LowCost SIV - Sistema Inteligente de Vigilância de baixo custo desenvolvido com ESP32

Luan Storh Gid
IFPR - Instituto Federal do Paraná
Quedas do Iguaçu, Brasil
luanstohrqui@gmail.com

João Paulo Ganhor
IFPR - Instituto Federal do Paraná
Quedas do Iguaçu, Brasil
joao.ganhor@ifpr.edu.br

Odair Moreira de Souza
IFPR - Instituto Federal do Paraná
Quedas do Iguaçu, Brasil
odair.desouza@ifpr.edu.br

Abstract — In recent years, Brazil has increased digital inclusion, in 2018, 70% of the population had devices to connect to the Internet. However, there is an exception in this inclusion, mainly in relation to the use of electronic security systems, even Brazil being considered a country with little security and high rates of robberies. Thus, the low adherence to these systems is due to their high values. Therefore, this work is based on the development of a low-cost intelligent surveillance system controlled by a Mobile application. The development of this system is done with ESP32 microcontrollers with the implementation of artificial intelligence algorithms made on the Tensor Flow platform.

Keywords — IoT; ESP32; TensorFlow; Electronic Security.

Resumo — Nos últimos anos o Brasil vem possibilitando cada vez mais a inclusão digital, por exemplo, em 2018, 70% da população possuía aparelhos para se conectar na Internet. Contudo há uma exceção nessa inclusão principalmente em relação a utilização de sistemas de segurança eletrônicos, mesmo com o Brasil sendo considerado um país com pouca segurança e altos índices de assaltos. Dessa maneira, a baixa adesão a esses sistemas se deve pelo fato de seus altos valores. Por isso, este trabalho baseia-se no desenvolvimento de um sistema inteligente de vigilância de baixo custo controlado por um aplicativo Mobile. O desenvolvimento desse sistema é feito com microcontroladores ESP32 com a implementação de algoritmos de inteligência artificial feitos na plataforma Tensor Flow.

Palavras-chave — Internet das Coisas; ESP32; Tensor Flow; Segurança eletrônica.

I. INTRODUÇÃO

Nos últimos anos, o Brasil vem passando por uma inclusão tecnológica e digital refletida pelo aumento de pessoas conectadas à Internet, dando destaque às parcelas mais pobres da população. Como Moura e Camargo [1] mostram, o acesso à Internet nas classes D e E passou de 13% em 2010 para 48% em 2018, ainda nesse ano 70% dos brasileiros estavam conectados à Internet.

Contudo, apesar dos números promissores na inclusão digital, aproximadamente três milhões de domicílios possuem algum tipo de sistema de segurança eletrônica no Brasil, com base na Associação Brasileira das Empresas de Sistemas Eletrônicos de Segurança (ABESE)[1]. Esse número é considerado baixo em contraste ao total de domicílios no Brasil, aproximadamente 74 milhões, segundo a Pesquisa Nacional de Amostras de Domicílios Contínua de 2019.

A baixa utilização dos sistemas de segurança eletrônica no Brasil não se dá por pouca relevância ou baixa necessidade no contexto brasileiro, já que o país encontra-se no pior ranking no quesito segurança entre os membros da Organização para a Cooperação e Desenvolvimento Econômico com base no OECD Better Life Index[2]. Então por que os sistemas de segurança são tão pouco empregados no Brasil? Uma das respostas são os altos custos desses sistemas. Isso impede que muitas pessoas tenham acesso a esses equipamentos, principalmente aquelas que vivem em zonas com alta criminalidade.

Quando se compara a renda média do brasileiro com os preços de câmeras de segurança fica evidente a razão da baixa utilização desses sistemas. Segundo Neri [2] a renda média do brasileiro é de R$995,00 no primeiro trimestre de 2021, em contraste, o preço de 2 câmeras de vigilância WiFi Intelbras[3], é de R$559,90 em setembro de 2021.

