EfficientNet para o Monitoramento Automático de Publicidades de Alimentos

Victória Pedrazzoli Ferreira
Paula Martins Horta (Coorientadora)
Sandra Eliza Fontes de Avila (Orientadora)

Relatório Técnico – IC-PFG-24-32
Projeto Final de Graduação
2024 – Outubro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Sumário

EfficientNet para o Monitoramento Automático de Publicidades de Alimentos

Victória Pedrazzoli Ferreira* Paula Martins Horta†
Sandra Eliza Fontes de Avila‡

Resumo

No cenário atual, nota-se cada vez mais o aumento do número de pessoas portadoras de obesidade, diabetes e outras doenças crônicas. Os especialistas em saúde e nutrição admitem que a combinação do consumo de alimentos processados, em conjunto com estratégias de publicidade, é uma das causas dessa epidemia global. Em resposta a esse reconhecimento, a Organização Mundial de Saúde (OMS) têm apontado a relevância de se monitorar e restringir esse tipo de publicidade. As pesquisas de monitoramento de publicidades televisivas revelam a prevalência de anúncios alimentícios sem o perfil nutricional favorável e a utilização de estratégias persuasivas. Este projeto de final de graduação investigou técnicas de aprendizado de máquina para o monitoramento automático de publicidades de alimentos em canais de televisão no Brasil. Mais especificamente, analisou a efetividade da EfficientNet como um método de classificação de publicidades alimentícias nos canais de TV brasileiros. Como conclusão, a rede analisada em conjunto com as técnicas estudadas (e.g., *data augmentation*, *batches* balanceados, adição de peso por classe) se mostram extremamente promissoras para a realização dessa classificação automática. Destacamos que este tipo de pesquisa é nova por si mesma e ainda não foi desenvolvida para o monitoramento automático de publicidades de alimentos.

1 Introdução

O excesso de peso é um problema de saúde pública que atinge, de maneira crescente, o mundo inteiro. Estimativas da Organização Mundial de Saúde mostram que, em 2016, mais de 1,9 bilhões de adultos estavam acima do peso, sendo mais de 650 milhões de obesos [12]. No Brasil, a prevalência de obesidade na população adulta também aumentou consideravelmente em menos de 20 anos [4].

O crescente aumento do consumo de alimentos ultraprocessados e, por consequência, da redução da participação de alimentos in natura, ou minimamente processados, na dieta dos brasileiros é uma das principais causas desse notável ganho de peso, visto que

*Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP. Pesquisa desenvolvida com suporte financeiro parcial do PIBIC/SAE/CNPq.

†Escola de Enfermagem, Departamento de Nutrição, Universidade Federal de Minas Gerais, Belo Horizonte, MG.

‡Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP.

temos uma participação excessiva de comidas que apresentam baixo teor de fibras e alta densidade energética, gorduras saturadas, açúcar e sódio [1,7–9].

A publicidade de alimentos tem grande participação nessa mudança de dieta pois ela pode influenciar negativamente o consumo alimentar dos indivíduos, por normalmente estar centrada em alimentos não saudáveis e visar estratégias de publicidade persuasivas, a fim de impactar no comportamento do consumidor, acarretando na fidelização pela marca desde a mais tenra idade [3,10,17,18].

Em 2010, a Organização Mundial da Saúde recomendou a redução da exposição das crianças à propaganda de alimentos, sobretudo aqueles ultraprocessados [11]. Mas, levando em conta as condições atuais, percebe-se que as iniciativas de regulação da propaganda para frear a massivo publicidade da indústria alimentícia não foram suficientes.

No caso do Brasil, apesar de existir uma regulamentação por parte da Agência Nacional de Vigilância Sanitária (Anvisa), os órgãos de fiscalização ainda não possuem força suficiente para colocá-la em prática. Atualmente, as publicidades classificadas como abusivas pela regulamentação são identificadas somente após já terem atingido um grande número de pessoas.

Visto que qualquer exposição a uma publicidade definida como abusivo é nociva, principalmente em uma idade jovem graças a fácil persuasão por parte da empresa para o consumidor, faz-se necessária a criação de um monitoramento automático de publicidades de alimentos e bebidas não alcoólicas.

Neste cenário, o principal objetivo deste projeto foi desenvolver um método que auxilie o monitoramento automático de publicidades de alimentos de canais de TV no Brasil. Mais especificamente, concentramos nossos esforços na investigação de técnicas de aprendizado em redes neurais profundas — e das adaptações necessárias dessas técnicas — para o monitoramento automático de publicidades de alimentos veiculadas na TV brasileira.

As principais contribuições são listadas a seguir:

1. Desenvolvimento de um método baseado nas redes EfficientNet para classificação de vídeos de publicidades de alimentos. Destacamos que este tipo de pesquisa é nova por si mesma e ainda não foi desenvolvida para o monitoramento automático de publicidades de alimentos.

2. Tratamento e análise de dados brutos de publicidades alimentícias coletados nos canais de TV brasileiros. Esta é a primeira base de dados de publicidades alimentícias e suas estratégias de marketing do país.

2 Fundamentação Teórica

Até onde sabemos, não existem trabalhos voltados para o monitoramento automático de publicidades de alimentos, o que ressalta a proposta original deste projeto de final de graduação. Portanto, visto que não existe um outro estudo diretamente relacionado a esta proposta, após uma extensa análise da literatura referente à classificação de imagens e vídeos de publicidades no âmbito do aprendizado de máquina, notou-se que o estado da arte para esse tipo de análise são as redes neurais profundas [2,14,15,24,28].

