PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA
ENGENHARIA DE COMPUTAÇÃO

LUCAS MEDEIROS RIBEIRO

DETECÇÃO DE DISPARO DE ARMA DE FOGO UTILIZANDO REDES NEURAIS CONVOLUCIONAIS

Porto Alegre
2022

LUCAS MEDEIROS RIBEIRO

DETECÇÃO DE DISPARO DE ARMA DE FOGO UTILIZANDO REDES NEURAIS CONVOLUCIONAIS

Trabalho de conclusão de curso de graduação apresentado na Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul, como requisito parcial para obtenção do grau de Engenheiro de Computação.

Orientador: Prof. Dênis Fernandes

Porto Alegre

2022

LUCAS MEDEIROS RIBEIRO

DETECÇÃO DE DISPARO DE ARMA DE FOGO UTILIZANDO REDES NEURAIS CONVOLUCIONAIS

Trabalho de conclusão de curso de graduação apresentado na Escola politécnica da Pontifícia Universidade Católica do Rio Grande do Sul, como requisito parcial para obtenção do grau de Engenheiro de Computação.

Aprovada em _____ de _____ de _____.

BANCA EXAMINADORA:

Nome do Professor

_____

Nome do Professor

_____

Nome do Professor

_____

Dedico este trabalho aos meus
pais, amigos e familiares que mesmo nos
meus momentos mais conturbados
sempre me incentivaram a continuar.

RESUMO

Sistemas para detecção de disparo de arma de fogo estão se tornando cada dia mais sofisticados e eficazes no combate à violência. Todavia, esta tecnologia é importada e possui um custo elevado, o que a torna pouco aderida pelas autoridades de segurança. Neste contexto, o presente trabalho pretende apresentar uma solução para auxiliar na detecção de disparos de armas de fogo utilizando redes neurais convolucionais (CNNs). Assim, serão apresentados conceitos sobre a classificação de áudios, meios de representar visualmente um áudio, o que são redes neurais convolucionais, uso do Keras e Tensorflow para geração e treinamento de redes neurais convolucionais. Posteriormente é descrita a solução proposta desde o processamento das amostras de áudio, o uso da biblioteca Librosa para geração dos espectrogramas, o uso dos modelos InceptionV3, ResNet50 e VGG16 para classificar espectrogramas e a geração de modelos refinados. As métricas para avaliação de resultado consideram acurácia, *loss*, precisão, *recall*, *f-score* e tabela de confusão. Os resultados adquiridos mostram uma acurácia acima dos 96,15% durante o treinamento no pior dos casos. No entanto, a validação do modelo pelo do *dataset* de teste resultou em uma precisão de 84% e *recall* de 73% no melhor dos casos. Percebe-se então, que para a classificação de áudio a solução proposta há pontos de melhoria no processamento das amostras para atingir um resultado melhor nos modelos usados.

Palavras-chave: Processamento de áudio. Librosa. Espectrograma. Redes neurais convolucionais. Keras. TensorFlow.

ABSTRACT

Systems to detect gunfire are becoming increasingly sophisticated and effective in combating violence. However, this technology is imported and has a high cost, which makes it little adhered to by security authorities. In this context, the present work intends to present a solution to assist in the detection of firearm shootings using convolutional neural networks (CNNs). Thus, concepts about audio classification, means of visually representing audio, what convolutional neural networks are, the use of Keras and Tensorflow for generating and training convolutional neural networks will be presented. Subsequently the proposed solution is described from the processing of the audio samples, the use of the library Librosa for spectrogram generation, the use of the InceptionV3, ResNet50 and VGG16 models to classify spectrograms, and the generation of refined models. The metrics for result evaluation consider accuracy, loss, precision, recall, f-score, and confusion table. The acquired results show an accuracy above 96.15% during worst case training. However, validation of the model by the test dataset resulted in a best-case accuracy of 84% and recall of 73%. It can be seen then, that for audio classification the proposed solution has points for improvement in the processing of the samples to achieve a better result in the models used.

Keywords: Audio processing. Librosa. Spectrogram. Convolutional neural networks. Keras. TensorFlow.

LISTA DE SIGLAS

ADC – Analog-to-Digital Converter

ADSR – Attack Decay Sustain Release

API – Application Programming Interface

DAC – Digital-to-Analog Converter

FFT – Fast Fourier Transform

MFCC – Mel-frequency cepstral coefficients

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Segurança pública é, sem dúvida, uma preocupação geral da sociedade. Como tal, a tecnologia para detecção de disparos de arma de fogo é um campo de pesquisa cada vez mais essencial. Uma solução existente no mercado é o ShotSpotter que identifica e localiza ocorrências de disparos em uma grande escala, combinando a análise automatizada de dados de áudio em uma cidade com a análise humana para diferenciação de disparos e outros ruídos. Embora esta tecnologia tenha aumentado o sucesso no reconhecimento de disparos de armas de fogo, essa solução tem um custo elevado para operar e manter. Deste modo, o uso de técnicas de *Deep Learning* para o reconhecimento e a classificação de disparos de arma de fogo pode ser uma alternativa para este problema.

Atualmente, há uma vasta quantidade de pesquisas e estudos sobre o uso de redes neurais convolucionais na classificação de sons, onde os trabalhos compararam meios alternativos para categorização de sons. No entanto, ainda há margem para melhoria na aplicação de redes neurais convolucionais para detecção de disparos de arma de fogo.

A abordagem deste trabalho pretende a utilização de espectrogramas para treinar uma rede neural convolucional na detecção de disparos de arma de fogo. A principal vantagem desta abordagem é o uso apenas de imagens para reconhecimento dos áudios. Além disso, comparar três modelos de redes neurais convolucionais para elencar qual possui o melhor desempenho, mas também identificar aquela que possui o menor número de previsões de falso-positivo.

1.1 MOTIVAÇÃO

O motivo pela escolha deste trabalho é particularmente aprofundar e contribuir intelectualmente com os estudos na área de processamento de sinais áudio e respectivamente *Machine Learning*. No entanto, também é necessário implementar uma solução simplificada ao sistema ShotSpotter, pois o objetivo é classificar se uma amostra de áudio é ou não é um disparo de arma de fogo. Por estas razões, o intuito deste trabalho é implementar uma solução utilizando redes neurais convolucionais para classificação e avaliar o quão confiável são os resultados obtidos.

1.2 OBJETIVO GERAL

O objetivo geral deste trabalho é apresentar uma solução para detecção de disparos de armas de fogo por meio de redes neurais convolucionais utilizando os modelos InceptionV3, ResNet50 e VGG16.

1.2.1 Objetivos Específicos

a) Compreender o funcionamento de redes neurais e sua aplicação em classificar áudios;
b) Treinar uma rede neural para classificar espectrogramas de áudios, identificando a qual classe pertencem;
c) Avaliar os resultados obtidos para mapear pontos de melhorias em trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção refere-se ao embasamento teórico dos assuntos relevantes para o estudo, assegurando a credibilidade e legitimidade dele. Nas subseções a seguir será tratado sobre: o que são sons impulsivos; como representar áudios visualmente e funcionamento de redes neurais convolucionais na classificação de imagens.

2.1 SONS IMPULSIVOS

Sons impulsivos são definidos, como sons de curta duração, com sua amplitude e duração aleatórias, ou mais especificamente, são gerados por uma mudança abrupta na pressão do ar, produzindo uma grande amplitude com uma curta duração. Geralmente essa classe de som acende em milissegundos e sua duração não ultrapassa a casa dos décimos de segundo. Eles são decorrentes de eventos onde há liberação de energia em um curto espaço de tempo, como no caso de batidas de portas, vidros quebrados, explosões, disparos de armas de fogo entre outros (REIS, 2015). Na figura 1 abaixo, é possível perceber como sons impulsivos são representados em função do tempo.