[1] **ABASE:** Disponível em: abese.org.br/. Acesso em: 16 set. 2021
[2] **OECD:** Disponível em: www.oecdbetterlifeindex.org/pt/quesitos/safety-pt/. Acesso em: 16 set. 2021
[3] **Intelbras:** Disponível em: loja.intelbras.com.br/camera-wifi-fullhd-im3-duo/p. Acesso em: 16 set. 2021

Baseando-se em Doneda [3] “Ainda que pese para o Brasil não ser um centro de inovação em tecnologia da informação e comunicações, temos um mercado relevante e uma sociedade com capacidade de elaboração crítica em face da adoção de novas tecnologias. Assim, para continuarmos com essa elaboração crítica temos que dar boas vindas a uma das tendências mais importantes da tecnologia da informação, a Internet das Coisas”.

A Internet das Coisas (IoT) vem revolucionando o cotidiano por conta de novas tecnologias que permitem a utilização de microcontroladores pequenos, potentes, com baixo gasto energético e acima de tudo com valores proporcionais aos das “coisas” que se quer conectar [4]. A IoT pode ser notada em vários setores da vida social e economia como agricultura, trânsito, saúde, produção industrial [5] e nos sistemas de segurança eletrônica.

Assim, espera-se por meio deste trabalho contribuir na resolução do problema dos altos custos dos sistemas de vigilância no Brasil, por meio da utilização de microcontroladores de hardware e software livres. O trabalho baseia-se na implementação de um sistema inteligente de vigilância. Para o desenvolvimento das câmeras, foram utilizadas placas ESP32 CAM que realizam a detecção de pessoas e a captura de imagens. As câmeras conectam-se a uma outra placa, o ESP-01, que armazena as imagens e gerencia o envio de notificações para uma aplicação Mobile desenvolvida em Android.

A criação do aplicativo deve-se pelo fato de amenizar o processamento das placas aumentando o desempenho do sistema, além disso, atualmente o Android é o sistema operacional mais presente no Brasil e é por meio deste que a maior parte da população acessa a Internet nos dias atuais. Dessa maneira o sistema requer um estrutura básica para sua implementação, a qual, além de um celular com Android, precisa de uma rede WiFi instalada na residência.

É importante ressaltar a utilização da plataforma Tensor Flow para a implementação de algoritmos de inteligência artificial, que permite a detecção de pessoas e avisa quem tiver o aplicativo de controle.

Nesse contexto, o objetivo geral é possibilitar uma alternativa de segurança de patrimônio de baixo custo e eficiente, por meio de microcontroladores de hardware e software livre, com aplicação de algoritmos de Inteligência Artificial e controle por um app mobile.

Os objetivos específicos são: i) Possibilitar aquisição de imagens para possível utilização como prova em alguma situação de invasão; ii) Manter o usuário do sistema atualizado quanto a circulação de pessoas em seu patrimônio; iii) Possuir um valor acessível para a realidade econômica brasileira; e, iv) Utilizar hardware e software livre no desenvolvimento do sistema inteligente de vigilância.

II. MATERIAIS UTILIZADOS

A. *Machine Learning em microcontroladores*

De acordo com Ada Lovelace, “quando os computadores programáveis foram desenvolvidos pela primeira vez, as pessoas já se perguntavam se essas máquinas poderiam se tornar inteligentes, mais de cem anos antes de uma ser construída”.

O Machine Learning costuma ser utilizado com o tipo de aprendizado Supervisionado, onde é necessário criar e treinar uma rede neural já mostrando o caminho. Esse modelo de inteligência artificial é relevante para algoritmos de classificação, regressão e previsão utilizando uma base de dados [6]. Não obstante, apesar do assunto Machine Learning ser conhecido pelo alto custo computacional, atualmente é possível levar essa tecnologia para os microcontroladores utilizando o Tiny ML. Segundo a organização TinyML.org[4] (traduzido), o Tiny Machine Learning é amplamente definido como um campo de tecnologias de aplicativos de aprendizado de máquina, incluindo circuitos integrados dedicados.

