UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DAINF - DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

FERNANDA HAUPTMANN DE ALMEIDA

# INTERPRETABILIDADE EM INTELIGÊNCIA ARTIFICIAL: UM ESTUDO DE CASO SOBRE TESTES COM VETORES DE ATIVAÇÃO DE CONCEITO (TCAV)

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2021

FERNANDA HAUPTMANN DE ALMEIDA

# INTERPRETABILIDADE EM INTELIGÊNCIA ARTIFICIAL: UM ESTUDO DE CASO SOBRE TESTES COM VETORES DE ATIVAÇÃO DE CONCEITO (TCAV)

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Gustavo Alberto Gimenez Lugo
DAINF - Departamento Acadêmico de Informática - UTFPR

Coorientadora: Veronica Ferreira Bahr Calazans
DAFCH - Departamento Acadêmico de Filosofia e Ciências Humanas - UTFPR

CURITIBA
2021

**FERNANDA HAUPTMANN DE ALMEIDA**

# INTERPRETABILIDADE EM INTELIGÊNCIA ARTIFICIAL: UM ESTUDO DE CASO SOBRE TESTES COM VETORES DE ATIVAÇÃO DE CONCEITO (TCAV)

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Sistemas de Informação da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 23/Agosto/2021

---

Gustavo Alberto Giménez-Lugo
Doutorado
Universidade Tecnológica Federal do Paraná

---

Myriam Regattieri De Biase da Silva Delgado
Doutorado
Universidade Tecnológica Federal do Paraná

---

Mayara Cristina Pereira Yamanoe
Doutorado
Universidade Tecnológica Federal do Paraná

**CURITIBA**

**2021**

Para meu pai Agnaldo Josimir de Almeida (*in memorian*).

## AGRADECIMENTOS

Agradeço a todos os professores e professoras que educam para a libertação.

Agradeço aos meus orientadores Gustavo e Veronica por me mostrarem a necessidade de recortes sem nunca me podar. Seu respeito, apoio, orientação e amizade foram essenciais para o desenvolvimento desse trabalho e profundamente tocantes na minha formação.

Agradeço a minha família e especialmente ao meu irmão Lucas por terem sonhado a conclusão dessa graduação junto comigo e por terem feito tudo em seu poder pra que eu tivesse as condições de realizar esse sonho.

Agradeço a todos os meus amigos e amigas pelos momentos de descontração e pelo acolhimento nos momentos difíceis. Sou grata especialmente aos meus amigos Mateus Belizario e Gabriel Biesek Regis pelo apoio imensurável durante toda a graduação.

*"E assim continuaram vivendo numa realidade escorregadia, momentaneamente capturada pelas palavras, mas que fugiria sem remédio quando fosse esquecido o valor da letra escrita." (MÁRQUEZ, 2019, p.56)*

# RESUMO

HAUPTMANN DE ALMEIDA, Fernanda. Interpretabilidade em Inteligência Artificial: Um Estudo de Caso sobre Testes com Vetores de Ativação de Conceito (TCAV). 2021. 86 f. Trabalho de Conclusão de Curso – Curso de Bacharelado em Sistemas de Informação, Universidade Tecnológica Federal do Paraná. Curitiba, 2021.

Interpretabilidade em Inteligência Artificial é a área de pesquisa que busca desenvolver métodos para a compreensão dos estados internos de modelos de aprendizagem de máquina, um desses métodos é a testagem com vetores de ativação de conceito (TCAV) apresentada por Kim et al. (2017). Essa pesquisa investiga a ferramenta a partir da análise do discurso sobre a ferramenta e da verificação empírica da mesma a partir de cenários de teste que nos permitiram verificar algumas fragilidades nos resultados obtidos por essa pesquisa. O propósito desse trabalho é buscar entender de que modo essas fragilidades limitam a construção da relação entre classe e conceito no método TCAV.

**Palavras-chave**: Interpretabilidade; Vetores de Ativação de Conceito; Inteligência Artificial.

## ABSTRACT

HAUPTMANN DE ALMEIDA, Fernanda. Interpretability in Artificial Intelligence: A Case Study on Tests with Concept Activation Vectors (TCAV). 2021. 86 f. Trabalho de Conclusão de Curso – Curso de Bacharelado em Sistemas de Informação, Universidade Tecnológica Federal do Paraná. Curitiba, 2021.

Interpretability in Artificial Intelligence is the research field which aims to develop methods for the comprehension of the internal states of machine learning models, one of these methods is Testing with Concept Activation Vectors (TCAV), presented by Kim et al. (2017). This research investigates the tool TCAV both from its presentation article and through empiric verification of it based on test cases that allowed for this research to verify some fragilities on the results obtained. The goal of this work is to understand how these fragilities limit the relation between class and concept implied by the TCAV method.


**Keywords**: Interpretability; Concept Activation Vectors; Artificial Intelligence.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## SUMÁRIO

# 1 Introdução

Desde a mitologia Grega, em Hefesto e Pigmaleão, incorpora-se a ideia de robôs inteligentes (BUCHANAN, 2005) e há todo tipo de especulações sobre quais os possíveis desdobramentos que os avanços em aprendizagem de máquina podem representar para a sociedade. Uma parcela dessas especulações argumenta que haverá (possivelmente) um momento de virada, onde essas máquinas inteligentes superarão a inteligência humana, atingindo um estado de superinteligência (BOSTROM, 2014, p.26). Ainda que provavelmente longe desse momento de virada, hoje, enquanto produtos da técnica contemporânea, aplicações de aprendizagem de máquina já promovem novos meios de relação entre sujeito e mundo.

