Universidade Estadual de Campinas
Instituto de Computação

Giovanna Nascimento Antonieti

Análise de Métodos de Explicabilidade de Redes Neurais Profundas para a Classificação de Elsagate

CAMPINAS
2021

Giovanna Nascimento Antonieti

Análise de Métodos de Explicabilidade de Redes Neurais Profundas para a Classificação de Elsagate

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Orientadora: Profa. Dra. Sandra Eliza Fontes de Avila

Este exemplar corresponde à versão final da Dissertação defendida por Giovanna Nascimento Antonieti e orientada pela Profa. Dra. Sandra Eliza Fontes de Avila.

CAMPINAS
2021

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

An88a

Antonieti, Giovanna Nascimento, 1998-
Análise de métodos de explicabilidade de redes neurais profundas para a classificação de elsagate / Giovanna Nascimento Antonieti. – Campinas, SP : [s.n.], 2021.

Orientador: Sandra Eliza Fontes de Avila.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Interpretabilidade (Aprendizado de máquina). 2. Explicabilidade (Aprendizado de máquina). 3. Elsagate. 4. Inteligência artificial. 5. Redes neurais profundas. 6. Aprendizado de máquina. I. Avila, Sandra Eliza Fontes de, 1982-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Analysis of explainability methods in deep neural networks for elsagate classification
Palavras-chave em inglês:
Interpretability (Machine learning)
Explainability (Machine learning)
Elsagate
Artificial intelligence
Deep neural networks
Machine learning
Área de concentração: Ciência da Computação
Titulação: Mestra em Ciência da Computação
Banca examinadora:
Sandra Eliza Fontes de Avila [Orientador]
Marley Maria Bernardes Rebuzzi Vellasco
Esther Luna Colombini
Data de defesa: 30-08-2021
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-8546-946X
- Currículo Lattes do autor: http://lattes.cnpq.br/6902189043900599

Universidade Estadual de Campinas
Instituto de Computação

Giovanna Nascimento Antonieti

Análise de Métodos de Explicabilidade de Redes Neurais Profundas para a Classificação de Elsagate

Banca Examinadora:

- Profa. Dra. Sandra Eliza Fontes de Avila (Orientadora)
  Universidade Estadual de Campinas
- Profa. Dra. Marley Maria Bernardes Rebuzzi Vellasco
  Pontifícia Universidade Católica do Rio de Janeiro
- Profa. Dra. Esther Luna Colombini
  Universidade Estadual de Campinas

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 30 de agosto de 2021

O que todos devemos fazer é nos certificar que estamos usando a inteligência artificial de uma maneira que beneficie a humanidade, e não que a deteriore.

(Tim Cook, atual CEO da Apple)

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal Nível Superior – Brasil (CAPES) – Código de Financiamento 001, ao qual sou imensamente grata.

Agradeço a todos os professores e professoras que me incentivaram, em especial a minha orientadora Profa. Dra. Sandra Avila por todo o suporte, confiança e parceria, mas também por me proporcionar a oportunidade de amadurecer e crescer como pessoa nos últimos anos. Agradeço também aos meus colegas do laboratório *Reasoning for Complex Data* (RECOD) pelo apoio.

Agradeço meus pais, Regiane Nascimento Antonieti e José Roberto Antonieti, pelo suporte durante meu mestrado e por me incentivarem a continuar estudando e buscando meus sonhos. Agradeço também a todos os meus amigos, por toda a ajuda e tempo gasto juntos, que com toda a certeza me ajudaram a seguir com esse trabalho.

Por fim, a Daniel Brai Gonzales Marcos, que conviveu comigo durante todo esse período, celebrando as conquistas e me motivando em todos os momentos desafiadores, que não foram poucos nesses últimos dezoito meses.

Resumo

Nos últimos anos, um número crescente de pesquisadores constatou que a interpretabilidade pode ter um valor significativo no processo de entendimento da causa de uma decisão tomada pelos modelos de aprendizado de máquina. Inferir os conceitos semânticos de altos nível a partir de dados visuais tem se mostrado uma tarefa difícil, como a tarefa classificação de conteúdo sensível, mais especificamente *Elsagate*, que são vídeos com personagens infantis em situações inapropriadas para crianças. A partir disso, as técnicas de explicabilidade podem se mostrar muito úteis para entender as decisões dos modelos para classificação de conteúdo sensível. Neste trabalho, investigamos a literatura a fim de adotar uma definição que diferencie os termos explicabilidade e interpretabilidade, propusemos uma taxonomia que unifica as principais taxonomias encontradas na literatura, realizamos uma revisão da literatura, além de classificar esses métodos na taxonomia proposta, e exploramos as seguintes técnicas de explicabilidade usando duas redes neurais profundas, a NASNet e a MobileNetv2: Gradiente Vanilla, Gradiente Integrado, SmoothGrad, LIME, GradCAM, GradCAM++ e ScoreCAM. Os resultados mostraram que nenhuma técnica possui o desempenho esperado para a explicabilidade de modelos treinados para a classificação de *Elsagate*. Além disso, identificamos uma grande dificuldade na avaliação desses métodos pela falta de métricas capazes de medir a explicabilidade das técnicas.

Abstract

In the last years, a growing number of researchers have found that interpretability can have significant value in understanding the cause of a decision made by machine learning models. Inferring high-level semantic concepts from visual data has proven to be a difficult task, such as the sensitive content classification task, specifically *Elsagate*, which are categorized as "child-friendly" but which contain inappropriate themes for children. Based on this, explainability techniques can be beneficial in understanding the decisions of models described in the literature for classifying sensitive content. In this work, we investigate the literature in order to adopt a definition that differentiates the terms explainability and interpretability, we propose a taxonomy that unifies the main taxonomies found in the literature, and we carry out a literature review, in addition to classifying these methods in the proposed taxonomy. We explore the following explanatory techniques using two deep neural networks, NASNet and MobileNetv2: Vanilla Gradient, Integrated Gradient, SmoothGrad, LIME, GradCAM, GradCAM++, and ScoreCAM. Thus, our results showed that no technique has the expected performance for the explicability of trained models for the *Elsagate* classification. Besides, we identify difficulty in evaluating these methods due to the lack of metrics capable of measuring the techniques' explainability.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

A Inteligência Artificial (IA) está no nosso cotidiano. Estima-se que a receita do mercado mundial de IA, incluindo setores de *software*, *hardware* e serviço cresça 16,4% nesse ano e que em 2024 atinja a marca de US$ 500 bilhões de dólares de receita e uma taxa de crescimento anual composta de cinco anos de 17,5%, segundo a *International Data Corporation* (IDC) [20]. O papel desses algoritmos na nossa vida tem crescido rapidamente, de uma simples recomendação de conteúdo ou resultados de pesquisa *online*, para áreas mais críticas, como diagnósticos médicos, riscos de seguro e até mesmo em avaliações de crédito para empréstimos.

Problematicamente, apesar desses modelos parecerem poderosos em termos de resultados e predições, esses algoritmos sofrem de opacidade, o que dificulta entender como os mecanismos internos funcionam, principalmente na área de aprendizado de máquina e *deep learning* [3]. Através de modelos sofisticados treinados em conjuntos de dados massivos, graças a escalabilidade e infraestruturas de alto desempenho, nos arriscamos a criar e usar modelos que não entendemos. Outros riscos inerentes deste modelo é a possibilidade de tomar decisões erradas [17], aprender artefatos ou correlações espúrias do conjunto de dados [10, 11], ou reconhecer objetos em imagens pelas propriedades do fundo ou iluminação devido a um viés no conjunto de dados de treinamento [17].

Para resolver esses problemas, a inteligência artificial explicável (XAI, do inglês, *eXplainable Artificial Intelligence*) visa criar técnicas que produzem modelos mais transparentes, explicáveis e interpretáveis, enquanto mantém o alto desempenho. É importante ressaltar que as definições de XAI são genéricas e por vezes não há consenso sobre sua definição [42], o que explica parcialmente o porquê dos métodos de interpretabilidade são tão diferentes entre si. Uma definição feita por Miller [30] é (em tradução livre): *Interpretabilidade é o grau o qual um humano pode entender a causa de uma decisão*. Outra definição feita por Kim *et al.* [23] é (em tradução livre): *Interpretabilidade é o grau ao qual um humano pode prever de forma consistente o resultado do modelo*. De forma genérica, interpretabilidade se refere a capacidade humana de entender e raciocinar um modelo [16].

Inferir conceitos semânticos de alto nível a partir de dados visuais tem sido um objetivo perseguido por muitos cientistas. A ausência de uma solução geral atesta a dificuldade de uma tarefa, que é trazida a partir da lacuna semântica entre a representação de baixo nível (pixels, frames) e os conceitos de alto nível que se deseja levar considerar [59]. Um dos problemas mais desafiantes envolvendo semântica de alto nível em dados visuais é a clas-

sificação de conteúdo sensível, como por exemplo, *Elsagate* [1]. O termo em si é composto por *Elsa* (uma personagem dos filmes da Disney, que foi um dos primeiros personagens retratados nesse tipo de vídeo) e *-gate* (um sufixo geralmente usados para escândalos). A maioria dos vídeos nessa categoria são animações grosseiras, que apresentam personagens infantis famosos em situações que envolvem violência, fetiches, drogas, álcool, atividades perigosas e/ou perturbadoras ou até mesmo conteúdo sexual [21]. A Figura 1.1 apresenta algumas imagens que exemplificam conteúdos do tipo *Elsagate*.

Figura 1.1: Exemplos de *frames* de vídeos anotados como *Elsagate*. Imagens da base de dados *Elsagate* [21].

A análise de métodos de explicabilidade para classificação de vídeos do tipo *Elsagate* é o foco principal desta dissertação.

1.1 Motivações e Desafios

Recentemente, o tópico XAI tem recebido atenção da academia. A Figura 1.2 ilustra o surgimento notável do termo *Explainable Artificial Intelligence* utilizando o *Google Trends*[1] e o ressurgimento em 2016. Esse ressurgimento do tópico é resultado direto da entrada da IA e o Aprendizado de Máquina no nosso cotidiano e seu impacto crucial em processos de tomada de decisão que são críticos, sem ser capaz de prover informações detalhadas sobre o processo que levou a certa decisão, recomendação ou predição feita pelo algoritmo.

As técnicas de interpretabilidade podem se mostrar muito úteis para entender as decisões de classificação tomadas pelos modelos descritos na literatura para a classificação de conteúdo sensível. No entanto, a grande dificuldade dessa investigação é que conteúdos sensíveis, como *elsagate*, são subjetivos, nem sempre explícitos nas imagens/vídeos que estão sendo classificados. Assim, o grande desafio desta dissertação é entender como os métodos de interpretabilidade podem ser aplicados no contexto de conteúdo sensível.

[1] `https://trends.google.com/trends/explore?date=all&q=Explainable%20Artificial%20Intelligence`

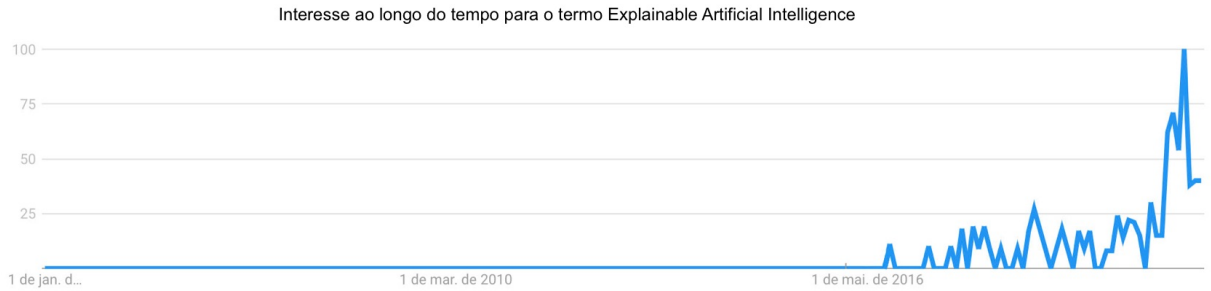


Figura 1.2: Resultado do *Google Trends* para pesquisa do termo <*Explainable Artificial Intelligence*> entre janeiro de 2004 e maio de 2021 (incompleto).

Outro desafio é que esses métodos podem identificar partes da imagem como relevantes para a classificação, que para nós humanos não faz sentido algum relacioná-las ao conteúdo que está sendo classificado, podendo ser apenas um meio que a rede aprendeu para classificar o conceito ou objeto, sem estar necessariamente errado, ou simplesmente pode ser algum viés do aprendizado da própria rede.

1.2 Objetivos

Os principais objetivos desta dissertação são:

O1. Investigar a taxonomia dos métodos de interpretabilidade e explicabilidade, investigar as vantagens e desvantagens de cada uma e propor uma taxonomia que integre as já existentes na literatura.

O2. Pesquisar as propriedades desejadas de um método de explicabilidade para que a sua explicação seja classificada como boa.

O3. Explorar técnicas de explicabilidade de modelos de redes neurais profundas.

O4. Analisar o desempenho dessas técnicas quando aplicadas à classificação de conteúdo sensível, do tipo *Elsagate*, a fim de possibilitar uma maior compreensão e encontrar possíveis melhorias para as mesmas.

1.3 Questões de Pesquisa

A principal questão de pesquisa que esta dissertação visa responder é:

Q1. Considerando problemas de classificação de conceitos abstratos, como *Elsagate*, que não apresentam um conceito bem definido, é possível aplicar e interpretar as técnicas de explicabilidade mais utilizadas na literatura (por exemplo, técnicas baseadas em gradiente, técnicas de mapa de ativação de classe)?

1.4 Contribuições

As principais contribuições são:

C1. Proposição de uma taxonomia que unifica as principais taxonomias de métodos de interpretabilidade e explicabilidade da literatura;

C2. Revisão da literatura dos métodos de explicabilidades, assim como a classificação dos mesmos de acordo com a taxonomia proposta;

C3. Avaliação dos métodos de explicabilidade comumente utilizados na área de classificação e identificação de objetos em um contexto mais subjetivo, para a classificação de conteúdos *Elsagate*.

1.5 Organização do Texto

O restante deste texto está organizado como segue.

No Capítulo Conceitos Relacionados, são apresentados os principais conceitos de XAI, as definições usadas nesta dissertação e explicação do funcionamento das técnicas utilizadas nos experimentos.

No Capítulo Taxonomias, são apresentadas as taxonomias da literatura e a taxonomia proposta nesta dissertação.

No Capítulo Trabalhos Relacionados, é apresentada a revisão da literatura que foi focada na proposta de novas técnicas de explicabilidade para aprendizado de máquina, mais especificamente para redes neurais convolucionais. Na busca feita no *Google Scholar* restringimos o número de artigos filtrando por trabalhos de explicações visuais e removendo trabalhos na área médica, com dados tabulares, de processamento de linguagem natural e textual, retornando 59 artigos, que foram lidos e selecionados de acordo com sua relevância.

No Capítulo Metodologia Proposta, são apresentados os materiais utilizados nos experimentos, conjunto de dados e modelos, além dos métodos utilizados nos experimentos.

No Capítulo Resultados, apresentamos os resultados obtidos nos experimentos e discutimos sobre os resultados obtidos e a avaliação desses métodos.

No Capítulo Conclusão, discutimos sobre como a literatura de XAI está se encaminhando, assim como a avaliação dos métodos propostos e os resultados desse trabalho, além de indicar passos futuros.

Capítulo 2

Conceitos Relacionados

Neste capítulo, vamos apresentar os conceitos e as definições importantes para a compreensão deste trabalho, como interpretabilidade (Seção 2.1). Além disso, detalharemos as técnicas que foram utilizados nos experimentos (Seção 2.2).

2.1 Interpretabilidade e Explicabilidade

A arquitetura de uma rede neural profunda é determinada por vários componentes (tipo de unidade, função de ativação, padrão de conectividade, mecanismos de bloqueio) e o resultado de um processo de aprendizagem, que também depende de várias propriedades (regularização, otimização, mecanismos adaptativos, função de custo). O resultado da rede de interação entre esses componentes não pode ser predito com exatidão, por esses motivos DNNs (*Deep Neural Networks*, ou redes neurais profundas) são por vezes tratados como modelos caixa-preta. Felizmente, esse problema não escapou da visão da comunidade de Aprendizado Profundo [43,48,67]. Pesquisas em como interpretar e explicar o processo de tomadas de decisões de redes neurais artificiais acontecem desde o final dos anos 80 [6].

Recentemente, houve uma explosão no número de pesquisas relacionados à inteligência artificial interpretável e explicável, apesar de não estar claro o que exatamente significa isso, uma vez que a definição de inteligência artificial interpretável e explicável é genérica e muitas vezes não há um consenso na literatura, o que explica parcialmente a diversidade que podemos encontrar nos métodos de interpretabilidade. Outro ponto a ser ressaltado é que alguns termos como interpretabilidade, explicabilidade e compreensibilidade podem ter significados redundantes [17] ou claramente distintos [7]. A seguir, apresentamos as definições adotadas nesta dissertação:

Definição 1. **Interpretabilidade**, segundo Arrieta *et al.* [7], é a habilidade de explicar ou fornecer significado em termos humanamente compreensíveis. Dessa forma, podemos entender que a interpretabilidade é uma característica *passiva* do modelo que é adicionada durante o desenvolvimento de sua arquitetura. Um exemplo de modelo interpretável seria uma árvore de decisão, onde um humano pode simular seu próprio comportamento, além do modelo ter regras que não se alteram de acordo com a natureza do dado e preservam sua legibilidade. Árvores de decisão, geralmente,

apresentam regras legíveis que explicam o conhecimento aprendido a partir dos dados e permitem uma compreensão direta da previsão do processo.

Definição 2. **Explicabilidade** está associada com a ideia da explicação ser uma interface entre humanos e o modelo tomador de decisões [17], onde temos os valores dos atributos sendo relacionados com a predição do modelo de forma humanamente inteligível. Assim, percebemos que a explicabilidade se refere a uma característica *ativa* do modelo que descreve o processo realizado para esclarecer o funcionamento interno do modelo. No caso de redes neurais, por exemplo, não conseguimos compreender o modelo apenas observando seu funcionamento, precisamos de métodos externos, normalmente relacionados à relevância de *features* ou outras técnicas de visualização, para entender o processo realizado pelo modelo. Assim, podemos nos referir a essas técnicas como técnicas de explicabilidade, já que são elas que agem como um mediador entre nós e o modelo.

Definição 3. Um modelo é considerado **caixa-preta** se o mapeamento entre seus parâmetros e a saída é escondido dos usuários [14].

Definição 4. Dada uma certa audiência, **inteligência artificial explicável** é uma entidade que fornece detalhes do seu funcionamento de forma facilmente compreensível.

Assim como muitos trabalhos tentam definir o que é interpretabilidade e explicabilidade, muitos outros tentam definir as características desejáveis nos métodos e em seus resultados. A seguir, listamos as características mais citadas e seus significados:

- *Robustez*: se refere a habilidade de manter certo nível de performance do método e do modelo, independente de pequenas variações na entrada [17].

- *Causalidade*: em algoritmos de aprendizado de máquina não supervisionado, não há garantias de que o modelo reflete relações de causa, já que sempre pode haver responsabilidade de causa não observadas para as variáveis. Usando modelos interpretáveis, conseguimos gerar hipóteses de causalidade que os cientistas podem testar de forma experimental [28]. Como a causa envolve correlação, um modelo explicável pode validar os resultados geradas por técnicas de inferência causais. Ou até mesmo dar uma primeira intuição de possíveis relações causais nos dados disponíveis [7].

- *Escalabilidade*: segundo Guidotti *et al.* [17], é oportuno que os modelos e métodos de explicabilidade sejam capazes de escalar para grandes volumes de dados, uma vez que vivemos na era do *big data*.

- *Generabilidade*: modelos portáteis que não requerem regime especial de treinamento ou que possuem restrições [17]. Explicabilidade é também um ponto que ajuda a generalidade, uma vez que torna mais fácil a tarefa de identificar os limites que afetam os modelos, possibilitando um melhor entendimento e implementação [7].

- *Informatividade*: uma vez que os modelos de aprendizado de máquina são usados a fim de apoiar o processo de tomada de decisão, é necessário uma grande quantidade

de informação para relacionar a tomada de decisão do usuário com a solução dada pelo modelo e evitar equívocos [7]. Exatamente por esse ponto, os modelos de aprendizado de máquina explicáveis devem fornecer informações sobre o problema que está sendo abordado.

- *Confiança*: para Lipton [28], pode ser a confiança de que o modelo vai apresentar um bom desempenho, sem nenhum viés ético, ou a confiança no desempenho do modelo que ficamos confortáveis em deixá-lo no controle de uma situação. É importante saber não só se o modelo acerta, mas também em que instâncias ele acerta, já que se ele acerta e erra as mesmas instâncias que um humano, podemos considerar confiável já que não estamos tendo custos de ceder o controle. Já em redes neurais profundas, em que o processo de tomada de decisão não precisa ser validado, segundo Xie *et al.* [64], pode ser considerado fidedigno. Essa confiança pode ser desenvolvida de duas maneiras: 1) através de testes satisfatórios, onde o modelo é executado em condições ideais e deve aproximar o seu desempenho na prática; e 2) experiência. Um usuário não precisa validar as ações de uma rede em que as entradas e saídas correspondem ao esperado.

- *Segurança*: redes neurais profundas que as decisões corretas ou incorretas impactam a vida humana, saúde ou políticas sociais devem ser seguras [64]. Ainda, segundo Xie *et al.*, uma rede profunda segura deve: 1) operar de forma consistente conforme o esperado; 2) proteger escolhas que podem impactar de forma negativa o usuário ou a sociedade; 3) exibir um alto grau de confiança em situações operacionais comuns e excepcionais; e 4) prover um *feedback* ao usuário de como uma condição operacional influencia na sua decisão. O primeiro aspecto de segurança alinha esse traço com a confiança, já que a confiança em um sistema é um pré-requisito para considerá-lo um sistema seguro para o usuário. O segundo e terceiro aspectos implicam que um sistema seguro possua mecanismos que aumentam seu processo de decisões para se distanciar de decisões com impactos negativos. O quarto aspecto se refere a necessidade de um *feedback* ao usuário sobre como as condições de operação do modelo influenciam suas decisões.