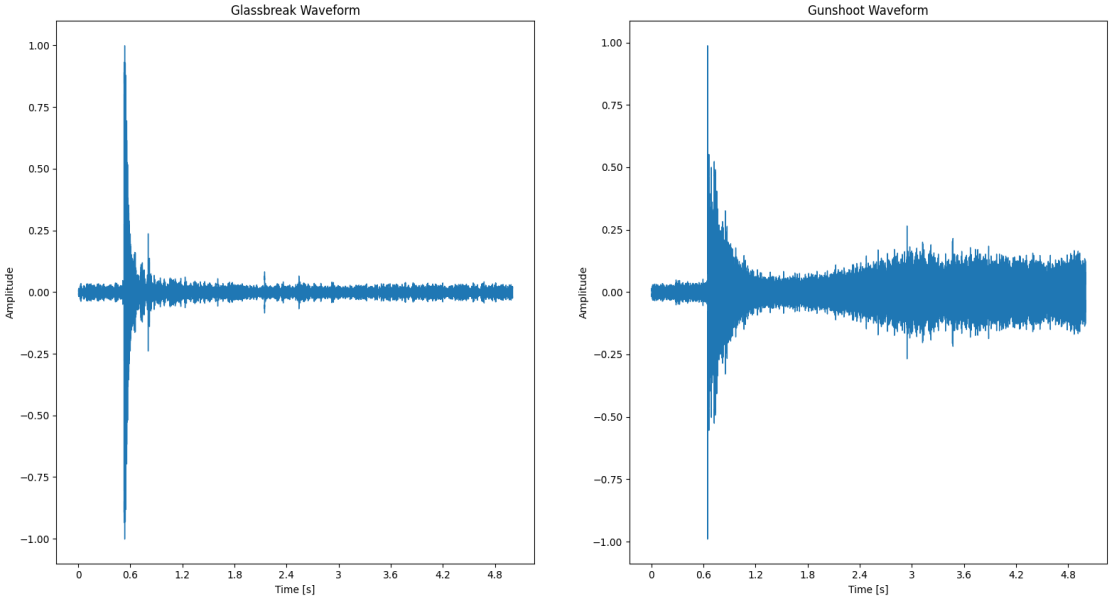


Figura 1 – Sons Impulsivos. (esquerda) Vidro quebrado. (direita) Disparo de arma de fogo.

Fonte: Elaborado pelo autor (2022).

Um conceito empregado para representar um som impulsivo é o ADSR (*Attack*, *Decay*, *Susain*, *Release*) presente no contexto musical. O ADSR é uma das formas comumente utilizadas para destacar as mudanças de amplitude de um som para produzir um timbre característico. A fase do ataque (*Attack*) determina o crescimento e a liberação de potência sonora. O decaimento (*Decay*) é representado por uma pequena perda de potência sonora inicial. A sustentação (*Sustain*) corresponde ao processo pelo meio material que rodeia a fonte de emissão da onda sonora, como, por exemplo, uma caixa de ressonância. Por fim, a vibração sonora perde energia até extinguir, se caracterizando a fase de repouso (*Release*).

Desta forma, um som impulsivo é caracterizado apenas pelas fases do ataque e repouso que ocorrem rapidamente, possuindo duração na ordem de décimos de segundos. As fases de decaimento e sustentação são praticamente inexistentes conforme se observa na figura 2.

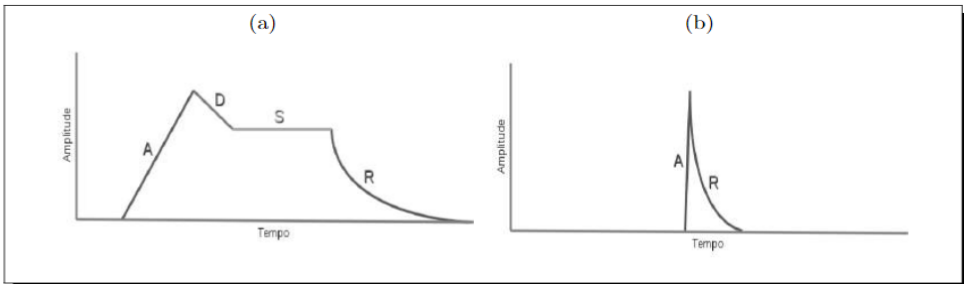


Figura 2 – Representação ADSR. (esquerda) ADSR de um som qualquer; (direita) ADSR de um som impulsivo.

Fonte: REIS, 2015, p. 5.

Portanto, um som impulsivo ideal possui um tempo de ataque curto, onde atinge seu máximo de energia muito rapidamente e não se mantém nesse estado por muito tempo, iniciando seu decaimento instantaneamente, dispersando exponencialmente.

2.2 REPRESENTAÇÃO DE UM ÁUDIO

O som é um fenômeno físico de pressão e descompressão de ar. Ao medir a pressão em um ponto do espaço pode-se representar a forma de um sinal digital de áudio. Os sinais digitais de áudio são normalmente representados através de uma

sequência de valores discretos de amplitude amostrados de um sinal analógico em instantes de tempo espaçados igualmente (MOURA, 2016).

O processo de amostragem é basicamente a discretização de um sinal analógico contínuo para digital, onde cada amostra é representada usando um conjunto finito de valores. A digitalização é realizada por um conversor analógico-digital (ADC) que possui três estágios: filtragem do sinal; amostragem do sinal; quantização das amostras e codificação em um sistema binário. O processo inverso é realizado por um conversor digital-analógico (DAC). A amostragem (discretização temporal) é realizada tomando periodicamente amostras dos valores instantâneos do sinal analógico. A taxa de amostragem é a frequência com que são tomadas as amostras. Por exemplo: A taxa de amostragem padrão de um CD de áudio é de 44.100 Hz.

Em um sinal digital as amostras são quantizadas e armazenadas usando um número determinado de bits para representar cada amostra. A quantidade de bits determina o número de valores possíveis a serem representados. Logo, cada amostra é representada por um valor aproximado. O erro de aproximação causa uma deformação do sinal em relação ao original (analógico), o ruído de quantização. A relação sinal/ruído é computada a partir dos números de bit que normalmente convenciona-se que os valores permitidos na representação da amplitude instantânea de som digital situam-se no intervalo de -1 e 1. Valores que ultrapassam os limites desse intervalo são substituídos resultando no efeito de distorção do sinal. Para evitar saturação, é preciso garantir que a amplitude do sinal não ultrapasse a amplitude máxima permitida.

O Teorema de Nyquist assume que a frequência de amostragem de um sinal analógico deve ser igual ou maior a duas vezes a maior frequência do espectro do sinal. Essa condição precisa ser respeitada para que posteriormente o sinal possa ser reconstituído perfeitamente. Desta forma, usando a taxa de amostragem de um CD, 44.100 Hz, não é possível representar frequências maiores que 22.050 Hz.

2.2.1 Representação Temporal

Esta representação relaciona a intensidade do sinal sonoro no tempo. Na figura 3 temos a visão temporal de uma amostra de um disparo de arma de fogo.

Nota-se que a parte azul representa a intensidade em cada instante de tempo e que neste exemplo a amplitude tem seu pico próximo a 1 segundo. Nesta representação é possível avaliar a magnitude do sinal.