Dentre as arquiteturas de redes neurais profundas que têm ganhado bastante destaque no âmbito da compreensão de vídeos estão as Transformers [27]. Estas consistem de um modelo de aprendizado profundo que procura pesar a influência de diferentes partes dos dados de entrada através de um mecanismo de atenção, conhecido como auto-atenção (*self-attention*).

Uma das aplicações mais atuais desse tipo de abordagem é o *Data-Efficient Image Transformers* (DeiT) [23], que busca acelerar o processo de treinamento de uma Transformer classificadora de imagens através da aplicação de uma estratégia de *teacher-student* e de um *token* de destilação.

De maneira geral, esse tipo de estratégia de treinamento parte do conceito de que um outro modelo, no caso do DeiT [23], ajuda a determinar a classificação durante o processo de aprendizado da Transformer, a fim de acelerar esse processo. Para tanto, foi necessária a inserção de um *token* de destilação, que carrega o rótulo estimado pelo *teacher* (uma rede neural profunda) durante o processo de treinamento e aplicação do mecanismo de atenção da Transformer. Os resultados do modelo DeiT, quando pré-treinados ou não na ImageNet [16], foram competitivos para as base de dados de imagens mais relevantes da literatura, como a própria ImageNet [16], a iNaturalist 2019 [25], a CIFAR-100 [6] e a CIFAR-10 [6].

Outro modelo bastante popular para a classificação de imagens são as redes neurais convolucionais, nas quais a EfficientNet [21] particularmente se destaca. Ela foi desenvolvida para se aproveitar da intuição de que quanto maior for a imagem de entrada, a rede precisaria de mais camadas para aumentar o campo receptivo e mais canais para capturar padrões mais refinados na imagem maior.

Dessa maneira o modelo procura escalar uniformemente todas as dimensões (profundidade, largura e resolução) das imagens usando um coeficiente composto, que é especificado para o usuário, e dessa maneira ajudar a controlar quantos recursos computacionais ainda estão disponíveis para o dimensionamento do modelo, deixando para outros três coeficientes especificarem como atribuir os recursos extras para cada dimensão.

Os resultados da EfficientNet também atingem o estado da arte para os principais benchmarks. A EfficientNet é a arquitetura explorada neste projeto e, portanto, é detalhada a seguir.

2.1 Arquitetura EfficientNet

O reconhecimento de imagem é um clássico problema de classificação, e as redes neurais convolucionais (*convolutional neural networks*, CNNs), especificamente as EfficientNets [21], são o estado da arte para esse problema.

A arquitetura EfficientNet foi construída a partir de algumas intuições, como por exemplo a ideia de quando aumentamos a resolução da imagem de entrada, o modelo se beneficiaria por um aumento de profundidade e largura. Também se baseia na noção de que e a fim de buscar melhor precisão e eficiência, é fundamental equilibrar todas as dimensões da rede.

Assim, a rede busca melhorar a precisão do modelo escalando uniformemente em todas as direções com grande eficiência através de uma constante (coeficiente composto). As fórmulas de escalonamento para cada direção levam como base o recurso computacional disponível. O coeficiente composto e um conjunto fixo de três coeficientes de dimensionamento (um para cada dimensão) são determinados por *grid search*.

Esse método de dimensionamento composto é feito em duas etapas. Na primeira etapa, um coeficiente composto é fixado como um (assumindo duas vezes mais recursos disponíveis) e os coeficientes para cada dimensão são determinados através do *grid search*. Em seguida, os valores encontrados são definidos como constantes e o escalonamento da rede é feito usando o coeficiente composto determinado empiricamente para a versão da EfficientNet que está sendo treinada.

A eficácia do modelo também depende muito da arquitetura base da rede (EfficientNet-B0). Para tanto, uma nova arquitetura usando a estrutura AutoML MNAS[1] foi proposta, o que também otimiza a precisão e a eficiência (FLOPS). A solução proposta pela rede é semelhante a MobileNetv2 [19] e a MnasNet [20], e também usa uma camada de *bottleneck convolution* invertida (MBConv) com convolução separável inteligente, mas com adição das técnicas de *squeeze-excitation*.

As convoluções separáveis adotam um truque para imobilizar uma única convolução 3×3 em duas outras a fim de reduzir o número de parâmetros. Primeiro se aplica um único filtro 3×3 aos canais de cada entrada, e depois um filtro 1×1 a todos os canais o que corresponde a uma convolução 3×3 direta, mas com menos quantidade de parâmetros. A primeira convolução é geralmente chamada de *depth*, e a segunda de *point*. Em seguida, a normalização é aplicada a ambas convoluções (Figura 1).

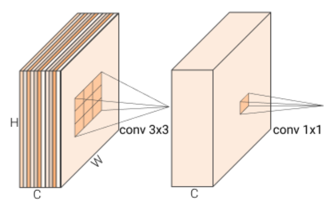


Figura 1: Convolução 3×3 feita em duas etapas. Figura reproduzida de TowardsDataScience (`https://towardsdatascience.com/mobilenetv2-inverted-residuals-and-linear-bottlenecks-8a4362f4ffd5`).

Dessa maneira, os mapas de ativação de entrada são primeiro expandidos usando convoluções 1×1 (para aumentar a profundidade), seguidos pelas convoluções *depth* e *point* respectivamente, o que reduz o número de canais na saída. As camadas mais estreitas são ligadas as largas por conexões chamadas *shortcut*, o que também ajuda a diminuir o número geral de operações necessárias, bem como o tamanho do modelo. É esta a estrutura dos principais blocos da rede que é retratada na Figura 2.

