Deep Learning and Mel-spectrograms for Physica Violence Detection in Audio

Tiago B. Lacerda[1], **Péricles Miranda**[2], **André Câmara**[2], **Ana Paula C. Furtado**[1,2]

[1]Centro de Estudos e Sistemas Avançados do Recife
CESAR.School
Recife, PE, Brazil

[2]Computing Department
Universidade Federal Rural de Pernambuco
Recife, Pernambuco, Brazil

tbl@cesar.school

{pericles.miranda, andre.camara, anapaula.furtado}@ufrpe.br

Abstract. *There is a growing interest in systems that detect violence automatically using ambient audio. In this work, we built and evaluated four classifiers with this proposal. However, instead of directly processing the audio signals, we converted them to images, known as Mel-spectrograms, and then used Convolutional Neural Networks (CNN) to treat as an image classification problem using pre-trained networks in this context. We tested the architectures Inception v3, VGG-16, MobileNet v2, and ResNet152 v2, and the classifier coming from the MobileNet architecture obtained the best classification results when evaluated on the HEAR Dataset created to carry out this research.*

Resumo. *Há um crescente interesse em sistemas de detecção de violência de forma automática por meio do áudio ambiente. Neste trabalho, construímos e avaliamos 4 classificadores com essa proposta. Porém, em vez de processar diretamente os sinais de áudio, nós os convertemos para imagens, conhecidas como mel-spectrograms, e em seguida utilizamos Redes Neurais Convolucionais (CNN) para tratar como um problema de classificação de imagens utilizando-se de redes pre-treinadas neste contexto. Testou-se as arquiteturas Inception v3, VGG-16, MobileNet v2 e ResNet152 v2, tendo o classificador oriundo da arquitetura MobileNet obtido os melhores resultados de classificação, quando avaliado no HEAR Dataset, criado para a realização desta pesquisa.*

1. Introdução

A detecção de violência física por meio do áudio é um problema complexo e pouco estudado, conforme [Durães et al. 2021], que identificou apenas 3 trabalhos entre 2015 e 2020 sobre o tema. Por outro lado, a área de Classificação de Cenas Acústicas (*Enviromental Sound Classification*, ESC), que trata de classificar sons e ruídos ambientais, é bastante desenvolvida contando com mais de 900 artigos, encontrados no Microsoft Academic. Muitos dos trabalhos de ESC utilizam a abordagem de extrair *mel-spectrograms* de áudios para tratar como um problema de classificação de imagens ([Piczak 2015],

[Nordby 2019]). Esta estratégia tem se mostrado promissora no domínio de ESC, e não foi investigada no contexto de detecção de violência em áudios. Isto nos fez levantar a seguinte questão de pesquisa que servirá de guia para o desenvolvimento deste trabalho: *É possível identificar cenas de violência física convertendo áudios em imagens e utilizando CNN's para realizar essa classificação?*

Na ausência de um *dataset* neste contexto para utilizar na pesquisa, foi desenvolvido um, sintético, chamado HEAR Dataset. Ele conta com mais 70000 áudios de 10 segundos divididos meio a meio entre duas classes: presença ou não de violência física praticada no segmento. Oportunamente, publicamos o HEAR Dataset em conjunto a este trabalho. Nos experimentos foram consideradas as seguintes arquiteturas de CNN: MobileNet [Sandler et al. 2018], Inception [Szegedy et al. 2015], VGG-16 [Simonyan and Zisserman 2015] e ResNet152 [He et al. 2015], e os resultados mostraram que a CNN MobileNet foi estatisticamente melhor que os demais modelos quando avaliada no HEAR DATASET, atingindo acurácia de 78.9% e f_1 score de 78.1%.

2. Background

Nesta Seção, serão introduzidos os conceitos de Redes Neurais Convolucionais (CNN), Espectrograma e *Mel-spectrogram* a serem utilizados neste trabalho.

2.1. Redes Neurais Convolucionais (CNN)

Uma Rede Neural Convolucional [Fukushima 1980], *Convolutional Neural Networks* ou CNN, consiste em uma arquitetura com uma camada de neurônios de entrada, seguida de uma série de camadas diferentes intercaladas sendo as mais comuns as camadas convolucionais, de agrupamento, um número de camadas de neurônios completamente conectados, além de uma camada de saída. A diferença entre a CNN e a perceptron multicamadas convencional consiste na introdução de uma combinação de convoluções e outras operações. Os filtros convolutivos ao serem aplicados à uma imagem, geram um conjunto de *features*, que são passadas para as camadas seguintes. Com isso, as camadas de neurônios completamente conectadas, ao invés de se conectarem a todos os pixels de uma imagem, limitam-se a processar apenas uma pequena parte da entrada (por exemplo, pequenos blocos de 3×3 pixels, definidos na camada de convolução), reduzindo-se o número de parâmetros de uma arquitetura CNN quando comparada a perceptron convencional. As redes neurais convolucionais passaram a contar com grande popularidade após o surgimento da arquitetura AlexNet [Krizhevsky et al. 2017], utilizada no desafio ImageNet de classificação de imagens. Desde então surgiram diversas outras arquiteturas passando das 8 camadas da AlexNet para mais de 150 na ResNet [He et al. 2015].