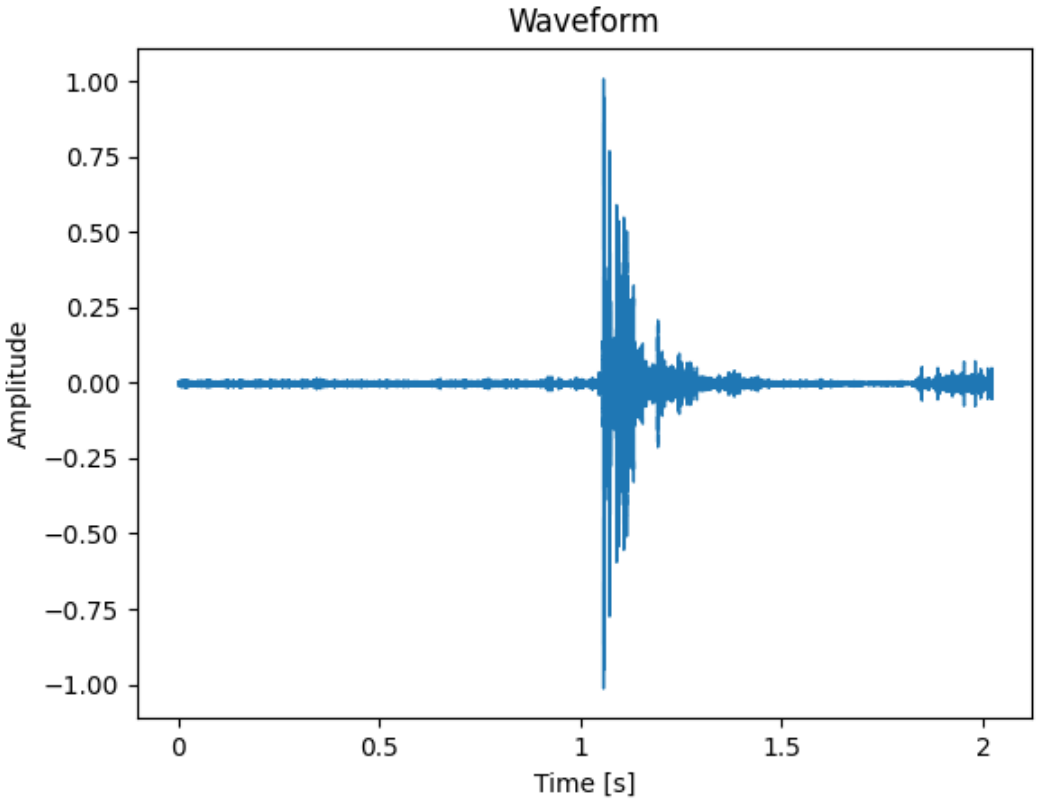


Figura 3 – Representação temporal.

Fonte: Elaborado pelo autor (2022).

2.2.2 Representação Espectral

A representação espectral diferentemente da representação temporal relaciona a intensidade do sinal sonoro na frequência. Esta representação é feita através da conversão do domínio do tempo para frequência a partir da Transformada de Fourier do sinal. Conforme o Teorema de Nyquist[1], assume que a frequência de

[1] O teorema de amostragem de Nyquist-Shannon expõe a relação entre a taxa de amostragem e a frequência do sinal medido. Esse teorema, considera que a taxa de amostragem fs deve ser maior que o dobro da componente de maior frequência que se pretende analisar no sinal medido. Essa frequência na maioria das vezes é chamada de frequência de Nyquist. Além de, ser fundamental no campo da teoria da informação, principalmente na área de telecomunicações e processamento de sinais. Assim sendo, de acordo com esta teoria, amostrar é o processo no qual se converte um sinal em uma sequência numérica (ENGINEER AMBITIOUSLY, 2022; WIKIPÉDIA, 2021).

amostragem de um sinal deve ser no mínimo duas vezes maior que a frequência do espectro, observa-se na figura 4 que um sinal amostrado em 44.100 Hz é representado até 22.050 Hz.

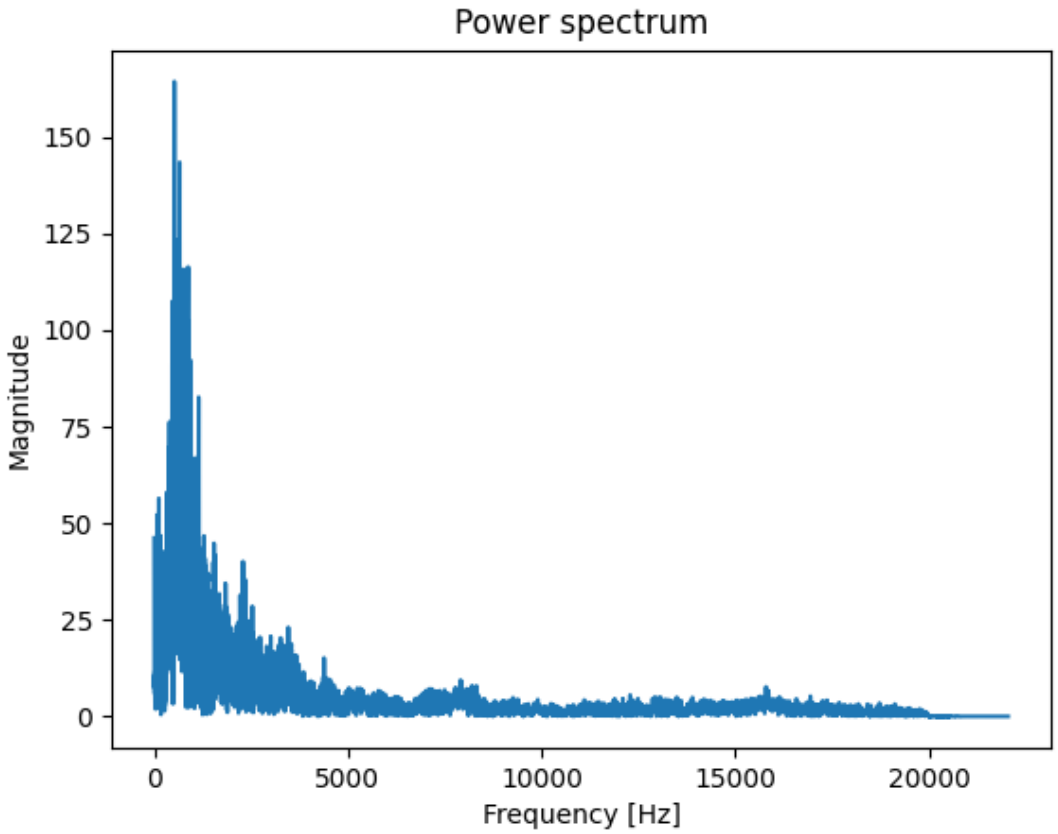


Figura 4 – Representação espectral.

Fonte: Elaborado pelo autor (2022).

2.2.3 Representação Tridimensional

A representação tridimensional, também conhecida como espectrograma, é a relação entre tempo e frequência graficamente. O espectrograma representa a intensidade de uma frequência em relação ao tempo. Através dessa representação é possível a análise dinâmica da densidade espectral de energia, ou seja, é possível avaliar a intensidade da energia dissipada relacionando tempo e frequência. Desta forma, conforme a figura 5 é possível visualizar representação de um disparo de arma de fogo.

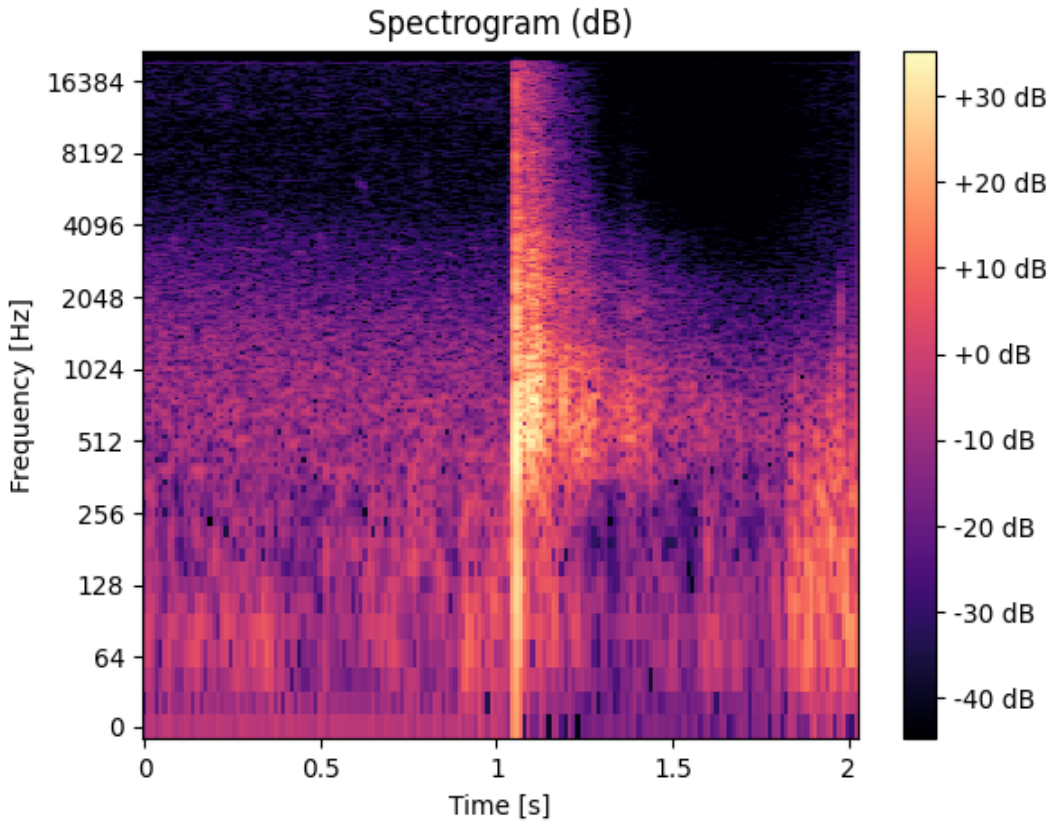


Figura 5 – Representação tridimensional (Espectrograma).