No caso da EfficientNet, a MBConv é um pouco maior devido a um aumento dos FLOPS, em seguida encontra-se o aumento da escala da rede. Com isso está formada a arquitetura do modelo, retratada na Figura 3.

[1] `https://www.automl.org/nas-overview`

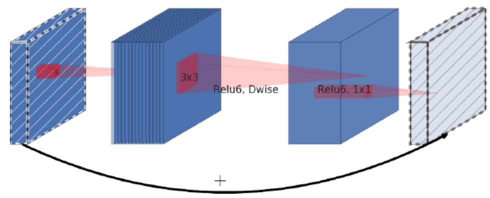


Figura 2: Bloco residual invertido de uma EfficientNet. Figura reproduzida de Towards-DataScience (`https://towardsdatascience.com/efficientnet-scaling-of-convolutional-neural-networks-done-right-3fde32aef8ff`).

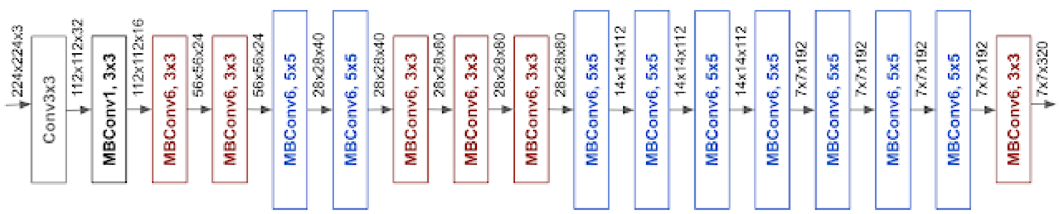


Figura 3: Arquitetura de uma EfficientNet. Figura reproduzida de Google AI blog (`https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html`).

3 Metodologia

Para todos os experimentos, utilizamos o código original da rede EfficientNet[2]. Nenhuma característica fundamental do modelo foi modificada, as únicas alterações durante o treinamento incluem a adição de técnicas para contornar o desbalanceamento de dados e testes de combinações diferentes de taxas de aprendizado e *dropout*.

Para a avaliação dos modelos, foram coletadas as curvas de acurácia e *loss*, tanto para o treinamento quanto para a validação. Extraímos também as matrizes de confusão antes e depois do *pooling*, precisão, *recall*, f1-score e suporte de cada classe.

3.1 Base de Dados

Este projeto foi realizado em parceria com o Departamento de Nutrição da Universidade Federal de Minas Gerais, que foi responsável pela construção, seleção e anotação da base de dados, e também pela validação dos resultados obtidos para a proposição de um novo método.

Para a construção da base de dados, foram escolhidos para monitoramento três canais da TV aberta (*Globo*, *Record* e *SBT*) examinados nos períodos de Abril de 2018, Maio de 2019 e Junho de 2020, e dois canais da TV fechada (*Discovery Kids* e *Cartoon Network*) monitorados nos períodos de Maio de 2019, Setembro de 2019 e Junho de 2020 (veja a Figura 4). Até o momento, a base de dados contém 603 vídeos distintos de publicidades alimentícias e mais de 20,000 vídeos não alimentícios passaram por análise.

[2]`https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet`

(a) Publicidades alimentícias

(b) Publicidades não alimentícias

Figura 4: Exemplos de publicidades coletadas e veiculadas nos canais abertos da TV brasileira.

Para a coleta de informações e anotação dos dados, foi aplicado o protocolo INFORMAS (*International Network for Food and Obesity/NCDs Research, Monitoring and Action Support*) [26]. Entre as informações coletadas estão: nome do canal, a data da gravação, o nome do programa, o horário de início e término do anúncio e o tipo de publicidade. Em seguida, especificamente para aquelas publicidades envolvendo alimentos, também foram anotadas o nome da marca ou empresa, nome e descrição do produto e a categoria do alimento, além de serem investigadas e classificadas as suas estratégias de publicidade, divididas entre 28 estratégias, pertencentes a um dos três seguintes grupos: poder das estratégias de publicidade, uso da oferta de prêmios e uso de alegações de benefícios da marca. As estratégias não foram utilizadas na metodologia proposta.

Preparação da Base de Dados

Após esse levantamento de estratégias, iniciamos a preparação da base. Como os vídeos entregues pela equipe da UFMG não estavam separados por publicidades, foi necessário automatizar o processo de corte dos vídeos da programação completa de cada canal, através das *timestamps* anotadas pela equipe da UFMG. Para tanto, foi criado um código em Python que utiliza a ferramenta `ffmpeg` [22] para auxiliar esse processo.

Com essa etapa pronta, identificamos a presença notável de diversas publicidades duplicadas nos vídeos recentemente cortados, correspondendo a cerca de 80% dos vídeos coletados. As cópias consistiam em dois tipos: algumas eram duplicatas perfeitas, enquanto outras seriam versões reduzidas de uma propaganda original. Para resolver esse problema, foi proposta uma maneira eficiente (descrita a seguir) de remover esses dois tipos de duplicatas.

Para remover as duplicatas, aplicamos funções de *average hash*[3], visto que se trata de uma abordagem barata, rápida e eficiente. A abordagem consiste em calcular um *hash* para cada *frame* do vídeo. Quando duas propagandas tiverem um número suficiente de *hashs* iguais, ou seja, quando esse valor fosse maior que um certo limiar, elas são consideradas cópias uma da outra.