2.2 Técnicas de Explicabilidade

Nesta seção, explicaremos de forma detalhada as técnicas que foram utilizadas nos experimentos, são elas Gradiente *Vanilla* [52], SmoothGrad [53], Gradiente Integrado [57], LIME [41], GradCAM [49], GradCAM++ [13] e ScoreCAM [63]. Estas técnicas são amplamente aplicadas na literatura e possuem código disponível, ou implementação em bibliotecas da própria linguagem de programação.

2.2.1 Gradiente Vanilla

Simonyan *et al.* [52] propuseram a utilização do gradiente para mostrar quais pixels precisam ser minimamente alterados para que afetem o resultado da classificação. Então, dada uma imagem I_0, uma classe c e uma rede neural convolucional de classificação treinada

com uma função de pontuação $F_c(I_0)$, a ideia é classificar os pixels da imagem I_0 baseado em sua influência na pontuação $F_c(I_0)$.

Para exemplificar, considere um modelo com uma função de pontuação linear para a classe c:

$$F_c(I_0) = w_c^T I + b_c, \tag{2.1}$$

onde a imagem I é representada de forma vetorizada e w_c e b_c, são respectivamente o vetor com os pesos e o viés do modelo. Nesse caso, é possível observar que a magnitude dos elementos de w definem a importância do pixel correspondente de I para a classe c.

Nos casos de redes convolucionais, a função de pontuação $F_c(I)$ não é linear. Dessa forma, o raciocínio feito anteriormente não pode ser imediatamente aplicado. Assim, dada a imagem I_0, pode-se aproximar a função $F_c(I)$ com uma função linear de vizinhança de I_0 computando a expansão da fórmula Taylor de primeira ordem:

$$F_c(I) \approx w^T I + b, \tag{2.2}$$

onde w é a derivada de F_c em relação à imagem I no ponto I_0:

$$w = \frac{\partial F_c}{\partial I} \Big|_{I_0}. \tag{2.3}$$

Para calcular o mapa de saliência $M \in \mathbb{R}^{m \times n}$ de uma imagem I_0, com m linhas e n colunas, encontra-se a derivada w (Equação 2.3) através da retropropagação. Depois disso, o mapa de saliência é obtido reorganizando os elementos do vetor w. No caso de imagens em preto e branco, o número de elementos em w é igual ao número de pixels em I_0. Assim, o mapa pode ser computado como $M_{ij} = |w_{h(i,j)}|$, onde $h(i,j)$ é o índice do elemento de w, correspondente ao pixel da imagem na i-ésima linha e j-ésima coluna. No caso de imagens coloridas, assumi-se o canal de cor c do pixel (i,j) da imagem I correspondente ao elemento de w com índice $h(i,j,c)$. Na Figura 2.1, podemos observar os mapas de saliência das classes com maior pontuação para imagens do conjunto de testes da base de dados ImageNet (ILSVRC 2013), que foram escolhidas de forma aleatória.

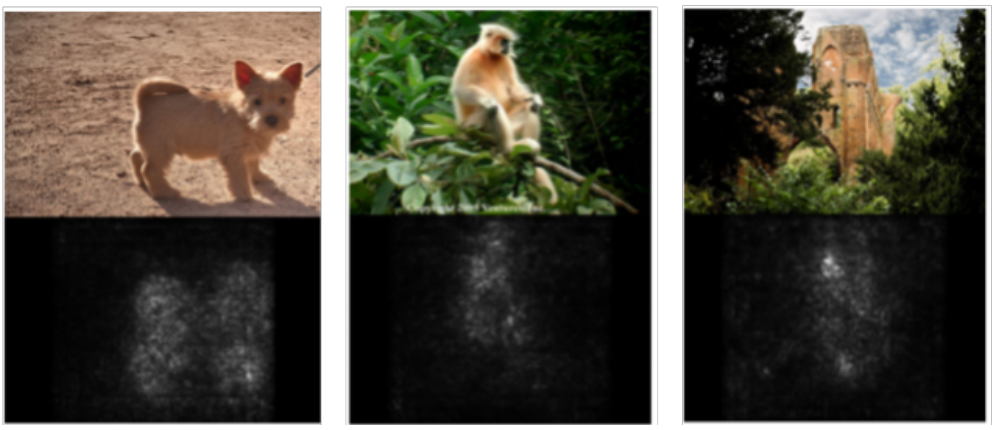


Figura 2.1: Mapa de saliência para a primeira classe predita nas imagens de teste da base de dados ImageNet (ILSVRC 2013). Figura reproduzida de Simonyan *et al.* [52].

2.2.2 SmoothGrad

A partir do Gradiente *Vanilla* [52], Smilkov *et al.* [53] propuseram a técnica *SmoothGrad*, que na prática consiste em adicionar ruído visual na imagem antes de calcular o mapa de sensibilidade, que buscam encontrar na imagem regiões particularmente relevantes para a classificação final [66], muitas vezes também são chamados de mapas de saliência ou mapa de atribuição de pixel. A ideia central é pegar a imagem de interesse, então criar amostras a partir da imagem original com a adição de ruído para então fazer a média dos mapas de sensibilidade resultante de cada amostra.

Podemos construir o mapa de sensibilidade $W_c(I)$ simplesmente pela diferenciação de W_c em relação à entrada I, o que pode ser apresentado da seguinte forma:

$$W_c(I) = \frac{\partial F_c(I)}{\partial I}, \tag{2.4}$$

onde ∂F_c representa a derivativa de F_c, em outras palavras o gradiente.

De forma intuitiva, W_c representa quanta diferença uma pequena mudança em cada pixel de I faria para a classificação para a classe c. Como resultado teríamos uma mapa resultante W_c que destaca regiões chave da imagem. No entanto, os mapas de sensibilidade baseados no gradiente puro são tipicamente ruidosos.

Então, para calcular o *SmoothGrad*, é utilizado amostras aleatórias na vizinhança da entrada I e a média dos seus mapas de sensibilidade. Matematicamente isso significa calcular:

$$W_c'(I) = \frac{1}{n}\sum W_c(I + N(0, \sigma^2)), \tag{2.5}$$

onde n é o número de amostras, e $N(0, \sigma^2)$ representa o ruído Gaussiano com desvio padrão σ. Na Figura 2.2, podemos observar 3 imagens e seus resultados variando o desvio padrão σ do ruído Gaussiano adicionado nas amostras, sendo 0% o gradiente padrão, também conhecido como *vanilla gradient*. Cada mapa de sensibilidade apresentado foi obtido aplicando o ruído Gaussiano $N(0, \sigma^2)$ para os pixels da imagem de entrada para 50 amostras e aplicando a média nos mapas resultantes.

2.2.3 Gradiente Integrado

Sundararajan *et al.* [57] identificaram dois axiomas que julgam fundamentais nos métodos de interpretabilidade, eles são:

- **Sensibilidade**: para cada entrada e *baseline* que difere em uma característica mas tem uma predição diferente, essa característica deveria ter um valor diferente de zero, em métodos de atribuição que satisfazem esse axioma.
- **Invariância de implementação**: para duas redes com funcionamento equivalente, arquiteturas diferentes com saídas iguais para todas as entradas, os resultados dos métodos de atribuição para elas devem ser sempre idênticos.

Com esses dois axiomas em mente e com base na técnica do Gradiente *Vanilla* [52], Sundararajan *et al.* [57] propuseram um método de atribuição que cumpre com esses axiomas, o Gradiente Integrado.

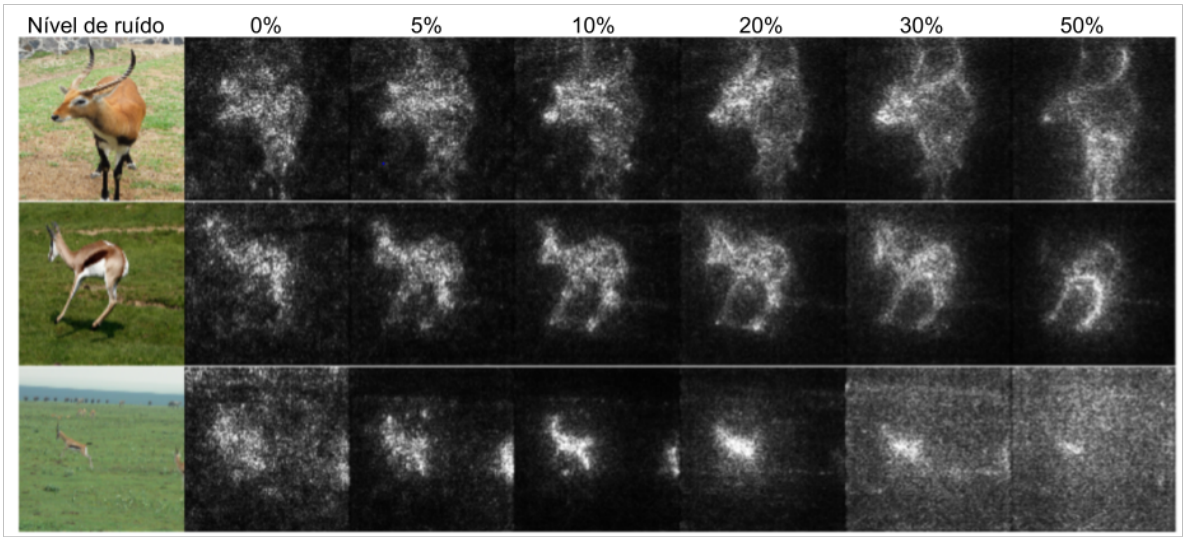


Figura 2.2: Apresentação dos efeitos que os diferentes nível de ruídos podem ter no resultado do método SmoothGrad. Figura reproduzida de Smilkov *et al.* [53].

Formalmente, suponha que temos a função $F : \mathbb{R}^n \to [0, 1]$ que representa uma rede neural profunda. Seja $I \in \mathbb{R}^n$ a entrada, e $I' \in \mathbb{R}^n$ seja o *baseline* da entrada, no caso de redes classificadoras de imagem, o *baseline* pode ser uma imagem preta.

Nesse método, os autores consideram um caminho em linha reta em $\mathbb{R}^n$ do *baseline* a entrada, e computam os gradientes em todos os pontos desse caminho. O gradiente integrado é obtido acumulando esses gradientes. Especificamente, os gradientes integrados são definidos como o caminho integral dos gradientes ao longo do caminho do *baseline* I' para a entrada I. A integral dos gradientes pode ser aproximada de forma eficiente com a somatória. Então, simplesmente somamos os gradientes em pontos de intervalos suficientemente pequenos ao longo do caminho em linha reta de I' até I. Dessa forma, o gradiente integrado ao logo da j-ésima dimensão para a entrada I e *baseline* I' é definido a seguir, onde $\frac{\partial F(I)}{\partial I^i}$ é o gradiente de $F(I)$ na j-ésima dimensão:

$$\text{IntegratedGrads}_j^{approI}(I) ::= (I_j - I'_j) \times \sum_{k=1}^{m} \frac{\partial F(I' + \frac{k}{m} \times (I - I'))}{\partial I_j} \times \frac{1}{m}, \tag{2.6}$$

onde m é o número de passos na aproximação de integral de Riemman. Na Figura 2.3, podemos ver como o método se comporta quando o mapa de atribuição é sobreposto na imagem original, destacando as áreas que mais contribuíram com a predição daquela classe.

2.2.4 LIME

Ribeiro *et al.* [41] propuseram o *Local Interpretable Model-Agnostic Explanations* (LIME), uma técnica que tenta encontrar a importância de *superpixels* (um conjunto de *pixels* similares) na imagem de entrada para a saída do modelo. Dessa forma, para explicar uma instância, o LIME gera os *superpixels* na imagem original e cria um conjunto de dados perturbados, onde alguns *superpixels* são pintados de preto/zerado. Com esse novo

Figura 2.3: Resultados do método gradiente integrado onde temos a imagem original, na primeira coluna à esquerda, e o resultado da sobreposição do retorno do método com a imagem original, na coluna à direita. Essas imagens foram rotuladas como ônibus escolar, na imagem de cima, e como mesquita, na imagem de baixo. Figura reproduzida de Sundararajan *et al.* [57].

conjunto de dado, um modelo linear é treinado com os dados perturbados e a distância da imagem original para a nova gerada. Assim, o método consegue identificar os *superpixels* mais influentes na classificação.

Formalmente, podemos definir $g \in G$ como a explicação como um modelo para uma classe de modelos potencialmente interpretáveis G. Esse modelo interpretável g pode ser uma árvore de decisão, um modelo linear ou qualquer outro modelo interpretável. Seja a complexidade do modelo interpretável medida por $\Omega(g)$. Se $\pi_I(z)$ é a proximidade entre as duas instâncias e $\iota(f, g, \pi_x)$ representa a fidelidade de g em aproximar localmente a função f do modelo medindo a proximidade por π_I, a explicação ξ para uma entrada I é dada pela equação:

$$\xi(I) = argmin_{g \in G} \iota(f, g, \pi_I) + \Omega(g). \tag{2.7}$$

onde o principal objetivo é minimizar o erro de aproximação de forma agnóstica ao modelo. Na Figura 2.4, podemos observar o resultado apresentado por Ribeiro *et al.* [41] para diferentes classes.

2.2.5 GradCAM

Gradient-weighted Class Activation Mapping (GradCAM), ou mapa de ativação de classe ponderada por gradiente, é uma técnica para produzir uma explicação visual para a decisões a partir de uma classe de modelos baseados em redes neurais convolucionais (CNN, *Convolutional Neural Networks*), que utilizam os gradientes do conceito que queremos

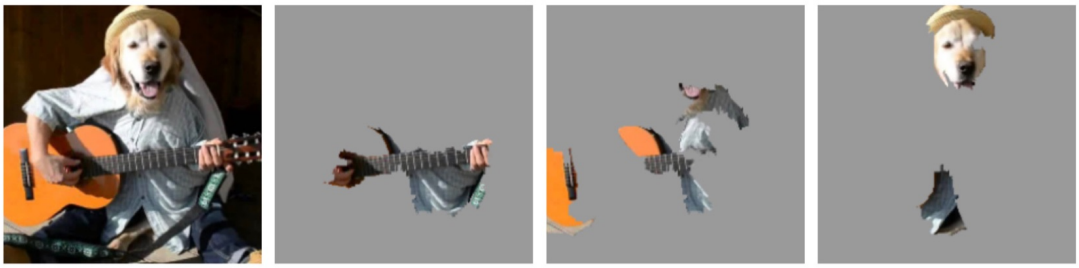


Figura 2.4: Resultado do método LIME: (a) imagem original e o resultado do método para a classificação da imagem como (b) guitarra, (c) violão e (d) labrador. Nessas imagens é possível observar os *superpixels* de maior relevância para a classificação de cada classe. Figura reproduzida de Ribeiro *et al.* [41].

identificar na imagem, fluindo para a camada final para produzir um mapa de localização, destacando as regiões importantes na imagem para prever os conceitos.

Selvaraju *et al.* [49] propuseram o GradCAM para mapear qualquer ativação discriminativa de classe das últimas camadas convolucionais nas imagens de entrada. Para encontrar o mapa de classe discriminativo, $L^c_{GradCAM} \in \mathbb{R}^{u \times v}$, de largura i e altura j, para qualquer classe c, primeiro calcula-se o gradiente para a classe c, f^c, sobre o mapa de ativação de características A^k de uma camada convolucional. Esse gradiente sofre a operação de *global average pooling* para obter os pesos de importância do neurônio. Matematicamente, esse processo pode ser formulado como:

$$\alpha^c_k = \frac{1}{Z} \sum_i \sum_j \partial \frac{f^c}{\partial A^k_{ij}}. \tag{2.8}$$

Esse peso α^c_k representa uma linearização parcial da rede neural a partir de A, e captura a importância do mapa de característica k para a classe alvo c. Então é realizado a combinação dos pesos com os mapas de ativação, seguido de uma função ReLU, como podemos ver na Equação 2.9.

$$L^c_{GradCAM} = ReLU(\sum_k (\alpha^c_k A^k)). \tag{2.9}$$

Na Figura 2.5, podemos observar o resultado do GradCAM aplicado ao problema de classificação de cachorros e gatos, que Selvaraju *et al.* [49] apresentam em seu trabalho. As regiões em vermelho representam regiões que possuem grande influência na classificação.

2.2.6 GradCAM++

Chattpadhyay *et al.* [13] propuseram o GradCAM++, um método que fornece melhores explicações visuais de predições em redes neurais convolucionais, em termos de uma melhor localização de objetos como também explicação de instâncias com ocorrências de múltiplas objetos em uma única imagem, quando comparado com os métodos estado-da-arte (por exemplo, GradCAM e variações). Os autores propuseram uma derivação

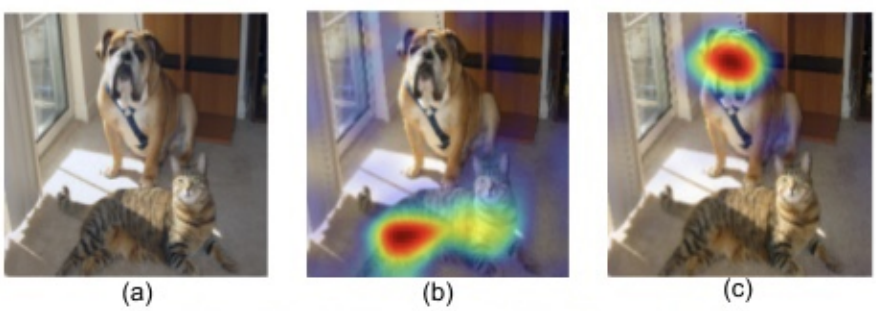


Figura 2.5: Resultado do método GradCAM: (a) imagem original; (b) resultado do método para classificação de gato; e (c) resultado do método na classificação de cachorro. As regiões em vermelho são as regiões com maior relevância para a classificação da imagem na respectiva classe. Figura reproduzida de Selvaraju *et al.* [49].

matemática para o método, que usa uma combinação de derivadas parciais positivas do mapa de atributos da última camada convolucional, relacionado a uma pontuação de classe específica como pesos para gerar uma explicação visual para a classe correspondente.

Em uma CNN com *global average pooling*, a pontuação final da classificação f^c para uma classe em particular c pode ser escrito como uma combinação linear dos mapas da última camada convolucional, $A^k_{i,j}$. Chattpadhyay *et al.* [13] definem os pesos w^c_k como:

$$w^c_k = \sum_i \sum_j \alpha^{kc}_{ij}.ReLU(\frac{\partial f^c}{\partial A^k_{ij}}), \tag{2.10}$$

sendo que w^c_k captura a importância de um mapa de ativação A^k em particular, e busca-se gradientes positivos para indicar os atributos que aumentam a ativação de saída dos neurônios da rede. A partir disso, os autores ainda derivam parcialmente duas vezes para obter o método final.

Os resultados obtidos por Chattopadhay *et al.* [13] podem ser observados na Figura 2.6.

2.2.7 ScoreCAM

Wang *et al.* [63] propuseram o ScoreCAM, um método que elimina a dependência de gradientes ao obter o peso de cada mapa de ativação por meio de sua pontuação de confiança na classe alvo. O resultado final é obtido por uma combinação linear de pesos e mapas de ativação. Formalmente, dado uma função geral $Y = f(X)$ que tem um vetor de entrada $X = [x_0, x_1, ..., x_n]^\top$ e uma saída escalar Y, para uma entrada base X_b, a contribuição de c_i de $x_i (i \in [0, n-1])$ em direção Y é a mudança da saída substituindo a i-ésima entrada em X_b por x_i. Formalmente, temos:

$$c_i = f(X_b \circ H_i) - f(X_b), \tag{2.11}$$

onde H_i é um vetor com o mesmo formato de X_b para cada entrada h_j em H_i, $h_j = \mathbb{I}[i = j]$. Dessa forma, dada uma rede neural convolucional representada por $Y = f(X)$, que tem como entrada X e saída um escalar Y, é escolhido uma camada convolucional l e sua ativação correspondente A. Defini-se o k-ésimo canal de A_l como A^k_l. Então, para uma

Figura 2.6: Resultado do método GradCAM++ para explicação visual das seguintes legendas: "Uma jovem acompanhada por uma pequena planta" e "Uma corrida de moto de motocross quatro crianças pequenas estão andando de uma corrida de bicicleta". Figura reproduzida de Chattopadhay *et al.* [13].

entrada base conhecida X_b, a contribuição de A_l^k para Y pode ser definida como

$$\begin{aligned} C(A_l^k) &= f(X \circ H_l^k) - f(X_b), \\ H_l^k &= s(Up(A_l^k)), \end{aligned} \tag{2.12}$$

onde $Up(\cdot)$ denota a operação de *upsampling* para o tamanho da entrada e $s(\cdot)$ a função de normalização dos mapas de cada elemento para uma matriz entre $[0, 1]$.

Finalmente, o método $L^c_{score-CAM}$ pode ser descrito como:

$$\begin{aligned} L^c_{score-CAM} &= ReLU(\sum_k \alpha_k^c A_l^c), \\ \alpha_k^c &= C(A_l^k). \end{aligned} \tag{2.13}$$

Na Figura 2.7, podemos observar os resultados apresentados por Wang *et al.* [63] na proposta do método.

2.3 Considerações

Podemos observar que das sete técnicas apresentadas, cinco se baseiam no gradiente para cálculo de importância dos *pixels* de uma imagem, uma vez que essa é uma medida bastante comum nos métodos de explicabilidade, por sua eficiência computacional, auxiliando em sua escalabilidade e generalização. Outro ponto é que essas técnicas foram propostas utilizando conceitos concretos, que é possível localizarmos na imagem, diferentemente

Figura 2.7: Resultado discriminativo de classe. Na imagem do meio temos o resultado para a classe "bull mastiff" e a da direita para a classe "gato tigrado", onde as regiões amareladas são as regiões com maior relevância. Figura reproduzida de Wang *et al.* [63].

do contexto que estamos propondo o teste desses métodos. Nos resultados experimentais, mostramos que estas técnicas não são adequadas para explicabilidade no contexto *Elsagate*.

Capítulo 3

Taxonomias

Nos últimos anos, muitos trabalhos foram publicados na área de interpretabilidade, propondo diferentes taxonomias de métodos de interpretabilidade para modelos de aprendizado de máquina. Neste capítulo, apresentaremos seis taxonomias [7, 14, 16, 17, 28, 39] e a taxonomia proposta, que engloba as taxonomias investigadas. Ressaltamos que ao longo desse capítulo os termos *features* e atributos serão usados como sinônimos.