Ao ínicio de sua obra, Bostrom (2014) conta uma pequena fábula interminada sobre pardais, a qual essa pesquisa se apropria agora para se introduzir. Na história da fábula, a maioria de um grupo de pardais acredita que precisa roubar um ovo de coruja e criá-la para que vigie seus ninhos por eles, livrando-os desse fardo. Há um pardal chamado Scronfickle que com a companhia de poucos alerta para a falha no plano "[...]não deveríamos nos dedicar primeiro ao aprendizado da arte de domesticar corujas antes de trazer uma ao nosso meio?". Sem fim conhecido, a fábula nos deixa com o questionamento de Scronfickle, à quem Bostrom (2014, p. IV) dedica sua obra e o qual serve como uma boa analogia ao processo de adoção quase que totalmente indiscriminado de aplicações de inteligência artificial nos mais diversos nichos sociais, ao mesmo tempo que não há uma contrapartida na cultura no que diz respeito ao entendimento do funcionamento dessas aplicações. Atualmente, um dos desdobramentos desse tipo de questionamento na ciência da computação é o desenvolvimento de métodos e ferramentas de interpretabilidade em redes neurais. Nesse contexto, entende-se interpretabilidade como "a habilidade de explicar ou apresentar em termos compreensíveis para humanos" (DOSHI-VELEZ; KIM, 2017, p.2) os processos internos de modelos de tomada de decisão.

Uma dessas ferramentas foi desenvolvida por cientistas do Google e apresentada ao mundo por Kim et al. (2017) no artigo "*Interpretability Beyond Feature Attribution: Testing With Concept Activation Vector* (TCAV)"[1]. De acordo com os autores, a ferramenta TCAV é capaz de prover uma "interpretação do estado interno de uma rede neural" a partir de conceitos humanos (como cor, textura, espécies, etc) verificando a partir de derivadas direcionais o grau de sensibilidade da predição de uma classe para um determinado conceito pré-selecionado por quem utiliza a ferramenta.

Por exemplo, a predição da classe "ursos polares" é relevante ao conceito "animais brancos peludos"? Descobrimos a partir de experimentos apresentados no desenvolvimento

[1] traduzido livremente como "Interpretabilidade Além da Atribuição de Características: Testando com Vetores de Ativação de Conceito".

dessa pesquisa que sim e que não. E essa divergência de resultados para essa classe e conceito é apenas um dos produtos da investigação empregada por essa pesquisa acerca das fragilidades na construção da relação entre classe e conceito realizada pela ferramenta TCAV. Entende-se aqui enquanto fragilidades conceituações que podem surgir a partir de uma conveniência visual e que não necessariamente refletem a realidade. Verifica-se ainda que se trata de uma interpretação especialmente frágil do estado interno do modelo por que não é possível hoje verificar o real estado interno de um modelo em seu processo de tomada de decisão.

## 1.1 Justificativa

Esse trabalho reconhece a necessidade sócio-científica de se caminhar na direção da compreensão dos processos internos desses algoritmos de tomada de decisão a partir de ferramentas de interpretabilidade. No entanto, deixar a definição do que constitui uma boa explicação para esses modelos matemáticos sob responsabilidade de seus desenvolvedores pode facilmente resultar em algo falho (MILLER; HOWE; SONENBERG, 2017, p.1). Desse modo, essa pesquisa verifica o método empregado na ferramenta TCAV, observando seus resultados para além das quantificações produzidas.

## 1.2 Objetivos

O objetivo deste trabalho é refletir, a partir de uma análise empírica da ferramenta TCAV, sobre a seguinte questão: quais os limites conceituais da relação construída para classe e conceito pela ferramenta TCAV? Buscando-se observar não só resultados referentes às operações quantitativas do método, mas também refletir sobre os possíveis desdobramentos sociais ocasionado pelas fragilidades desses resultados.

### 1.2.1 Objetivos Específicos

Para buscar compreender de que modo as limitações na relação construída para classe e conceito pela ferramenta TCAV se apresentam nos resultados impingidos a presente pesquisa se vale dos seguintes objetivos específicos.

a) Verificar a influência que tipos de cenários distintos podem provocar no funcionamento da ferramenta. Ou seja, dado o processo da ferramenta na formulação das relações entre classe e conceito, como determinadas entradas podem impingir vieses nos resultados?;
b) Verificar se discurso sobre a ferramenta condiz com a sua realidade de uso;
c) Discutir possíveis desdobramentos do uso da ferramenta TCAV;

## 1.3 Organização da Obra

Em um primeiro momento, essa pesquisa buscava alçar uma disucussão dos resultados sob à luz da filosofia da tecnologia, desse modo, no capítulo 2 dedicado à fundamentação teórica dessa pesquisa, há uma uma seção dedicada a construção de um pequeno arcabouço teórioco acerca da relação entre humanidade, técnica e cultura. No entanto, o processo de execução dos testes propostos centrou a presente pesquisa frente a um construto computacional que impôs desafios técnicos à realização dos experimentos, de modo que por questão de tempo, essa pesquisa abriu mão dessa investigação extradisciplinar de forma aprofundada, mas ainda se valendo do referencial teórico construído e apresentado na seção 2.4 para ajudar em embasar a discussão da conclusão dessa pesquisa. O capítulo 3 se dedica à apresentação de trabalhos correlatos, ou seja, buscamos na literatura obras que se dedicam à investigação de ferramentas artificilmente inteligentes e seus desdobramentos na sociedade. O capítulo seguinte (4) apresenta uma revisão do artigo de Kim et al. (2017) sobre a ferramenta TCAV. Os capítulos 5 e 6 apresentam respectivamente a metodologia utilizada por essa pesquisa e o desenvolvimento do trabalho proposto. Os testes apresentados no capítulo 6 são resultados de 10 rodadas de treinamento para o modelo Inception 5h, pois a máquina disponível para execução dos testes não dispunha de capacidade de processamento suficiente para mais rodadas de treinamento.