Fonte: Elaborado pelo autor (2022).

Neste exemplo, por se tratar de um som impulsivo no instante 1 segundo vemos uma grande liberação de energia em todas as frequências e logo em seguida a sua dissipação.

2.3 REDES NEURAIS CONVOLUCIONAIS

As redes neurais convolucionais são redes neurais artificiais projetadas para localizar, modelar e prever padrões presentes em dados de entrada, como, por exemplo, imagens coloridas ou vídeos. Elas fazem isso deslizando iterativamente sobre pequenas regiões dos dados de entrada e mapeando quaisquer propriedades inerentes em uma região para uma camada de processo na rede pelo uso de filtros. Os filtros, ou *kernels*, são matrizes tipicamente quadradas em dimensionalidade, sendo multiplicadas por trechos dos quadrados dos dados de entrada para reduzir qualquer característica presente em uma nova representação menor. Observa-se na

figura 6 que os filtros mapeiam as características entre as camadas reduzindo o seu tamanho até chegar na camada de saída da rede.

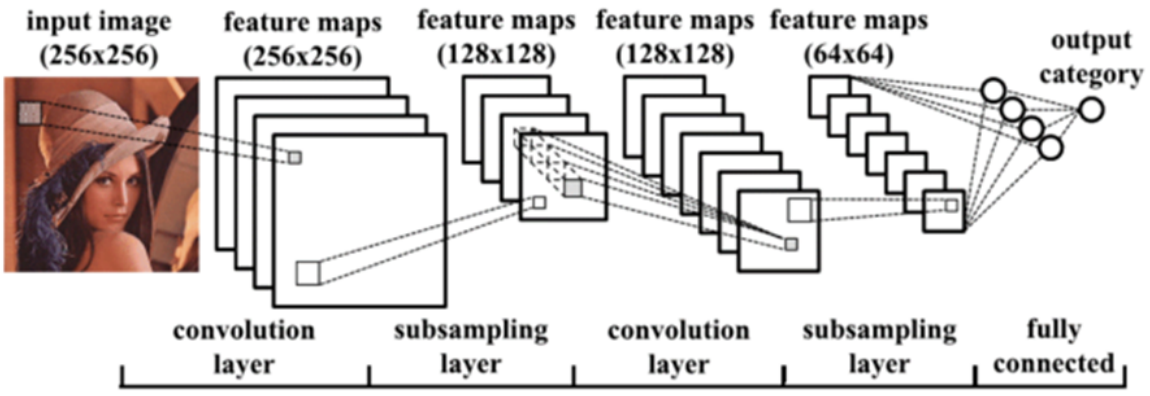


Figura 6 – Rede Neural Convolucional.

Fonte: Data Science Academy (2022).

Este processo de mapear as características dos dados de entrada em um espaço reduzido é repetido diversas vezes até a camada de saída da rede. O que produz um vetor de valores de probabilidade correspondentes às possíveis classes às quais os dados de entrada possam pertencer.

2.4.1 Camadas de uma Rede Neural Convolucional

A arquitetura de uma rede neural convolucional pode ser representada por diversas camadas combinadas entre si para a finalidade de reconhecer padrões. Uma rede neural convolucional pode ser dividida em dois módulos distintos: convolução e classificação.

O módulo de convolução é responsável pela extração de elementos da imagem que descrevem o seu conteúdo. O módulo de classificação é responsável por classificar os dados extraídos durante a etapa de convolução. Isto é, uma rede neural convolucional é composta por diversas camadas intermediárias, sendo cada uma responsável por uma tarefa específica.

As principais camadas presentes em uma rede neural convolucional são convolução, *pooling* e *fully connected*. A camada de convolução é responsável por extrair e mapear o conteúdo da imagem que está sendo processada, transformando toda essa informação em dados. Assim sendo, este processo ocorre por meio da aplicação de pequenos filtros que permitem a obtenção de informações de pequenas partes da imagem. A camada de *pooling* recebe os dados extraídos na etapa de

convolução e simplifica a informação, reduzindo os dados em um único valor. Com relação à camada *fully connected*, responsáveis por classificar os dados extraídos nas camadas anteriores, ou seja, é a transformação de todas as informações colhidas em uma única linha. Observa-se na figura 7, a função das camadas citadas anteriormente demonstrando a conexão entre camadas dentro de uma rede neural convolucional.

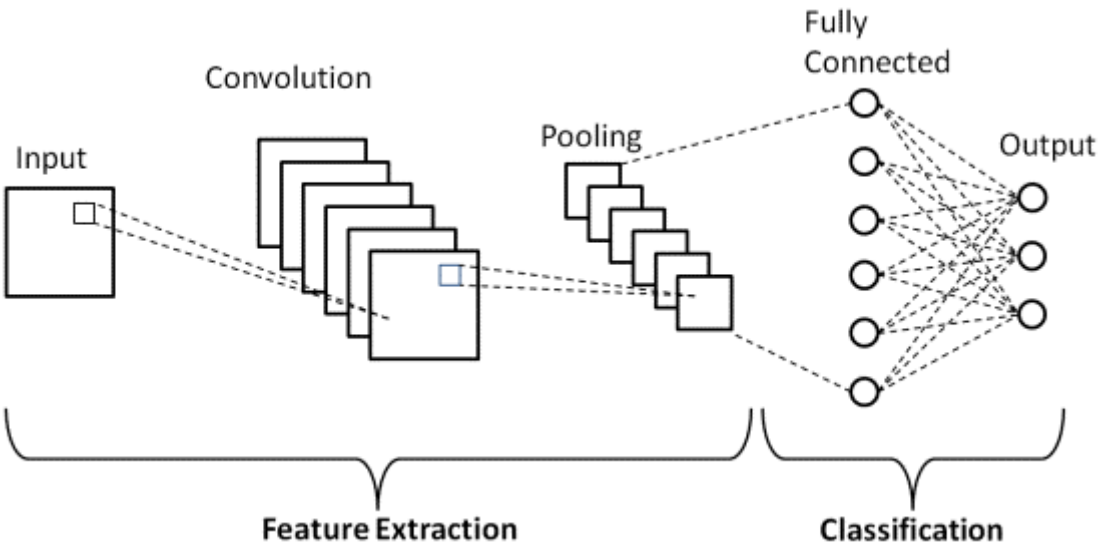


Figura 7 – Principais camadas de uma rede neural convolucional.

Fonte: Gurucharan (2022).

Além das camadas citadas acima, existem uma técnica e uma camada extras fundamentais na composição de uma rede neural convolucional, são elas: *dropout* e de ativação, respectivamente. O *dropout* como o nome diz é desativado aleatoriamente e temporariamente algumas camadas intermediárias para reduzir o *overfitting* da rede. O *overfitting* é quando uma rede decora os dados utilizados no aprendizado e não consegue classificar novos dados. Logo, este cenário representa um modelo de uma rede treinada que não tem a capacidade de generalização e não apresenta bom desempenho. A camada de ativação é responsável pelo aprendizado da rede, bem como sua relação entre as variáveis e quais neurônios são ativados.

2.4 FERRAMENTAS

Para o desenvolvimento do projeto foi utilizado a linguagem de programação Python por conter em seu acervo bibliotecas para processamento de áudio e *Machine Learning*. Desta forma, as bibliotecas e ferramentas escolhidas facilitam o

desenvolvimento da solução. Isto é, já possuem a maioria das funções necessárias permitindo o foco apenas na implementação dos algoritmos de pré-processamento e treinamento e teste dos modelos de redes neurais utilizadas. A solução proposta foi implementada utilizando as seguintes bibliotecas e ferramentas que serão descritas abaixo para facilitar o entendimento.