3.1 Taxonomias da Literatura

A taxonomia proposta por Lipton [28] (Figura 3.1) se baseia na classificação dos métodos de acordo com a explicação gerada, chamadas de explicações *post-hoc*, e transparência de modelos. A transparência conota um senso de entendimento dos mecanismos de como o modelo funciona e pode ser considerada em nível de modelo por completo (simulabilidade), em nível de componentes individuais, onde cada parte do modelo — entrada e parâmetro — admite uma explicação intuitiva (capacidade de decomposição) e em nível de treinamento (transparência algorítmica). Já as explicações *post-hoc* não elucidam precisamente como o modelo funciona, mas podem conferir informações úteis para os usuários de IA. Essa última categoria pode ser dividida em:

- *Explicações textuais*: nessa categoria, podemos treinar um modelo para predições e um separado para gerar explicações textuais;
- *Visualização*: renderizamos visualização na esperança de determinar de forma quantitativa o que o modelo aprendeu;
- *Explicações locais*: alguns trabalhos focam em explicar a dependência de redes neurais localmente, já que descrever tudo o que foi aprendido pela rede pode ser difícil;
- *Explicação por exemplos*: nesse conjunto, os métodos tendem a reportar, além da predição, quais outros exemplos os modelos consideram mais similares [12].

Por sua vez, Guidotti *et al.* [17] (Figura 3.2) propõem uma taxonomia baseada nos problemas que tentamos solucionar ao adicionar interpretabilidade aos modelo. Como o trabalho de Lipton [28], Guidotti *et al.* também têm a separação para *design* transparente,

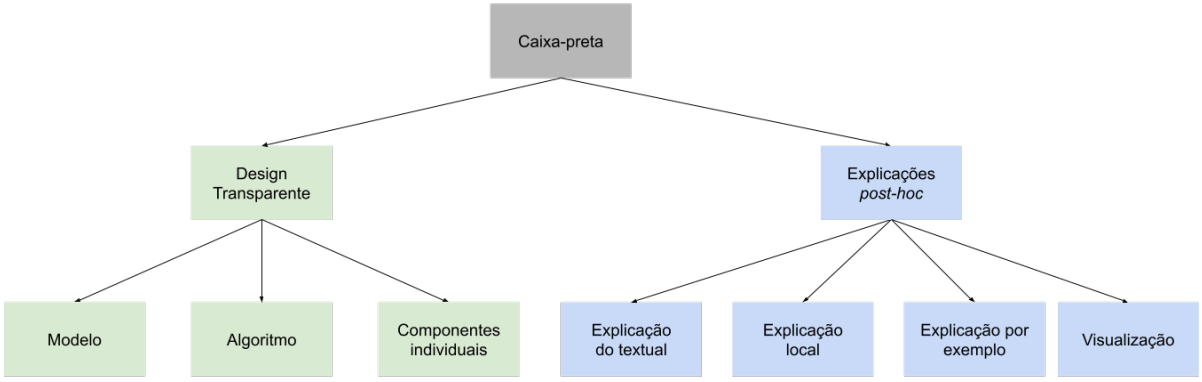


Figura 3.1: Taxonomia proposta por Lipton [28].

porém ele divide essa categoria em duas outras diferentes: 1) extração de regra, onde o classificador global possui um conjunto de regras que o levam a todas as possíveis decisões; e 2) seleção de protótipo, onde temos um modelo compreensível equipado com uma função global de explicação entendível. Ter um protótipo, objeto representativo de um conjunto de instâncias similares, no meio do conjunto de treinamento ajuda na interpretabilidade. A outra categoria proposta por Guidotti *et al.* é a de explicação do modelo caixa-preta, que tem três subcategorias:

- *Explicação do modelo*, onde provemos uma explicação global do modelo caixa-preta através de um modelo interpretável e transparente capaz de imitar o comportamento do modelo que queremos compreender;
- *Explicação da saída*, diferente da subcategoria anterior, buscamos a explicação sobre a saída do modelo para apenas uma instância, não é um requisito explicar toda a lógica do modelo e sim só para a predição de uma saída específica;
- *Inspeção de modelo* consiste em prever uma representação, seja visual ou textual, para entender como um modelo funciona ou porquê certa decisão foi tomada.

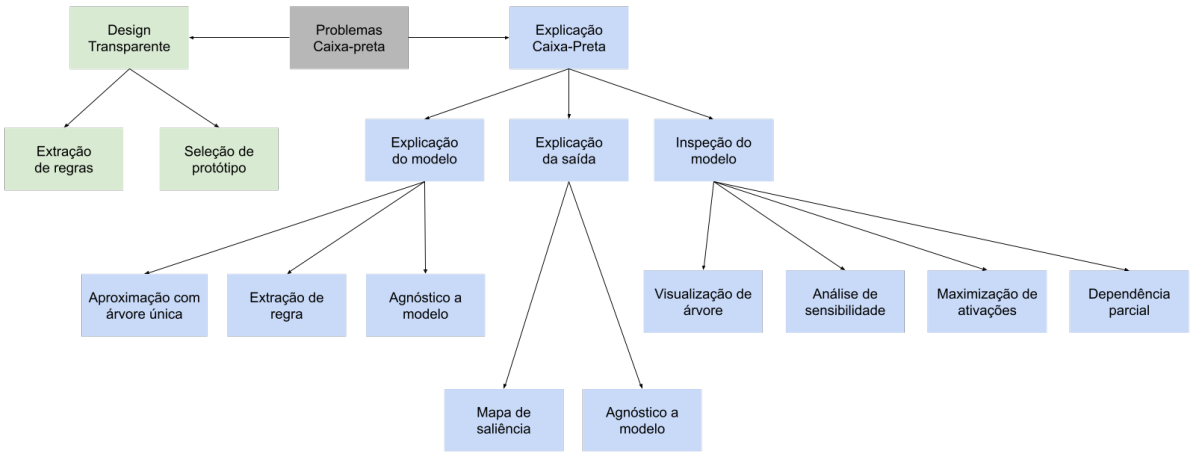


Figura 3.2: Taxonomia proposta por Guidotti *et al.* [17].

Ras *et al.* [39] (Figura 3.3) propõem um taxonomia mais simples e que consiste em métodos de extração de regra, métodos de atribuição e métodos intrínsecos. Os métodos de extração de regra buscam regras para aproximar o processo de tomada de decisão usando entrada e saída da rede profunda. Essa abordagem pode ser dividida em três categorias: 1) abordagem com decomposição, onde trabalhamos com partes menores dos modelo; 2) abordagem pedagógica, onde a tarefa de extração de regra é vista como uma tarefa de aprendizado; e 3) abordagem eclética, que utiliza conhecimentos da arquitetura e/ou vetores de pesos da rede como complemento de um algoritmo simbólico. Os métodos de atribuição medem a importância de um componente alterando a entrada e gravando o quanto a mudança afeta a performance do modelo; métodos de oclusão, perturbação, exemplos adversariais, análise de diferença de predição e deleção se encaixam nessa categoria. Por último, os métodos intrínsecos buscam aumentar a interpretabilidade dos componentes internos com métodos que são parte da estrutura, alterando a função de *loss* e/ou adicionando modelos que adicionam função ou como parte da estrutura do modelo a ser explicado. Um ponto importante a ser ressaltado é que os métodos intrínsecos não explicam nada por si só, mas tentam fazer o modelo inerentemente interpretável alterando a arquitetura das redes.

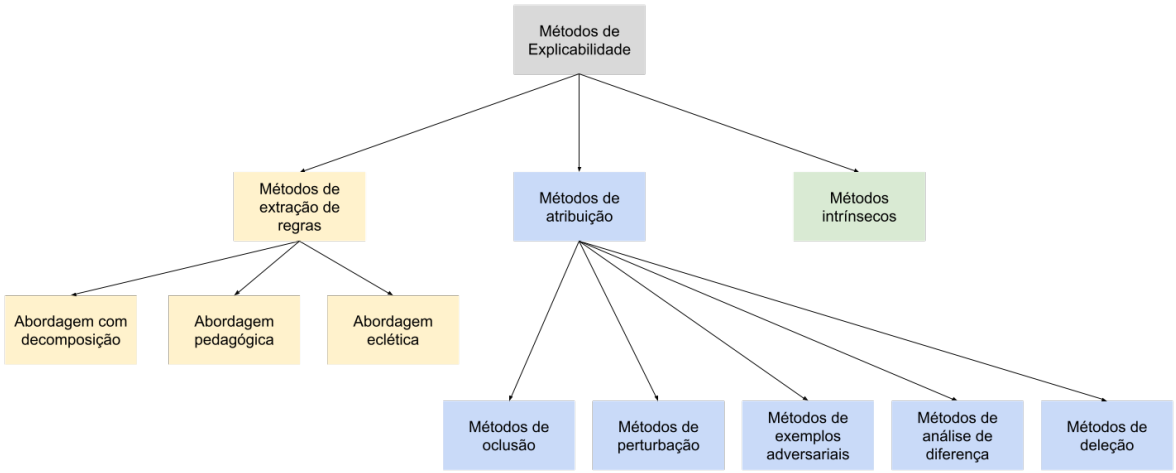


Figura 3.3: Taxonomia proposta por Ras *et al.* [39].

Já no trabalho de Arrieta *et al.* [7] (Figura 3.4), os autores propõem uma taxonomia que pode ser dividida entre modelos transparentes e métodos de explicação *post-hoc*. Os modelos transparentes incluem os algoritmos que possuem as três propriedades propostas por Lipton [28]: simulabilidade, capacidade de decomposição e transparência algorítmica. Já os métodos de explicação *post-hoc* podem ser agnósticos a modelo ou específicos a modelo, os métodos agnósticos a modelo podem ser divididos nas seguintes categorias:

- *Explicação por simplificação.* Uma das técnicas mais difundidas segundo Arrieta *et al.*, nessa categoria buscamos modelos que sejam mais simples e representem o modelo caixa-preta que queremos compreender. Explicações locais se encaixam nessa categoria, já que algumas vezes os modelos simplificados são apenas representativos de partes do modelo mais complexo;

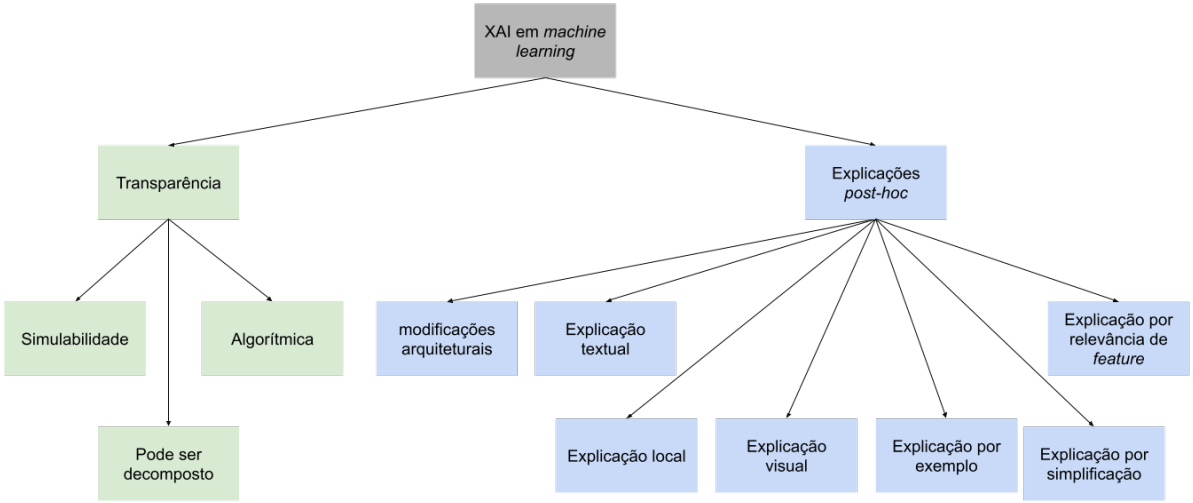


Figura 3.4: Taxonomia proposta por Arrieta *et al.* [7].

- *Explicação por relevância de feature* buscam explicar o funcionamento dos modelos opacos medindo a influência, relevância ou importância que cada *feature* tem na saída da predição do modelo a ser explicado;

- *Explicações visuais* são um meio de atingirmos as explicações agnósticas a modelo, criando visualizações da entrada para a saída dos modelos opacos. Por esse motivo, a maioria dos trabalhos que se encaixam nessa categoria também trabalham com técnicas de relevância de *feature*, que provê informações que podem ser mostradas ao usuário final.

As técnicas consideradas específicas ao modelo por Arrieta *et al.* possuem uma divisão de acordo com o tipo do modelo a que se aplicam. Esses são: conjuntos e sistemas de múltiplos classificadores, SVM (*support vector machines*), redes neurais multi camadas, redes neurais convolucionais e redes neurais recorrentes. Dentro de cada um desses tipos de modelos podemos encontrar ainda uma divisão de acordo com como o método funciona, que são as mesmas subcategorias dos métodos agnósticos a modelo acrescidas, em alguns casos, das categorias de explicações textuais e modificações de arquitetura.

Em seu trabalho, Fan *et al.* [16] (Figura 3.5) propõem uma divisão entre análise de interpretabilidade *post-hoc* e modelagem interpretável *ad-hoc*. A análise de interpretabilidade *post-hoc* explica modelos que já existem e esses métodos podem ser agrupados nas seguintes categorias:

- *Análise de feature* são técnicas centradas em comparar, analisar e visualizar *features* de neurônios e camadas da rede, o que permite encontrar *features* sensíveis e formas de processá-las de tal forma que a lógica do modelo pode ser explicada;

- *Inspeção do modelo* são métodos que usam algoritmos externos para entrar na rede neural, extraindo de forma sistemática estruturas importantes e informações paramétricas dos mecanismos internos da rede;

- *Métodos de saliência* identificam atributos dos dados de entrada que são mais relevantes para a predição ou são uma representação latente do modelo;
- *Proxy*, nessa categoria os métodos constroem modelos mais simples e interpretáveis que possuem grande semelhança com o modelo caixa-preta treinado. Esses métodos podem ser tanto locais quanto globais;
- *Análise física/matemática avançada* colocam o modelo em um quadro teórico matemático, no qual os mecanismos da rede podem ser entendidos através de ferramentas matemáticas avançadas;
- *Explicação por caso* prove exemplos representativos que capturam a essência de um modelo. No entanto, essa prática é uma *sanity check* do que é uma explicação já que não conseguimos muitas informações do funcionamento da rede a partir dos exemplos selecionados;
- *Explicação textual* geram um texto descritivo em tarefas de imagem-texto que são condutoras para entender o comportamento do modelo.

As técnicas de modelagem interpretável *ad-hoc* buscam construir modelos interpretáveis e podem ser categorizadas em:

- *Métodos de representações interpretáveis* empregam técnicas de regularização para orientar a otimização de uma rede neural em direção a uma representação mais interpretável;
- *Métodos de renovação de modelo* buscam interpretabilidade por meio de projetar e implantar mecanismos mais interpretáveis na rede.

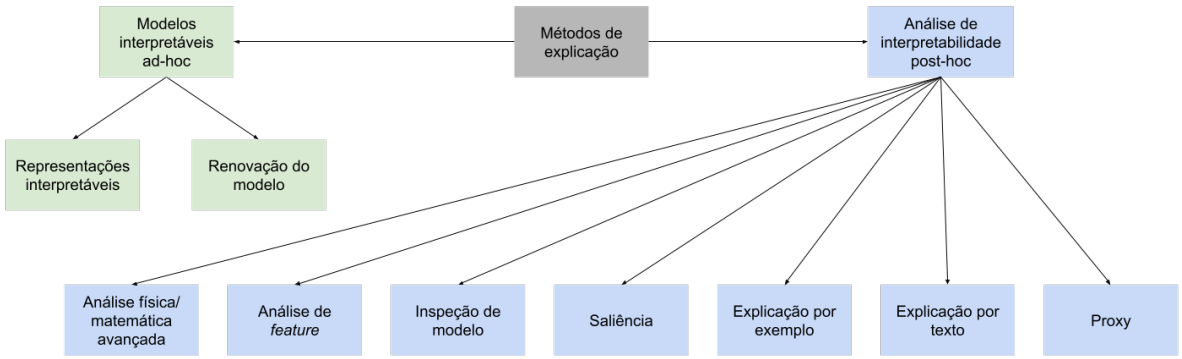


Figura 3.5: Taxonomia proposta por Fan *et al.* [16].

Das e Rad [14] (Figura 3.6) propõem uma categorização geral dos métodos de inteligência artificial de acordo com *escopo*, *metodologia* e *uso*. O escopo se refere aonde o método é focado: em uma instância (local) ou ao modelo como um todo (global). Já a metodologia compete ao qual a abordagem algorítmica é utilizada: 1) retropropagação, onde o algoritmo faz uma ou várias passagens pela rede e geram as atribuições durante a retropropagação, ou 2) baseado em perturbação, onde o algoritmo é focado em perturbar

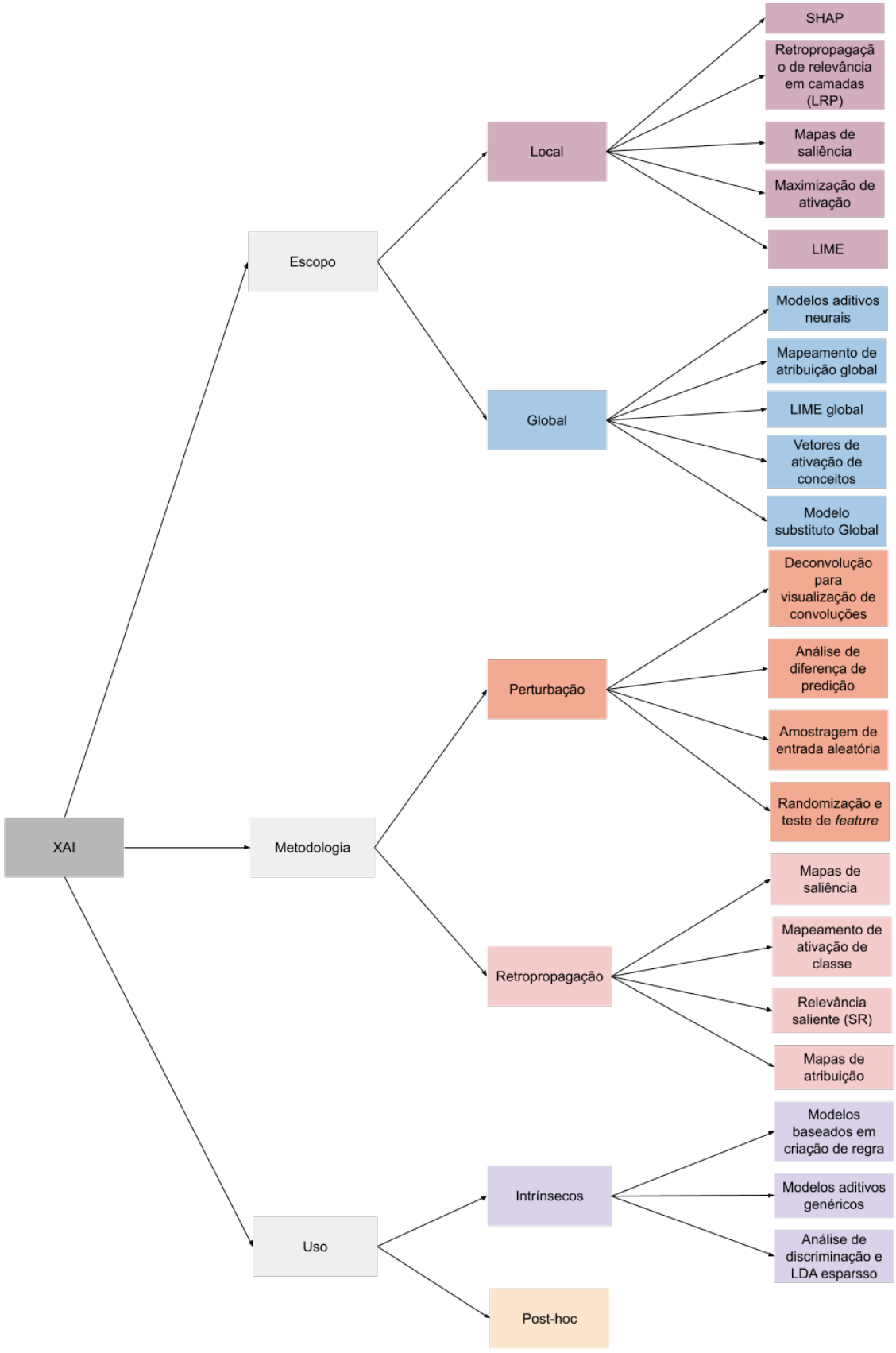


Figura 3.6: Taxonomia proposta por Das e Rad [14].

as *features* de uma entrada para gerar as representações das atribuições sem a necessidade de retropropagação de gradientes. Quanto ao uso, é sobre se o método foi construído na arquitetura da rede (intrínseco) ou se não depende da arquitetura do modelo e pode ser aplicado a modelos já treinados (*post-hoc*). Cada uma dessas subcategorias ainda pode ser dividida de acordo com os métodos e resultados de cada técnica.

3.2 Taxonomia Proposta

Como podemos observar na seção anterior, temos várias taxonomias propostas na literatura, cada uma com estrutura diferente. Apesar de sempre terem algumas similaridades, elas nunca são agrupadas em uma única taxonomia. Isso pode ocorrer porque por vezes os autores divergem sobre os conceitos básicos dentro da área de interpretabilidade. Em virtude dessas diversas classificações, decidimos nos basear nas taxonomias descritas na seção anterior e criar uma nova taxonomia que as associe, mantendo as divisões principais de modelos transparentes, que muitas vezes aparecem sob outros nomes, e as explicações *post-hoc*, agrupando os métodos de acordo com as técnicas de explicação utilizada, e que leve em consideração as definições adotadas no decorrer desta dissertação. Na Figura 3.7, ilustramos a taxonomia proposta.

Os modelos transparentes podem contar com transparência em nível de:

- *Modelo*: o ser humano consegue visualizar o modelo completo de uma vez. Essa propriedade também é chamada de simulabilidade, já que em modelos com esse nível de transparência o usuário é capaz de simular o comportamento do modelo. Um exemplo de modelo simulável são os modelos bayesianos onde a relação estatística modelada entre as variáveis e as próprias variáveis devem ser diretamente compreensível para determinada audiência.