A conclusão dessa pesquisa apresentada no capítulo 7 sumariza os cenários observados nos resultados dos testes realizados e aponta os possíveis desdobramentos do uso a ferramenta TCAV, especialmente no que diz respeito à despreocupação em Kim et al. (2017) para com os dados não observados. Esses dados, por vezes completamente desconhecidos por quem utiliza a ferramenta, ainda assim podem ser relevantes ao processo interno dessa e por vezes podem ser críticos, de modo que o “nosso entendimento falho pode levar a decisões com consequência dramáticas”Hand (2020, p. 305) nas mais diversas aplicações sociais no caso desse tipo de ferramenta passar a ser utilizada comercialmente.

# 2 Aprendizagem de Máquina e Humanidades

Esse capítulo se dedica à apresentação do arcabouço teórico utilizado por esta pesquisa.

## 2.1 Aprendizagem de Máquina

Russell e Norvig (2010, p. 2–4) apresentam a definição de Inteligência Artificial em quatro perspectivas de pesquisa categorizadas da seguinte forma:

- **Pensando Humanamente**: onde se pretende uma modelagem baseada nas ciências cognitivas e que almeja automatizar atividades que envolvem o pensamento humano (tomada de decisão, por exemplo) (Cf. RUSSELL; NORVIG, 2010).
- **Pensando Racionalmente**: nessa categoria, a tradição lógica define inteligência artifical como "o estudo das faculdades mentais a partir do uso de modelos computacionais" (WINSTON, 1992 apud RUSSELL; NORVIG, 2010, p. 2).
- **Agindo Humanamente**: onde segundo RUSSELL; NORVIG o objeto artificialmente inteligente precisaria deter habilidades específicas como:
  - Processamento de linguagem natural;
  - Representação de conhecimento para armazenamento;
  - Habilidade de responder perguntas e chegar a novas conclusões usando as informações armazenadas; e
  - Aprendizagem de máquina para se adaptar a novas circunstâncias, além de detectar e extrapolar padrões.
- **Agindo Racionalmente**: nessa categoria, o autores afirmam que se reconhece Inteligência Artificial quando o agente computacional é capaz de operar de forma autonoma, perceber o ambiente, persistir por um longo período de tempo, se adaptar a mudança e criar e buscar alcançar objetivos (Cf. RUSSELL; NORVIG, 2010).

Desse modo, como é possivel interpretar o termo Inteligência Artificial de mais de uma forma, entendemo-nos nesse estudo como o termo guarda-chuva usado para referir-se a todo tipo de máquinas com algum nível de autonomia em seus processos de tomada de decisão. Nesse contexto, aprendizagem de máquina é um conjunto de técnicas para se produzir inteligência artificial. Essas técnicas surgem da necessidade de resolver problemas que são difíceis de modelar, o que com os algoritmos aprendizes se torna mais alcançável.

Segundo Minsky (1961), como o conceito de aprendizagem pode ser dado de formas diferentes, a forma mais justa de usar o termo "aprendizagem" em computação seria no sentido de que sistemas aprendizes são capazes de utilizar registros obtidos no passado como evidência para a proposições mais genéricas ao longo do tempo. Desse modo, Paluszek e Thomas (2016) definem que a partir de um conjunto de dados de entrada,

sistemas de aprendizagem de máquina buscam realizar predições sobre, ou responder a, dados futuros. Esses dois tipos de problemas são chamados respectivamente de regressão e classificação.

Problemas de classificação contemplam os modelos que pretendem predizer uma classe discreta para um exemplo novo de dado; enquanto modelos para regressão consistem em utilizar dados do passado para fazer uma predição contínua de saída sobre novos dados, os quais o agente em questão ainda não tem conhecimento sobre. Para tanto é preciso que o modelo encontre a função $f(x)$ que melhor mapeie a aproximação entre os dados de entrada e saída. Desse modo, o processo de treinamento do modelo tem o propósito de definir essa função $f(x)$.

O conjunto de dados iniciais disponibilizado ao algoritmo de treinamento pode ser construído de duas formas: histórico (dados obtidos no passado) ou atualizado em tempo real com novos dados. A qualidade desse conjunto é diretamente ligada à qualidade das respostas e predições da ferramenta.

### 2.1.1 Sobre o Conjunto de Dados

O conjunto de dados é parte substancial do estado da arte em pesquisas de reconhecimento de imagens, não só no que diz respeito ao processo de treinamento, mas também como ferramenta que permite comparar e medir a performance de algoritmos (Torralba; Efros, 2011).

Segundo Cortiz (2020, p. 2), "o algoritmo de treinamento é um conjunto de regras que não faz juízo de valor nem apresenta viés de qualquer natureza", no entanto, se o modelo de saída desse algoritmo apresenta "comportamentos enviesados", isso se deve aos dados utilizados no treinamento. Desse modo, o mesmo algoritmo de treinamento alimentado com diferentes conjuntos de dados, resulta em modelos que se comportam de formas diferentes.