2.2. Domínio do Áudio

De acordo com [Mesaros et al. 2016, Abeßer 2020], existem duas classes de problemas distintas: Classificação de Eventos sonoros (*Acoustic Scene Classification*, ASC), que consistem em adicionar um ou mais marcadores, ou *tags*, ao áudio em análise para indicar a presença de determinadas características. A outra é Detecção de Eventos Sonoros (*Acoustic Event Detection*, AED) similar a anterior, porém determina-se os momentos de início e fim de cada um dos eventos de interesse sendo, portanto, mais complexo que o primeiro. O problema tratado neste artigo é de ASC, com o objetivo de detectar em um áudio se há violência ou não, não importando em que momento do áudio ela ocorra.

2.2.1. Espectrograma

Um espectrograma é uma representação visual do espectro de frequências de um sinal variando no tempo como, por exemplo, na Figura 1. Ele é construído tomando-se a *Short-Time Fourier Transform* (STFT) do sinal, que converte um sinal do domínio do tempo para o domínio da frequência. Porém ao invés de se calcular a Transformada de Fourier (TF) do sinal completo, ela é calculada em segmentos de intervalos curtos e depois dispostos em sequencia. Por exemplo, em um áudio de 1 segundo, pode-se dividi-lo em 10 intervalos de 100 ms, calcula-se a TF de cada intervalo e monta-se a sequencia espectral encadeada. Há possibilidade de haver sobreposição de intervalos, por exemplo, com o primeiro intervalo de [0 ; 100ms] e o segundo de [80 ; 180ms], representando uma sobreposição de 20ms ou 20%.

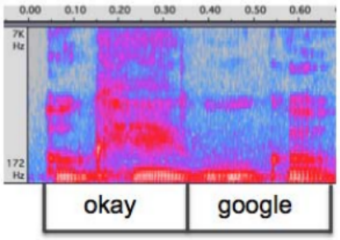

Figura 1. Exemplos de espectrograma utilizado no framework inicial de *Keyword Spotting* do "Ok, Google". O eixo horizontal é o tempo e o eixo vertical é a frequência. A cor representa a intensidade (energia) em cada frequência, a cada instante. Fonte: [Chen et al. 2014]

2.2.2. Mel-spectrogram

O *Mel-spectrogram* difere-se do espectrograma por ter as frequências convertidas para a escala de Mel, que é uma transformação logarítmica aplicada aos componentes do domínio da frequência, cuja ideia principal é representar os componentes do som dando importância maior aos que mais sensibilizam o ouvido humano. A ideia subjacente é reduzir a quantidade de informação correlacionada, sem perda de informação relevante. A transformação utilizada para converter do domínio da frequência para a escala de Mel é dada por:

$$Mel(f) = 2595 \times \log_{10}(1 + \frac{f}{700})$$

.

Para ilustrar, observa-se que a diferença de frequência entre 2000 e 2500Hz e de 15.5kHz e 15kHz é a mesma, 500Hz, porém a segunda é imperceptível ao ouvido humano. Essa questão de sensibilidade maior nas frequências mais baixas é que a escala de Mel se propõem a modelar. Por fim, esta escala é a mais utilizada em problemas de classificação de cenas acústicas de acordo com [Abeßer 2020]. Após a extração do *mel-spectrogram*, uma imagem em duas dimensões pode ser criada para uso em classificadores de imagens. A Figura 2 ilustra o *mel-spectrogram* de dois áudios de violências físicas (um "soco" e um "tapa"), ambos caracterizados por curta duração ($<$ 200ms) e alta intensidade.

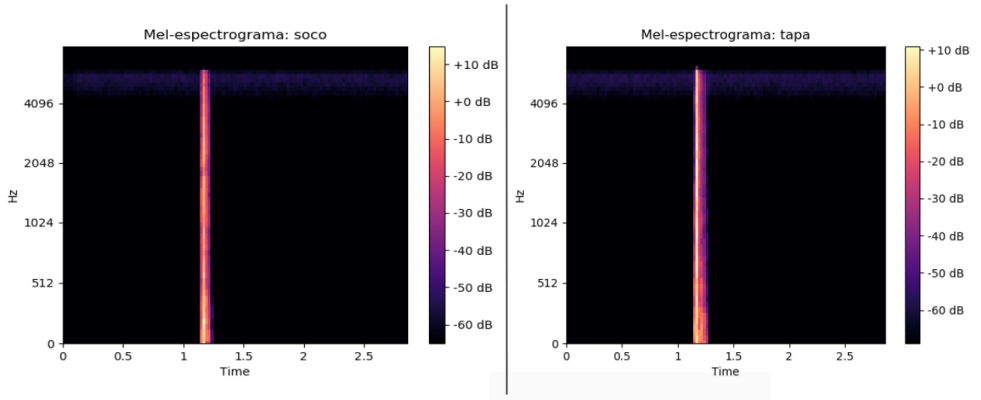

Figura 2. Exemplos de *mel-spectrograms* de duas violências físicas distintas. Esse tipo de evento se caracteriza por curta duração e alta intensidade, que se traduz em uma linha vertical do *mel-spectrogram*.