- *Componentes*: também referente a capacidade de decompor o modelo e cada uma de suas partes, por exemplo entrada, parâmetros e pesos, ser interpretável. Um exemplo de modelo transparente a nível de componente é o *k-Nearest Neighbors*, em alguns casos a quantidade de variáveis analisadas é tão alta e/ou as métricas de similaridade são tão complexas que esses parâmetros podem ser decompostos e analisados separadamente.

- *Algoritmo*: também denotada por transparência algorítmica, pode ser vista de diferentes maneiras, mas se refere prioritariamente a capacidade do usuário final entender o processo seguido pelo modelo para produzir uma saída a partir de um dado de entrada. Segundo Arrieta *et al.* [7], a principal restrição de modelos algoritmicamente transparentes é a capacidade de ser completamente explorável através de métodos e análises matemáticas. Um exemplo de modelo transparente a nível algorítmico são árvores de decisão, onde as regras humanamente inteligíveis explicam o conhecimento adquirido dos dados e permite para um entendimento direto do processo de predição.

Quanto as técnicas de explicabilidade *post-hoc* propomos três categorias:

- *Inspeção do modelo*: consiste em prover uma representação para entender alguma propriedade específica do modelo caixa-preta ou suas predições. Esses métodos podem usar algoritmos externos para aprofundar no modelo extraindo sistematicamente informações estruturais e paramétricas dos mecanismos internos da rede [16]. Essa explicação pode ocorrer através de:

  - *Modificações arquiteturais*: nessa categoria, se encaixam as técnicas que propõem a adição ou alteração de algum componente da rede. Pode ser uma nova função de erro [15], um módulo que adiciona capacidades a mais ao modelo [45] ou até mesmo uma alteração em parte da estrutura arquitetural do modelo, como operações entre as camadas de uma rede neural [26], por exemplo.
  - *Dependência parcial*: esse grupo de métodos utiliza o gráfico de dependência parcial (PDP, do inglês, *partial dependence plot*), que é uma ferramenta de visualização da relação entre as variáveis responsáveis e as variáveis preditoras em um espaço reduzido de *features*. Krause *et al.* [25] adicionam uma perturbação aleatória nos valores de entrada do modelo para entender como cada *feature* impacta a predição, para isso utilizaram um gráfico de dependência parcial.
  - *Visualização de árvore*: nessa categoria, os métodos buscam extrair uma interpretação visual de redes neurais profundas utilizando árvores de decisão. Por exemplo, Thiagarajan *et al.* [58] propõem o *TreeView*, que dado um modelo caixa preta, ele primeiro decompõe o espaço do recurso em k (definidos pelo usuário) fatores de sobreposição. Em seguida, ele cria um meta-recurso para cada um dos k *clusters* e classificador *random forest* que prevê os rótulos de *cluster*. Finalmente, ele mostra uma árvore de decisão substituta como uma aproximação da caixa preta.

- *Extração de regras*: esses métodos aproximam o processo de tomada de decisão de uma rede neural profunda utilizando a entrada e saída da rede para extrair regras que sejam humanamente interpretáveis. Métodos de extração de regras podem validar se a rede está funcionando como o esperado quanto ao fluxo lógico do modelo e também para explicar quais aspectos do dado de entrada tem efeitos que levam a determinada saída [39].

- *Métodos de atribuição*: medem a importância de determinado componente alterando a entrada ou componentes internos do modelo e gravando o quanto essa alteração impactou o desempenho do algoritmo. Essa categoria pode refletir intuitivamente com explicações visuais quais fatores da dimensão de entrada tem um impacto significativo na saída do modelo [39]. Dentre dos métodos de atribuição podemos ter as seguintes categorias:

  - *Vetores de ativação de conceito*: CAVs (*concept activation vectors*) é um método de explicabilidade local que busca interpretar estados internos da rede neural em domínio de conceitos amigáveis para humanos. Kim *et al.* [24] propuseram um novo método TCAV (*testing with CAVs*) que usa derivadas

direcionais semelhantes aos métodos baseados em gradiente para avaliar a sensibilidade das previsões de classe para mudanças em dados em relação à direção do conceito para uma camada específica.

- *Mapas de saliência*: identificam quais atributos do dado de entrada são mais relevantes para a predição ou uma representação latente do modelo. Esses métodos são populares nas pesquisas sobre interpretabilidade [16] e extensivos testes aleatórios podem ser independentes de modelo e independentes dos dados [5]. Nessa categoria temos os métodos de gradiente [52, 53, 57] e técnicas de CAM [13, 49, 63].
- *Explicações locais*: consiste em prover uma explicação para a saída do modelo caixa-preta para uma instância. Nessa categoria de métodos não é necessário explicar a lógica inteira por trás do modelo mas apenas o raciocínio para a predição de uma instância de entrada específica. As pesquisas nessa área usam mapas de calor, técnicas bayesianas e técnicas de importância de *features* para entender a correlação dos atributos e a importância na predição. Por exemplo, Batch *et al.* [8] propuseram a técnica LRP (*Layer-wise Relevance BackPropagation*) que encontra pontuação de relevância para atributos individuas de um dado de entrada decompondo a saída da predição de redes neurais profundas. A pontuação de relevância é calculada para cada entrada através da retropropagação da pontuação de classe de um nó de saída através da camada de entrada.
- *Explicação textual*: geram uma descrição textual em tarefas conjuntas imagem-linguagem que são condutoras para entender o comportamento do modelo. Nessa categoria temos métodos de atenção como no trabalho de Xu *et al.* [65], onde os atributos são alinhados as descrições textuais correspondentes por uma rede neural recursiva como as redes LSTMs (*Long short-term memory*) [18].
- *Análise de feature*: técnicas centradas em comparação, análise e visualização de *features* dos neurônios e camadas. Atributos sensíveis e as formas de processar elas são identificadas de forma que a lógica do modelo pode ser compreendida. Os métodos de maximização de ativação, que se encaixam nessa categoria também, se dedicam a sintetizar imagens que maximizam a saída de uma rede neural ou neurônios de interesse. As imagens resultantes são chamadas de "*deep dream*", pois podem ser consideradas imagens oníricas de uma rede neural ou de um neurônio. Zeiler *et al.* [66] modelaram uma rede deconvolucional que consiste em operações de *unpooling*, retificação e operações de deconvolução, para emparelhar com a rede original para que as *features* possam ser invertidas sem a necessidade de retreinar o modelo.
- *Explicação por exemplo*: apresentam um exemplo que é considerado como o mais semelhante a instância de consulta que precisa de uma explicação. Encontrar uma instância similar e selecionar uma instância representativa dos dados como um protótipo [9] são basicamente a mesma coisa e só usam métricas diferentes. Enquanto a seleção de protótipos busca encontrar um subconjunto de instâncias que representam a base de dados inteira, explicações baseadas em

exemplos usam métricas de similaridade baseada em proximidade de representações de uma rede neural, de forma a expor as informações de representação interna da rede. Wallace *et al.* [61] empregam a técnica KNN para obter o caso mais similar do caso de consulta em um espaço de *features* e então computam a porcentagem de vizinhos próximos que pertencem a classe esperada como uma medida para interpretabilidade, o que sugere o quanto uma predição é suportada pelo dado.

– *Métodos de oclusão*: buscam encontrar as regiões mais importantes da imagem adicionando elementos que escondem partes da imagem e observando como isso afeta o resultado das predições. Zeiler *et al.* [66] também buscam visualizar ativação de camadas individuais internas da rede através da oclusão de diferentes regiões da imagem de entrada, gerando visualizações com a rede deconvolucional (DeConvNet).

– *Proxy*: constroem um modelo mais simples e interpretável que assemelha-se a um modelo caixa-preta treinado, grande e complexo. Esses métodos podem ser locais, em um espaço parcial, ou global, em um espaço de solução inteiro. Um exemplo de técnica de Proxy é o LIME [41], que busca aproximar o modelo que queremos explicar através de um modelo de regressão linear, que imita o comportamento do modelo alvo.

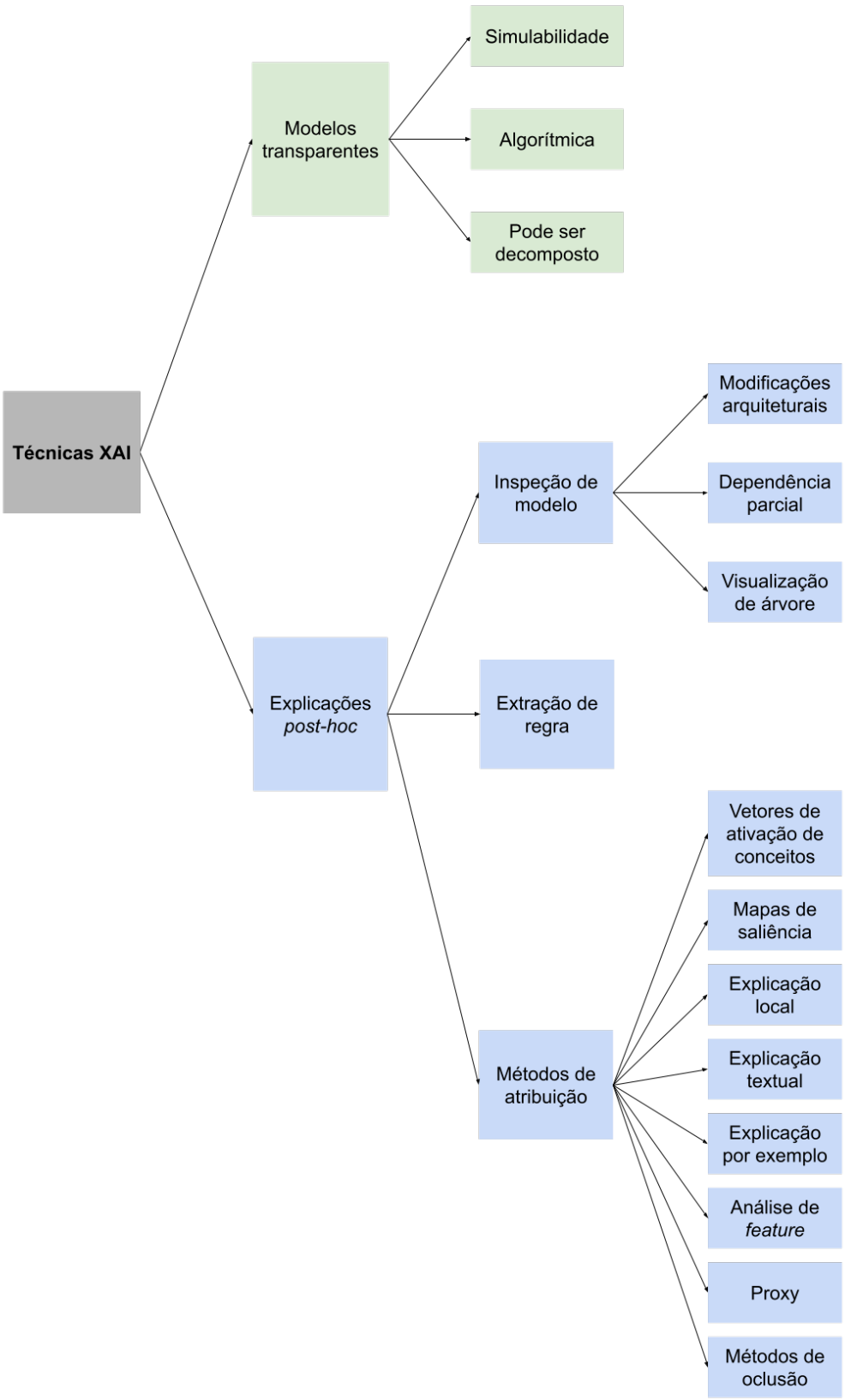


Figura 3.7: Taxonomia das técnicas de inteligência artificial explicável proposta com base nos taxonomias e trabalhos investigados.

Na Figura 3.8, comparamos as categorias do último nível de cada taxonomia da seção anterior e a categoria correspondente da taxonomia proposta. Um ponto importante a ressaltar é que nenhuma taxonomia possui todas as divisões que estamos propondo e que temos algumas categorias que aparecem em diversos trabalhos com nomes diferentes.

<table>
<tr><th>Taxonomia proposta</th><th>Taxonomia Lipton [25]</th><th>Taxonomia Guidotti et al [16]</th><th>Taxonomia Ras et al [33]</th><th>Taxonomia Arrieta et al [7]</th><th colspan="2">Taxonomia Fan et al [15]</th><th colspan="2">Taxonomia Das e Ras [13]</th></tr>
<tr><td>Simulabilidade</td><td>Modelo</td><td rowspan="3"></td><td rowspan="3">Modelos Intrínsecos</td><td>Simulabilidade</td><td colspan="2" rowspan="3">Representação interpretáveis</td><td colspan="2" rowspan="3"></td></tr>
<tr><td>Algorítmica</td><td>Algoritmo</td><td>Algoritmo</td></tr>
<tr><td>Pode ser decomposto</td><td>Pode ser decomposto</td><td>Pode ser decomposto</td></tr>
<tr><td>Visualização de árvore</td><td rowspan="6"></td><td>Aproximação com árvore única</td><td rowspan="3"></td><td rowspan="2"></td><td rowspan="4">Inspeção do modelo</td><td rowspan="2"></td><td rowspan="3">Análise de discriminação e LDA esparso</td><td rowspan="2"></td></tr>
<tr><td>Modificação arquitetural</td><td></td><td>Modificação arquitetural</td><td rowspan="2">Renovação do modelo</td><td rowspan="2">Modelos Aditivos Neurais e Genéricos</td></tr>
<tr><td>Dependência parcial</td><td>Dependência parcial</td><td rowspan="4"></td><td rowspan="2"></td></tr>
<tr><td>Extração de regra</td><td>Extração de regra</td><td>Extração de regra</td><td colspan="2">Modelo baseado em regras</td></tr>
<tr><td>Vetores de ativação de conceito</td><td></td><td rowspan="3"></td><td colspan="2">Vetores de ativação de conceitos</td></tr>
<tr><td>Mapas de saliência</td><td>Visualização</td><td>Mapa de saliência</td><td colspan="2">Saliência</td><td colspan="2">Mapas de saliência, SR, Deconvolução para visualização de deconvolução, CAM</td></tr>
<tr><td>Explicação local</td><td>Explicação local</td><td rowspan="4"></td><td rowspan="3"></td><td>Explicação local</td><td colspan="2"></td><td colspan="2">SHAP, LIME, LPR, Amostragem de entrada aleatória</td></tr>
<tr><td>Explicação textual</td><td>Explicação textual</td><td>Explicação textual</td><td colspan="2">Explicação textual</td><td colspan="2" rowspan="2"></td></tr>
<tr><td>Explicação por exemplo</td><td rowspan="4"></td><td>Modelo de exemplo adversarial</td><td>Explicação por exemplo</td><td colspan="2">Explicação por exemplo</td></tr>
<tr><td>Análise de feature</td><td>Maximização de ativação, Análise de sensibilidade</td><td>Método deleção, Análise de diferença</td><td>Explicação por relevância de feature</td><td colspan="2">Análise de feature</td><td colspan="2">Análise de diferença de predição, Randomização de teste de feature</td></tr>
<tr><td>Proxy</td><td>Seleção de protótipo</td><td></td><td>Explicação por simplificação</td><td colspan="2">Proxy</td><td colspan="2">Modelo substituto</td></tr>
<tr><td>Método de oclusão</td><td></td><td>Método de oclusão</td><td></td><td colspan="2"></td><td colspan="2"></td></tr>
</table>

Figura 3.8: Comparativos das taxonomias da literatura com a taxonomia proposta.

Capítulo 4

Trabalhos Relacionados

Para realizar o levantamento dos trabalhos relacionados foi feita uma revisão do estado da arte utilizando-se o *Google Scholar*. Como termo de busca foi utilizado "*image_ classification interpretability explanation cnn visual_ explanation -npl -tabular*", que retornou 357 trabalhos. Esse conjunto foi reduzido alterando-se o termo de busca para excluir trabalhos relacionados a área médica e trabalhos voltados para dados textuais, o que nos restou 57 trabalhos. A partir da análise do conteúdo do trabalho, realizamos uma filtragem final, sobrando 13 artigos. Adicionamos mais 5 trabalhos a esse conjunto, que foram os métodos avaliados que não apareceram na busca: SmoothGrad [53], Gradiente Integrado [57], LIME [41] e Gradiente *Vanilla* [52]. Ao total, analisamos 18 trabalhos da literatura.

Na Tabela 4.1, sumarizamos os trabalhos resultantes desta pesquisa, considerando as seguintes informações: 1) tipo de publicação (periódico, conferência ou *preprint*), 2) nome da técnica, 3) bases de dados utilizadas nos experimentos, 4) método de avaliação utilizado (qualitativo e/ou quantitativo), e se for quantitativo, listamos 5) as métricas utilizadas, 6) tipo dos dados em que o método foi proposto (imagem, vídeo e/ou texto), 7) técnica que o método utiliza para gerar os resultados e, por último, 8) classificação do método de acordo com a taxonomia proposta na Seção 3.2.

De maneira geral, como pode ser visto na Tabela 4.1, os métodos de explicabilidade propostos são aplicados em base de dados genéricas (por exemplo, ImageNet, VOC 2007, MS COCO 2014) e as análises são feitas de forma qualitativa (ou seja, comparações visuais). Além disso, a maioria dos trabalhos na área de explicabilidade está voltada para classificação de imagens e poucos trabalhos se arriscam nos modelos de classificação de vídeos. A seguir, sumarizamos as principais contribuições dos 18 trabalhos, de forma cronológica (2021 a 2014). Os detalhes podem ser encontrados na Tabela 4.1.

Sudhakar *et al.* [56] propuseram o ADA-SISE (*Adaptative Semantic Input Sampling for Explanation*), uma melhoria do método SISE [46], que permite a seleção de forma adaptativa das informações sobre as *features* mais importantes para a predição, isso age como um filtro que transforma o método existente em uma solução unificada e automatizada eliminando a necessidade de um usuário alterar os hiperparâmetros. O tempo de execução, em relação ao algoritmo base, é reduzido em até 30%, enquanto melhora a interpretabilidade sem comprometer a qualidade da explicação gerada.

Li *et al.* [27] exploram um método de perturbação genérico e propõem uma nova função de erro que permite a suavização dos atributos na dimensão espacial e temporal, o que

ajuda no desempenho do modelo na classificação de vídeos através dessa suavização. Essa nova função permite a comparação de resultados entre diferentes redes e pode evitar a geração de explicações adversas para as características de entradas de um vídeo.

Sattarzadeh *et al.* [47] apresentam o *Integrated* GradCAM, uma combinação dos métodos gradiente integrado e GradCAM, a fim de resolver o problema que esse último método pode ter por subestimar a contribuição de representações descobertas pelo modelo em relação a suas predições, devido a sua implementação de termos baseados no gradiente médio. O método *Integrated* GradCAM tenta resolver os problemas de gradiente usando os benefícios das técnicas de retropropagação.

Jung e Oh [22] construíram um modelo de explicação como uma função linear que denotam a existência de um mapa de ativação. Dessa forma, o modelo pode determinar uma explicação visual de CAM (*class map activation*), que será utilizado para formular os valores SHAP [29] como uma solução unificada, que permite a estimativa desses valores SHAP a partir dos mapas de ativação com uma única retropropagação. O método é denominado como LIFTCAM por ser baseado na técnica DeepLIFT [51].

Shi *et al.* [50] propuseram o ZoomCAM que vai além da última camada convolucional integrando os mapas de importância por todas as camadas intermediárias. Assim, o método captura objetos de pequena escala refinados para várias instâncias de classe discriminativas, que são comumente perdidas pelos métodos de visualização tradicionais, como GradCAM++ [13] e o ScoreCAM [63].

Muhammad e Yeasin [33] apresentaram o método EigenCAM, que computa e usa os principais componentes aprendidos das camadas convolucionais. Esse processo para obter o CAM é independente da pontuação de relevância de classe. O método assume que todas as características espaciais relevantes na entrada aprendidas na hierarquia da rede convolucional vai ser preservada no processo de otimização, já as características menos importantes vão ser regularizadas e suavizadas.

Stergiou *et al.* [54] propuseram um método transversal a toda estrutura da rede neural e incrementalmente descobre *kernels* de diferentes profundidades que são informativos para uma classe específica. Esse método, denominado pirâmide de características de classe, busca os atributos e suas hierarquias correspondentes, baseadas em ativações das camadas prévias da rede neural.

Stergiou *et al.* [55] propuseram um método de retropropagação de mapas de ativação para explicação de vídeos, chamado de tubos de saliência, que buscam pontos e regiões que são consideradas pontos de foco da rede, tanto em nível de *frame* quanto a nível temporal. Essa é uma técnica discriminativa de classe que gera explicações visuais de qualquer rede convolucional 3D, sem a necessidade de retreinar ou fazer mudanças arquiteturais no modelo.

Omeiza *et al.* [35], com a intenção de melhorar uma explicação visual em termos de nitidez visual, localização de objetos e múltiplas ocorrências de objetos em uma imagem simples, apresentaram o SmoothGradCAM++, uma técnica que combina as técnicas SmoothGrad e GradCAM++.

Nauta *et al.* [34] apresentaram a árvore de protótipo neural (*ProtoTree*), um método de aprendizado profundo que inclui protótipos de uma árvore de decisão interpretável para visualizar o modelo inteiro de forma fiel. Nessa árvore binária, cada nó possui uma

parte do protótipo treinável. A presença ou ausência desse protótipo em uma imagem determina o roteamento através do nó. Uma *ProtoTree* tem um poder representacional de uma rede neural e também contém uma estrutura de árvore de decisão binária, que aproxima a acurácia de um classificador não interpretável e oferece explicações locais e globais.

Wang *et al.* [62] propuseram a interpretação CHAIN (*Concept-harmonized Hierarchical Inference Interpretation of Deep Convolutional Neural Network*) em que a decisão da rede pode ser hierarquicamente deduzida em conceitos visuais de alto a baixo nível semântico. Esse processo acontece em três etapas: 1) o modelo de harmonização de conceito conceitos de um nível semânticos de a baixo são alinhados com os neurônios da rede de um modelo profundo para um modelo caixa-branca; 2) o modelo de inferência hierárquica, o conceito dos neurônios do modelo profundo é desmontado nas unidades da camada do modelo caixa-branca; e 3) o modelo de inferência hierárquica de conceitos harmonizados, o conceito de uma camada profunda é inserido em um conceita de uma camada rasa. Finalmente, a rede tomadora de decisão é explicada como uma forma de inferências hierárquica de conceitos harmonizados.