Por exemplo, segundo Giest e Samuels (2020, p. 1) um conjunto de dados que falha em representar grupos margilizados resulta em modelos que afetam negativamente suas "oportunidades economicas, mobilidade social e participação democratica". Segundo as autoras, esse cenário se forma porque "grupos marginalizados produzem menos dados". Os motivos para tanto de acordo com Selbst (2017, p. 685) seriam porque as pessoas pertencentes a esse grupo "estão menos envolvidos na economia formal e suas atividades produtoras de dados", "têm acesso desigual e menos fluência nas tecnologias necessárias para o engajamento online", ou ainda são "consumidores menos lucrativos e portanto menos interessantes à observação".

Desse modo, se um modelo é o "conjunto acumulado de relações descobertas" (SELBST, 2017, p.677), ele também imprime em suas aplicações todas as relações não descobertas, um processo que pode resultar em "saídas adversas desproporcionalmente concentradas em grupos historicamente em desvantagem" (SELBST, 2017, p.673). Hand

(2020, p. 6) se refere a esses dados desconhecidos como *"dark data"*.

Em sua obra Hand (2020) produz uma taxonomia a fim de melhor abordar os tipos de dados desconhecidos. Segundo Hand (2020, p. 24), dados desconhecidos são onipresentes, "podendo surgir em qualquer e todo lugar", e é ao não saber que os dados não estão lá que os torna tão "perigosos".

### 2.1.2 Sobre as Formas de Treinamento do Modelo

A partir então desses conjuntos de dados é possível treinar o modelo, ou seja, o arcabouço matemático que promove a aprendizagem da ferramenta. As possibilidades de treinamento para promover o aprendizado da ferramenta acerca dos dados podem ser dos seguintes tipos:

- aprendizagem supervisionada: primeiro treina-se o conjunto a partir de imagens onde as localização desses objetos foi marcada e classificada, e após aprender a partir desses exemplos, o algoritmo é capaz de localizar e classificar um conjunto de dados novo;
- aprendizagem não-supervisionada: quando não há uma determinação humana do que deve ser a saída da operação, de modo que o algoritmo busca aprender propriedades dos dados que sejam relevantes aos seus propósitos de classificação ou regressão (Cf. STENROOS et al., 2017);
- aprendizagem semi-supervisionada, quando parte dos conjuntos de treinamento está categorizada e parte não, geralmente a menor parte está categorizada e é utilizada como vantagem para interpretar a outra parte; (Cf. PALUSZEK; THOMAS, 2016) e

Esses conjuntos de dados quase sempre precisam de algum tipo de pré-processamento. Uma possibilidade é a extração de características relevantes a partir de detectores. Esse modelo de pré-processamento permite a geração de um conjunto de entrada inicial para o modelo de aprendizagem de máquina, "não sendo prático ou possível usar conjuntos de dados inteiros diretamente" (STENROOS et al., 2017). Esses detectores inicialmente eram manualmente elaborados, no entanto nem sempre se sabe quais são as características importantes as quais eles devem buscar, desse modo, os modelos de aprendizagem também têm buscado métodos para ensinar a esses detectores o que buscar (Cf. STENROOS et al., 2017).

Modelos de aprendizagem de máquina precisam encontrar um ajuste, comumente chamado de *fitting*, que permita ao modelo não deixar de capturar dados importantes, ao mesmo tempo que não seja rigoroso (*overfitted*) a ponto de incluir detalhes irrelevantes e ruído. Esse processo de ajuste é o que permite à máquina aprendiz ser capaz de deduzir de forma mais genérica acerca dos dados não conhecidos.

Para calcular a performance de um algoritmo aprendiz é possível utilizar uma função de perda, a qual busca quantificar a discrepância entre as previsões obtidas pelo modelo e as saídas que eram esperadas, no caso de treinamento por apredizagem

supervisionada. “O objetivo da fase de treinamento é minimizar essa perda” (STENROOS et al., 2017), permitindo que o algoritmo aprendiz produza conjunto de dados de saída mais robusto.

## 2.2 Redes Neurais

Dentro do arcabouço de técnicas disponíveis a partir de aprendizagem de máquina, um subconjunto bastante popular refere-se às técnicas de redes neurais (RN). Assis et al. (2016, p. 3) definme redes neurais como "uma técnica de Análise Multivariada de Dados, que utiliza o processamento de informações em paralelo, visando simular o comportamento de uma rede neural biológica".

De acordo com Anderson (1995, p. 8), neurocomputação é "uma tentativa de se construir computadores que sejam um pouco parecidos com o cérebro e que talvez possam fazer um pouco do que o cérebro faz". Segundo o autor, almeja-se a partir deste campo de pesquisa que uma rede neural seja boa como um cérebro é em atividades como "reconhecer padrões, controle motor, percepção, [...], intuição e bons palpites", mas que cérebros também são "lentos, imprecisos, generalizam erroneamente, são preconceituosos e em geral incapazes de explicar suas próprias ações" e que redes neurais também se parecem muito com cérebros no que diz respeito a essas "propriedades menos desejáveis" (ANDERSON, 1995).

