os objetivos foi realizada a implementação sobre os dispositivos a conexão socket TCP que permitiu a transferência de imagens entre os dispositivos ESP32 e Raspberry Pi. Foi utilizada uma rede pré-treinada retirando a necessidade de treinar a rede do zero, esse pré-treinamento foi executado na plataforma em nuvem *Google Collab*, que acelerou o processo de treinamento. Outra técnica utilizada foi o uso de cruzamento de informações, que com sucesso aumentou o nível de detecção da rede, permitindo vários dispositivos colocados em ângulos diferentes dar um veredicto de uma única cena.

Os estudos e implementações realizados neste trabalho, permitem a criação de dispositivos de segurança que podem ser colocados em locais fechados e abertos. Posteriormente também em aplicações derivadas para áreas diferentes, como na agricultura, saúde e IoT.

Como trabalhos futuros, pode-se sugerir a investigação da rede SqueezeNet que torna a implementação de rede neural dentro dos dispositivos ESP32, o que tornaria a latência de resposta mais imediata. Além disso, o uso de outras arquiteturas de dispositivos como FPGAs pode aumentar a flexibilidade do sistema e diminuir o tamanho das redes convolucionais implementadas.

REFERÊNCIAS

- APPEN. **What is Training Data?** Sydney. 2020. Disponível em: <https://appen.com/blog/training-data/>. Acesso em: 05 out. 2021. não paginada.
- ALVES, Gisely. Entendendo Redes Convolucionais(CNNs). **neurônio.ai**. [S. l.] . 2018. Disponível em: <https://medium.com/neuronio-br/entendendo-redes-convolucionais-cnns-d10359f21184>. Acesso em: 10/10/2021. não paginada.
- BLAIR, J. P, SCHWEIT, K. W. **A Study of Active Shooter Incidents, 2000 - 2013**. Texas State University and Federal Bureau of Investigation, U.S. Department of Justice, Washington D.C. p. 5, 2014. Disponível em: . Acesso em: 06 mar. 2021.
- BRAMBERGER, Michael et al. Real-time video analysis on an embedded smart camera for traffic surveillance. In: **Proceedings. RTAS 2004. 10th IEEE Real-Time and Embedded Technology and Applications Symposium, 2004**. IEEE, 2004. p. 174-181.
- FERREIRA, Vitor Finotti. **Towards FPGA-embedded CNNs: Network quantization and HDL infrastructure for bringing CNNs into FPGAs**. Orientador: Dr. Bruno de Carvalho Albertini. 2021. 106 p. Dissertação (Master degree on Computer Engineering) - Escola Politécnica da Universidade de São Paulo, São Paulo, p. 34, 2021.
- Zeiki**, Barra do Bugres, v. 3, n. 2, p. 27-40, (2022).

KANAMI, Pratik; PADOLE, Mamta. **Deep learning to detect skin cancer using google colab. International Journal of Engineering and Advanced Technology Regular Issue**, v. 8, n. 6, p. 2176-2183, 2019.

KATARIA, Jatin. Carnivores Image Classification using Google Colab. **Analytics Vighya**. [S. l.]. 2020. Disponível em: <https://medium.com/analytics-vidhya/carnivores-imageclassification-using-google-colab-add14f7c3e4f>. Acesso em: 27 out. 2021. não paginada.

LUKETINA, Jelena et al. A survey of reinforcement learning informed by natural language. **arXiv preprint arXiv:1906.03926**, 2019.

MACHADO, Arlindo. A cultura da vigilância. [S. l.], 1991. Disponível em: <https://artepensamento.com.br/item/a-cultura-da-vigilancia/>. Acesso em: 16 out. 2021. não paginada.

MADEIRO, Carlos. **Mortes por arma de fogo no país atingem o maior percentual desde 1980**. Maceió, 2019. Disponível em: <https://noticias.uol.com.br/cotidiano/ultimasnoticias/2019/06/05/mortes-por-arma-de-fogo-no-pais-atingem-maior-percentual-desde1980.html>. Acesso em: 16 out. 2021. não paginada

MILANO, Danilo de; HONORATO, Luciano Barrozo. **Visão Computacional**. p. 1, 2014 Universidade Estadual de Campinas (UNICAMP), Disponível em: <https://docplayer.com.br/3058305-Visao-computacional-danilo-de-milano-luciano-barrozohonorato-unicamp-universidade-estadual-de-campinas-ft-faculdade-de-tecnologia.html>. Acesso em: 25 out. 2021.

MOREIRA, Eliana Alves; CARRIEL, Gerson Nunho. **Paralelismo em Nível de Thread**. p. 1, [entre 2015 a 2020]. Disponível em: <https://www.ic.unicamp.br/~ducatte/mo401/1s2012/T2/G01-120437-123789-134097-t2.pdf> Acesso em: 02 nov. 2021.

NOGARE, Diego. **Performance de Machine Learning – Matriz de Confusão**. [S. l.]. 2020. Disponível em: <http://diegonogare.net/2020/04/performance-de-machine-learning-matriz-deconfusao/>. Acesso em: 08 nov. 2021. não paginada.

OSÓRIO, Fernando S., BITTENCOURT, João R. Sistemas Inteligentes baseados em Redes Neurais Artificiais aplicados ao Processamento de Imagens. **I WORKSHOP DE INTELIGÊNCIA ARTIFICIAL UNISC – Universidade de Santa Cruz do Sul** Departamento de Informática- Junho ,p. 18, 2000. Disponível em: <https://docplayer.com.br/3402894-Sistemas-inteligentes-baseados-em-redes-neuraisartificiais-aplicados-ao-processamento-de-imagens.html>. Acesso em: 29 out. 2021.

PACHECO, André GC. Classificação de espécies de peixe utilizando redes neurais convolucionais. **arXiv preprint arXiv:1905.03642**, p. 1-6, 2016.

Realengo, Janaúba e outros: episódios de ataques em escolas no Brasil. **G1**, [S. l.], Não paginada, 13 mar. 2019. Disponível em: <https://g1.globo.com/sp/mogi-das->

Zeiki, Barra do Bugres, v. 3, n. 2, p. 27-40, (2022).

cruzessuzano/noticia/2019/03/13/episodios-de-ataques-em-escolas-no-brasil.ghtml. Acesso em: 28 out. 2021. não paginada

RUDEK, Marcelo, COELHO, Leandro dos Santos, CANGIHLIERI JUNIOR, Osiris, Visão Computacional Aplicada a Sistemas Produtivos: Fundamentos e Estudo de Caso, **XXI 54 Encontro Nacional de Engenharia de Produção** - 2001, Salvador, p. 1, 2001.

SARKER, MO Faruque; WASHINGTON, Sam. Learning Python Network Programming. **Packt Publishing Ltd**, p. 178, 2015.

SILVA, IN da; SPATTI, Danilo Hernane; FLAUZINO, Rogério Andrade. **Redes neurais artificiais para engenharia e ciências aplicadas**. São Paulo: Artliber, v. 23, n. 5, p. 33-111, 2010.

TALAVIYA, Tanha et al. Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides. **Artificial Intelligence in Agriculture**, 2020. Acesso em: 28 out. 2021. não paginada

Recebido em: novembro de 2021.

Aceito em: fevereiro de 2022.