2.4.1 Google Colab

O Google Colab é um serviço de nuvem gratuito hospedado pelo Google para incentivar pesquisas na área de *Machine Learning*. É uma ferramenta que permite criar e executar códigos na linguagem Python diretamente do navegador, de forma simples e rápida. O Google Colab é hospedado em um serviço baseado no Jupyer notebook que não requer nenhuma configuração para uso, enquanto disponibiliza o acesso livre a recursos computacionais (GOOGLE, 2022).

2.4.2 Keras

O Keras é uma biblioteca de *software* livre que fornece uma API para a biblioteca de manipulação de redes neurais artificiais TensorFlow. Keras contém várias implementações de blocos de construção de rede neural comumente aplicados, como camadas, objetivos, funções de ativação, otimizadores e uma série de ferramentas para facilitar o trabalho com dados de imagem e texto para simplificar a codificação necessária para escrever código de rede neural profunda (CHOLLET, 2015).

2.4.3 Librosa

O Librosa é uma biblioteca em Python que dispõe de ferramentas para análise e manipulação de arquivos de áudio. O foco do Librosa é extrair características de áudio para análise de músicas, no entanto, ela é amplamente usada para os mais variados propósitos, como: processamento de áudio (KNEIPP, 2019; MCFEE *et al.*, 2015).

2.4.4 OpenCV

OpenCV (Open Source Computer Vision) é uma biblioteca de programação construída para fornecer uma API comum para manipulação de imagens em tempo real. Originalmente, desenvolvida pela Intel em 2000, é totalmente livre para uso acadêmico e comercial. Tem como objetivo tornar a visão computacional mais acessível aos desenvolvedores, possuindo mais de 500 funções e pode ser utilizada em diversas linguagens de programação (C++, Python, Ruby, Java). Além de, ser usada para diversos tipos de análise em imagens e vídeos, como a detecção, tracking e reconhecimento facial, edição de fotos e vídeos, detecção e análise de textos, etc (CEDRO TECHNOLOGIES, 2018; WIKIPÉDIA, 2022).

2.4.5 Scikit-learn

O Scikit-learn[2] é uma biblioteca em Python que disponibiliza um conjunto de algoritmos para *Machine Learning* e outras funções estatísticas que visam facilitar a análise do treinamento realizado.

2.4.6 TensorFow

O TensorFlow[3] é uma biblioteca de *software* livre e código aberto para aprendizado de máquina aplicável a uma ampla variedade de tarefas. O TensorFlow

[2] Disponível em: https://scikit-learn.org/stable/. Acesso em: 29 out. 2022.
[3] TENSORFLOW: OPEN SOURCE MAXHINE LEARNING. [S. l.: s. n.], 2016. 1 vídeo (2 min 17 seg). Publicado pelo canal Google. Disponível em: https://www.youtube.com/watch?v=oZikw5k_2FM. Acesso em: 1 nov. 2022.

é um sistema para criação e treinamento de redes neurais para detectar e decifrar padrões e correlações.

2.5 TRABALHOS RELACIONADOS

No desenvolvimento deste trabalho, foi feita uma revisão de artigos relacionados ao objetivo para embasar a abordagem escolhida para solução. Dois artigos relacionados ao uso de redes neurais convolucionais para detecção de disparos de arma de fogo foram usados como base no desenvolvimento da solução proposta.

O artigo “Gunshot Detection Using Convolutional Neural Networks” (BAJZIK; PRINOSIL; KONIAR, 2020) procura analisar o uso de métodos para processamento de imagens na área de reconhecimento de eventos de áudio. Portanto, a abordagem usada é dada a partir da combinação de espectrogramas, coeficientes MFCC e matriz de *self-similarity* em uma única imagem colorida para treinamento de redes neurais convolucionais. Para essa tarefa, três modelos convencionais foram testados, InceptionV3, ResNet16 e VGG16 no reconhecimento dos áudios. Além disso, é usado um pré-processamento no áudio para adicionar ruídos de fundo aleatoriamente para estender o *dataset*. As amostras de áudio foram obtidas por *datasets* de áudios gratuitos. Os modelos foram testados submetidos a diversas configurações considerando apenas espectrogramas, com efeito de *downsampling* no sinal de áudio e com todas as características mencionadas anteriormente. Notou-se que usando a combinação de todas as características foi possível classificar melhor as amostras onde não há disparo, reduzindo a detecção de falsos positivos. A melhor precisão encontrada foi usando o modelo ResNet18 com efeito de *downsampling* no sinal de entrada.

O artigo “Low Cost Gunshot Detection using Deep Learning” (MOREHEAD *et al.*, 2019) o objetivo é propor uma tecnologia acessível e precisa na detecção e localização de disparos de arma de fogo. A pesquisa realizada visa criar modelos de redes neurais convolucionais capazes de identificar espectrogramas com precisão e embarcar em um microcomputador Raspberry Pi. Os modelos propostos foram otimizados a algoritmos de pré-processamento do áudio para identificar com precisão disparos e serem implantados em um Raspberry PI. Neste trabalho, por se

tratar de uma solução robusta, o *software* embarcado é um pipeline separado em 3 estágios: avaliar continuamente o áudio captado em dois segundos, caso um disparo seja detectado um alerta é gerado e então encaminhado mensagens para determinado número de telefone informando sobre a ocorrência. No que diz respeito a localização é realizado uma avaliação através do uso de triangulação para identificar as diferenças no tempo de ocorrência de alertas entre os dispositivos Raspberry Pi para determinar onde o tiro pode ter ocorrido. Esta abordagem é assistida por um modelo de inteligência artificial que usa a diferença entre tempo e volume dos vários dispositivos como entrada para prever a localização do disparo. A avaliação da solução teve sucesso na detecção de disparos em ambientes controlados, no entanto, ainda deve ser testado em um cenário mais realístico, como um centro urbano de alta criminalidade para avaliar a robustez do projeto.

3 SOLUÇÃO PROPOSTA

Esta seção refere-se à descrição da solução proposta para o estudo, assegurando a credibilidade e legitimidade dele. Nas subseções a seguir será tratado sobre: como foi composto o *dataset*; o pré-processamento das amostras; quais são os modelos usados e o treinamento respectivo dos modelos.

3.1 DATASET

A primeira etapa da solução foi obter gravações de áudios que contenham amostras de disparos de arma de fogo e sons similares. O *dataset* de armas de fogo é o Gunshot Audio Forensics[4] contendo aproximadamente 10.000 amostras gravados em diversos ângulos por quatro dispositivos diferentes (LILIEN, 2018). As amostras têm entre um e dois segundos de duração com uma frequência de amostragem de 48.000 Hz. Além disso, as amostras são gravações de diversos tipos de armas como pistolas, revólveres, carabinas e rifles.

No que diz respeito, ao outro *dataset* utilizado foi o UrbanSound8K[5] contendo 8732 amostras de áudios similares subdivididos em 10 classes distintas: ar-condicionado, buzina, criança brincando, latido, perfuração, motor, disparo, britadeira, sirene e música de rua. As amostras presentes neste *dataset* tem entre um e quatro segundos de duração com uma frequência de amostragem de 44.100 Hz.

O *dataset* usado foi dividido em dois conjuntos, um para treinamento, validação e outro para teste para avaliação de resultados. O conjunto de treinamento e validação é composto de 9003 amostras de áudio, onde 4645 são da classe de disparo e 4358 da classe não é disparo. O conjunto de teste contém 3096 amostras de áudio, com 2210 da classe disparo e 859 da classe não é disparo. Vale ressaltar que as amostras de áudio foram selecionadas aleatoriamente a partir dos *datasets* de referência Gunshot Audio Forensics e UrbanSound8K.

---

[4] GUNSHOT AUDIO FORENSIS DATASET. Gunshot audio forensics dataset. **Cadreforensics**, 2022. Disponível em: http://cadreforensics.com/audio/. Acesso em: 10 out. 2022.
[5] URBAN SOUND DATASETS. Urbansound8k dataset. **Urban Sound Datasets**, 2022. Disponível em: https://urbansounddataset.weebly.com/urbansound8k.html. Acesso em: 30 set. 2022.

3.2 PRÉ-PROCESSAMENTO

O objetivo do pré-processamento é preparar as amostras de áudio que serão utilizadas no treinamento da rede neural convolucional. O pré-processamento realizado consiste em normalizar o áudio e gerar o respectivo espectrograma.