Kubat e Haykin (1999, p. 24) explicam que para atingir uma boa performance, uma rede neural emprega uma série de interconexões a partir de células computeiras chamadas "neurônios" (ou "unidades de processamento") e, segundo os autores, as redes neurais se parecem com cérebros humanos em dois aspectos:

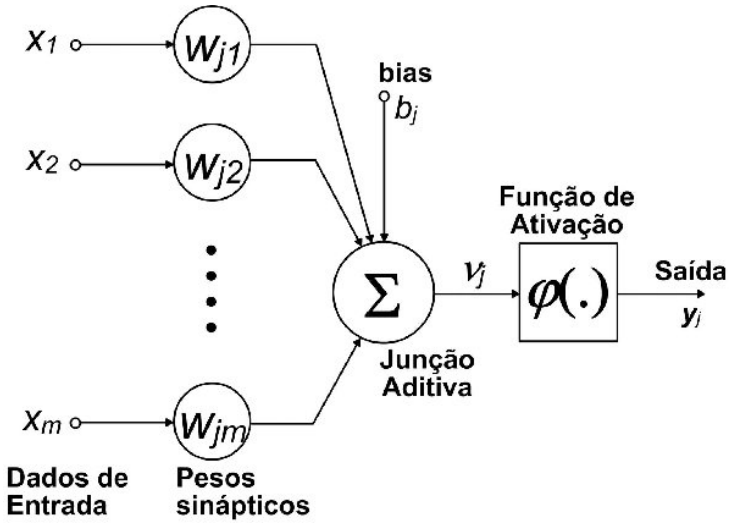
1. "Conhecimento é adquirido pela rede a partir do seu ambiente através do processo de aprendizagem." (KUBAT; HAYKIN, 1999)
2. "A força da conexões intraneural, conhecida como peso sináptico, é utilizada para guardar o conhecimento obtido." (KUBAT; HAYKIN, 1999)

O processo de aprendizagem de redes neurais se dá a partir de um algoritmo de aprendizagem, cuja abordagem tradicional é modificar os pesos sinápticos da rede de modo a obter um desejado objetivo inserido em seu design (KUBAT; HAYKIN, 1999) . Esses processo pode ser descrito através do modelo apresentado na figura 1.

Na figura 1, um conjunto de sinapses (conexões) existe caracterizada cada uma delas por um peso (força) que pode ser negativo ou positivo. A equação se dá, de modo que em uma sinapse $j$, o sinal $x_j$ ligado ao nerônio $k$ é mutiplicada pelo peso sináptico $w_{kj}$. O somatório computa os valores de entrada em uma combinação linear, enquanto a função de ativação garante um cárater não linear ao resultado.

A função de ativação tem relação direta com a efetividade da rede uma vez que transforma um sinal de entrada em um sinal de saída a partir de relações não-lineares. De acordo com Sharma, Sharma e Athaiya (2020, p. 310 - 311), se funções de ativação não fossem usadas em uma rede neural os sinais de saída seriam simplesmente uma função de grau um, dada sua baixa complexidade não são capazes de aprender e reconhecer mapeamentos de dados complexos. Desse modo, a ausência da função de ativação em

Figura 1 – Modelo Não-linear de Neurônio.




Fonte: Adaptado de Haykin et al. (2009)

> "O neurônio $k$ recebe $m$ entradas $x_j$. O neurônio tem também $m$ parâmetros de pesos $w_{kj}$ . Os parâmetros geralmente incluem um viés (*bias*) $b_j$ setado com o valor inicial de 1. As entradas e pesos são linearmente combinados e somados. A soma então é alimentada em uma função de ativação que produz a saída $y_j$ do neurônio: $y_k = \varphi(s_k) = \varphi \sum_{j=0}^{m} w_{kj} x_j$. O neurônio é treinado cuidadosamente, selecionando os pesos para produzir a saída desejada para cada entrada." (STENROOS et al., 2017, p. 13-14)

uma rede neural a torna incapaz de realizar tarefas complicadas, como modelar conjuntos de "dados complicados como imagens, videos, audio, discurso, texto, etc" (SHARMA; SHARMA; ATHAIYA, 2020).

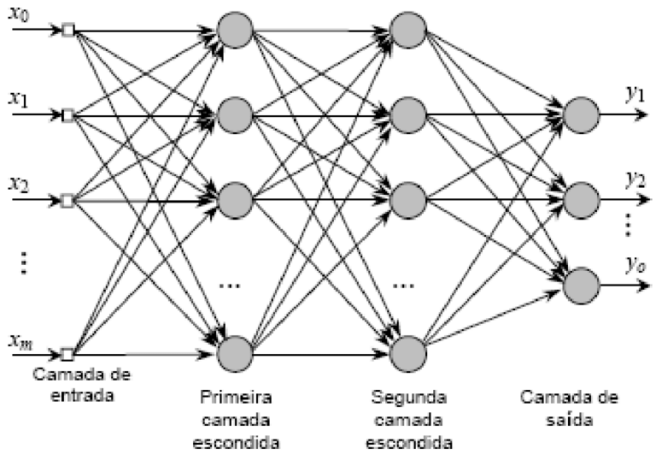
A função de ativação pode ser de diversos tipos (Sigmóide, Tanh, ReLU. etc), desde que seja diferenciável (para algoritmos baseados em gradiente). Sharma, Sharma e Athaiya (2020, p. 311) explicam que essa é "uma característica porque permite que possamos calcular erros e perdas com respeito aos pesos através do algoritimo conhecido como *back-propagation*, o qual será apresentado a seguir. Uma função de ativação diferenciável também possibilita otimizar pesos usando técnicas como descida de gradiente (ou qualquer outra técnica de otimização) para reduzir erros.

Esses neurônios funcionam como diversos processadores paralelos conectados que produzem cada um uma sequência de ativações com valores reais. "Alguns neurônios podem ser ativados a partir de sensores em contato com o meio, outros a partir de conexões ponderadas com neurônios predecessores" e por fim "alguns podem influenciar o meio através do acionamento de ações" (SCHMIDHUBER, 2015).

A combinações desses neurônios agrupados em camadas foma uma rede neural multicamada. Tipicamente, se tratam de três tipos de camadas: camada de entrada, uma

ou mais camadas escondidas e camada de saída. Esse tipo de estrutura também pode ser chamada de rede neural profunda, com o comportamento explicado na figura 2.