Os métodos *Vanilla Gradient*, Gradiente Integrado, SmoothGrad, LIME, GradCAM, GradCAM++ e ScoreCAM foram explicados com mais detalhes na Seção 2.2. Em resumo, Simonyan *et al.* [52] propuseram a utilização do gradiente como pesos para identificar quais *pixels* possuem mais impacto na classificação quando são alterados. Essa técnica é referenciada como *Vanilla Gradient* em outros trabalhos. Sundararajan *et al.* [57] propuseram o Gradiente Integrado, que é obtido a partir da integral dos gradientes acumulados entre a amostra que queremos explicar e o *baseline*, no caso de uma imagem, seria uma imagem com todos os *pixels* zerado. Smilkov *et al.* [53] propuseram a técnica *SmoothGrad*, que busca melhorar o resultado da técnica proposta em [52] adicionando ruído gaussiano na amostra que queremos explicar. Ribeiro *et al.* [40] propuseram o LIME (*Local Interpretable Model-Agnostic Explanations*), uma técnica que tenta aproximar um modelo interpretável para explicar a instância alvo, através do cálculo de importância de cada *superpixel* para a classificação. Selvaraju *et al.* [49] propuseram o GradCAM, uma técnica para produzir explicações visuais para as decisões de CNN, utilizando o gradiente do conceito que queremos identificar na imagem. Chattpadhay *et al.* [13] propuseram uma melhoria na técnica de Selvaraju *et al.* [49], o GradCAM++, utilizando uma combinação de derivadas parciais dos mapas de atributos da última camada convolucional para gerar uma explicação visual para a classe alvo. Por fim, Wang *et al.* [63] propuseram o ScoreCAM, um método que obtém o peso de cada mapa de ativação por meio de sua pontuação de confiança, eliminando a dependência do gradiente.

Ref.$_{\text{Ano}}$	Publicação	Nome	Base de dados	Avaliação	Métricas	Dados	Técnicas	Classificação
[56]$_{2021}$	ICASSP	Ada-SISE	VOC 2007	quant.	energy-based pointing game, bounding box, drop rate, increase rate	imagem	retropropagação e SISE	análise de features
[27]$_{2021}$	WACV	–	UFC101-24, EPIC-Kitchens	qual. + quant.	spatial pointing game	vídeo	perturbação combinada com uma nova função de erro	modificações arquiteturais
[47]$_{2021}$	ICASSP	Integrated GradCAM	VOC 2007	qual., quant.	energy-based pointing game, bounding box, drop rate, increase rate	imagens	gradiente	mapas de saliência
[22]$_{2021}$	preprint	LIFTCAM	ImageNet, MSCOCO 2014, VOC 2007	qual.	–	imagem	gradiente e valores SHAP	mapas de saliência
[50]$_{2020}$	ICPR	ZoomCAM	ImageNet, VOC 2012	quant.	intersecção sobre união	imagem	gradiente	mapas de saliência
[33]$_{2020}$	IJCNN	EigenCAM	ILSVRC 2014	quant.	taxa de erro de intersecção sobre união	imagens	projeção dos pesos da ultima camada na imagem	análise de features
[63]$_{2020}$	CVPR	ScoreCAM	ImageNet	qual. + quant.	average drop rate, average increase rate	imagem	aumento de confiança	mapas de saliência
[34]$_{2020}$	preprint	ProtoTree	CUB-200-2011, Stanford Cars	qual.	–	imagem	modelo intrinsecamente interpretáveis, utiliza retropropagação e extração de regras	modelo transparente (simulabilidade)
[62]$_{2020}$	preprint	CHAIN	ImageNet, Places365	qual. + quant.	inference distance	imagem	inferência hierárquica de conceitos visuais	vetores de ativação de conceitos
[55]$_{2019}$	ICIP	Tubos de Saliência	Kinetics, UCF-101 e EPIC-kitchens	qual.	–	vídeo	retropropagação de mapas de ativação	análise de features
[35]$_{2019}$	IntelliSys	SmoothGradCAM++	ImageNet	qual.	–	imagem	gradiente	mapas de saliência
[54]$_{2019}$	ICCVW	Pirâmide de Características de Classe	Kinetics-400, HMDB-51, EPIC-kitchens, EGTEA Gale	qual.	–	vídeo	retropropagação de features e dependência de canal da classe e features	análise de features
[13]$_{2018}$	WACV	GradCAM++	ImageNet	quant.	average drop rate, % increase in confidence, win %	imagem	gradiente	mapas de saliência
[53]$_{2017}$	preprint	SmoothGrad	ILSVRC-2013	qual.	–	imagem	gradiente	mapas de saliência
[57]$_{2017}$	ICML	Gradiente Integrado	ImageNet	qual.	–	imagem, texto	gradiente	mapas de saliência
[49]$_{2016}$	NIPS	GradCAM	ImageNet, VOC 2017	qual.	–	imagem	gradiente	mapas de saliência
[41]$_{2016}$	KDD	LIME	Husky vs Lobo	qual.	–	imagem, tabular, texto	interpretação local, com aproximação de um modelo interpretável	explicação local
[52]$_{2014}$	preprint	Gradiente *Vanilla*	ILSVRC-2013	qual.	–	imagem	gradiente	mapas de saliência

Tabela 4.1: Tabela comparativa dos trabalhos encontrados na revisão da literatura.

Capítulo 5

Metodologia Proposta

Neste capítulo, vamos apresentar a metodologia proposta para analisar os métodos de explicabilidade de redes neurais profundas no problema de classificação de *Elsagate*. Na Seção 5.1, descrevemos a base de dados utilizada e, em seguida, na Seção 5.2, detalhamos a metodologia proposta para a geração dos modelos utilizados. Tanto a base de dados quanto os modelos treinados foram propostos por Ishikawa *et al.* [21]. Por fim, na Seção 5.3, especificamos os métodos de explicabilidade analisados.

5.1 Base de Dados

Conteúdo sensível pode ser definido como qualquer material que possa ser desagradável, até mesmo uma ameaça, para sua audiência [32]. Dessa forma, violência e pornografia são considerados conteúdos sensíveis, mas, por exemplo, também devemos considerar conteúdos grotescos ou perturbadores como tal.

Ishikawa *et al.* [21] propuseram a primeira base de dados com desenhos animados do tipo *Elsagate* (conteúdo perturbador para crianças). A base contém 285 horas, sendo 1396 vídeos contendo *Elsagate* e 1898 vídeos com conteúdo não-sensível. Para criar essa base de dados, Ishikawa *et al.* baixaram os vídeos rotulados como *Elsagate* de canais do YouTube reportados pelos usuários do Reddit no fórum "*What Elsagate*?" [1], e os vídeos que não possuem conteúdo sensível foram coletados de canais oficiais do YouTube, como *Cartoon Network* e *Disney*.

Na Figura 5.1, apresentamos algumas amostras dessa base de dados. Na primeira linha, temos alguns *frames* de vídeos anotadas como *Elsagate*, onde podemos ver que por vezes temos personagens infantis com traços diferentes do original, ou até mesmo a adição de rostos grotescos ou de personagens não infantis em situações esquisitas, que nem parecem fazer sentido. Na segunda linha da Figura 5.1, vemos *frames* de desenhos ou filmes retirados de canais oficiais de desenhos animados, que foram anotados como conteúdo não sensível, ou seguro.

Para analisar os métodos de explicabilidade, utilizamos os vídeos do conjunto de teste do trabalho de Ishikawa *et al.* [21], que contém 331 vídeos considerados não-sensível e 278 vídeos classificados como *Elsagate*. Alguns vídeos dessa classe estão disponíveis para

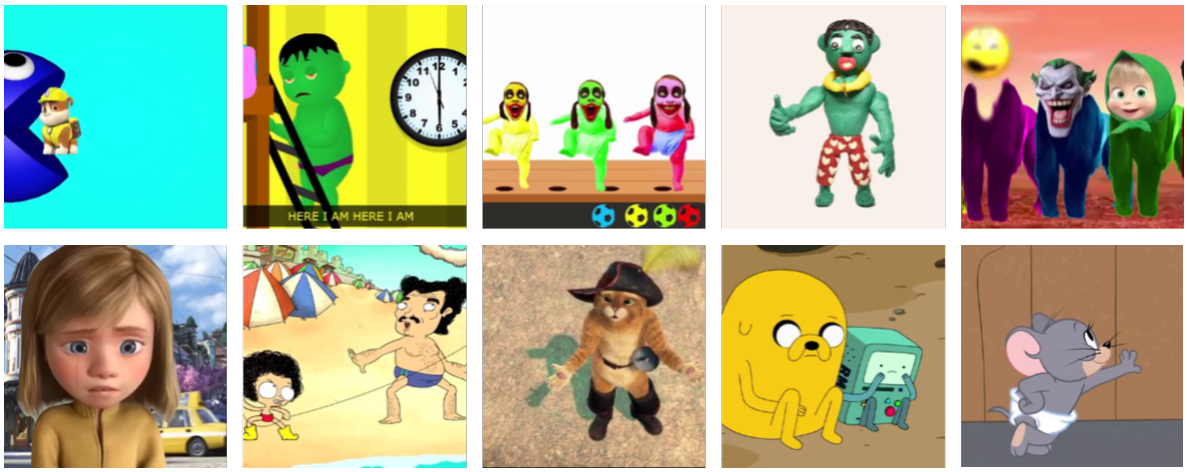


Figura 5.1: *Frames* presentes na base de dados apresentada por Ishikawa *et al.* [21], na linha de cima temos alguns exemplos de *frames* retirados de vídeos classificados como *Elsagate*, enquanto na de baixo alguns exemplos de *frames* de vídeos classificados como não sensíveis.

visualização[1]. Para facilitar a interpretação dos resultados dos métodos de explicabilidade foi utilizado o primeiro, o último e o *frame* do meio de cada vídeo, resultando em 1827 imagens no total para aplicarmos os métodos de explicabilidade. Optamos por utilizar apenas os *frames* de cada vídeo, uma vez que os resultados apresentados por Ishikawa *et al.* [21] não apresentaram diferenças significativas quando as informações estáticas e de movimento foram utilizadas para a classificação. Diante disso, optamos por analisar métodos de explicabilidade voltados para imagens ao invés de métodos voltados para informações temporais.

5.2 Modelos

Ishikawa *et al.* [21] propuseram um método, baseado no trabalho de Perez *et al.* [38], para classificar vídeos com conteúdo do tipo *Elsagate*. A visão geral do método é ilustrado na Figura 5.2.

Inicialmente, como informação estática, os *frames* de cada vídeo são extraídos numa amostragem de um *frame* por segundo. Esses *frames* são redimensionados para o tamanho de entrada das redes utilizadas (224×224 *pixels*), mantendo a razão de aspecto. Também foi utilizado a informação de movimento, uma vez que incorporar essas informações em redes neurais profundas leva a classificadores de vídeos sensíveis mais eficazes. Para isso, são extraídos vetores de movimento [2], que podem ser decodificados diretamente do arquivo compactado de vídeo com baixo custo computacional. Esse processo de decodificação inclui muitos subprocessos, sendo o principal deles a compensação de movimento por predição entre quadros, o que resulta no vetor de movimento, que é um descolamento de translação do *frame* de referência para o *frame* alvo, representando a movimentação de pequenas regiões de cada *frame*.

[1] `https://tinyurl.com/ratc3wju`

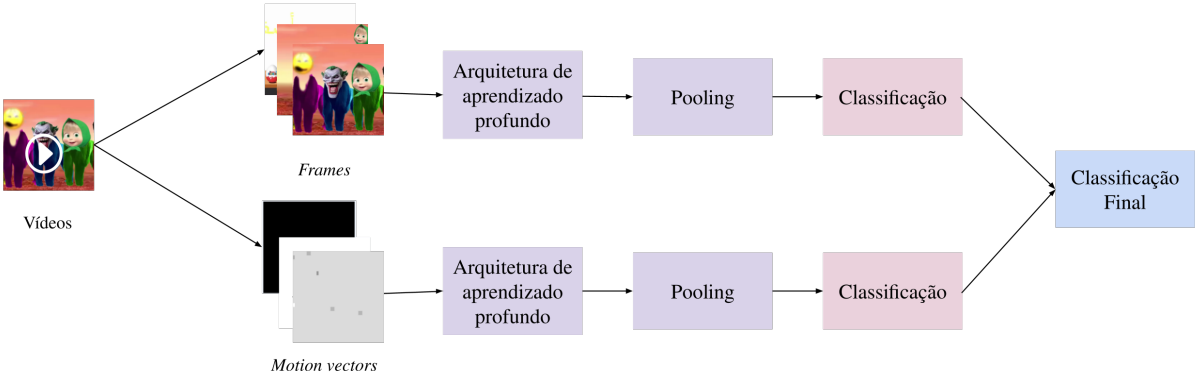


Figura 5.2: Visão geral do método proposto por Ishikawa *et al.* [21] para classificação de conteúdo *Elsagate*.

Em seguida, as informações extraídas dos vídeos (estáticas e de movimento) são dadas como entrada para redes neurais profundas. Ishikawa *et al.* [21] treinaram diversas arquiteturas (por exemplo, NASNet [68], SqueezeNet [19] e MobileNetv2 [44]). Para gerar a predição para os vídeos, os resultados para os *frames* e os vetores de movimento são agrupados (*pooling*).

A fusão da informação estática com a informação de movimento é realizada utilizando a abordagem de *late fusion*, ou fusão tardia, onde cada informação é processada por classificadores diferentes, gerando classificações independentes que são combinadas em uma pontuação final de classificação. Por fim, as máquinas de vetores de suporte (*support vector machines*, SVM) foi utilizado para o processo final de tomada de decisão.

O melhor resultado para Elsagate, acurácia de 92.6%, foi obtido com a NASNet no conjunto de teste. Para a SqueezeNet, a acurácia foi de 62.0%. Como a MobileNetv2 teve um pior desempenho nos dados de treinamento, o modelo não foi avaliado no conjunto de teste.

Para analisar os métodos de explicabilidade, os modelos escolhidos do trabalho de Ishikawa *et al.* foram a NASNet e a MobileNetv2, os modelos com melhor e pior desempenho, respectivamente. Os modelos já estão treinados.

5.3 Métodos de Explicabilidade

Para os métodos de explicabilidade, utilizamos a implementação disponível no `tensorflow explain` versão 0.0.2 dos métodos: *Vanilla Gradient*, Gradiente Integrado, SmoothGrad e GradCAM. Entre esses, alteramos a parametrização padrão apenas do método SmoothGrad, alterando o nível de ruído para 50%, uma vez que foi o nível de ruído com melhor resultado apresentado pelos autores do método em [53]. O GradCAM++ foi implementado no próprio arquivo de experimento utilizando algumas funções presentes no `tensorflow` com base no repositório oficial do `github`[2]. O LIME possui um pacote próprio, disponibilizado pelos autores da técnica. A versão utilizada nos experimentos foi a 0.1.1.137. Por último, a implementação do ScoreCAM que utilizamos está disponível no

[2] `https://github.com/adityac94/Grad_CAM_plus_plus`

repositório `scam-net`[3]. Para esse método, também utilizamos a parametrização padrão. Importante ressaltar que a versão utilizada do `keras` foi a 2.3.0 e a do `tensorflow` foi a 2.0.2.

Para conseguirmos aplicarmos os métodos nos modelos de validação disponibilizados por Ishikawa *et al.*, foi necessário remover a última camada dos mesmos. No primeiro momento, utilizamos a parametrização padrão dos pacotes onde tínhamos a implementação das técnicas, com exceção do SmoothGrad, como já dito anteriormente. Esses parâmetros padrões são:

- Gradiente *Vanilla*: não possui nenhum hiperparâmetro a ser ajustado;
- SmoothGrad: utilizamos 5 amostras por imagem e ruído igual a 0.5 ou 50%;
- Gradiente Integrado: número de passos entre o *baseline* e a imagem alvo igual a 10 passos;
- LIME: número de *features*, que indica o número de *superpixels* a serem incluídos na explicação, igual a 5;
- GradCAM: o hiperparâmetro de peso da imagem, que indica o peso da imagem que queremos entende para sobrepor com o mapa de atribuição calculado, foi de 0.7;
- GradCAM++: não possui nenhum hiperparâmetro a ser ajustado;
- ScoreCAM: o hiperparâmetro tamanho de *batch* de inferência por padrão é 32.

Como os resultados do LIME, como pode ser visto no próximo capítulo, ficaram difíceis de interpretar, pensamos que poderia ter sido a técnica de segmentação dos *superpixels* que podia ter atrapalhado. A técnica padrão do pacote é a *quickshift*, e portanto, realizamos um segundo experimento onde utilizamos o método *slic* na busca de uma divisão mais simples e fácil de interpretar.

[3] `https://github.com/andreysorokin/scam-net`

Capítulo 6

Resultados

Neste capítulo, vamos apresentar os resultados obtidos com as técnicas descritas na Seção 2.2. Os resultados estão agrupados em técnicas baseadas principalmente em gradiente (Seção 6.2.1), o LIME (Seção 6.2.2) e técnicas de mapa de ativação de classe (Seção 6.2.3). A seguir, vamos apresentar a ferramenta de visualização que foi desenvolvida para facilitar a análise dos resultados.

6.1 Ferramenta de Visualização

Para conseguirmos avaliar as sete técnicas aplicadas de forma mais simples, desenvolvemos uma ferramenta utilizando `React`[1] para exibir o resultado de todas as técnicas para os dois modelos simultaneamente (NASNet e MobileNet).

O desenvolvimento da ferramenta surgiu a partir da inexistência de ferramentas e métricas que permitem essa comparação. Atualmente, não temos métodos que quantificam a explicabilidade de um modelo. Alguns trabalhos até tentam avaliar os resultados dos métodos de visualização com técnicas que necessitam de um trabalho adicional para criarmos um *groundtruth* das regiões mais importantes da imagem. No entanto, no contexto analisado, não temos uma explicação objetiva nas imagens que possam ser ligadas diretamente com a classe *Elsagate* ou a classe não sensível; logo, não é possível criarmos um *groundtruth*.

Na Figura 6.1, podemos observar uma captura de tela da ferramenta. Rapidamente, conseguimos comparar as sete técnicas e os dois modelos. A ferramenta também permite a navegação direta para a primeira imagem classificada como não sensível e a primeira imagem classificada como *Elsagate*. Além disso, é possível navegar para uma imagem digitando seu índice na caixa de texto junto dos botões. Na figura também podemos observar, na parte inferior, os botões de navegação (anterior e próxima) do conjunto de dados.

A ferramenta foi desenvolvida utilizando `React`, uma biblioteca em JavaScript que facilita o desenvolvimento de aplicações *front-end*. Para executá-la, utilizamos um servidor `Flask`[2] que nos retorna o nome do arquivo da imagem, uma vez que não seria possível

[1] `https://reactjs.org`
[2] `https://flask.palletsprojects.com`

Figura 6.1: Captura de tela da ferramenta de visualização desenvolvida para avaliar o resultado das técnicas de explicabilidade.

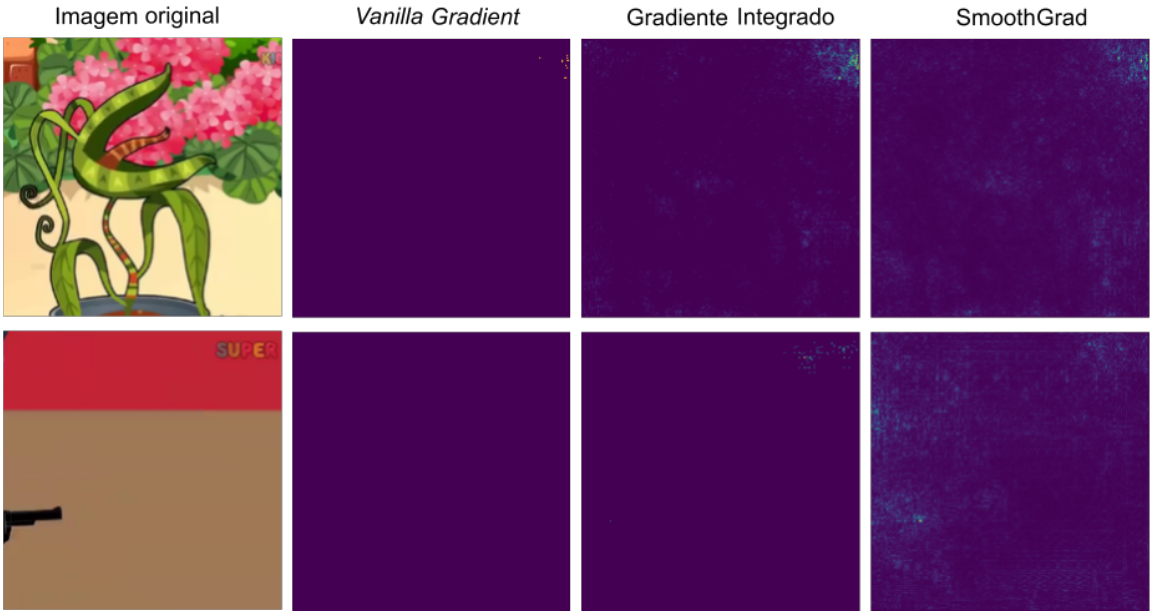
navegar pelos diretórios base diretamente utilizando `TypeScript`[3]. Assim, fazemos uma junção do diretório da técnica com o nome do arquivo. Dessa forma, podemos recuperar a imagem de um servidor local que subimos usando `Python` no diretório base das pastas das técnicas. É importante ressaltar que para conseguirmos visualizarmos as imagens no navegador foi necessário ter as imagens com a extensão PNG.