Primeiramente, para realizar o pré-processamento do áudio foi necessário ajustar todas as amostras para a mesma frequência de amostragem de 44.100 Hz e garantir a mesma duração em segundos. Utilizando as funções da biblioteca Librosa foi manipulada a amostra, modificando a frequência de amostragem e mapeado quando há disparo de arma de fogo, marcando o instante em que ocorre. O algoritmo implementado usa funções do Librosa para mapear o instante do disparo dado a partir dos seguintes passos: primeiro a amostra de áudio é lida; depois, as amplitudes são normalizadas e foi calculada a média geral da magnitude do sinal. Em seguida, foi preciso dividir a amostra em janelas de tempo de um segundo com um deslocamento de um décimo. A seguir é calculada a média da magnitude para as janelas. Por fim, é avaliado qual janela tem a maior média de magnitude e a marca. Portanto, para garantir que as amostras tenham a mesma duração, foi desenvolvida uma função para a partir dos áudios marcados obter somente o segundo onde ocorre o disparo.

O algoritmo utilizado para o processamento das amostras áudio foi implementado a partir de funções do Librosa e realiza os seguintes passos: primeiro, a amostra de áudio é lida; depois, é obtido apenas o trecho onde contém o disparo; em seguida, cada trecho é dividido em janelas; a seguir, em cada janela do sinal será aplicado uma FFT, transformando cada janela em um conjunto de valores representados por magnitude e frequência; por fim, converte se a magnitude em dB e, associando todas as janelas obtemos a representação do espectrograma do trecho de áudio contendo o disparo. Vale ressaltar que, por utilizar modelos de redes neurais convolucionais pré-definidos, os espectrogramas foram redimensionados para 224 x 224 *pixels*.

3.3 MODELOS

Um modelo de uma rede neural convolucional trata-se de uma arquitetura previamente conhecida e validada em um determinado problema, mas que pode ser utilizada para avaliar outros casos. No presente trabalho a abordagem escolhida foi de realizar o *fine-tuning* (ajuste fino) dos modelos pré-definidos InceptionV3, ResNet50 e VGG16.

O *fine-tuning* é uma técnica que reduz o tempo de treinamento, pois, ele mantém os modelos pré-treinados e adiciona camadas extras para refinar as características aprendidas pelo modelo. Nas subseções será apresentado como cada modelo foi adaptado para a detecção de disparos.

3.3.1 InceptionV3

O modelo InceptionV3[6] é uma rede neural convolucional desenvolvida pela equipe do Google em 2015 que possui um total de 42 camadas e uma baixa taxa de erro. Na figura 8, observa-se que a arquitetura deste modelo é segmentada em níveis e que no início temos camadas de diferentes tamanhos e no final camadas paralelas.

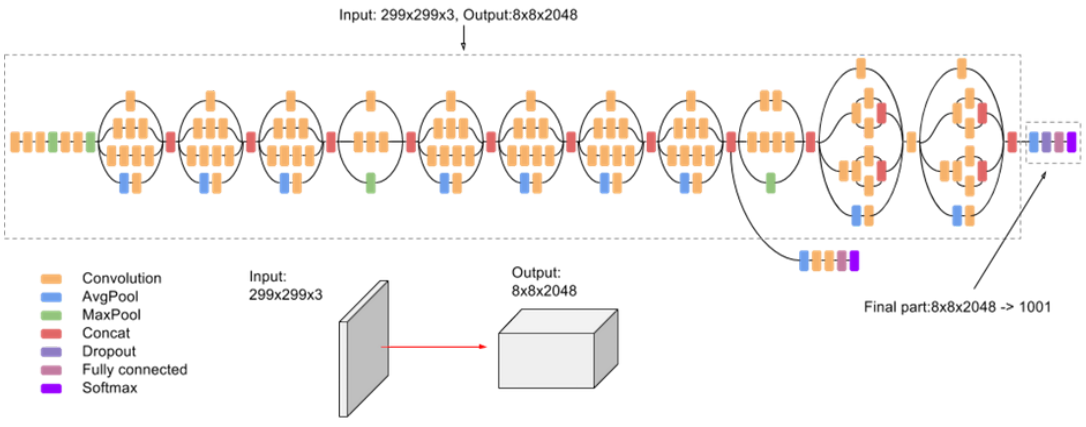


Figura 8 – Arquitetura do modelo InceptionV3.

Fonte: Google Cloud (2022).

Usando a API do Keras foi carregado o modelo InceptionV3 e então adicionado uma camada *flatten*, uma camada de *dropout* entre duas camadas

[6] Disponível em: https://keras.io/api/applications/inceptionv3/. Acesso em: 28 out. 2022.

dense. A camada *flatten* funciona como entrada do dado retornada pelo modelo para alimentar as camadas subsequentes. A arquitetura do modelo na figura 8 retrata que a dimensão do vetor de valores é de 2048. Visto que o *fine-tuning* tem como objetivo refinar os valores obtidos nas camadas *dense,* reduzindo o vetor de valores na saída.

3.3.2 ResNet50

O modelo ResNet50[7] é uma variante dos modelos *Residual Network* desenvolvido em 2015 inovando o uso de redes neurais que ao invés de aprender características entre as camadas, é apreendido somente resíduos. Isto é, entende-se resíduo como uma simples subtração da característica aprendida na entrada da camada.

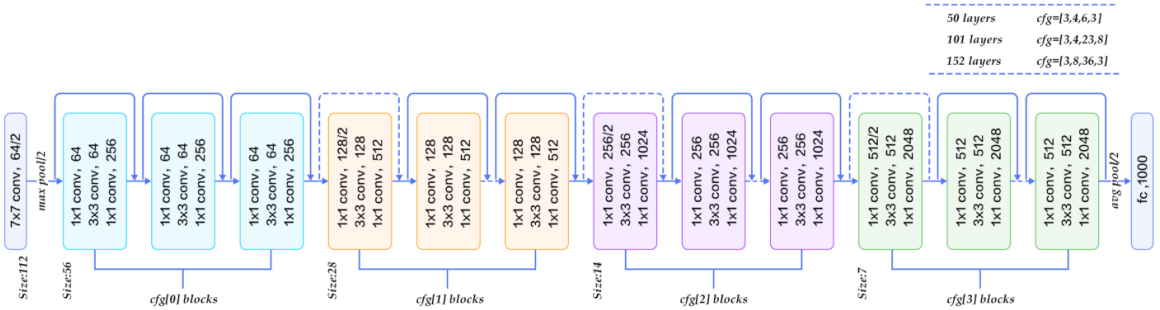


Figura 9 – Arquitetura do modelo ResNet50.

Fonte: Wangenheim (2018).

Conforme a figura 9, percebe-se que o modelo ResNet faz uma conexão entre a entrada e a saída de todas as camadas, indicando o aprendizado por resíduo entre as camadas presentes na arquitetura.

Usando o Keras para carregar o modelo do ResNet50 foi adicionado uma camada *flatten*, duas camadas *dropout* entre três camadas *dense*. O objetivo desta abordagem, adicionando mais camadas *dense*, é estender o comportamento da arquitetura residual e, ao mesmo tempo, refinar o vetor de valores.

[7] Disponível em: https://keras.io/api/applications/resnet/. Aceso em: 26 out. 2022.

3.3.3 VGG16

O modelo VGG16[8] trata-se de uma variante do VGG com 16 camadas convolucionais e é muito interessante devido à arquitetura uniforme, como observamos na figura 10.

Figura 10 – Arquitetura do modelo VGG16.

Fonte: Wangenheim (2018).

Esta arquitetura é muito utilizada na extração de recursos de imagem, entretanto, ela mapeia diversos parâmetros. Assim, o uso do *fine-tuning* nesta arquitetura é justamente condensar os valores para obter uma melhor precisão. A configuração realizada utilizando o Keras foi adicionar uma camada *flatten*, uma camada *dropout* e quatro camadas *dense*. A camada *dropout* foi adicionada no centro das camadas *dense* para ter uma melhor distribuição entre as camadas que serão desabilitadas aleatoriamente.