Figura 2 – Arquitetura de uma rede perceptron multicamadas com duas camadas escondidas.




Fonte: Adaptado de Haykin et al. (2009)

> "neurônios recebem como entrada as ativações dos neurônios da camada anterior e executam algum tipo simples de computação, como uma soma ponderada, por exemplo, seguida de uma ativação não-linear (porém em algumas redes a saída pode ser linear). Os neurônios da rede juntamente implementam um mapeamento não-linear complexo da entrada para a saída. Esse mapeamento é apreendido a partir dos dados através da adaptação dos pesos de cada neurônio usando uma técnica chamada *back-propagation*." (MONTAVON; SAMEK; MÜLLER, 2018, p. 2)

O algoritmo que apresenta uma solução para a definição iterativa dos pesos na rede neural de forma é chamado *back-propagation*. O algoritmo utiliza o valor da função de perda para verificar a diferença entre o resultado esperado por quem treina a rede, e a saída da rede, então a partir da noção dessa discrepância reformula os pesos das camadas internas começando a partir da camada mais próxima da saída e propagando-se pra trás, em direção a camada de entrada (WERBOS et al., 1990).

Os estados internos de redes neurais profundas são comumente referidos como "caixa-preta", pois seus métodos de decisão e organização internas não são claros, apesar do grande sucesso desse tipo de modelo (SHWARTZ-ZIV; TISHBY, 2017). Esse tipo de tecnologia tem aumentado em grau de complexidade interna e em número de aplicações, ao mesmo tempo que a dificuldade de desvendar seu processos internos também passa a ser cada vez mais influente no cotidiano. Nas palavras de Castelvecchi (2016, p. 2), "aprendizagem de máquina está se tornando ubíqua na pesquisa e na indústria. Mas para cientistas confiarem em, primeiro precisam entender o que as máquinas estão fazendo". Para tratar desse assunto, essa pesquisa reserva a sessão 2.3.

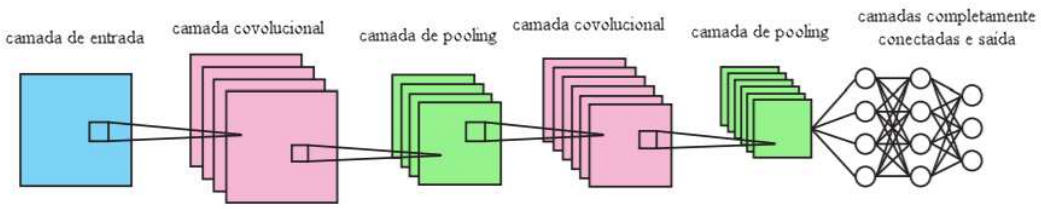
### 2.2.1 Redes Neurais Convolucionais

A essa pesquisa torna-se relevante uma subseção dedicada a rever a técnica utilizada em redes neurais profundas, a qual se dedica ao campo de visão computacional, chamadas Rede Neurais Convolucionais (CNN).

De forma geral, redes neurais convolucionais buscam extrair alguma informação específica de imagens ou vídeos digitais. A ideia de convolução foi inspirada no campo de recepção do córtex visual animal, o qual é formado por detectores que se sobrepõem. Em operações de convolução, essa função biológica é aproximada a partir da sobreposição de filtros (matrizes) que produzem efeitos visuais distintos (STENROOS et al., 2017).

A arquitetura de uma CNN consiste em mútiplas camadas com funções distintas, geralmente alternando entre camadas de convolução (formadas pela combinação de filtros convolucionais) e camadas de *pooling*, usadas para reduzir o volume de dados (que cresce no processo de aplicação dos filtros). Vargas, Paes e Vasconcelos (2016, p. 2) explicam que a "combinação das entradas de um neurônio, utilizando os pesos respectivos de cada uma de suas conexões, produz uma saída passada para a camada seguinte".

Figura 3 – Exemplo de Rede Neural Convolucional.



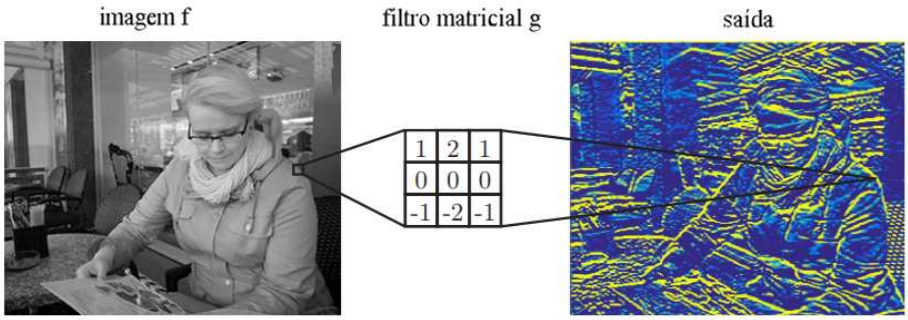

Fonte: Adaptado de Stenroos et al. (2017)

Cada camada de convolução forma-se a partir de diversos neurônios, cada qual "responsável por aplicar um filtro em um pedaço específico da imagem" (VARGAS; PAES; VASCONCELOS, 2016, p. 2), desse modo, diferentemente do que acontece em uma rede de perceptrons clássicas, em uma CNN cada neurônio se conecta a um conjunto de pixels da camada anterior que representa um apenas subconjunto da entrada. A cada uma dessa conexões se atribui um filtro (que também pode ser chamado de kernel ou máscara) e que podem ser interpretado como uma matriz .