6.2 Resultados das Técnicas

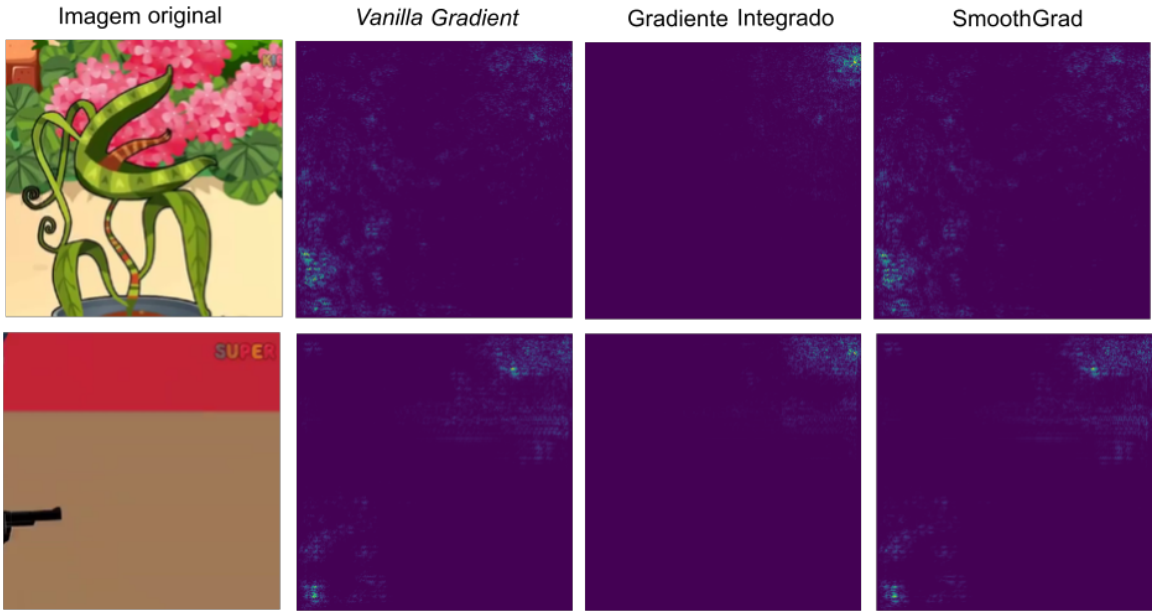
Para facilitar a visualização dividiremos os resultados em três categorias: técnicas baseadas principalmente em gradiente (Gradiente *Vanilla* ou *Vanilla Gradient* [52], Gradiente Integrado [57] e SmoothGrad [53]), o LIME [41] e técnicas de mapa de ativação de classe (GradCAM [49], GradCAM++ [13] e ScoreCAM [63]), isso nos ajudará a discutir os resultados.

Cabe relembrar que o modelo com a NASNet obteve uma acurácia de 92.6% (conjunto de teste) e com a MobileNetv2 obteve uma acurácia de 95.0% (conjunto de treinamento/validação[4]).

Para a avaliação das técnicas, vamos utilizar as mesmas imagens. Os dois *frames* apresentados são de vídeos do tipo *Elsagate*. Ressaltamos que na imagem que tem uma planta aparentemente não tem nenhum conteúdo sensível. No entanto, no decorrer do vídeo, a planta carnívora acaba comendo alguns personagens de um famoso desenho infantil.



(a) NASNet



(b) MobileNetv2

Figura 6.2: Resultados das técnicas de gradiente (*Vanilla Gradient*, Gradiente Integrado e SmoothGrad) para as redes (a) NASNet e (b) MobileNetv2.

6.2.1 Técnicas de Gradiente

Na Figura 6.2, mostramos os resultados dos métodos *Vanilla Gradient*, Gradiente Integrado e SmoothGrad, dos modelos NASNet (Figura 6.2a) e MobileNetv2 (Figura 6.2b). Optamos por não fazer a sobreposição dos mapas de saliência com as imagens uma vez que regiões escuras podem nunca ser ressaltados. Na figura temos dois *frames* de vídeos diferentes, ambos anotados como *Elsagate*, e nos dois casos temos logos no canto superior direito das imagens, referente ao canal de onde foram tirados os vídeos. A diferença entre os *frames* é que temos no primeiro *frame* bastante informação e cores, principalmente ao fundo, enquanto no segundo *frame* temos um fundo de cor única predominante e apenas um objeto em cena, algo que aparenta ser uma arma de fogo.

Analisando os resultados, para a NASNet, conseguimos ver uma concentração de *pixels* destacado exatamente nessa região superior à direita, quando aplicamos as técnicas de explicabilidade. No caso do Gradiente Integrado temos alguma informação destacada no meio e borda à esquerda da primeira imagem, enquanto no segundo exemplo não há outra região de destaque, nem mesmo onde temos o objeto. O SmoothGrad continua destacando esse canto, mas conseguimos observar que ele captura alguns pontos ao longo da imagem. No caso da imagem com a planta, já no segundo exemplo, vemos uma grande atenção para o lado que temos a arma de fogo, no entanto, não é concentrado na região onde temos o objeto, como era de se esperar. Já no caso da MobileNetv2, vemos que a região onde temos as letras são destacadas nas duas imagens, em todas as técnicas. Nesse caso, não temos outras regiões de destaque no Gradiente Integrado, enquanto no *Vanilla Gradient* e SmoothGrad temos uma concentração grande de *pixels* em destaque do lado esquerdo da primeira imagem. Na segunda imagem, essa região de destaque fica no canto inferior esquerdo, logo abaixo do objeto presente na imagem. Apesar dos resultados — não interpretáveis — das técnicas de explicabilidade, os dois modelos, tanto a NASNet quanto a MobileNetv2, acertaram a predição de ambos os vídeos.

Como podemos observar, essas técnicas são difíceis de avaliar por conta dos pontos esparsos. Mas, conseguimos ver que a concentração desses pontos parece não estar em regiões importantes, ou pelo menos, em regiões que esperávamos, como no centro da imagem, onde temos a região da planta na primeira linha, e na região da arma de fogo, no caso da segunda imagem. Na Seção 6.2.4, discutiremos mais resultados das técnicas em outras imagens, tanto classificadas como *Elsagate*, como classificadas como conteúdo não sensível.

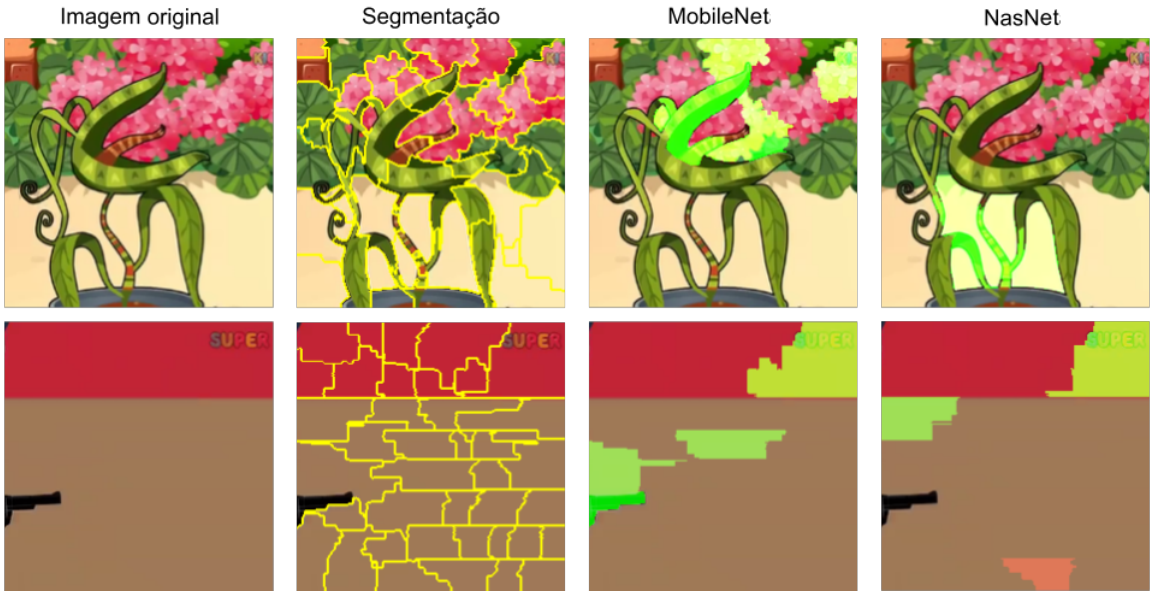
6.2.2 LIME

Para entender o resultado do LIME, é importante destacar que a técnica ressalta regiões em tons de verde e vermelho/laranja. As regiões em verde indicam regiões que influenciaram de forma positiva na classificação daquela imagem na classe correta, enquanto as regiões em vermelho/laranja influenciaram de forma negativa na classificação, ou seja, aquela região da imagem influencia o modelo a classificar a imagem na classe incorreta.

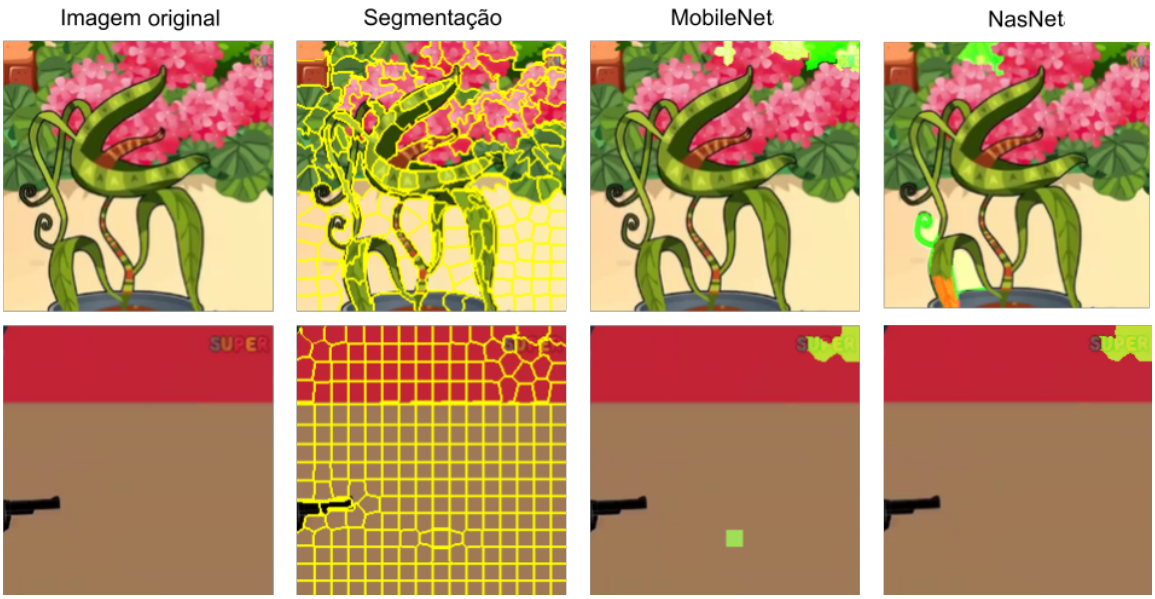
[3] `https://www.typescriptlang.org`

[4] O modelo não foi avaliado no conjunto de teste no trabalho de Ishikawa *et al.* [21].

Na Figura 6.3a, podemos observar tanto a divisão dos *superpixels* como as regiões que contribuem positivamente para a classificação daquela imagem na classe, destacadas em verde nas imagens, e regiões que contribuem de forma negativa para a classificação naquela classe, regiões laranjas e vermelhas. Como podemos notar nas imagens, o método LIME destaca regiões diferentes do que as técnicas de gradiente e diversas vezes destacam regiões de fundo como regiões importantes de forma positiva para a classificação.



(a) *quickshift*



(b) *slic*

Figura 6.3: Resultados do LIME para os dois modelos, MobileNetv2 e NASNet, utilizando a técnica de segmentação (a) *quickshift* e (b) *slic*.

Além disso, tanto para a NASNet quanto para a MobileNetv2 (na Figura 6.3a) ressaltam a região onde temos as letras presentes na imagem, mas apenas a MobileNetv2 ressalta essa região na primeira imagem, a que tem a planta. No caso da segunda ima-

gem, temos uma área que é destacada em laranja, que seria uma região que contribui negativamente para a classificação como *Elsagate*.

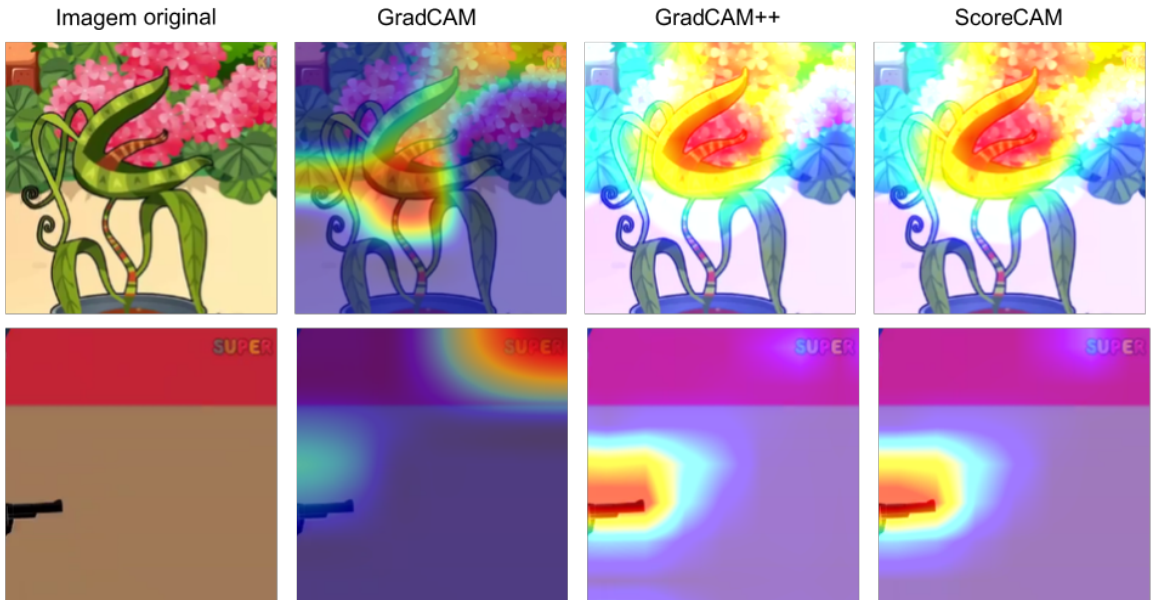
Na Figura 6.3b, temos o resultado de ambos os modelos utilizando o método *slic* de segmentação. Ainda é possível ver que a MobileNetv2 ressalta sempre as regiões com logos, e no caso da segunda imagem, o objeto antes destacado perde a importância com esse novo método de segmentação, ressaltando uma região de fundo que não tem informações aparentes. No caso da NASNet, continuamos vendo que ela ressalta regiões em laranja, mas nesse caso, esse destaque ocorre na imagem da planta, enquanto na segunda imagem temos apenas a região do logo ressaltada novamente.

6.2.3 Técnicas de CAM

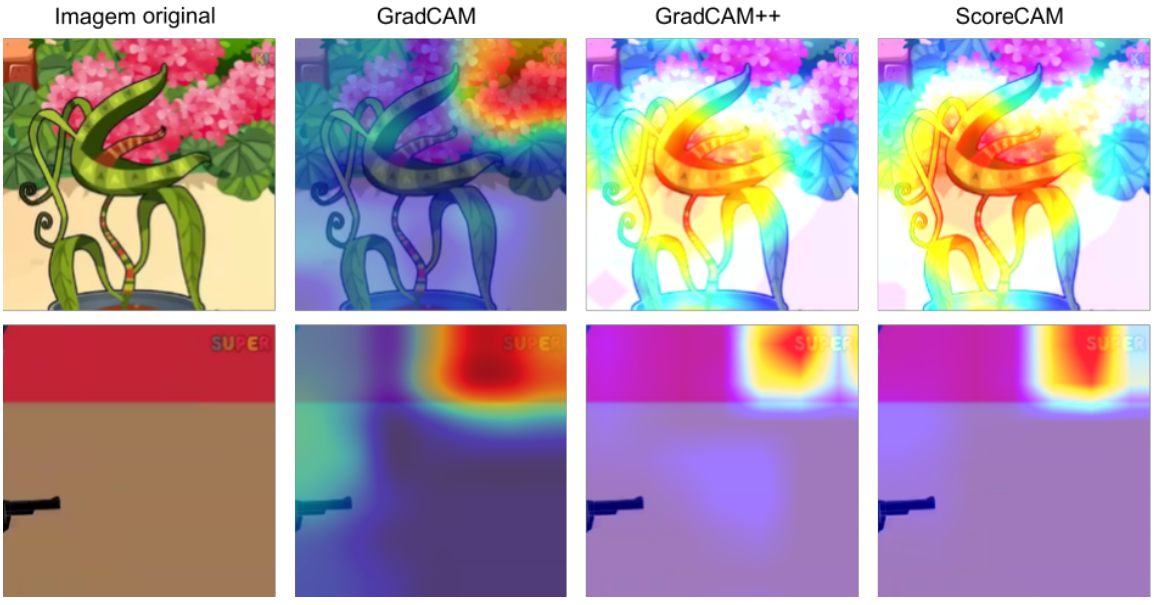
Para as técnicas de CAM, temos o GradCAM, GradCAM++ e ScoreCAM. Essas técnicas tendem a concordar quando as regiões importantes, como podemos observar nas Figuras 6.4a e 6.4b. As técnicas também tendem a focar principalmente em olhos de personagens quando temos ao menos um personagem no *frame*, como podemos ver nas figuras apresentadas na discussão da próxima seção. Mas, curiosamente, mesmo com personagens, os métodos tendem a destacar as letras coloridas que aparecem nos cantos superiores, aquelas que estão ligadas ao canal em que o vídeo foi disponibilizado. Esse resultado pode indicar um possível viés dos modelos/dados, já que imagens com textos na parte de baixo, como legendas, ou textos no meio da imagem, como créditos, não parecem ter o foco da rede.

Considerando as redes analisadas, na Figura 6.4a, podemos observar o resultado das três técnicas quando aplicadas na NASNet. O GradCAM sempre ressalta a região do logo do canal presente nas duas imagens, além de ressaltar outra região da imagem. No caso da segunda imagem, podemos ver que temos um ponto de ativação próximo ao objeto, mas que não possui a mesma importância que a região das letras. Quanto aos outros métodos, tanto o GradCAM++ como o ScoreCAM ressaltam os objetos presentes na imagem e o logo do canto superior esquerdo de ambas a imagem. Porém, no segundo exemplo temos exatamente o oposto do que acontece com o GradCAM, temos uma ativação mais forte em cima do objeto e uma ativação quase inexistente em cima do logo.

Na Figura 6.4b, temos o resultado das técnicas para a MobileNetv2. Nesse caso conseguimos observar que o GradCAM ressalta com maior intensidade a região do logo de ambas as imagens, até há algumas ativações mais fracas espalhadas pelas as imagens, mas o grande foco da rede nesse caso seria no logo. Para os outros dois métodos, GradCAM++ e ScoreCAM, na primeira imagem, onde vemos a planta, temos uma região no centro da imagem que possui grande ativação. Essa região também coincide com a parte da imagem onde temos maior concentração de informação da planta. No caso da imagem com a arma, vemos que a ativação principal está sendo na região do logo, até temos outras regiões com algum destaque, mas esse destaque é quase nulo.



(a) NASNet



(b) MobileNetv2

Figura 6.4: Resultados das técnicas de mapa de ativação de classe para as redes (a) NASNet e (b) MobileNetv2.

6.2.4 Discussão

Não surpreendentemente, as técnicas analisadas não parecem ter um bom resultado no contexto de classificação de *Elsagate*. Podemos ver que muitas vezes os métodos focam em regiões inesperadas da imagem, por vezes se concentrando em cantos superiores onde temos letras, ou em pixels aparentemente aleatórios da imagem. As técnicas aqui classificadas como técnicas de gradiente podem possuir uma visualização mais complexa, o que torna seus resultados difíceis de entender. E podemos ver que em alguns casos essas técnicas também ressaltam esses escritos superiores.

Um ponto a ser discutido sobre esse comportamento é se esse canto superior com letras seria um viés dos dados, o que parece pouco provável já que aparece tanto em imagens classificadas como *Elsagate* ou não sensível. Outra hipótese que surge é se pode ser considerado um possível viés do modelo, que está prestando atenção durante o processo de classificação em regiões que não são importantes, ou ao menos nós humanos não consideramos como regiões importantes que deveriam ter atenção. Apesar de levantarmos esses pontos, se o modelo não está focando nas regiões que esperamos, não significaria que o modelo está incorreto, mas que apenas não está de acordo com a nossa expectativa. Mas, claramente, não é o caso do logo. Focar no logo é um indicativo que tem algo errado nos modelos.

Essa última observação também nos leva a questionar se nenhum método de explicabilidade funciona bem no nosso contexto ou se apenas não nos mostram o que queremos ver como regiões importantes. Essa é uma questão que surge quando estamos vendo o resultado de métodos de explicabilidade, principalmente em contextos subjetivos como é o caso do contexto analisado.

No caso do LIME, por exemplo, sem a segmentação da imagem lado a lado, é difícil de observar se ele está destacando um *superpixel* ou mais de um. Quando possuímos essas regiões destacadas próximas, é importante saber se são dois *superpixels* diferentes ou se pode ser uma região apenas por conta da segmentação. Essa informação pode facilitar nossa interpretação dos resultados. Ao decorrer desta dissertação, percebemos que o LIME sempre ressalta regiões de fundo ou regiões que não esperamos que tenha tanta importância na classificação daquela determinada imagem. Esse comportamento nos leva novamente ao questionamento se esse método não funciona ou se ele não nos mostra o que eu gostaríamos de ver.

Para fins de comparação, mostramos lado a lado os resultados de todos os métodos, *Vanilla Gradient*, Gradiente Integrado, SmoothGrad, LIME (utilizando o método de segmentação padrão, *quickshift*), GradCAM, GradCAM++ e o ScoreCAM, para a rede NASNet (Figuras 6.5 e 6.7, anotadas como *Elsagate* e conteúdo não sensível, respectivamente) e para a rede MobileNetv2 (Figuras 6.6 e 6.8, anotadas como *Elsagate* e conteúdo não sensível, respectivamente). Como podemos observar para diferentes imagens obtivemos os mesmos comportamentos.