3. Trabalhos relacionados

Nesta seção, são apresentados inicialmente os trabalhos relacionados com detecção de violência por meio de áudio. Em seguida, são detalhados trabalhos de classificação de cenas acústicas (ESC) que converteram áudio em imagens para uso de CNNs.

Em [Durães et al. 2021] foram apresentados três trabalhos relacionados a detecção de violência por meio do áudio (sendo os demais por meio de vídeo ou áudio e vídeo - chamados multimodais) sendo publicados entre 2015 e 2020. O primeiro deles, desenvolvido por [Rouas et al. 2006], propôs um sistema de detecção de gritos em um vagão de metrô utilizando as seguintes características, MFCC, energia, delta e delta-deltas das características. Em seguida dois modelos foram criados, sendo um utilizando *Gaussian Model Mixture* (GMM) e outro *Support Vector Machine* (SVM). Neste trabalho foi utilizado um *dataset* próprio com cerca de 2500 segundos de áudio. Note-se que [Durães et al. 2021] incluiu esse trabalho mesmo sendo fora do período estabelecido (2015 a 2020).

[Crocco et al. 2016] apresenta uma revisão sistemática cujo título é *Audio Surveillance: A Systematic Review* onde são apresentados as características mais utilizadas, uma taxonomia dos problemas de classificação, problemas em aberto dentre outros.

Em [Souto et al. 2019] foram consideradas as características MFCC, Energia e *zero-crossing rate* (ZCR) extraídas dos áudios por meio de um sequenciamento de janelas de tempo de curta duração, posteriormente agrupadas por média e desvio-padrão em janelas de tempo maiores. Essas características foram então utilizadas em um classificador SVM. O de melhor desempenho utilizou o MFCC obtendo acurácia de 73,14% quando avaliado em um *dataset* próprio.

De forma complementar a revisão realizada por [Durães et al. 2021], realizamos uma pesquisa no Microsft Academic por meio do VOSViewer [1] com os termos "*violence*", "*detect*" and "*audio*" retornando 47 artigos, sendo o *survey* de Durães (2021) um deles.

Dessa pesquisa, se destacou o trabalho de [Giannakopoulos et al. 2006] como um dos primeiros a utilizar características como energia, entropia, ZCR em conjunto com

[1] URL: https://www.vosviewer.com/

classificador SVM para realizar detecção automática de cenas acústicas de violência. Em comum a estes trabalhos, verificou-se que não existe um *dataset* publicado para o problema da detecção de violência por meio do áudio. Em segundo lugar, apesar dos nossos melhores esforços de pesquisa, não identificamos até o momento trabalhos que utilizem *deep learning*, notadamente CNN's, para classificação de cenas acústicas contendo violência.

Por outro lado, a *Classificação de Cenas Acústicas* (ASC) é um campo ativo de pesquisa, que conta com 909 artigos, conforme busca realizada no Microsoft Academic. De acordo com [Nordby 2019], a utilização de CNN em conjunto com espectrogramas ou *mel-spectrograms* para classificação de sons ambiente é bastante comum. O próprio [Nordby 2019] utilizou esta abordagem em conjunto com o *dataset* UrbanSound8k ([Salamon et al. 2014]). O primeiro uso dessa abordagem para ESC foi por [Piczak 2015] também no *dataset* UrbanSound8k tendo obtido 81% de acurácia.

Portanto, o nosso objetivo com este trabalho é utilizar as técnicas e ferramentas do contexto do ESC aplicada ao contexto de classificação de violência por meio do áudio, utilizando para tanto de uma representação visual do áudio (mel-espectrograma) como entrada para as CNNs. Desta forma, aproveita-se todo o avanço e desenvolvimento das CNN, inclusive os modelos pré-treinados, neste novo contexto.

4. Materiais e Métodos

Esta seção apresenta o arranjo experimental adotado nesta pesquisa. Serão detalhados o processo de criação do HEAR dataset, os modelos de CNNs adotados bem como a estratégia de treino e teste, as métricas de avaliação dos modelos e o setup para a realização do experimento.