Dessa maneira, fizemos a uma série de experimentos para determinar uma taxa eficaz para realizar a remoção de duplicatas. Através desses experimentos, percebemos que a natureza das propagandas dos canais abertos e fechados é um pouco diferente, visto que as duplicatas são mais comuns nos canais abertos. Assim, inicialmente, foram definidas duas taxas, uma para canais abertos e outra para canais fechados. Em seguida, notou-se que as propagandas também se repetem entre canais, então uma terceira taxa foi definida, que é aplicada para todos os vídeos coletados.

No entanto, apenas essa definição não bastou para que obtivéssemos um bom resultado, uma taxa restritiva demais mantinha um número alto de duplicatas, enquanto uma mais abrangente removia propagandas parecidas, com mesmo atores, produtos ou cores parecidas, mas que não eram duplicatas.

Para resolver esse impasse, adotamos o seguinte pipeline. Primeiramente, removemos as cópias existentes em um mesmo canal, passando as publicidades por 3 taxas distintas (de maneira subsequente). As taxas são definidas pela natureza do canal (aberto ou fechado) e também se tornam cada vez menos restritivas. Para os canais abertos *Record*, *Globo* e *SBT* o valor de cada taxa é 40, 20 e 10 enquanto para os canais fechados *Discovery Kids* e *Cartoon Network* determinou-se 50, 25 e 13.

Contudo, quando um vídeo é identificado como a versão original de algum outro, ele não mais poderá ser considerado duplicata de nenhuma publicidade, ou seja, apenas outras propagandas que não foram definidas como ”originais”podem ser cópias dele. Dessa maneira, conseguimos remover todas as duplicatas dentro de um mesmo canal, perdendo quase nenhuma informação. Em seguida, repetimos o mesmo processo, agora com outras 4 taxas específicas (60, 30, 25, 20) para tratar as publicidades selecionadas para os 5 canais, a fim de remover as cópias entre canais.

Assim, ao final do processo, todos os vídeos duplicados são removidos, com a menor perda de dados possível. Essa divisão em duas etapas (uma por canal e só depois a coleta completa) foi necessária pois, dado a natureza de cada canal, tentar fazer a remoção em uma única etapa se provou (de forma empírica) menos eficaz através dos nossos experimentos.

Destacamos que essa é a primeira base de dados desse tipo no Brasil. Um dos trabalhos futuros deste projeto é disponibilizá-la publicamente. Para isso, seguiremos as recomendações feitas no artigo *Datasheets for Datasets* [5].

Separação da Base de Dados

Após a preparação da base de dados, foi iniciada a divisão da base de dados para o treinamento dos modelos. Para tanto, a “base inicial de dados” foi dividida em três novas

[3] `https://github.com/gklc811/duplicate_video_finder`

identificadas na Figura 5: uma foi destinada para treinar os algoritmos (Base 1, que contém grande parte dos vídeos de 2020), uma foi destinada para validação dos resultados (Base 2, que contém os vídeos de 2019) e a última foi destinada ao teste final do modelo (Base 3, que contém os vídeos de 2018 e uma pequena parcela dos vídeos de 2020). Essa divisão é necessária para evitar o enviesamento dos modelos ao serem executados em novos dados. Ela foi feita de maneira que a grande parte dos dados foram destinados ao treinamento do modelo, mas de forma que tanto a Base 2, quanto a Base 3 ainda são capazes de representar a totalidade destes. Mesmo assim, o modelo foi treinado de maneira a contornar desbalanceamentos, visto a maioria das publicidades são de propagandas não alimentícias. Isso foi feito através de estratégias, que serão discutidas posteriormente.

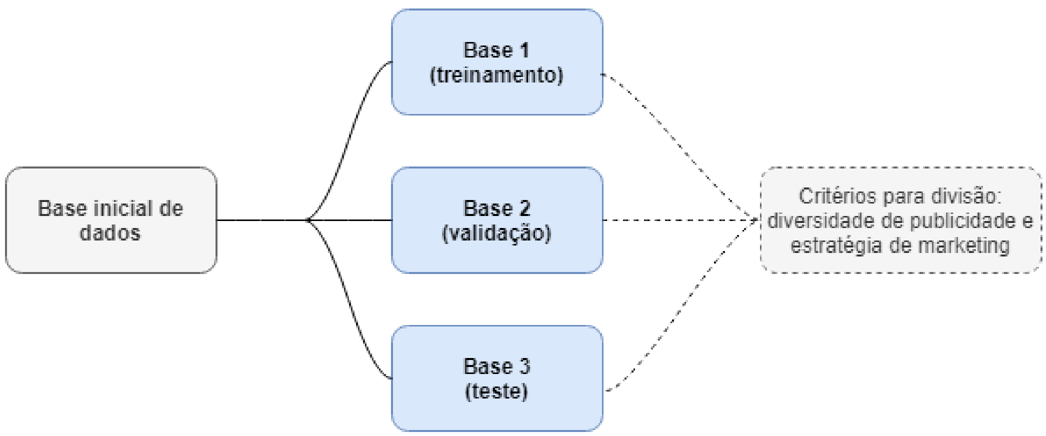


Figura 5: Síntese da metodologia que compreende a etapa de divisão da base de dados.