3.4 TREINAMENTO

O treinamento dos modelos foi implementado na plataforma do Google Colab, ferramenta essa usada para treinar, validar e avaliar os resultados. O algoritmo de treinamento pode ser dividido em algumas etapas, A primeira delas é mapear os *datasets* de treino, validação e testes, onde representam 60%, 30% e 20% das amostras respectivamente. A etapa subsequente é carregar um modelo e adicionar as camadas de *fine-tuning* mencionadas na seção anterior. A partir do modelo carregado, configurado e compilado é iniciado treinamento em cima dos *datasets* de

[8] Disponível em: https://keras.io/api/applications/vgg/. Acesso em: 27 out. 2022.

treino e validação. A duração do treinamento foi estipulada em 10 épocas para comparação posterior com os trabalhos relacionados.

4 AVALIAÇÃO DE RESULTADOS

Esta seção refere-se aos resultados obtidos a partir do treinamento realizado. Nas subseções a seguir será tratado sobre: o formato de avaliação; revisão dos resultados obtidos e a comparação entre os resultados de trabalhos relacionados.

4.1 FORMATO DE AVALIAÇÃO

O método para avaliação dos modelos foi realizado a partir das métricas de acurácia, precisão, *recall*, *f-score* e matriz de confusão.

A acurácia é uma métrica obtida a partir do treinamento do modelo, cuja representação é a quantidade de amostras classificadas corretamente. A precisão é a métrica gerada após os testes realizados com o *dataset* de treino e representa a quantidade de amostras classificadas, que realmente pertencem a essa classe. A *recall* é a métrica gerada após os testes realizados com o *dataset* de treino e representa a quantidade de amostras classificadas corretamente. O *f-score* é uma métrica que considera a taxa de precisão e *recall*, representando a média harmônica entre os dois valores.

A tabela de confusão é uma ferramenta que auxilia na análise rápida do desempenho do treinamento. Os valores representados na matriz são obtidos a partir do resultado da classificação do *dataset* de teste e compara o valor da predição com a classe correta.

4.2 RESULTADOS

A partir do treinamento dos modelos InceptionV3, ResNet50 e VGG16 o resultado obtido ficou abaixo do trabalhos relacionados. No caso, os resultados gerados durante o treinamento foram bons, sendo que a menor taxa de acurácia é de 96,15% dentre os modelos. Porém, os resultados pós-treinamento dados a partir da avaliação da capacidade de predição do modelo sobre o *dataset* de teste não foram satisfatórios. Diante disso, os valores de precisão, *recall* e *f-score* ficaram abaixo dos 84% o que se tratando de redes neurais não é um valor confiável, pois está abaixo dos resultados encontrados nos trabalhos relacionados. Este resultado

indica *overfitting*, ou seja, os modelos treinados apenas distinguem as amostras usadas durante o treinamento e nos testes têm dificuldades em classificar corretamente. A causa do *overfitting* são os dados de treinamento que não possuem um pré-processamento adequado propagando ruído para aprendizado.

Tabela 1: Comparação entre os Modelos

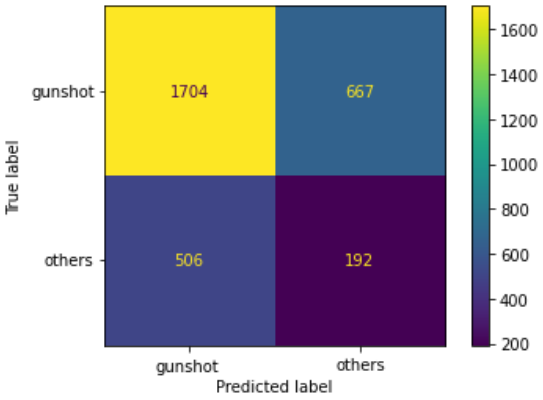
Modelo	Acurácia	Precisão (disparo)	*Recall*	*F-Score*
InceptionV3	96,15%	77%	72%	74%
ResNet50	99,43%	82%	73%	77%
VGG16	99,46%	84%	73%	78%

Fonte: Elaborado pelo autor (2022).

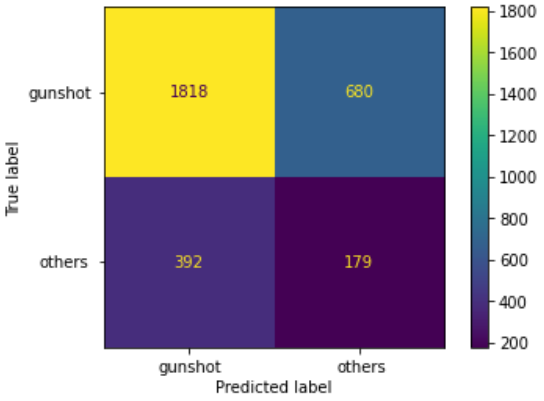
Analisando a Tabela 1, observamos que durante o treinamento os modelos alcançaram altos valores de acurácia, entretanto, as outras métricas obtidas após os modelos serem submetidos a classificar o *dataset* de treinamento os resultados não atingiram um valor aceitável. Isto é, uma precisão máxima de 84% com o modelo VGG16 e 77% com o InceptionV3, representando cerca de 20% de queda da acurácia. Na métrica de recall, temos uma queda de porcentagem, uma vez que considera os acertos de verdadeiros positivos para ambas classes. Neste cenário, um dos motivos para essa queda de rendimento é o *dataset* usado para teste, pois não possui muitas amostras da classe não é disparo. De forma que, aproximadamente 70% das amostras são disparos e os outros 30% não são disparos. Esta distribuição do *dataset* acaba não favorecendo a análise do resultado obtido.

Na figura 11, é representado a matriz de confusão para cada modelo, a matriz (a) InceptionV3 mostra que 1704 entre 2210 amostras de disparo foram classificadas corretamente, entretanto, 667 entre 859 amostras de não disparo foram classificadas como disparo, ou seja, uma taxa de 77,64% de falso positivo. A matriz (b) ResNet50 também possui uma alta taxa de falso positivo, classificando 680 amostras de não disparo como disparo. O pior caso é na matriz (c) VGG16 que teve 701 amostras de não disparo classificadas como disparo, sendo uma taxa de 81,6% de falso positivo. Percebe-se, que os modelos tendem a ter uma taxa semelhante entre as amostras

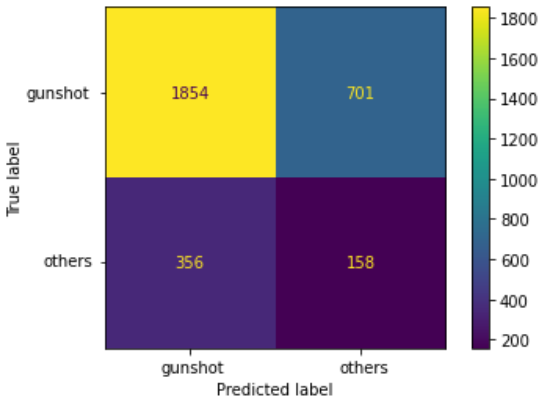
classificadas como verdadeiro positivo e falso positivo. Sendo uma solução não confiável na detecção de falsos positivos.



(a) InceptionV3



(b) ResNet50



(c) VGG16

Figura 11 – Matrizes de Confusão.

Fonte: Elaborado pelo autor (2022).

Outro fator que deve ser considerado para justificar este resultado é o pré-processamento realizado sobre o *dataset* de áudio. Em um cenário ideal, as amostras usadas seriam processadas para remover quaisquer efeitos sonoros como reverberação ou eco e adicionado um ruído de fundo para torná-lo mais realístico em situações reais. O uso de outras características como coeficientes MFCC poderiam ter sido extraídas do áudio para auxiliar na identificação de disparos. Desta forma, tornaria a solução mais sofisticada no âmbito de processamento de áudio em si, devido ao fato que os modelos tiveram um rendimento aceitável durante o treinamento. Isso evita o *overfitting* nos modelos, uma vez que, os ruídos são removidos durante o processamento do áudio, reduzindo o aprendizado de dados incorretos.

Ao avaliar os trabalhos relacionados seguem algumas diretrizes de preparar o áudio antes de processar os espectrogramas, entretanto, o presente trabalho não realiza nenhum pré-processamento para remover quaisquer efeitos sonoros. Também, não é levado em consideração outras características como coeficientes MFCC, matriz de *self-similarity* em um dos trabalhos, onde o resultado é melhor que o obtido na solução proposta. Ao comparar os resultados obtidos, observa-se que no primeiro trabalho a precisão, 98,6% de precisão usando todas as características citadas anteriormente. No caso, o segundo trabalho a precisão é de 98% apenas pré-processando a amostra de áudio. Percebe-se, que o pré-processamento geraria um resultado satisfatório tanto na taxa de acerto de verdadeiro positivo quanto falso positivo. Em comparação com os estudos citados anteriormente, tivemos um decréscimo na precisão de 14% no pior resultado.