4.1. Dataset

Apesar de existirem *datasets* no contexto do áudio como o AudioSet [Gemmeke et al. 2017], o XD-Violence [Wu et al. 2020] (com áudios e vídeos de violência) ou FSD50K [Fonseca et al. 2020], derivado da base do Freesound [Fonseca et al. 2017], não encontramos nenhum que se adequasse ao contexto de detecção de violência com as características desejáveis a seguir: i) Base com volume suficiente para treinamento de modelos de *Deep Learning*, ii) Classes com e sem a ocorrência de violência física, e iii) Tamanho fixo, em segundos.

Diante do exposto, decidimos fabricar a nossa própria base de dados, chamada **HEAR DATASET**, atendendo aos requisitos acima descritos[2]. Para tanto, foi utilizada a ferramenta *Scaper* ([Salamon et al. 2017]), que é uma biblioteca Python, *open source*, para síntese sonora. Com ela é possível sintetizar sons a partir de uma coleção de eventos sonoros isolados, "misturando" múltiplos sons a partir de uma única "especificação", definida de forma probabilística. Para aumentar a variabilidade da saída, *Scaper* suporta a aplicação de transformações de áudio tais como mudança de tom e alongamento de tempo para cada evento.

Para criação da base, foram utilizadas como fonte três conjuntos de áudios extraídos das bases AudioSet e Freesound, conforme descrito a seguir:

[2]URL: https://drive.google.com/drive/folders/1Ijo3Tb52GzNzSo35Vl8rsZIdu691rzYG?usp=sharing

- **15 mil áudios de primeiro plano:** Extraídos do Google AudioSet, têm 10s e pertencem às classes *Inside*, *Conversation* e *Speech*. Esses áudios são os de *foreground*, primeiro plano. Detalhes sobre a descrição das classes pode ser obtida no site[3]. Chamaremos esses áudios de *conversas*.
- **20 mil áudios de fundo:** Também extraídos do Google Audioset porém das classes *pets, television, toilet flush, door, radio, water, vacuum cleaner, sobbing, noise, sink e frying* e todos com 10s de duração. Chamaremos esse áudios de *contexto*.
- **45 sons de violência física de curta duração:** São áudios curtos (menores que 250ms de duração) de violência física, notadamente socos e tapas, da base Freesound. Chamaremos estes áudios de *sons de violência*.

Para elaboração da base de dados, primeiramente separamos os conjuntos de *contexto* e *conversas* em 90% para a criação da base de treinamento e os outros 10% para a base de teste. Em seguida os áudios de *contexto* e *conversas* são mixados de forma aleatória utilizando o *Scaper*, dando origem a 30 mil áudios na base de treino e 5 mil na base de teste. Esses dois conjuntos de áudios compõem a classe de *não violência* do *dataset*. Em seguida, uma cópia desses áudios foi mixada novamente com os *sons de violência* de curta duração dando origem a classe *violência* do *dataset*. Todo o processo está descrito na Figura 3 e a Figura 4 mostra um exemplo de como foi gerado um áudio contendo uma violência utilizando o *Scaper*.

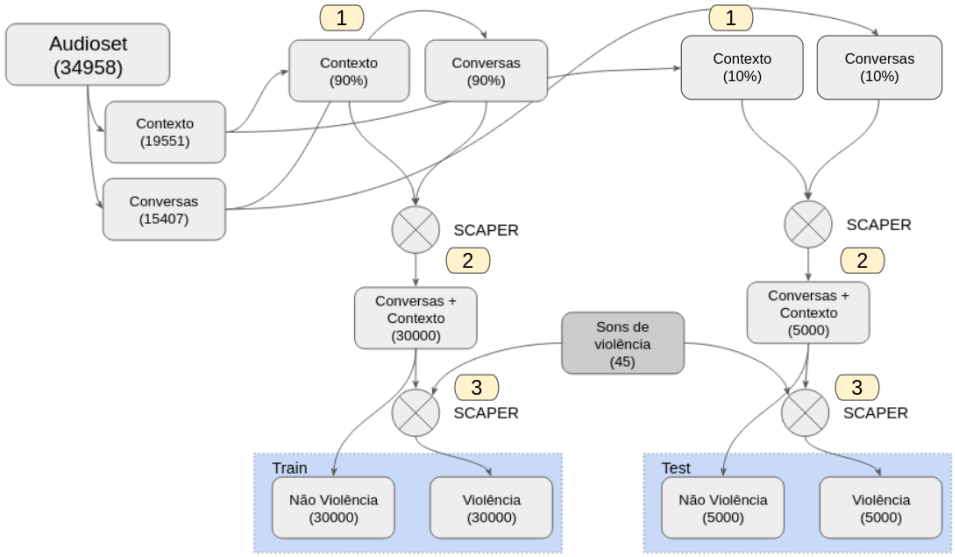

Figura 3. Detalhamento de como foi gerada a base de dados utilizando a base do Audioset e sons de violência. (1) Os áudios de *contexto* e *conversas* são separados na primeira etapa, em seguida (2) são mixados dando origem a base de treinamento e de teste, ambas sem sons de violência presentes. Em uma nova etapa, (3) de 1 a 3 sons de violência física são mixados dando origem aos exemplos em que há violência presente. Os mesmos sons de violência foram utilizados para gerar os exemplos das bases de treino e teste.