Na Figura 6.5, apresentamos cinco *frames* de diferentes vídeos anotados como *Elsagate*. Desses vídeos, temos apenas a segunda imagem, que apresenta o *pac-man* e o cachorro, que foi classificada errada como conteúdo não sensível pela rede NASNet. Todos os outros quatro *frames* presentes na figura foram classificados de forma correta pelo modelo. Outro

aspecto interessante nos *frames* é que na última coluna, temos uma imagem que possui o texto no canto superior direito, referente ao canal que se foi tirado o vídeo. Como já observado nas seções anteriores, as técnicas de gradiente possuem a interpretação dos resultados dos métodos de explicabilidade mais complicada, uma vez que não temos como apontar ao certo onde estão os pontos destacados nos mapas de saliência na imagem original. Apesar disso, podemos observar que no método *Vanilla Gradient* temos mais regiões destacadas, que ficaram na cor amarela devido à conversão da imagem para PNG e a aglomeração dos pontos destacados por quase toda a imagem no caso do primeiro, segundo e quarto *frame*. No caso do LIME, o método parece ter um bom desempenho na segunda imagem, onde temos poucos objetos presentes na imagem, enquanto nos outros *frames* até temos regiões destacadas em cima dos objetos, mas temos bastante regiões destacadas ao fundo que não parecem ser influentes na classificação. Ainda nessa técnica, podemos observar que no quinto *frame*, temos uma região de destaque no canto superior direito, exatamente onde temos as letras coloridas, que são quase imperceptíveis devido ao fundo da imagem. Já nas técnicas de CAM, as regiões de destaque tendem a se concentrar próximo a faces e olhos, além de na última imagem termos o destaque também no canto superior direito, nas letras.

No caso da MobileNetv2, podemos observar as técnicas de explicabilidade na Figura 6.6. Dentre os cinco vídeos, o modelo só conseguiu classificar corretamente os representados pelos *frames* da terceira e quinta coluna. Os métodos de gradiente continuam tendo uma interpretação não clara, apesar de não termos mais aquela quantidade de ruído presentes no resultado do *Vanilla Gradient*. No caso do LIME, continuamos vendo que esse método destaca regiões de fundo, que aparentemente não parecem conter alguma informação relevante para a classificação do vídeo. Para o método GradCAM, podemos ver que temos regiões maiores destacadas em vermelho, que seriam regiões de maior importância, o que poderia indicar que o modelo está "perdido". Nos últimos dois métodos de CAM, continuamos observando o comportamento de destacar regiões próximas de faces e olhos, e na última imagem ainda temos o destaque na região das letras presentes na imagem.

Nas Figuras 6.7 e 6.8, temos os resultados das técnicas de explicabilidade para a NASNet e MobileNetv2, respectivamente, para cinco *frames* de diferentes vídeos anotados como *não sensível*. Um aspecto importante é que ambos os modelos acertaram a classificação dos cinco vídeos. Podemos observar que as técnicas de CAM continuam focando em regiões de rosto e olhos. Na quarta imagem, temos muitas informações e cores presentes na imagem, e continuamos percebendo que as técnicas acabam se perdendo. Quanto ao LIME, percebemos que ele destaca alguns objetos presentes na última imagem, e nas três primeiras percebemos que ele ressalta bastante o fundo da imagem. Um comportamento que surgiu neste método na segunda imagem foi o surgimento de áreas destacadas em vermelho, que seriam áreas que influenciam de forma negativa para a classificação da classe alvo, no caso como conteúdo não sensível. Infelizmente, inspecionando a imagem, não conseguimos elaborar justificativas para esse destaque. Quanto as técnicas de gradiente continuamos vendo os mesmos comportamentos presentes nas outras imagens, sendo que o *vanilla gradient* para a NASNet continua destacando muitas áreas, quase a imagem inteira. E temos concentração de *pixels* destacados em diversas

regiões, nas outras técnicas, inclusive para o outro modelo, que aparentemente não contém informações importantes para a classe alvo. Para as técnicas de CAM, observamos o contraste entre os resultados da NASNet (Figura 6.7), mais concentrados, e MobileNetv2 (Figura 6.8), mais difusos.

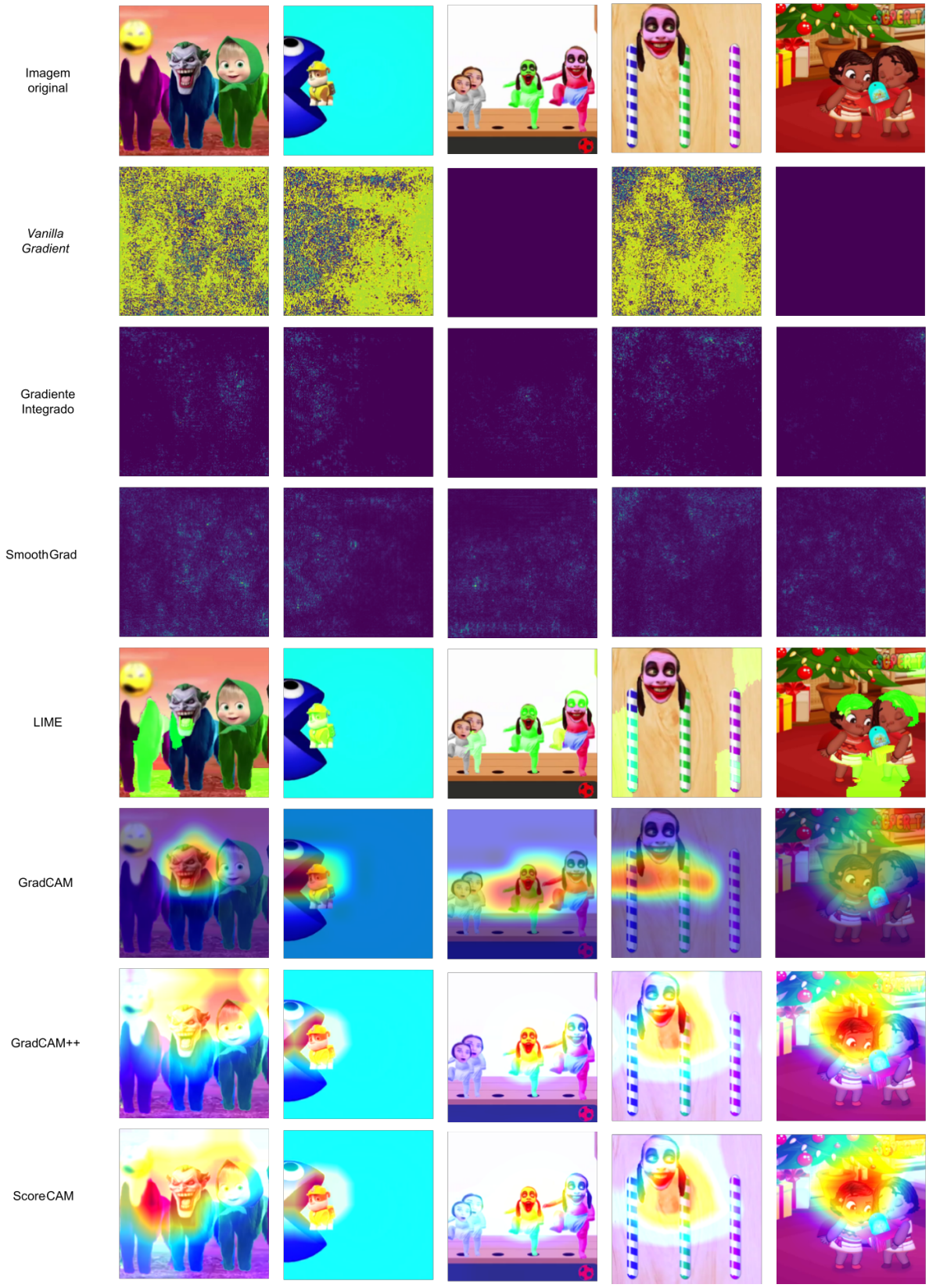


Figura 6.5: Resultados das técnicas de explicabilidade para a rede NASNet, para imagens classificadas como *Elsagate*.

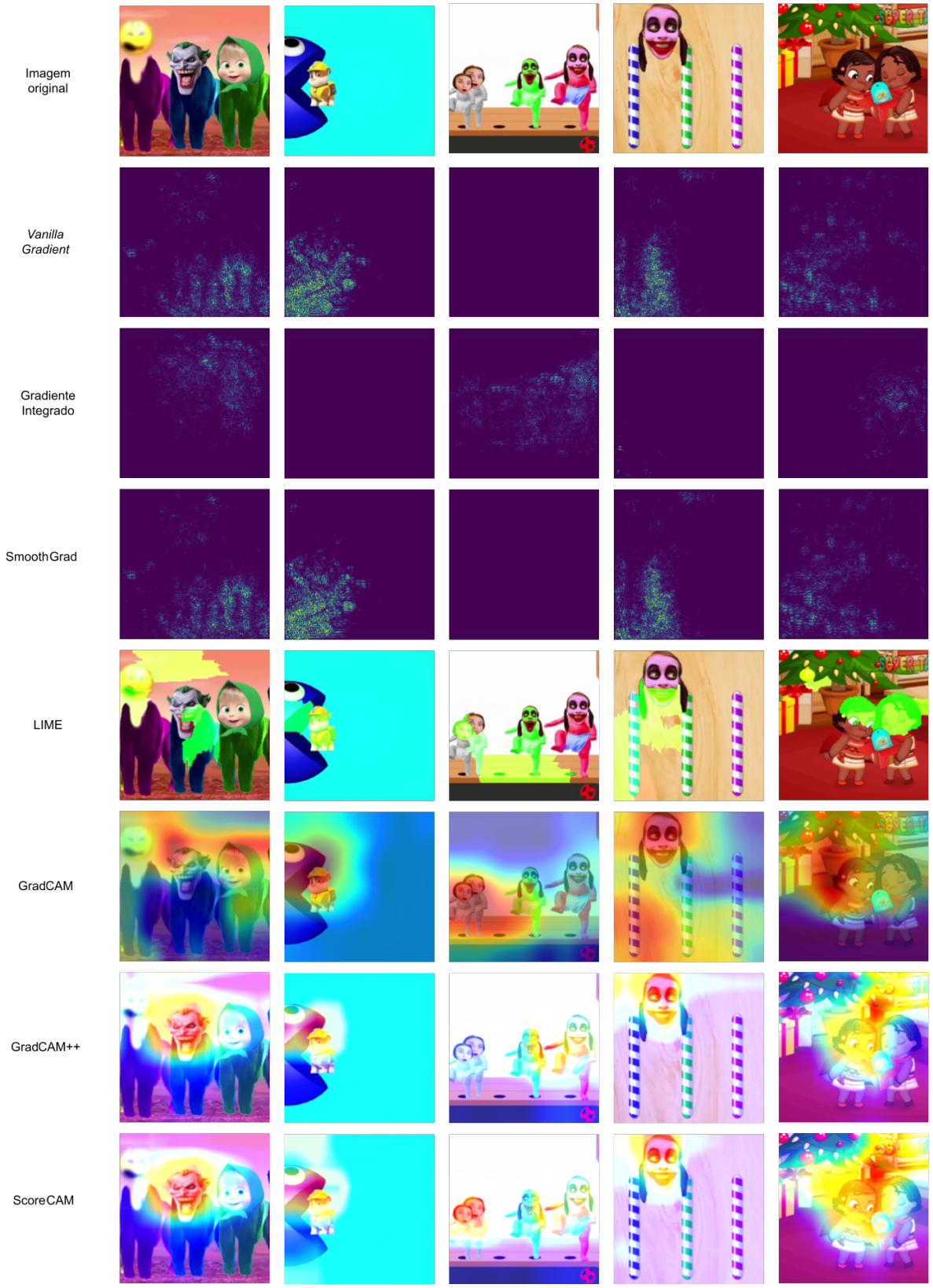


Figura 6.6: Resultados das técnicas de explicabilidade para a rede MobileNetv2, para imagens classificadas como *Elsagate*.

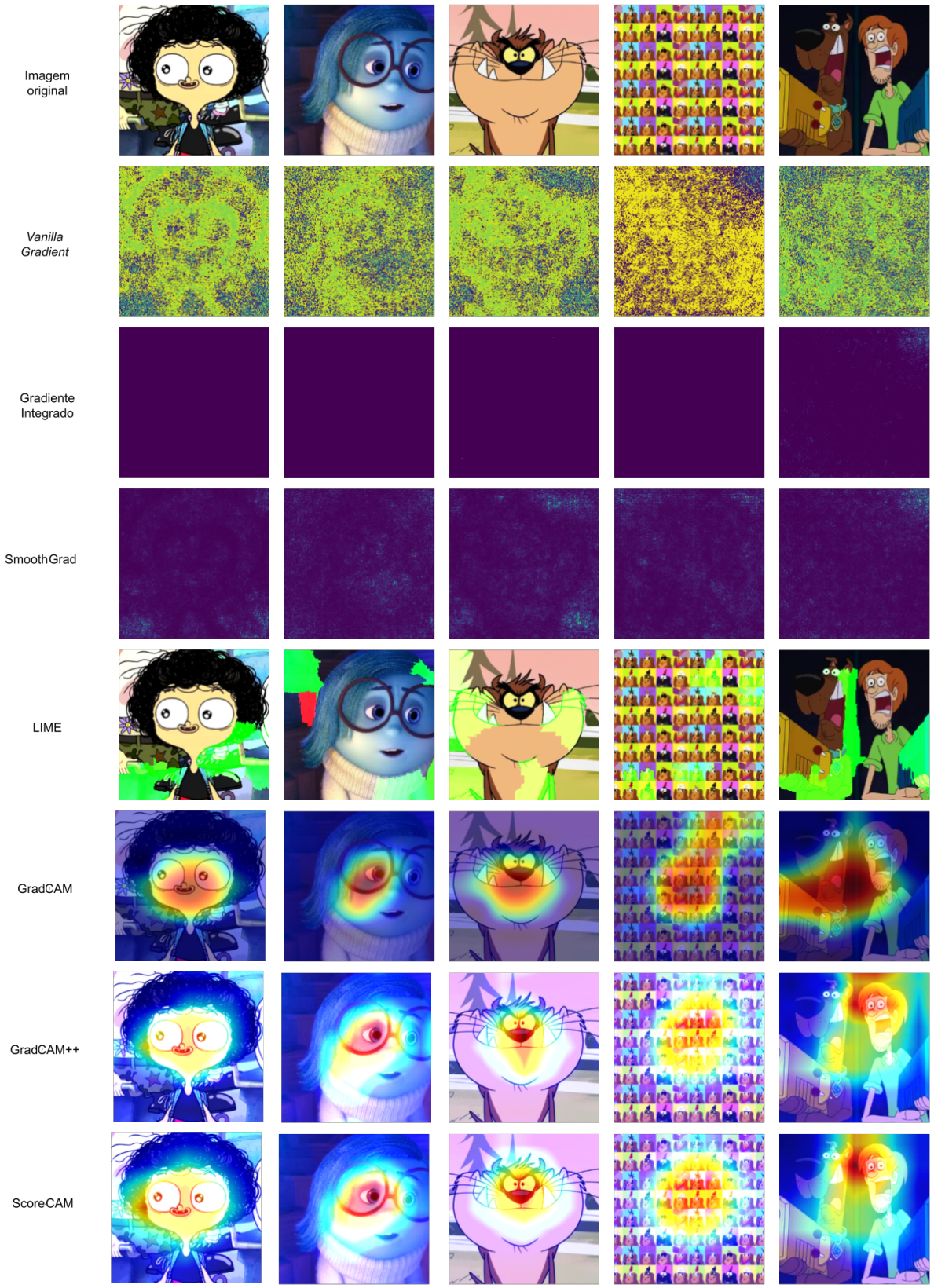


Figura 6.7: Resultados das técnicas de explicabilidade para a rede NASNet, para imagens classificadas como *não sensível*.

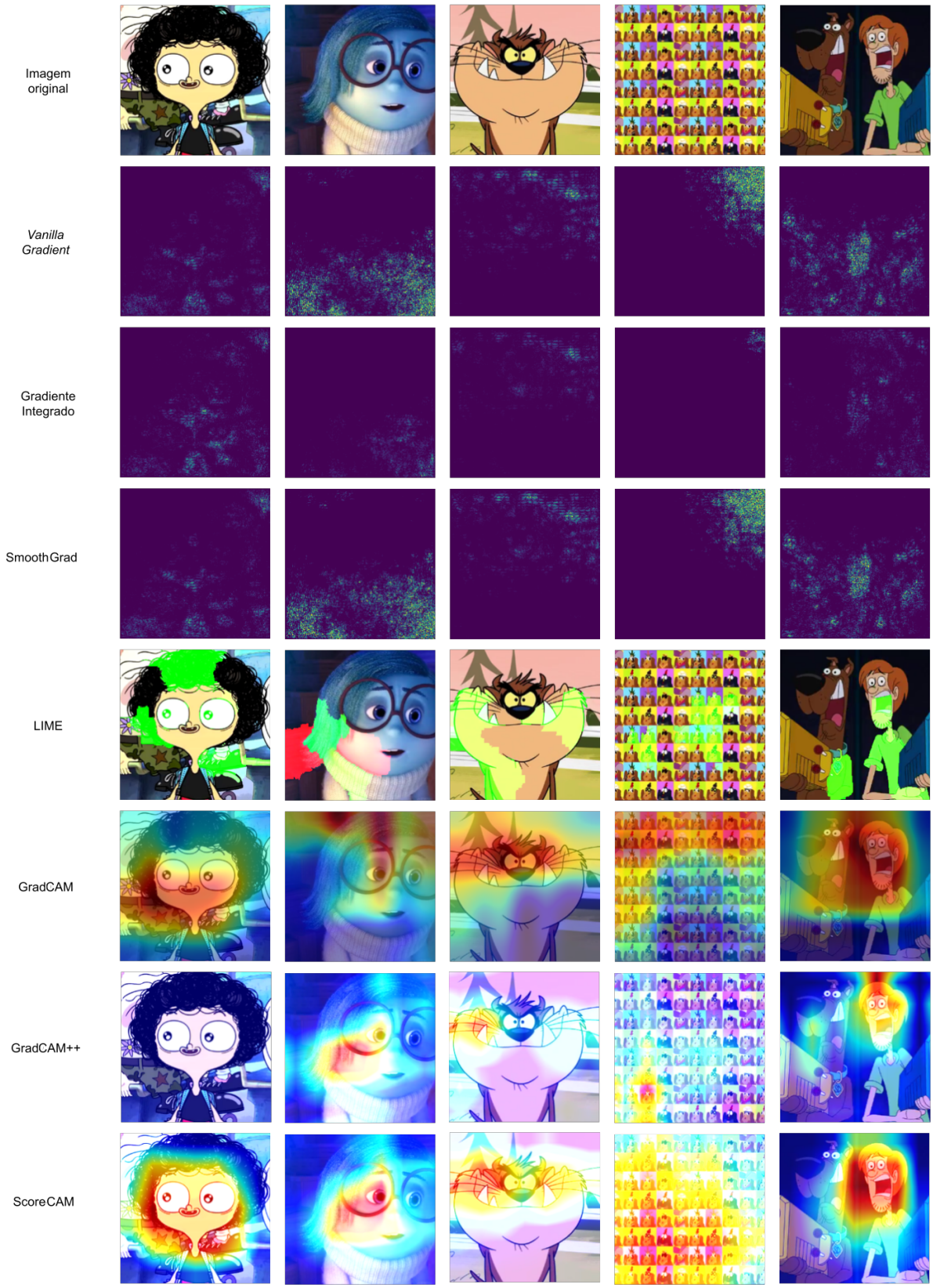


Figura 6.8: Resultados das técnicas de explicabilidade para a rede MobileNetv2, para imagens classificadas como *não sensível*.

Para analisarmos de forma mais precisa os resultados as técnicas, apresentamos nas Figuras 6.9 e 6.10 os resultados estratificados pelos acertos (verdadeiros positivos e verdadeiros negativos) e erros (falsos positivos e falsos negativos). Para as duas figuras, para simplificar as análises, apresentamos os resultados para três técnicas (SmoothGrad, LIME e ScoreCAM) e para o modelo com a melhor taxa de acerto, a NASNet.

Na Figura 6.9a, podemos observar os *frames* retirados de vídeos classificados corretamente como *Elsagate*. Mas, como podemos observar na segunda e terceira figura, o SmoothGrad parece destacar informações na região superior da imagem, enquanto o LIME destaca regiões de fundo, onde a princípio não tem objetos importantes para a classificação, e o ScoreCAM mantém o foco na região no centro da imagem, ainda sem informação aparente para a classificação. Na primeira imagem também conseguimos ver que o LIME e o ScoreCAM destacam rostos (diferentes) na imagem, enquanto o SmoothGrad parece destacar grande parte da imagem.

Na Figura 6.9b, temos *frames* de vídeos classificados de forma incorreta como *Elsagate*. Aqui, podemos observar que temos uma variação do desenho *Turma da Mônica*, onde possuímos traços diferentes do comum (ou seja, traços mais simples, com personagens com corpos menores e cabeças maiores e arredondadas). Podemos observar o mesmo comportamento da Figura 6.9a, o LIME tende a destacar regiões de fundo nas imagem, e na última imagem até destaca o corpo da personagem, mas ainda tem uma atenção no fundo. O ScoreCAM possui o destaque na região central das imagens, que no primeiro e segundo caso são regiões de rosto ou olhos, enquanto na última imagem a região destacada fica acima do rosto da personagem. Por fim, o SmoothGrad acaba tendo várias regiões em destaque, mas no primeiro caso temos a região da cabeça do personagem em destaque, na segunda imagem parece se concentrar no fundo e não necessariamente no personagem na cena, e no caso da última imagem a grande concentração de *pixels* destacados parece estar em cima da única personagem em cena, mais especificamente em cima do rosto da mesma.