Conclui-se que apesar de não ser o resultado similar aos trabalhos relacionados, o modelo que teve a melhor performance foi o ResNet50 avaliando a precisão obtida entre a classificação de verdadeiro positivo e falso positivo.

5 CONCLUSÃO

Neste trabalho foram expostas técnicas de processamento de áudio, treinamento de redes neurais convolucionais e o seu uso para classificar se uma amostra de áudio é um disparo de arma de fogo. As amostras de áudio utilizadas para treinamento, validação e avaliação foram obtidas de dois *datasets* o Gunshot Audio Forensics e o UrbanSound8K. Também, foram pré-processadas as amostras de áudio para normalizar e marcar os instantes de disparos. Foi utilizado apenas o espectrograma como *feature* para tarefa de classificação das amostras. A partir da técnica de *fine-tuning* os modelos InceptionV3, ResNet50 e VGG16 foram refinados para identificação de disparos a partir dos áudios. Todo o desenvolvimento feito para realização do projeto pode ser acessado no Github[9] e Google Colab.

Conclui-se que apenas utilizando espectrogramas é possível obter um resultado aceitável e em alguns casos superior a modelos que se usam de mais *features* para classificação.

Este trabalho abordou uma solução de *Machine Learning* no reconhecimento a partir de áudios. Em projetos futuros, seria pertinente e interessante um estudo abordando as técnicas de processamento de áudio para a remoção de ruídos. Desta forma, melhorar a qualidade das amostras assim gerando melhores resultados em modelos pré-treinados. Também seria interessante adicionar algum pré-processamento na imagem destacando a área em que ocorre o disparo, facilitando assim o treinamento e melhorando a distinção do que não é um disparo.

Esperamos que o presente trabalho, com os resultados obtidos, possibilite o desenvolvimento de outras pesquisas, porém considerando outros aspectos, visões e abordagens. E que este estudo contribua para a área de *Machine Learning*, mas também inspire outras pesquisas considerando outros aspectos e caminhos que podem ser seguidos.

[9] Disponível em: https://github.com/. Acesso em: 2 nov. 2022.

REFERÊNCIAS

BAJZIK, Jakub; PRINOSIL, Jiri; KONIAR, Dusian. Gunshot detection using convolutional neural networks. *In:* **24th International Conference Electronics**, 2020. Disponível em: https://doi.org/10.1109/IEEECONF49502.2020.9141621. Acesso em: 1 nov. 2022.

CEDRO TECHNOLOGIES. OpenCV: uma breve introdução à visão computacional com python. **Cedro Tecnologies**, 3 out. 2018. Disponível em: https://blog.cedrotech.com/opencv-uma-breve-introducao-visao-computacional-com-python. Acesso em: 20 out. 2022.

CHOLLET, Francois. Keras. **GitHub**, 2015. Disponível em: https://github.com/fchollet/keras. Acesso em: 10 out. 2022.

DATA SCIENCE ACADEMY. O que é visão computacional? **Data Science Academy**, 24 jun. 2022. Disponível em: https://blog.dsacademy.com.br/o-que-e-visao_computacional/. Acesso em: 30 nov. 2022.

ENGINEER AMBITIOUSLY. Acquiring an analog signal: bandwidth, Nyquist Sampling Theorem, and Aliasing. **NI**, 10 ago. 2022. Disponível em: https://www.ni.com/pt-br/innovations/white-papers/06/acquiring-an-analog-signal--bandwidth--nyquist-sampling-theorem-.html. Acesso em: 5 set. 2022.

GOOGLE. O que é o colaboratory?. **Colaboratory**, 2022. Disponível em: https://research.google.com/colaboratory/intl/pt-BR/faq.html#:~:text=O%20Colaboratory%20ou%20%E2%80%9CColab%E2%80%9D%20%C3%A9,an%C3%A1lise%20de%20dados%20e%20educa%C3%A7%C3%A3o.. Acesso em: 15 out. 2022.

GOOGLE CLOUD. Guia avançado do Inception v3. **Google Cloud**, 6 jul. 2022. Disponível em: https://cloud.google.com/tpu/docs/inception-v3-advanced. Acesso em: 3 nov. 2022.

GUNSHOT AUDIO FORENSIS DATASET. Gunshot audio forensics dataset. **Cadreforensics**, 2022. Disponível em: http://cadreforensics.com/audio/. Acesso em: 10 out. 2022.

GURUCHARAN, Marthi. Basic CNN Architecture: explaining 5 layers of convolutional neural network. **UpGrad**, 28 jul. 2022. Disponível em: https://www.upgrad.com/blog/basic-cnn-architecture/. Acesso em: 30 nov. 2022.

KNEIPP, Eduardo Campos. **Análise de áudios para predição de no-show**. 2019. 48 f. Trabalho de Conclusão de Curso (Bacharel em Engenharia de controle e Automoção) - Departamento de Automoção e Sistemas, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, 2019. Disponível em: https://repositorio.ufsc.br/bitstream/handle/123456789/200059/PFC%20Eduardo%20Campos%20Kneipp_2019-1.pdf?sequence=1&isAllowed=y. Acesso em: 26 out. 2022.

LILIEN, Ryan. Development of computational methods for the audio analysis of gunshots. **National Criminal Justice Reference Service**, jun. 2018. 30 p. Disponível em: https://www.ojp.gov/ncjrs/virtual-library/abstracts/development-computational-methods-audio-analysis-gunshots. Acesso em: 5 nov. 2022.

MCFEE, Brian *et al*. Librosa: audio and music signal analysis in Python. *In:* **Proceedings of the 14th Python in Science Conference**, p. 18-25. 2015. Disponível em: https://doi.org/10.25080/Majora-7b98e3ed-003. Acesso em: 31 out. 2022.

MOREHEAD, Alex *et al*. Low cost gunshot detection using deep learning on the Raspberry Pi. *In:* **IEEE International Conference on Big Data (Big Data)**, p. 3038-3044, 2019. Disponível em: https://doi.org/10.1109/BigData47090.2019.9006456. Acesso em: 1 nov. 2022.

MOURA, Shayenne da Luz. **Separação de sinais de áudio em melodia e acompanhamento**. 2016. 37 f. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) - Instituto de Matemática e estatística, universidade de São Paulo, São Paulo, 2016. Disponível em: https://linux.ime.usp.br/~shayenne/mac0499/monografia.pdf. Acesso em: 26 out. 2022.

REIS, Clovis Ferreira dos. **Sistema modular para detectação e reconhecimento de disparos de armas de fogo**. 2015. 110 f. Dissertação (Mestrado em Informática) – Programa de Pós-Graduação em Informática, Centro de Informática, Universidade Federal da Paraíba, João Pessoa, 2015. Disponível em: https://repositorio.ufpb.br/jspui/handle/tede/9244. Acesso em: 25 out. 2022.

URBAN SOUND DATASETS. Urbansound8k dataset. **Urban Sound Datasets**, 2022. Disponível em: https://urbansounddataset.weebly.com/urbansound8k.html. Acesso em: 30 set. 2022.

WANGENHEIM, Aldo von. Deep learning: reconhecimento de imagens. Redes para classificação de imagens e reconhecimento de objetos em cenas. **LAPIX**, 2018. Disponível em: https://lapix.ufsc.br/ensino/visao/visao-computacionaldeep-learning/deep-learningreconhecimento-de-imagens/. Acesso em: 4 nov. 2022.

WIKIPÉDIA. OpenCV. **Wikipédia:** a enciclopédia livre, 13 out. 2022. Disponível em: https://pt.wikipedia.org/wiki/OpenCV. Acesso em: 20 out. 2022.

WIKIPÉDIA. Teorema da amostragem de Nyquist–Shannon. **Wikipédia:** a enciclopédia livre, 27 jul. 2021. Disponível em: https://pt.wikipedia.org/wiki/Teorema_da_amostragem_de_Nyquist%E2%80%93Shannon. Acesso em: 10 set. 2022.