Diversos parâmetros podem ser configurados no *Scaper* durante e elaboração da base onde, através de experimentos, determinamos o o seguinte. Para mixagem dos áudios

[3]URL: https://research.google.com/audioset/

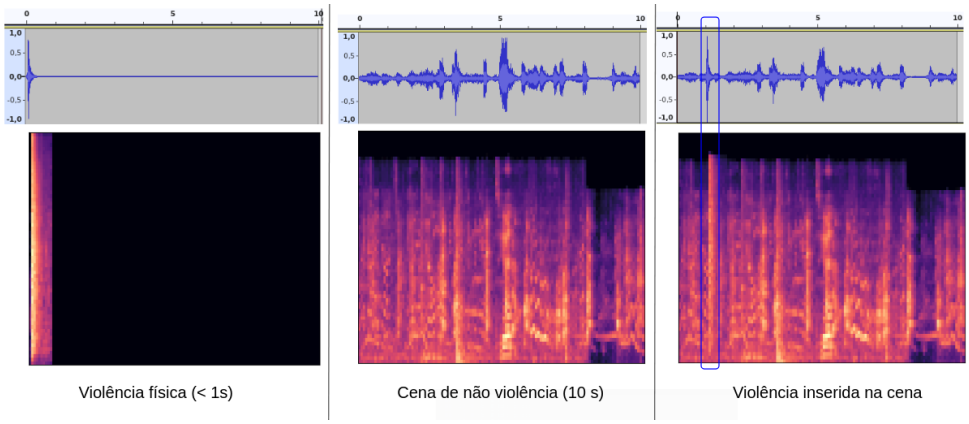

Figura 4. Na primeira Figura, à esquerda, há um áudio com menos de 1s de duração referente à uma agressão isolada. Na Figura central, há um áudio de um homem e uma mulher conversando. No áudio à direita, o som da agressão física foi aleatoriamente inserido no áudio central.

de *contexto* com os áudios de *conversas* para dar origem aos áudios da classe de *não violência*, utilizamos:

- **SNR (relação sinal-ruído):** Neste contexto, o sinal se refere aos áudios de *conversas* e o ruído ao *contexto* ou sons presentes no fundo. Os valores utilizados foram entre 4 e 6 dB com distribuição uniforme. O efeito do valor do SNR pode ser mais facilmente visualizado na Figura 5.
- **Pitch:** Aumento da base por meio de variação do tom. Utilizamos uma distribuição uniforme com valores entre -3 e +3 dB.
- **Time stretch:** Aumento da base por meio de variação na velocidade do áudio. Utilizamos uma distribuição uniforme do fator multiplicador entre 0.7 e 1.3.

Para a criação dos áudios da classe de *violência*, utilizamos novamente os parâmetros acima porém considerando os 45 áudios de violências físicas isoladas como primeiro plano e os áudios da classe de não violência (sintetizados no passo anterior) como segundo plano. De 1 a 3 áudios de violência foram inseridos nas cenas de não violência considerando uma localização média de 5s e desvio padrão de 3s dentro do quadro de 10s. Observa-se que ao variar o *SNR*, *Pitch* e *Time strech* potencializa-se os 45 exemplos de violência física disponíveis.

Os áudios gerados para ambas as classes têm as características abaixo:

- **Canais:** Mono canal
- **Taxa de amostragem:** 16000 (amostras por segundo)
- **Precisão:** 16-bit (cada amostra com 16 bits de resolução)
- **Bit Rate:** 256kbps (16k amostras x 16-bit por segundo)
- **Duração:** 10 segundos
- **Tamanho do arquivo:** 320kB (16k amostras x 16 bits por amostra x 10 s / 8 bits)

Por fim, a base criada conta com 60000 áudios de treinamento (50% de cada classe) e 10000 de teste (novamente, 50% de cada classe). Porém os 10000 áudios de teste foram divididos ao meio para criar uma base de avaliação, restando portanto 5000