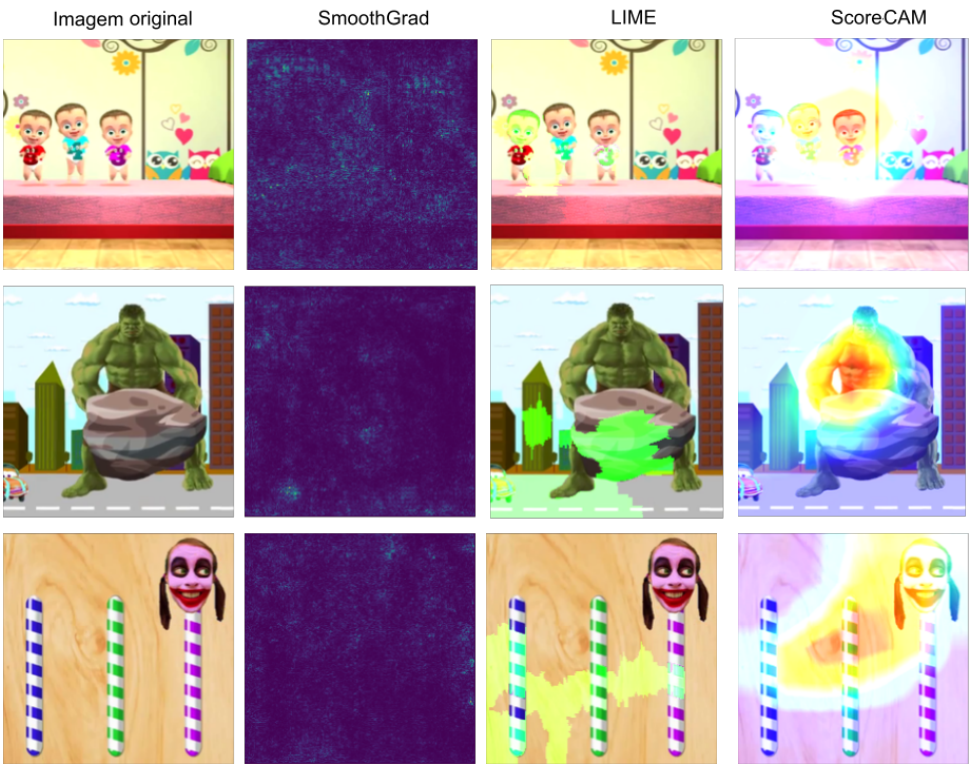
Na Figura 6.10a, podemos observar *frames* de vídeos classificados corretamente como não sensíveis. Nesse caso, também vídeo da *Turma da Mônica*, mas classificado corretamente. Podemos observar que a técnica de gradiente nesses casos destaca bastante informação quando comparada aos exemplos anteriores e que esses destaques parecem coincidir com a posição dos personagens na cena. No caso do LIME, podemos ver que apenas no primeiro caso temos regiões em vermelho, que representa regiões que contribuíram negativamente para a atribuição do rótulo de não sensível. Apesar de serem regiões que não possuem informações relevantes para nós, nos outros casos temos regiões de fundo destacadas e parte dos personagens também. Na segunda imagem ainda temos uma região de um arbusto que é apresentada em laranja, indicando uma baixa contribuição negativa para a classificação daquela imagem. Por último, no caso do ScoreCAM, podemos observar regiões de rostos destacados na segunda e terceira imagem, enquanto na primeira imagem temos que a região destacada como negativa pelo LIME é a região em destaque nessa técnica.

Na Figura 6.10b, temos *frames* de vídeos que foram classificados erroneamente como não sensíveis. Aqui, podemos observar os mesmos comportamentos, vemos que nos três exemplos o LIME destaca regiões de fundo, mas na segunda e terceira imagem, temos um destaque nas regiões que parecem ter alguma informação importante, o rosto dos

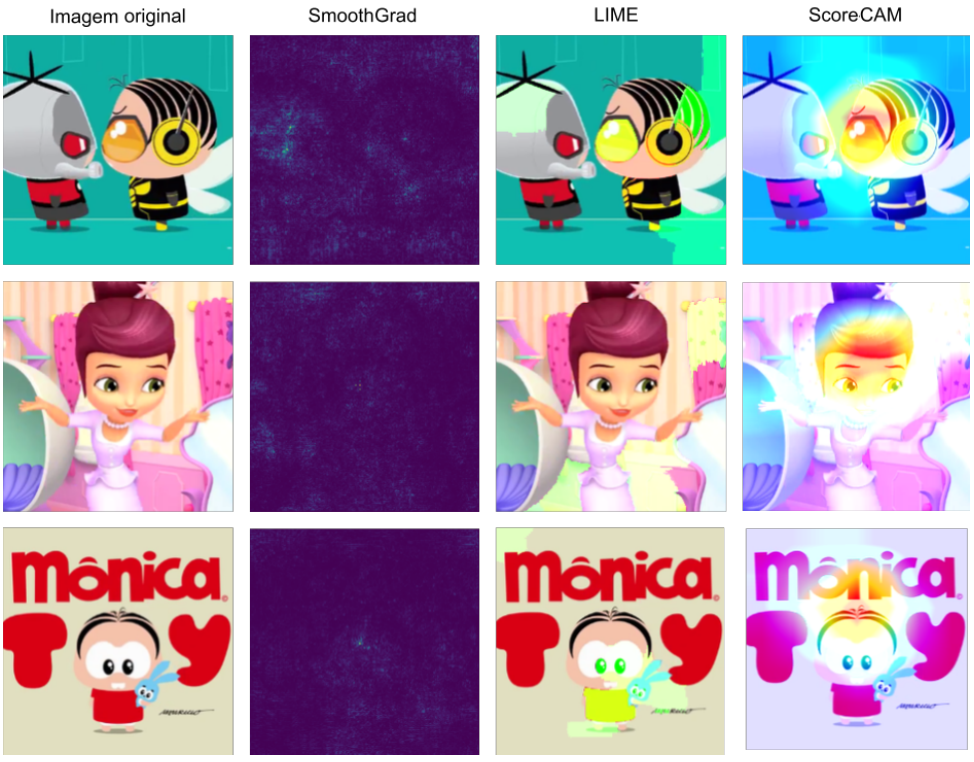
personagens. O ScoreCAM continua destacando regiões centrais da imagem. Por fim, o SmoothGrad destaca muitas regiões da imagem o que dificulta a interpretação desses resultados.

Em resumo, como podemos observar nas imagens, não há uma diferença clara no resultado das técnicas de explicabilidade para imagens de vídeos classificados corretamente e incorretamente. Também, é possível observar que não há um consenso entre as técnicas, regiões importantes que aparecem em uma técnica não são necessariamente destacadas pela outra técnica. Além disso, observarmos que os vídeos classificados corretamente e incorretamente às vezes têm o mesmo traço ou apenas um rosto diferente, como é o caso do último exemplo da imagem 6.9a e 6.10b.

Por fim, dentre todas as técnicas analisadas, o SHAP [29] foi a única que não conseguimos rodar nos modelos. Apesar de ser uma técnica extremamente difundida, enfrentamos diversos problemas para executá-la.



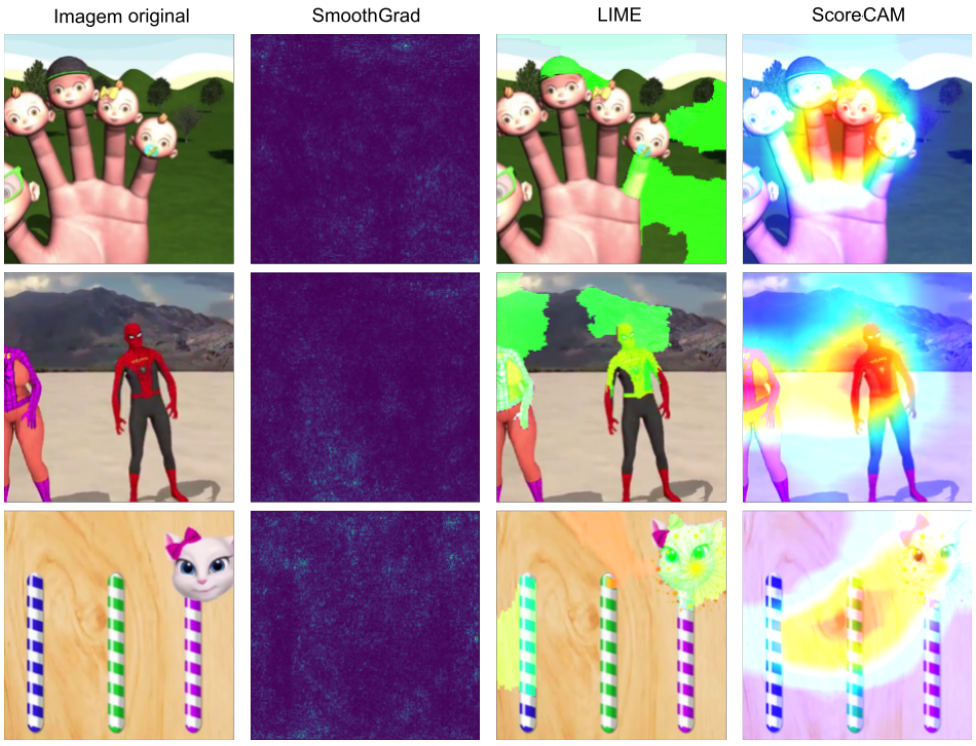
(a) Verdadeiro Positivo



(b) Falso Positivo

Figura 6.9: Quadros de exemplo de vídeos classificadas como *Elsagate* para a NASNet, sendo (a) exemplos de verdadeiro positivo e (b) exemplos de falso positivo.

(a) Verdadeiro Negativo



(b) Falso Negativo

Figura 6.10: Quadros de exemplo de vídeos classificadas como *não sensível* para a NAS-Net, sendo (a) exemplos de verdadeiro negativo e (b) exemplos de falso negativo.

Capítulo 7

Conclusão

Confiar cegamente em resultados de modelos classificativos com alto desempenho é, hoje, desaconselhável, devido à influência de vieses nos dados e correlações espúrias no treinamento de modelos de Aprendizado de Máquina.

Nesta dissertação, discutimos e analisamos vários artigos na área de Inteligência Artificial Explicável (*Explainable Artificial Intelligence*, XAI). Como resultado desta análise, observamos que ainda não temos concordância na comunidade sobre o que é uma explicação, o que é um modelo explicável, o que é interpretabilidade e alguns outros conceitos importantes. É evidente que as atividades de pesquisas nessa área não têm sido suficientemente importantes para definir formalmente o que é uma explicação e identificar quais são as propriedades desejadas em um método de explicação. Consequentemente, também não há um consenso sobre como classificar essas técnicas que estão emergindo, e por esse motivo propomos uma nova taxonomia que engloba as principais existentes. As descobertas mostram que as pesquisas em XAI se concentram principalmente em métodos de explicabilidade agnósticos a modelo e *post-hoc* que utilizam diversas técnicas diferentes. Talvez, a falta de formalismo seja um dos motivos para existir métodos de explicabilidade tão diversos na literatura.

Na revisão da literatura, observamos que temos diversos artigos propondo diferentes técnicas, que por vezes não parecem levar em conta trabalhos anteriores, seja porque as técnicas anteriores não possuem uma boa generalização ou porque não cumprem os requisitos que aquele autor leva em consideração para uma técnica de explicabilidade com bom desempenho. Além disso, todos os métodos presentes hoje na literatura são focados na classificação de objetos concretos em imagens (por exemplo, cachorros, gatos, pessoas), em sua grande maioria, o que pode explicar porque nenhum dos sete métodos mais conhecidos da literatura tiveram um bom desempenho quando aplicados a modelos de classificação de *Elsagate*, um conteúdo subjetivo. O ponto mais interessante dessa investigação foi poder observar que os modelos parecem focar em regiões específicas da imagem que não contêm a informação principal da imagem, o que poderia indicar um possível viés do modelo.

Como trabalhos futuros, vislumbramos alguns possíveis caminhos:

- Aplicar técnicas baseadas com conceitos para avaliar seu desempenho no contexto proposto. Por exemplo, o método TCAV (*Testing with Concept Activation Vectors*) [24], que usa derivadas semelhantes aos métodos baseados em gradiente para

avaliar a sensibilidade das previsões em relação à direção do conceito para uma camada específica;

- Investigar técnicas que não utilizam o gradiente e propõem novos cálculos para a medida de importância de uma região. Por exemplo, o método EigenCAM [33] leva em conta todas as características espaciais da entrada do modelo;

- Combinar diversas técnicas, preferencialmente com funcionamentos diversos, que se encaixem em diferentes categorias da taxonomia proposta. Por exemplo, seria interessante combinar o ADA-SISE [56] ou o EigenCAM [33], que se encaixam na categoria de análise de *feature*, com os métodos que utilizam Vetores de Ativação de Conceito [24], que poderiam utilizar o cálculo adaptativo da ativação de um conceito de acordo com os novos cálculos de importância de uma região;

- Avaliar os métodos através da técnica de *sanity checks* proposto por Adebayo *et al.* [5] para os métodos mais recentes, para podermos avaliar a invariância dos métodos quanto a modelo e dados.

Por fim, vislumbramos avaliar os possíveis caminhos em diversos tipos de conteúdo sensível, como pornografia [31, 32, 38], violência [4, 36, 37] e abuso sexual infantil [60].

Referências Bibliográficas

[1] What is elsagate?. reddit. Disponível em: `https://www.reddit.com/r/ElsaGate/comments/6o6baf/what_is_elsagate/`. Accesso em: 16-05-2019. 15, 45

[2] ISO/IEC 14496-10:2014. Information technology — coding of audio-visual objects — part 10: Advanced video coding. *Standard*, 9, 2018. 46

[3] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018. 14

[4] Marcos Vinícius Adão Teixeira and Sandra Avila. What should we pay attention to when classifying violent videos? In *Proceedings of the 16th International Conference on Availability, Reliability and Security*, pages 1–10, 2021. 68

[5] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Neural Information Processing Systems*, 2018. 37, 68

[6] Robert Andrews, Joachim Diederich, and Alan B Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based systems*, 8(6):373–389, 1995. 18

[7] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020. 9, 18, 19, 20, 29, 31, 32, 35

[8] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015. 37

[9] Jacob Bien and Robert Tibshirani. Prototype selection for interpretable classification. *The Annals of Applied Statistics*, pages 2403–2424, 2011. 37

[10] Alceu Bissoto, Michel Fornaciali, Eduardo Valle, and Sandra Avila. (De)Constructing bias on skin lesion datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 14

[11] Alceu Bissoto, Eduardo Valle, and Sandra Avila. Debiasing skin lesion datasets and models? not so fast. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 740–741, 2020. 14

[12] Rich Caruana, Hooshang Kangarloo, John David Dionisio, Usha Sinha, and David Johnson. Case-based explanation of non-case-based learning methods. In *Proceedings of the AMIA Symposium*, page 212, 1999. 29

[13] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847, 2018. 9, 20, 25, 26, 27, 37, 42, 43, 44, 50

[14] Arun Das and Paul Rad. Opportunities and challenges in explainable artificial intelligence (xai): A survey. *arXiv preprint arXiv:2006.11371*, 2020. 9, 19, 29, 33, 34

[15] Yinpeng Dong, Hang Su, Jun Zhu, and Bo Zhang. Improving interpretability of deep neural networks with semantic information. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4306–4314, 2017. 36

[16] Feng-Lei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 2021. 9, 14, 29, 32, 33, 36, 37

[17] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018. 9, 14, 18, 19, 29, 30

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 37

[19] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 47

[20] International Data Corporation IDC. Worldwide semiannual cognitive artificial intelligence systems spending guide. `https://www.idc.com/tracker/showproductinfo.jsp?containerId=IDC_P33198`, 2018. Último acesso em: 02/08/2021. 14

[21] Akari Ishikawa, Edson Bollis, and Sandra Avila. Combating the elsagate phenomenon: Deep learning architectures for disturbing cartoons. In *IAPR/IEEE International Workshop on Biometrics and Forensics*, 2019. 9, 10, 15, 45, 46, 47, 52

[22] Hyungsik Jung and Youngrock Oh. Lift-cam: Towards better explanations for class activation mapping. *arXiv preprint arXiv:2102.05228*, 2021. 42, 44

[23] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*, pages 2280–2288, 2016. 14

[24] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 36, 67, 68

[25] Josua Krause, Adam Perer, and Kenney Ng. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 5686–5697, 2016. 36

[26] Xilai Li, Tianfu Wu, Xi Song, and Hamid Krim. Aognets: Deep and-or grammar networks for visual recognition. *arXiv preprint arXiv:1711.05847*, 2017. 36

[27] Zhenqiang Li, Weimin Wang, Zuoyue Li, Yifei Huang, and Yoichi Sato. Towards visually explaining video understanding networks with perturbation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1120–1129, 2021. 41, 44

[28] Zachary C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31—-57, 2018. 9, 19, 20, 29, 30, 31

[29] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Neural Information Processing Systems*, pages 4765–4774. 2017. 42, 64

[30] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 2018. 14

[31] Daniel Moreira, Sandra Avila, Mauricio Perez, Daniel Moraes, Vanessa Testoni, Eduardo Valle, Siome Goldenstein, and Anderson Rocha. Pornography classification: The hidden clues in video space–time. *Forensic Science International*, 268:46–61, 2016. 68

[32] Daniel Moreira, Sandra Avila, Mauricio Perez, Daniel Moraes, Vanessa Testoni, Eduardo Valle, Siome Goldenstein, and Anderson Rocha. Multimodal data fusion for sensitive scene localization. *Information Fusion*, 45:307–323, 2019. 45, 68

[33] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020. 42, 44, 68

[34] Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 42, 44

[35] Daniel Omeiza, Skyler Speakman, Celia Cintas, and Komminist Weldermariam. Smooth Grad-CAM++: An enhanced inference level visualization technique for deep convolutional neural network models. *arXiv preprint arXiv:1908.01224*, 2019. 42, 44

[36] Bruno Peixoto, Bahram Lavi, João Paulo Pereira Martin, Sandra Avila, Zanoni Dias, and Anderson Rocha. Toward subjective violence detection in videos. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8276–8280, 2019. 68

[37] Bruno Malveira Peixoto, Sandra Avila, Zanoni Dias, and Anderson Rocha. Breaking down violence: A deep-learning strategy to model and classify violence in videos. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, pages 1–7, 2018. 68

[38] Mauricio Perez, Sandra Avila, Daniel Moreira, Daniel Moraes, Vanessa Testoni, Eduardo Valle, Siome Goldenstein, and Anderson Rocha. Video pornography detection through deep learning techniques and motion information. *Neurocomputing*, 230:279–293, 2017. 46, 68

[39] Gabriëlle Ras, Marcel van Gerven, and Pim Haselager. Explanation methods in deep learning: Users, values, concerns and challenges. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 19–36, 2018. 9, 29, 31, 36

[40] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Nothing else matters: model-agnostic explanations by identifying prediction invariance. *arXiv preprint arXiv:1611.05817*, 2016. 43

[41] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016. 9, 20, 23, 24, 25, 38, 41, 44, 50

[42] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. 14

[43] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *ITU Journal: ICT Discoveries - Special Issue 1 - The Impact of Artificial Intelligence (AI) on Communication Networks and Services*, 1:1–10, 10 2017. 18

[44] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 47

[45] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Neural Information Processing Systems*, pages 4967–4976, 2017. 36

[46] Sam Sattarzadeh, Mahesh Sudhakar, Anthony Lem, Shervin Mehryar, Konstantinos N Plataniotis, et al. Explaining convolutional neural networks through attribution-based input sampling and block-wise feature aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11639–11647, 2021. 41

[47] Sam Sattarzadeh, Mahesh Sudhakar, Konstantinos N Plataniotis, Jongseong Jang, Yeonjeong Jeong, and Hyunwoo Kim. Integrated grad-cam: Sensitivity-aware visual explanation of deep convolutional networks via integrated gradient-based scoring. *arXiv preprint arXiv:2102.07805*, 2021. 42, 44

[48] Christin Seifert, Aisha Aamir, Aparna Balagopalan, Dhruv Jain, Abhinav Sharma, Sebastian Grottel, and Stefan Gumhold. Visualizations of deep neural networks in computer vision: A survey. In *Transparent Data Mining for Big and Small Data*, pages 123–144. 2017. 18

[49] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision*, pages 618–626, 2017. 9, 20, 25, 26, 37, 43, 44, 50

[50] Xiangwei Shi, Seyran Khademi, Yunqiang Li, and Jan van Gemert. Zoom-cam: Generating fine-grained pixel annotations from image labels. In *25th International Conference on Pattern Recognition (ICPR)*, pages 10289–10296, 2021. 42, 44

[51] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017. 42

[52] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *Computing Research Repository*, abs/1312.6034, 2013. 9, 20, 21, 22, 37, 41, 43, 44, 50

[53] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017. 9, 20, 22, 23, 37, 41, 43, 44, 47, 50

[54] Alexandros Stergiou, Georgios Kapidis, Grigorios Kalliatakis, Christos Chrysoulas, Ronald Poppe, and Remco Veltkamp. Class feature pyramids for video explanation. In *IEEE/CVF International Conference on Computer Vision Workshop*, pages 4255–4264, 2019. 42, 44

[55] Alexandros Stergiou, Georgios Kapidis, Grigorios Kalliatakis, Christos Chrysoulas, Remco Veltkamp, and Ronald Poppe. Saliency tubes: Visual explanations for spatio-temporal convolutions. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1830–1834. IEEE, 2019. 42, 44

[56] Mahesh Sudhakar, Sam Sattarzadeh, Konstantinos N Plataniotis, Jongseong Jang, et al. Ada-sise: Adaptive semantic input sampling for efficient explanation of convolutional neural networks. *arXiv preprint arXiv:2102.07799*, 2021. 41, 44, 68

[57] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, volume 70, pages 3319–3328, 2017. 9, 20, 22, 24, 37, 41, 43, 44, 50

[58] Jayaraman J Thiagarajan, Bhavya Kailkhura, Prasanna Sattigeri, and Karthikeyan Natesan Ramamurthy. Treeview: Peeking into deep neural networks via feature-space partitioning. *arXiv preprint arXiv:1611.07429*, 2016. 36

[59] Eduardo Valle, Sandra de Avila, Antonio da Luz Jr, Fillipe de Souza, Marcelo Coelho, and Arnaldo Araújo. Content-based filtering for video sharing social networks. In *Brazilian Symposium on Information and Computer System Security*, 2012. 14

[60] Paulo Vitorino, Sandra Avila, Mauricio Perez, and Anderson 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. 68

[61] Eric Wallace, Shi Feng, and Jordan Boyd-Graber. Interpreting neural networks with nearest neighbors. *arXiv preprint arXiv:1809.02847*, 2018. 38

[62] Dan Wang, Xinrui Cui, and Z Jane Wang. Chain: Concept-harmonized hierarchical inference interpretation of deep convolutional neural networks. *arXiv preprint arXiv:2002.01660*, 2020. 43, 44

[63] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 24–25, 2020. 9, 20, 26, 27, 28, 37, 42, 43, 44, 50

[64] Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran. Explainable deep learning: A field guide for the uninitiated. *arXiv:2004.14545*, 2020. 20

[65] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015. 37

[66] Matthew Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833, 2014. 22, 37, 38

[67] Haipeng Zeng. Towards better understanding of deep learning with visualization. *The Hong Kong University of Science and Technology*, 2016. 18

[68] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018. 47