Dessa forma, algoritmos de inteligência artificial podem ser implementados em hardware baratos com conexões de Internet simples. Com base em Tensor Flow [7] bilhões de dispositivos presentes em nossas vidas podem ser otimizados com Inteligência Artificial por meio do Machine Learning para microcontroladores.

B. *TensorFlow*

Os sistemas de Inteligência Artificial podem ser considerados complexos de implementar, contudo, a plataforma Tensor Flow vem auxiliando os desenvolvedores a criar modelos e treiná-los de forma otimizada e simples.

O Tensor Flow é uma plataforma completa de código aberto para machine learning, tem um ecossistema abrangente e flexível de ferramentas, bibliotecas e recursos da comunidade que permite aos pesquisadores utilizar ML de última geração e aos desenvolvedores criar e implantar aplicativos com tecnologia de ML [8]. Exemplificando, conforme Google Developers: "Não esperamos que você seja um especialista em ML; em vez disso, construímos nossa plataforma para que você não precise ser” (traduzido) [9].

C. *ESP32*

O ESP32 é um chip dual core de 32 bits, sendo um dos chips para circuitos integrados de maior custo benefício e desempenho. Várias placas que utilizam o chip ESP32 vêm com conexão híbrida WiFi e Bluetooth integrada possibilitando aplicações de Internet das Coisas. Vale ressaltar que placas como as ESP32 CAM e ESP32 WROOM possuem memória PSRAM de 512 KB, suficiente

[4] ***TinyML.org***: Disponível em https://www.tinyml.org/about/. Acesso em 16 set. 2021.

para captar imagens com até 2 milhões de pixels e executar algoritmos com Inteligência Artificial.

III. METODOLOGIA

O sistema de vigilância será constituído de 3 partes: (i) a parte fundamental são as câmeras desenvolvidas com placas ESP32 CAM; (ii) a central de armazenamento que está sendo implementada com uma placa ESP32 WROOM DEVKIT juntamente com um módulo leitor de cartão de memória; e, (iii) aplicativo (app) mobile de controle, o qual será desenvolvido em Android por meio da plataforma Android Studio. Na Fig. 1 apresenta-se o esquema de comunicação entre os componentes do sistema.

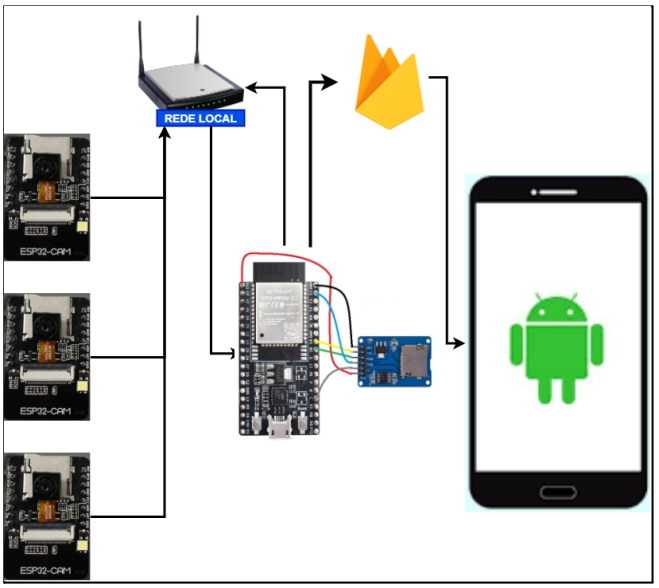

Fig. 1. Esquema da infraestrutura de comunicação do sistema.

A. *A implementação do ESP32 CAM*

As câmeras detectam a presença de pessoas para posteriormente notificar o aplicativo, para isso foi utilizado o modelo de detecção de pessoas presente como exemplo da biblioteca TensorFlowLite para ESP32. Esse modelo funciona com imagens de 1 byte de escalas de cinza e com resolução de 96x96px e retorna dois valores que são respectivamente a chance de ter ou não alguém na imagem. Ressalta-se que as câmeras estarão conectadas na rede local e se comunicam com a central por meio de requisições HTTP, que estão do lado do cliente.