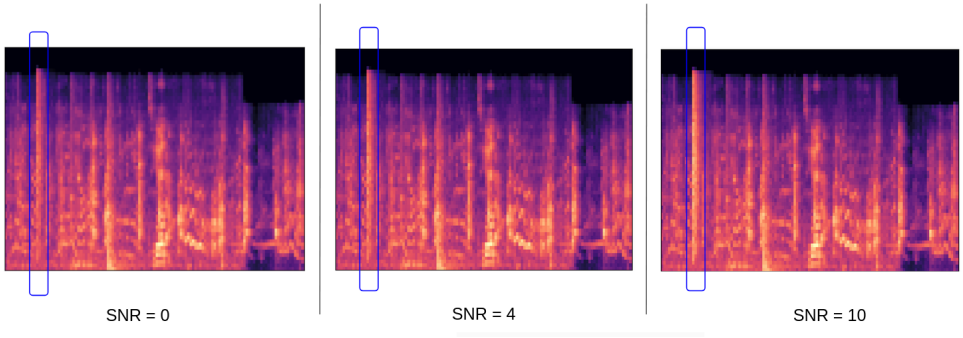

Figura 5. Demonstração do efeito de valores de SNR = 0, 4 e 10dB. Quanto mais intenso o evento de interesse, no caso a violência, mais facilmente ele seria detectado.

para teste e 5000 para avaliação, ambas com 50% de áudios com violência presente e 50% sem violência presente.

No entanto, a geração dos arquivos *waveform* foi um passo intermediário, pois utilizou-se representações desses áudios sobre a forma de *mel-spectrograms* (imagens) com entrada para os modelos de Redes Neurais Convolucionais. Para obter essa transformação, utilizou-se a biblioteca Librosa, que é um pacote Python para processamento de sinais de áudio e música. Após experimentação realizada com o software *Sonic Visualizer*, chegamos aos valores abaixo como os que foram capazes de gerar *mel-spectrograms* com maior definição.

- **n_fft:** 2048, que significa que 2048 amostras compõem uma janela curta para a transformada de Fourier. Como a taxa de amostragem é de 16KHz, 2048 amostras correspondem à 128ms do áudio.
- **hop_length:** 1796, passo para construção do *mel-spectrogram*. Como esse valor é menor que n_fft, significa que há sobreposição de 12,3%.
- **n_mels:** 64, número de *features* a serem consideradas.
- **Tamanho das imagens:** 640 x 480 x 3 (imagens coloridas).

4.2. Métodos Comparados

Neste trabalho, foram consideradas quatro arquiteturas de CNN's para comparação descritas a seguir, e que foram escolhidas por serem recentes (2015 em diante) e representativas de acordo com a taxonomia realizada por [Khan et al. 2020].

- **VGG-16** [Simonyan and Zisserman 2015]: Propôs uma ideia de campo receptivo eficaz e de uma topologia simples e homogênea porém se utiliza de 3 camadas completamente conectadas que são computacionalmente caras.
- **Inception v3** [Szegedy et al. 2015]: Utiliza filtros assimétricos e camada de gargalo para diminuir o custo computacional. Porém apresenta uma arquitetura complexa.
- **MobileNet v2** [Sandler et al. 2018]: Em 2017 um grupo de pesquisadores do Google publicou um artigo que introduziu uma arquitetura de rede neural otimizada para dispositivos móveis. A arquitetura proporciona alta precisão, mantendo os parâmetros e operações matemáticas o mais baixo possível, com objetivo de permitir o uso em Smartphones.

- **ResNet152 v2** [He et al. 2015]: Deve seu nome a seus blocos residuais com "saltos" ou "*skip connection*" que permitem que o modelo seja extremamente profundo. A inclusão do "*skip connection*" foi uma inovação que permitiu que a ResNet alcançasse até 152 camadas, sem problemas de gradiente de fuga (*vanishing gradient*).

Para realização dos experimentos, obteve-se os pesos dos modelos pré-treinados no *dataset* ImageNet, que é um *dataset* com mais de 1.4 milhões de imagens e 1000 classes. Essa técnica é conhecida como *transfer learning*, que consiste em aproveitar os pesos treinados em outro contexto e realizar apenas um ajuste em camadas adicionais da rede. Ao final de cada arquitetura, adicionamos uma camada de achatamento ou *Flatten*, seguida por uma camada de *dropout* de 20%, para evitar *overfitting*. Em seguida, adicionamos uma camada completamente conectada de neurônios, *dense layer*, com 1024 neurônios e função de ativação RELU. Por fim, outra camada completamente conectada para saída com apenas 2 neurônios referentes as duas classes que deseja-se classificar (violência e não violência) com função de ativação SOFTMAX, para o retorno das probabilidades das classes. Em relação à execução das CNNs, cada indivíduo foi treinado por 40 épocas (por razões computacionais) com um tamanho de lote de 20. Não foram definidos critérios de parada. O código fonte está disponível no GitHub[4].