3.2 Abordagem Proposta

Uma visão geral do fluxograma da abordagem proposta é ilustrada na Figura 6. Primeiramente, utilizando os dados da Base 1 (dados de treinamento), extraímos um certo número de *frames* para cada vídeo. Em seguida, cada um desses *frames* passa pela rede neural escolhida (a EfficientNet) gerando uma classificação para cada *frame*. Para obtenção de uma única classificação para o vídeo, o resultado obtido para cada *frame* de um mesma publicidade deve passar por um processo de fusão através de um método de *pooling*. Com isso, determinamos os melhores modelos e estratégias aplicados para serem validados com os vídeos separados na Base 2 (dados de validação). Essa última etapa serve para indicar a necessidade de alguma adaptação nas técnicas aplicadas. Caso não seja necessário, o melhor modelo obtido até o momento é selecionado para ser testado na Base 3 (dados de teste) e assim finalmente se tem acesso aos resultados finais que serão analisados por especialistas da área de nutrição.

O código desenvolvido neste projeto está disponível em `https://github.com/vic-pf/PFG`.

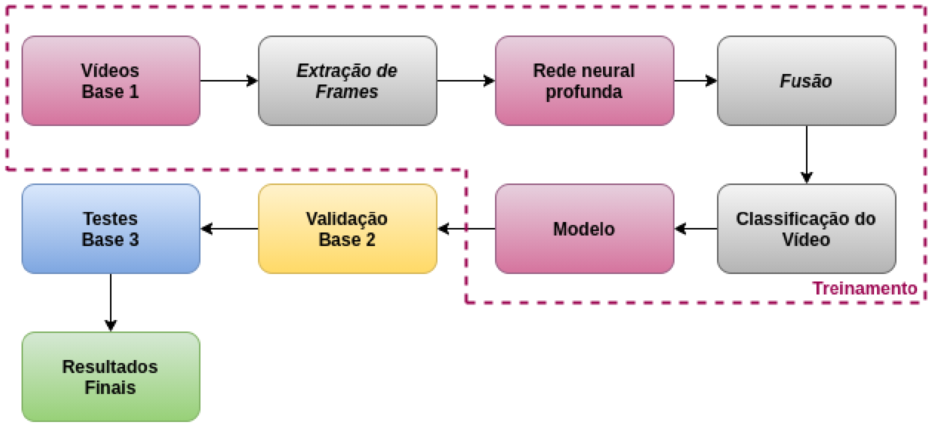


Figura 6: Fluxograma da abordagem proposta.

3.2.1 Rede Neural Profunda: EfficientNet

Optamos por utilizar a rede neural profunda EfficientNet [21] como modelo, visto que como a mesma é o estado da arte na tarefa de classificação de imagens. Além disso, ela é uma boa referência comparativa para uma abordagem mais recente como o *Data-Efficient Image Transformers* (DeiT) [23].

Para validar a eficiência do modelo em um cenário conhecido, realizamos alguns experimentos com a base de dado TF Flowers[4] que contém 3670 imagens divididas nas seguintes classes: *dandelion*, *daisy*, *tulips*, *sunflowers* e *roses*. Fizemos experimentos iniciais (como *baseline*) na Base 1, a fim de averiguar a performance da mesma e levantarmos algumas métricas úteis. Para tanto, executamos a extração de apenas um (*frame*) por vídeo (exatamente o do *frame* do meio) e em seguida fizemos a classificação destes para obtermos um bom resultado comparativo para as técnicas aplicadas posteriormente. Os principais resultados estão descritos na Seção 4.

3.2.2 Abordagem para Classificação

Usando Base 1, propomos a abordagem para classificação de imagens e vídeos de publicidades de alimentos. Primeiramente, separamos uma pequena parcela dos vídeos da Base 1 para fazer uma pré-validação dos resultados, a fim de diminuir o enviesamento do modelo e selecionar os melhores modelos para serem validados. Para a divisão desses dados, analisamos duas separações: uma aleatória e outra manual. A segunda teve como critério de separação garantir que nosso grupo de pré-validação alimentício não contivesse apenas propagandas de um único anunciante, ou seja, que representasse bem o nosso conjunto de treinamento e sua diversidade. Além disso, garantimos que em ambos casos, o grupo destinado ao treinamento do modelo tivesse a mesma porcentagem de publicidades alimentícias e não alimentícias, ou seja, separamos os conjuntos de forma balanceada.

[4] https://www.tensorflow.org/datasets/catalog/tf_flowers

Como entrada da rede neural, foram usados os quadros brutos dos vídeos de publicidades alimentícias em uma amostra de um *frame* por segundo, enquanto que para os vídeos de publicidades não-alimentícias os frames foram retirados da seguinte forma, onde T é a duração do vídeo em segundos:

- 1/2 *frames* por segundos, se $T \leq 10$;
- 1/4 *frames* por segundos, se $10 < T \leq 25$;
- 1/6 *frames* por segundos, se $25 < T \leq 35$;
- T/15 *frames* por segundos, se $35 < T \leq 60$;
- T/30 *frames* por segundos, se $60 < T \leq 120$;
- T/45 *frames* por segundos, se $120 < T$.

Essa decisão foi tomada a fim de minimizar o desbalanceamento notado durante os testes anteriores. Todos os *frames* foram padronizados e centralizados de acordo com a métrica compatível ao tamanho de entrada do modelo escolhido: para a EfficientNet (considerando as suas diferentes versões), a resolução varia de 224×224 a 600×600 pixels.

Dessa forma, cada *frame* retirado das publicidades teve sua própria classificação e, para determinar a categoria de um vídeo, passamos os resultados de cada *frame* por uma abordagem de tomada de decisão separada. Optamos por uma abordagem de *pooling* para combiná-las em uma única classificação final por vídeo, através de uma média simples das classificações.