B. *O desenvolvimento da central*

A central atuará como um servidor WEB para as câmeras e será responsável por receber as imagens, convertê-las para JPEG e armazená-las na memória. Na sequência as imagens serão codificadas em uma lista de caracteres de base 64 para serem enviadas para um banco de dados de tempo real no Firebase, e assim poderem ser acessadas no app. Em caso de detecção de presença em uma das câmeras, a central enviará um email para o usuário do sistema utilizando a plataforma If This Then That (IFTTT), e o Firebase desencadeará um push notification para o app Android.

C. *O aplicativo de controle*

O aplicativo deverá possibilitar o envio do nome e senha da rede local e um login de email e senha. Essas informações serão enviadas para a central que utilizará para se conectar na rede local e no Firebase respectivamente. Outro fator será a visualização das imagens armazenadas na central. Para melhor organização, o app contará com um sistema de diretórios listados por data e número da câmera. Quando a visualização de uma imagem específica for necessária, será passado o nome do arquivo com data e número da câmera para a central por meio Firebase e ainda por meio deste a central enviará a respectiva imagem.

O aplicativo também deverá possibilitar a customização dos modos de funcionamento das câmeras, os quais poderão ser captação de imagens por detecção de pessoas ou por detecção de movimento, além disso, haverá a definição dos horários de funcionamento. O app também possibilitará o recebimento de push notifications as quais poderão ser configuradas da seguinte maneira: sem envio de notificação; envio de notificação em caso de detecção de pessoa; e, envio de notificação em caso de detecção de movimento.

Na Fig. 2 apresenta-se o diagrama de atividades do sistema no modo de funcionamento de detecção de presença. São mostradas as atividades realizadas pela câmara, central de armazenamento e pelo app quando a presença de uma pessoa é detectada.

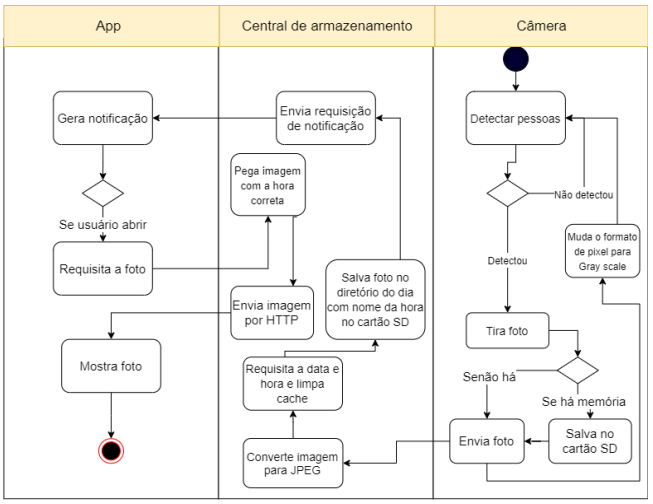

Fig. 2. Diagrama de atividades do monitoramento do sistema.

IV. RESULTADOS E DISCUSSÕES

O primeiro resultado foi a obtenção de um algoritmo funcional de detecção de pessoas utilizando o modelo treinado que está presente na biblioteca do Tensor Flow Lite para microcontroladores, implementado no ESP32 CAM. A

Fig. 3 mostra uma imagem feita de caracteres mostrada no monitor serial da IDE Arduino, nela há duas pontuações: *person score*, a qual é a probabilidade de haver alguém da imagem; e, *no person score*, que se refere à probabilidade de não ter ninguém na imagem. Como na Figura Fig. 3 o valor de *person score* é maior que *no person score*, significa que há alguém.