O tamanho do filtro, ou seja, o tamanho dessa matriz define o tamanho da vizinhança que cada neurônio da camada irá processar, além disso, a variável importante para a camada convolucional é o "passo", valor que representa quantos pixels serão pulados entre cada janela, o que nos diz qual será o tamanho da camada seguinte desta rede (VARGAS; PAES; VASCONCELOS, 2016, p. 2).

Assim como em redes neurais clássicas, aplica-se também às CNNs a função de ativação geralmente logo após a convolução. De acordo com Vargas, Paes e Vasconcelos (2016, p.2), a " não linearidade das camadas intermediárias permite que as aplicações

Figura 4 – Detecção de bordas horizontais em uma imagen utilizando filtro convolucional.




Fonte: Adaptado de Stenroos et al. (2017)
Matriz representando o filtro de uma convolução de imagens no domínio espacial.

sucessivas dessas distorcões tornem as categorias de saída linearmente separaveis". Para modelos de classificação, os autores explicam que "após o conjunto das camadas de convolucão e pooling ao menos uma camada totalmente conectada" é acrescentada.

## 2.3 Interpretabilidade em Aprendizagem de Máquina

Interpretar importa porque no "contexto social, a razão para uma decisão geralmente importa", explica Lipton (2018, p. 2), por exemplo, "causar uma morte intencionalmente (assassinato) *vs.* não intencionalmente (homício culpososo) são crimes distintos" ou então, "a decisão de contratar alguém baseado (direta ou indiretamente) em uma característica protegida [por lei] como raça tem uma consequência legal" e ainda assim, os modelos preditivos de hoje em dia não são capazes de explicar o racional de seus processos internos (LIPTON, 2018).

A fé cega em algoritmos com estados internos opacos, ou *black box algorithms*, como são conhecidos, traz à tona questões críticas à segurança de suas aplicações, como em veículos autônomos ou diagnósticos médicos, por exemplo (GUIDOTTI et al., 2018 apud MARTIN, 2019, p.16). As questões éticas e de legalidade também motivam a pesquisa em interpretabilidade, como na Europa, por exemplo, onde a partir da regulamentação do direito à proteção de dados (*General Data Protection Regulation*, ou GDPR), os e as cidadãs europeias têm o "direito à explicação", o que significa que uma pessoa pode pedir informações relevantes que explique o processo de tomada de decisão algorítmica sobre essa mesma pessoa (GOODMAN; FLAXMAN, 2017, p.1).

Mas o que interpretar significa? Um das definições possíveis para o termo[1] é "dar certo sentido a; entender; julgar", e no contexto de interpretabilidade em redes neurais artificiais, como Doshi-Velez e Kim (2017, p. 3) explicam, isso significa "dar um sentido humano à". Interpretabiliade em RN difere de explicabilidade, porque não se preocupa

[1] Definições de *Oxford Languages*.

com o porquê das decisões do modelo, mas tenta compreender qual o esquema interno daquele modelo.

E quando se faz necessário interpretar? De acordo com as autoras, "em aprendizagem de máquina, distinguem-se dois tipos de incerteza", a primeira, quantificável e formalmente explicável, enquanto a segunda é a incerteza pela "incompletude", o que produz um viés intangível no conjunto de saída do modelo. Para exemplificar como a incompletude se dá, Doshi-Velez e Kim (2017, p. 3) apresentam alguns cenários, e que "na presença de uma incompletude, explicações são uma das formas de garantir que o efeito do buraco na formalização do problema seja visível para nós". Alguns exemplos trazidos pelas autoras estão citados abaixo.

- "Entendimento científico", quando o objetivo humano é obter conhecimento, a melhor forma de agir é "pedir por uma explicação que possamos converter em conhecimento" (DOSHI-VELEZ; KIM, 2017, p. 3);
- "Segurança", quando somos incapazes de prever todos os cenários negativos possíveis (DOSHI-VELEZ; KIM, 2017, p. 3);
- "Ética", pois pode haver aspectos enviesados que não foram considerados a priori (DOSHI-VELEZ; KIM, 2017, p. 3);

Miller, Howe e Sonenberg (2017) argumentam sobre a relevância de se considerar os métodos e modelos em interpretabilidade que reduzam o risco de "deixar os internos coordenando o hospício" em referência ao livro de Cooper et al. (2004). A obra em questão faz referência sobre os riscos de se colocar programadores para executar o papel de designers, e que serve de analogia segundo Miller, Howe e Sonenberg (2017), para os riscos enfrentados pelas ferramentas de interpretabilidade, uma vez que redes neurais se aplicam às mais diversas áreas e nem sempre (ou quase nunca) a pessoa responsável pelo desenvolvimento modelo é especialista no domínio da aplicação para validar se realmente a explicação obtida confere com a realidade.

Nesse contexto, entende-se interpretabilidade como a habilidade de explicar ou apresentar em termos compreensíveis para humanos os processos internos referentes à tomada de decisões nesse tipo de algoritmo. No entanto, essa definição não é um consenso e também não existe um paradigma claro sobre como avaliá-la (DOSHI-VELEZ; KIM, 2017). Para Miller, Howe e Sonenberg (2017), a melhor opção seria construir esses modelos de explicabilidade para IA a partir de um "entendimento de compreensão da explicação", e essa explicação deve ser avaliada a partir de dados obtidos em estudos acerca do comportamento humano.