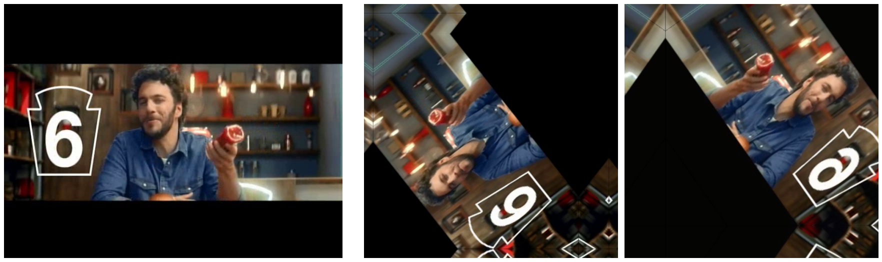
No entanto, no decorrer deste projeto, percebemos que apenas esse tipo de abordagem de retirar os *frames* dos vídeos a taxas diferentes dependendo da classe não foi suficiente para contornar o desbalanceamento de dados. Para solucionar o problema, aplicamos três estratégias distintas a fim de encontrar o melhor resultado: *data augmentation* [13], *batches* balanceados e adição de peso por classes. Vale destacar que o código foi desenvolvido de maneira a poder aplicar qualquer combinação dessas técnicas como for desejado, apesar de não necessariamente todas as combinações fazerem sentido. A seguir, detalhamos cada uma das estratégias.

Data Augmentation

Para o processo de *data augmentation*, aplicamos as seguintes transformações para os *frames* extraídos dos vídeos de publicidades alimentícias: rotação, translação, zoom, adição ou remoção de contraste e giros no eixo vertical e/ou horizontal (Figura 7). Isso foi feito através do uso da classe `keras.Sequential`[5], que além de permitir uma definição de taxa para realizar cada uma essas transformações de forma aleatória, também possibilita a aplicação de mais de uma transformação por imagem/*frame*.

Dessa maneira, ao fim desse processo, diminui-se o desbalanceamento de dados entre as classes de forma aleatória, sem que o conteúdo carregado por cada *frame* aumentado seja exatamente o mesmo, e nem se torne irreconhecível.

[5] `https://keras.io/api/models/sequential`



(a) *Frame* original de uma publicidade alimentícia

(b) Mesmo *frame* após transformações

Figura 7: Exemplos das transformações de *data augmentation* realizadas.

Batches Balanceados

Para obter *batches* balanceados, criamos o nosso próprio *data generator*, ou seja, ao definir o tamanho do *batch* como X, a quantidade de imagens para cada classe estaria próxima ou exatamente igual a X/2.

Para tanto, seguimos exemplos de abordagens populares na internet, entre elas destacamos o `BalancedDataGenerator`[6], que se propõe a solucionar essa questão através da biblioteca Python `imbalanced-learn`[7]. O código em questão cria um *data generator* balanceado, repetindo amostras aleatórias da base de dados, caso não houvessem imagens suficientes para atingir um equilíbrio entre as classes dentro do *batch* em questão.

Dessa maneira, ao fim desse processo, contornamos o desbalanceamento de dados entre as classes de forma diferente a discutida anteriormente. Também destacamos a possibilidade de aplicar ambas as técnicas em conjunto, diminuindo o número de amostras iguais que seriam necessárias para garantir o balanceamento dos *batches*. No entanto, a aplicação desta técnica necessita que todo o conjunto de dados seja carregado inteiramente na memória, o que se torna a sua aplicação restrita ao tamanho da memória.

Adição de Peso por Classe

Também decidimos adicionar pesos para cada classe durante o processo de classificação de uma imagem. Assim, podemos contornar o desbalanceamento adicionando um peso maior para a classe alimentícia (que possui menos dados), e um peso menor para a não alimentícia (visto que ela tem mais entradas). Isso foi feito através do uso da biblioteca Python `scikit-learn`[8] que possui uma função que faz esse cálculo de peso automaticamente e de maneira balanceada.

Novamente, destacamos a possibilidade de aplicar essa técnica em conjunto com *data augmentation*. Combinada com esta outra abordagem, primeiramente podemos aumentar

[6] https://gist.github.com/arnaldog12/16efc663c869b35e2479bd607d56c1da

[7] https://imbalanced-learn.org/stable

[8] https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

o número de imagens da classe desbalanceada, o que por consequência diminui diferença de peso entre cada classe, mas sem a necessidade de repetir dados até atingir o equilíbrio, visto que a classe desbalanceada ainda pode ser favorecida através da aplicação dos pesos. No entanto, a aplicação de peso em conjunto com o balanceamento de *batches*, apesar de ser possível, não faz tanto sentido, já que são propostas exclusivas para solucionar um mesmo problema.

3.3 Validação do Modelo

Por fim, validamos os melhores modelos na Base 2, a fim de certificar se a taxa de acerto destes se mantém. Além disso, buscamos encontrar fontes de erros preditivos do modelo, que serão discutidas posteriormente.

As métricas utilizadas para avaliar os modelos são a acurácia balanceada e matriz de confusão.

4 Resultados

Os principais resultados dos modelos treinados foram retratados pelas matrizes de confusão por vídeo dos conjuntos de validação a seguir. Estão retratados nessa matriz a porcentagem de número de falsos positivos, falsos negativos, verdadeiros positivos e verdadeiros negativos, onde a classe positiva é a alimentícia e a classe negativa a não alimentícia. Os experimentos seguem a ordem das abordagens descritas na seção anterior. Cada tabela representa a execução de um experimento específico.