4.3. Métricas de Avaliação

A seguir, definimos as métricas utilizadas para avaliação dos modelos desenvolvidos neste trabalho. A acurácia é o número de exemplos previstos corretamente entre todos os exemplos, e pode ser calculada pela seguinte equação: Acuracia $= \frac{TP+TN}{TP+FP+FN+TN}$, onde TP e TN significam verdadeiros positivos e negativos, e FP e FN significam falsos positivos e negativos. A acurácia é uma das métricas mais comuns para avaliar o desempenho do algoritmo em uma tarefa de classificação. No entanto, é comumente usada quando a distribuição de classes é balanceada, quando quando o TP e o TN estão presentes na mesma proporção no *dataset* (caso do HEAR DATASET). De forma complementar e informativa, também calculamos as demais métricas a seguir: F_1-score, que é a média harmônica entre a sensibilidade (revocação)$= \frac{TP}{TP+TN}$ e a precisão$= \frac{TP}{TP+FP}$. Esta métrica é obtida pela seguinte equação: F_1-score = $\frac{2 \times revocacao \times precisao}{revocacao+precisao}$. Como pode ser visto, F_1-score é adotado quando o FN e FP são cruciais.

4.4. Setup experimental

Os experimentos foram executados em um ambiente Linux POP!OS 64-bits com com processador AMD® Ryzen 5 3500u (3.7GHz, 8 Threads), 16GB de RAM, 256 SSD e uma GPU Nvidia GeForce MX250 com 2GB. Os registros foram feitos com auxilio do MLFlow, que é uma plataforma popular de código aberto para gerenciar o desenvolvimento de soluções de AM, incluindo o rastreamento de experimentos. Utilizamos a linguagem Python e o framework TensorFlow para o desenvolvimento dos modelos de CNN.

5. Resultados

Os modelos de CNN foram avaliados no HEAR DATASET, e os resultados são apresentados na Tabela 1. Esta mesma tabela também apresenta o tempo total de execução

[4]URL: http://github.com/tblacerda/HEAR_ENIAC

(em horas) e o total de parâmetros de cada modelo. Como se pode ver, a MobileNet obteve melhor desempenho em termos de acurácia (78.9%), f_1 score (78.1%), e precisão (80.9%). Esta arquitetura foi vencida pela ResNet-152 em revocação, que atingiu 87.8%, superando os 75.4% da MobileNet.

Visando realizar uma comparação justa, foi executado o teste estatístico de McNemar conforme descrito em [Dietterich 1998]. Esse teste é recomendado quando é inviável se treinar e avaliar modelos diversas vezes, como neste caso, onde o menor tempo de obtido foi de 17h para uma única avaliação do modelo VGG-16 (como mostrado na Tabela 1). Como a MobileNet obteve o melhor resultado, este teste foi executado de forma pareada, comparando a MobileNet com todos os demais modelos. A hipótese nula é que os dois algorítimos testados devem ter o mesmo desempenho. Por outro lado, rejeitar a hipótese nula, significa que os classificadores apresentam desempenhos diferentes na base de avaliação.

Tabela 1. Resultados obtidos bem como total de parâmetros das arquiteturas, tempo de execução e número de épocas utilizado.

Arquitetura	Épocas	Tempo de execuçao (horas)	Total de parâmetros (x10^6)	Acurácia	F1-score	Precisão	Revocação
MobileNet	40	17.3	28.7	0.789	0.781	0.809	0.754
VGG16	40	17.0	22.5	0.728	0.718	0.741	0.697
RestNet-152	40	23.2	100.2	0.682	0.733	0.629	0.878
Inception v3	40	17.2	34.4	0.560	0.613	0.546	0.700

O teste de McNemar é baseado em um teste χ^2 para verificar se os valores obtidos na tabela de contingência são esperados sob a hipótese nula. O valor $\frac{(|N_{01}-N_{10}|-1)^2}{N_{01}+N_{10}}$ é distribuído de acordo com χ^2 com 1 grau de liberdade. Neste casso, N_{00} é o número de exemplos classificados incorretamente pelos dois modelos, N_{01} é o número de exemplos classificados corretamente pelo segundo classificador e incorretamente pelo primeiro, N_{10} é o número de exemplos classificados corretamente pelo primeiro classificador e incorretamente pelo segundo, e N_{11} é o número de exemplos corretamente classificados pelos dois. Se a hipótese nula estiver correta, então a probabilidade que esse valor seja maior que $\chi^2_{0.95} = 3.841459$ é menor que 0.05. Como pode-se ver na Tabela 2, o valor obtido pelo teste de McNemar em cada uma das comparações do MobileNet com os demais modelos foi maior que o valor 3.841459. Isto significa que a hipótese nula foi rejeitada e que a MobileNet possui um desempenho diferente estatisticamente dos demais modelos. Em outras palavras, a MobileNet alcançou resultados estatisticamente superiores aos das demais arquiteturas quando avaliadas no HEAR DATASET.

Tabela 2. Resultados dos testes de McNemar comparando o modelo gerado com a arquitetura MobileNet com as demais, demonstrando que o modelo gerado pela MobileNet foi, de fato, superior.