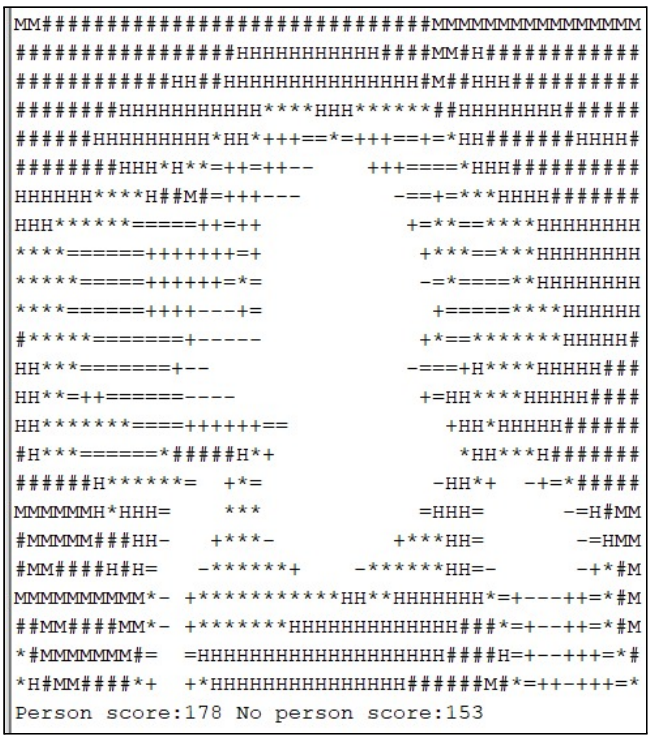

Fig. 3. Caso de presença detectada pela inteligência artificial.

Desenvolveu-se um código para codificar a imagem em caracteres de base 64 e enviá-la para um banco de dados em tempo real no Firebase.

V. CONSIDERAÇÕES FINAIS

Notou-se nos testes preliminares, o funcionamento adequado do sistema inteligente de vigilância em desenvolvimento com base na arquitetura de hardware livre por meio da placa ESP32, que possui uma placa de rede integrada, possibilitando utilizar o sistema mesmo em desenvolvimento com ajustes dos parâmetros para receber requisições HTTP, desse modo permite realizar a otimização nos testes. Apesar do bom processamento do ESP32, observou-se a falta de memória para realizar codificação da imagem em JPEG e processar o algoritmo de detecção em conjunto. Outra dificuldade encontrada foi em relação ao armazenamento de vídeo, o qual funciona com um taxa de 1,5 imagens por segundos, por isso essa funcionalidade foi descartada.

No início do desenvolvimento esperava-se repassar a imagens da placa central para o app por meio de requisições HTTP, mas ao pesquisar sobre as funcionalidades da plataforma Firebase foi definido a utilização de uma tabela compartilhada.

REFERÊNCIAS

[1] L. Moura and G. Camargo, “Impacto econômico e social do Android no Brasi,” Bain & Company, Inc. São Paulo: 2019.

[2] M. Neri, “Bem-estar trabalhista, felicidade e pandemia”. Praia do Botafogo: FGV Social, 2021. Sumário Executivo.

[3] D. Doneda, Prefácio. In: MAGRANI, Eduard. “A Internet das Coisas”. Rio de Janeiro: FGV Editora, 2018, E-Book, p. 20.

[4] S. Oliveira, “Internet das Coisas com ESP8266, ARDUINO E RASPBERRY PI”. São Paulo: Novatec Editora Ltda., 2017. *E-Book*, p. 55.

[5] E. Magrani. “A Internet das Coisas”. Rio de Janeiro: FGV Editora, 2018, *E-Book*, p. 20.

[6] F. Feltrin, “Inteligência Artificial com Python”. [S.I]: 2020, 182p, E-book.

[7] TensorFlow, “TensorFlow para microcontroladores”. atualizado em: 2 mai. 2021, disponível em: https://www.tensorflow.org/lite/microcontrollers?hl=es-419. Acesso em: 21 ago. 2021.

[8] TensorFlow, “Uma plataforma completa de código aberto para machine learning”. [S.I], [S/D], disponível em: https://www.tensorflow.org/, acesso em: 21 ago. 2021.

[9] Google Developers, “Developer Keynote (Google I/O '21) - American Sign Language”. [S.I] 2021, disponível em: https://www.youtube.com/watch?v=D_mVOAXcrtc&t=2417s, acesso em 25 ago. 2021.