4.1 Treinamento dos Modelos na Base 1

4.1.1 Base separada aleatoriamente

A Tabela 1 representa o primeiro experimento (Baseline). Enquanto que as Tabelas 2 e 3 são os resultados para *data augmentation* com balanceamento de *batches* e *data augmentation* com adição de peso por classe, respectivamente.

4.1.2 Base separada manualmente

A Tabela 4 representa os resultados para *data augmentation* com adição de peso por classe na base separada manualmente.

4.2 Validação na Base 2

A Tabela 5 apresenta os resultados da validação do melhor modelo separado aleatoriamente.

Classe	**Alimentícia**	**Não Alimentícia**
Alimentícia	23%	77%
Não Alimentícia	8%	92%

Tabela 1: **Baseline.** Matriz de confusão da classificação de cada classe. EfficientNet-B0 para classificação de um *frame* por vídeo (o do meio).

Classe	**Alimentícia**	**Não Alimentícia**
Alimentícia	73%	27%
Não Alimentícia	0%	100%

Tabela 2: ***Data augmentation* com Balanceamento de *batches*.** Matriz de confusão da classificação de cada classe. EfficientNet-B4 para classificação de um *frame* por vídeo.

Classe	**Alimentícia**	**Não Alimentícia**
Alimentícia	86%	14%
Não Alimentícia	0%	100%

Tabela 3: ***Data augmentation* com Adição de peso por classe.** Matriz de confusão da classificação de cada classe. EfficientNet-B7 para classificação de mais de um *frame* por vídeo.

Classe	**Alimentícia**	**Não Alimentícia**
Alimentícia	100%	0%
Não Alimentícia	1%	99%

Tabela 4: ***Data augmentation* com Adição de peso por classe.** Matriz de confusão da classificação de cada classe. EfficientNet-B7 para classificação de um *frame* por vídeo.

Classe	**Alimentícia**	**Não Alimentícia**
Alimentícia	87%	13%
Não Alimentícia	1%	99%

Tabela 5: ***Data augmentation* com Adição de peso por classe.** Matriz de confusão da classificação de cada classe. Validação do modelo 3 no conjunto de 2019.

4.3 Acurácia Balanceada

A Tabela 6 apresenta a acurácia balanceada de todos os experimentos retratados nesse projeto.

Tipo de Separação	**Técnica**	**Acurácia Balanceada (%)**
Aleatória	Baseline	57.5
	Augmentation com Balanceamento de batches	86.5
	Augmentation com Adição de peso	93.0
	Validação na Base 2	93.0
Manual	Augmentation com Adição de peso	99.5

Tabela 6: Acurácia balanceada de cada um dos experimentos apresentados.

5 Discussão

Diante dos resultados, notamos que a utilização da EfficientNet combinada com estratégias para amenizar o desbalanceamento de dados trouxe grandes melhorias quando levamos em conta nosso resultado inicial (Baseline, representado na Tabela 1).

Entre essas técnicas, a aplicação de peso por classe combinada com aumentação de dados é a que apresentou o melhor resultado visto a sua performance durante nossa validação. A técnica de balanceamento de *batches* não foi tão promissora porque há necessidade de reduzir o modelo para a versão B4, justamente por conta do carregamento em memória imperativo para essa abordagem. Ou seja, a memória disponível foi fator limitante, o que indica a necessidade de refatoração do código de maneira a evitar o carregamento completo do dataset em memória, para que se possa aplicar o modelo EfficientNet-B7 e se reconsiderar o uso da técnica de *batches* balanceados. Mesmo assim, os resultados obtidos pelas outras abordagens já foram por si só satisfatórios.

Em relação às técnicas de separação da base de dados para o treinamento, visto o grande desbalanceamento, o modelo se beneficia de uma separação manual feita para a classe alimentícia, pois dessa maneira garantimos que a base de treinamento representa bem a totalidade dos nossos dados.

Em suma, a EfficientNet combinada com as técnicas estudadas durante esse projeto se mostram extremamente promissoras para classificação automática de publicidades alimentícias.

6 Próximos Passos

É fundamental continuar a validação dos nossos melhores resultados na Base 2 e encontrar nosso "modelo ideal" para finalmente testá-lo na Base 3. Somente assim pode-se confirmar com propriedade a performance da EfficientNet e a necessidade de aplicação de outras estratégias, como o DeiT [23], mesmo os resultados iniciais indicando que a aplicação de outro modelo provavelmente não será necessária.

Ainda é necessário dar início a uma fase de reconhecimento dos principais padrões das estratégias persuasivas mais comuns empregadas em publicidades brasileiras, e por esse motivo pode se fazer necessária uma nova revisão bibliográfica.

Recomenda-se que essa nova etapa também seja implementada e testada para o contexto de maneira semelhante a este projeto, sendo treinada na Base 1, validada na Base 2 e testada na Base 3. Sempre indicando se existe possibilidade de melhoria e quais abordagens podem ser utilizadas para esse aprimoramento caso elas existam.

Agradecimentos

Agradeço a professora Sandra Eliza Fontes de Avila, por ter sido minha orientadora e ter desempenhado tal função com dedicação e amizade.

A Paula Martins Horta, minha co-orientadora, e a Michele Bittencourt Rodrigues, pela oportunidade de participar desse projeto e pelos conhecimentos que me foram repassados.