Modelo A	Modelo B	Valor	Resultado
MobileNet	ResNet	171.648	Hipótese nula rejeitada
MobileNet	VGG	67.507	Hipótese nula rejeitada
MobileNet	Inception	556.726	Hipótese nula rejeitada

6. Conclusão e Trabalhos Futuros

Este trabalho se propõe a investigar o desempenho de CNNs na detecção de violência física por meio do áudio ambiente. No entanto, ao invés de serem treinadas com o sinal de áudio diretamente, os áudios foram convertidos em *mel-spectrograms* (imagens) usados no treinamento dos modelos. Com isso, torna-se possível usar técnicas e ferramentas consolidadas da área de imagens, mas para o contexto de áudio. Esta abordagem não havia sido investigada ainda no domínio de detecção de violência em áudios. Nesta pesquisa, foram considerados quatro modelos de CNN: Inception v3, VGG-16, MobileNet v2 e ResNet 152 v2. Para fins de treino e teste dos modelos, foi criada a HEAR DATASET, uma base de dados sintética balanceada, composta de áudios de violência e não-violência com origem no AudioSet e FreeSound Dataset e que foram "mixadas" com uso do Scaper. Os resultados mostraram que a MobileNet obteve um desempenho estatisticamente superior aos demais, alcançando valores de acurácia, f_1 score e precisão promissores. Como trabalhos futuros, pretende-se comparar o modelo acima desenvolvido com outro desenvolvido por meio de *Transfer Learning* a partir de uma *Large-Scale Pretrained Audio Neural Networks* (PANNs) [Kong et al. 2020]. Além disto, pretende-se testar os modelos em áudios de violência real, embarcar o modelo em um aplicativo e, por fim, desenvolver estudos sobre *Federated Learning* neste contexto.

Referências

Abeßer, J. (2020). A review of deep learning based methods for acoustic scene classification. 10(6):2020.

Chen, G., Parada, C., and Heigold, G. (2014). Small-footprint keyword spotting using deep neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4087–4091. IEEE.

Crocco, M., Cristani, M., Trucco, A., and Murino, V. (2016). Audio surveillance: A systematic review. 48(4):1–46.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. 10(7):1895–1923.

Durães, D., Marcondes, F. S., Gonçalves, F., Fonseca, J., Machado, J., and Novais, P. (2021). Detection violent behaviors: A survey. In Novais, P., Vercelli, G., Larriba-Pey, J. L., Herrera, F., and Chamoso, P., editors, *Ambient Intelligence – Software and Applications*, pages 106–116. Springer International Publishing.

Fonseca, E., Favory, X., Pons, J., Font, F., and Serra, X. (2020). FSD50k: an open dataset of human-labeled sound events.

Fonseca, E., Pons, J., Favory, X., Font, F., Bogdanov, D., Ferraro, A., Oramas, S., Porter, A., and Serra, X. (2017). Freesound datasets: A platform for the creation of open audio datasets. page 8.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. 36(4).

Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. (2017). Audio set: An ontology and human-labeled dataset for audio events.

In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780. IEEE.

Giannakopoulos, T., Kosmopoulos, D., Aristidou, A., and Theodoridis, S. (2006). Violence content classification using audio features. In Antoniou, G., Potamias, G., Spyropoulos, C., and Plexousakis, D., editors, *Advances in Artificial Intelligence*, volume 3955, pages 502–507. Springer Berlin Heidelberg. Series Title: Lecture Notes in Computer Science.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Khan, A., Sohail, A., Zahoora, U., and Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. 53(8):5455–5516.

Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., and Plumbley, M. D. (2020). PANNs: Large-scale pretrained audio neural networks for audio pattern recognition. 28:2880–2894.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. 60(6):84–90.

Mesaros, A., Heittola, T., and Virtanen, T. (2016). Metrics for polyphonic sound event detection. 6(6):162.

Nordby, J. (2019). Environmental sound classification on microcontrollers using convolutional neural networks. page 70.

Piczak, K. J. (2015). ESC: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018. ACM.

Rouas, J.-L., Louradour, J., and Ambellouis, S. (2006). Audio events detection in public transport vehicle. In *2006 IEEE Intelligent Transportation Systems Conference*, pages 733–738. IEEE.

Salamon, J., Jacoby, C., and Bello, J. P. (2014). A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044. ACM.

Salamon, J., MacConnell, D., Cartwright, M., Li, P., and Bello, J. P. (2017). Scaper: A library for soundscape synthesis and augmentation. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 344–348. IEEE.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520. IEEE.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition.

Souto, H., Mello, R., and Furtado, A. (2019). An acoustic scene classification approach involving domestic violence using machine learning. In *Anais do ENIAC*, pages 705–716.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). Rethinking the inception architecture for computer vision.

Wu, P., Liu, J., Shi, Y., Sun, Y., Shao, F., Wu, Z., and Yang, Z. (2020). Not only look, but also listen: Learning multimodal violence detection under weak supervision.