A UNICAMP, ao SAE e seu corpo docente, que foram essenciais no meu processo de formação e pelos aprendizados ao longo dos anos do curso.

Ao PIBIC e ao CNPq pelo financiamento que me possibilitou dedicar-me ao projeto.

Aos amigos/familiares por todo o apoio e pela ajuda, que muito contribuíram para a realização deste trabalho.

Aos meus pais e a minha avó Silvandira Pedrazzoli, que me incentivaram sempre a dedicar-me por completo e compreenderam a minha ausência durante a realização deste trabalho.

A todos aqueles que contribuíram, de alguma forma, direta ou indiretamente, para a realização deste trabalho, os meus agradecimentos. Obrigada pelo enriquecendo o meu processo de aprendizado.

Referências

[1] R. M. Bielemann, J. V. S. Motta, G. C. Minten, B. L. Horta, and D. P. Gigante. Consumption of ultra-processed foods and their impact on the diet of young adults. *Revista de saude publica*, 49:28, 2015.

[2] A. Bissoto, F. Perez, V. Ribeiro, M. Fornaciali, S. Avila, and E. Valle. Deep-learning ensembles for skin-lesion segmentation, analysis, classification: Recod titans at isic challenge 2018. *arXiv preprint arXiv:1808.08480*, 2018.

[3] E. J. Boyland, S. Nolan, B. Kelly, C. Tudur-Smith, A. Jones, J. C. Halford, and E. Robinson. Advertising as a cue to consume: a systematic review and meta-analysis of the effects of acute exposure to unhealthy food and nonalcoholic beverage advertising on intake in children and adults, 2. *The American journal of clinical nutrition*, 103(2):519–533, 2016.

[4] I. B. de Geografia e Estatística. Pesquisa nacional de saúde - 2019. percepção do estado de saúde, estilos de vida e doenças crônicas: Brasil, grandes regiões e unidades da federação, 2020.

[5] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. Iii, and K. Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

[6] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

[7] R. B. Levy, R. M. Claro, D. H. Bandoni, L. Mondini, and C. A. Monteiro. Disponibilidade de"açúcares de adição"no brasil: distribuição, fontes alimentares e tendência temporal. *Revista Brasileira de Epidemiologia*, 15:3–12, 2012.

[8] R. B. Levy, R. M. Claro, and C. A. Monteiro. Sugar and overall macronutrient profile in the brazilian family diet (2002-2003). *Cadernos de saude publica*, 26(3):472–480, 2010.

[9] M. L. d. C. Louzada, A. P. B. Martins, D. S. Canella, L. G. Baraldi, R. B. Levy, R. M. Claro, J.-C. Moubarac, G. Cannon, and C. A. Monteiro. Ultra-processed foods and the nutritional dietary profile in brazil. *Revista de Saúde Pública*, 49, 2015.

[10] S. Mills, L. Tanner, and J. Adams. Systematic literature review of the effects of food and drink advertising on food and drink-related behaviour, attitudes and beliefs in adult populations. *Obesity Reviews*, 14(4):303–314, 2013.

[11] W. H. Organization et al. Set of recommendations on the marketing of foods and non-alcoholic beverages to children, 2010.

[12] W. H. Organization et al. World health organization obesity and overweight fact sheet, 2016.

[13] F. Perez, C. Vasconcelos, S. Avila, and E. Valle. Data augmentation for skin lesion analysis. In *Third International Workshop, ISIC 2018, Held in Conjunction with MICCAI*, pages 303–311, 2018.

[14] M. Perez, S. Avila, D. Moreira, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha. Video pornography detection through deep learning techniques and motion information. *Neurocomputing*, 230:279–293, 2017.

[15] R. Pires, S. Avila, J. Wainer, E. Valle, M. D. Abramoff, and A. Rocha. A data-driven approach to referable diabetic retinopathy detection. *Artificial intelligence in medicine*, 96:93–106, 2019.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[17] S. J. Russell, H. Croker, and R. M. Viner. The effect of screen advertising on children's dietary intake: A systematic review and meta-analysis. *Obesity reviews*, 20(4):554–568, 2019.

[18] B. Sadeghirad, T. Duhaney, S. Motaghipisheh, N. Campbell, and B. Johnston. Influence of unhealthy food and beverage marketing on children's dietary intake and preference: a systematic review and meta-analysis of randomized trials. *Obesity Reviews*, 17(10):945–959, 2016.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[20] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019.

[21] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[22] S. Tomar. Converting video formats with ffmpeg. *Linux Journal*, 2006(146):10, 2006.

[23] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[24] E. Valle, M. Fornaciali, A. Menegola, J. Tavares, F. V. Bittencourt, L. T. Li, and S. Avila. Data, depth, and design: Learning reliable models for skin lesion analysis. *Neurocomputing*, 383:303–313, 2020.

[25] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

[26] S. Vandevijvere, G. Sacks, and B. Swinburn. International network for food and obesity/ncd research, monitoring and action support: benchmarking food environments towards healthier diets. In *Annals of Nutrition & Metabolism: Abstracts of the 20th International Congress of Nutrition 2013*, pages 865–865. S Karger AG, 2013.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polosukhin. Attention is all you need. In *International Conference on Neural Information Processing Systems*, 2017.

[28] P. Vitorino, S. Avila, M. Perez, and A. Rocha. Leveraging deep neural networks to fight child pornography in the age of social media. *Journal of Visual Communication and Image Representation*, 50:303–313, 2018.