Essa afirmação de Miller, Howe e Sonenberg (2017) embasa-se na a ideia de explicação casual de Hilton (1990 apud MILLER; HOWE; SONENBERG, 2017), a qual envolve sempre alguém explicando algo para um segundo sujeito. Nesse sentido, Miller propõe um experimento para demonstrar o fato de que as ciências sociais e os estudos em comportamento humano não estão tendo o devido impacto nessas ferramentas de

interpretabilidade. O autor ressalta que se trata de um modo simplista de olhar para os artigos, e não nega que existem ótimas produções sobre o assunto, mas ainda assim os resultados evidenciam o fato de que a maioria dos trabalhos limita a influência ou até ignora completamente essas áreas do conhecimento nas suas produções.

Para além da limitação ou completa alienação das ciências sociais e humanas nas produções de interpretabilidade, um segundo desafio na área se dá também no que Breiman et al. (2001 apud HALL; GILL, 2019, p. 11) chamou de "a multiplicidade de bons modelos", o que significa que "para o mesmo conjunto de variáveis de entrada e predições esperadas, algoritmos complexos de aprendizagem de máquina podem produzir múltiplos modelos precisos, com arquiteturas internas similares, mas não idênticas". Desse modo, Hall e Gill (2019, p. 11) sugerem que diversas técnicas em interpretabilidade sejam usadas para derivar explicações para um único modelo, buscando-se encontrar resultados consistentes a partir de cada técnica.

Quando falamos em técnicas de interpretabilidade em apredizagem de máquina é importante definir alguns conceitos, o primeiro deles no que diz respeito ao modo de interpretação dos dados: local *Vs.* global. De acordo com Hall e Gill (2019, p. 19), "interpretabilidade global nos ajuda a entender a relação modelada entre as entradas e a predição esperada", no entanto esse geralmente é um processo de alta aproximação. Por outro lado, interpretação local promove o entendimento de pequenas regiões do modelo que relaciona entradas e saídas, podendo oferecer uma explicação mais precisa. Também é importante identificar dois tipos de técnicas em interpretabilidade: aquelas desenhadas para um "tipo ou classe específico de algoritmo", chamadas *model-specifc*; e aquelas as quais podem ser aplicadas em diferentes modelos de algoritmos de aprendizagem de máquina, conhecidas pelo termo *model-agnostic*.

### 2.3.1 Algumas Técnicas de lnterpretabilidade

Hall e Gill (2019, p. 23) dividem as técnicas de interpretabilidade em dois grupos: projeções bidimensionais (projeções 2D) e grafos de correlação. Os autores explicam que ainda que seja técnicamente possível plotar muitas dimensões a fim de se buscar compreender a relação modelada para entradas e saídas, fazendo-o é mais comum que se diminua o entendimento o humano sobre que se observa do que o contrário.

Sobre as técnicas de projeção 2D, os autores explicam que a ideia é representar o conjunto de dados em um espaço de baixa dimensão, mas ainda significativo. Entre as técnicas possíveis para se alcançar essa visualização estão os gráficos de dependência parcial, ou *partial dependence plots* (PDP). Essa técnica de projeção 2D consiste em um gráfico que demonstra a relação funcional entre um pequeno número de variáveis e uma predição.

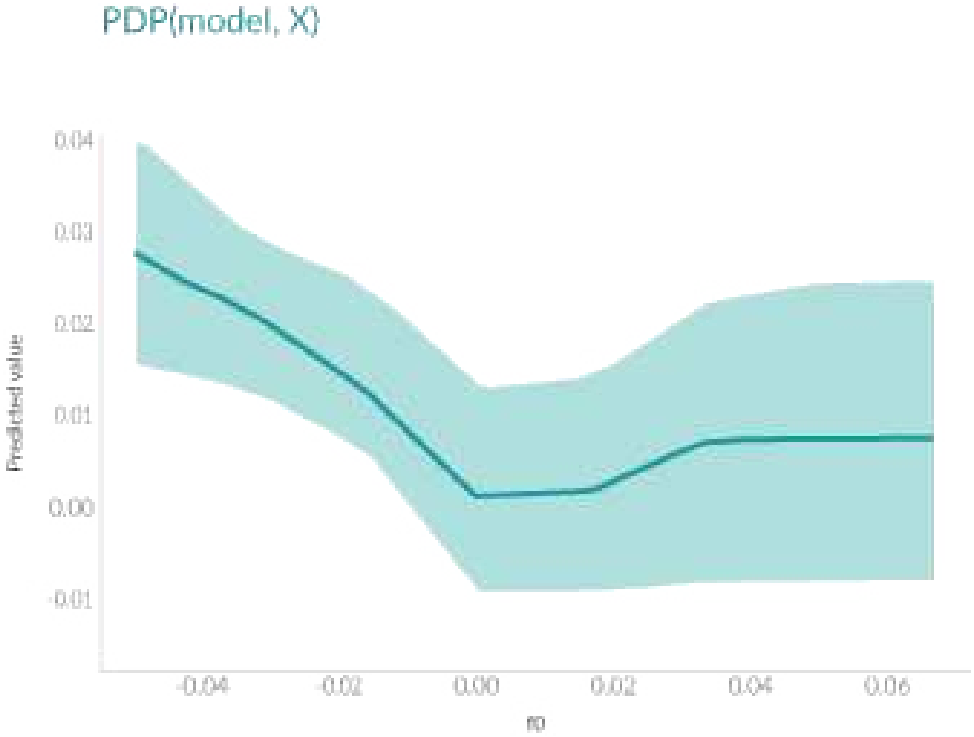
Esse modelo foi apresentado por Friedman (2001, p. 1219) há mais de duas décadas e apresenta um conceito menos abrangente de interpretabilidade. Segundo o

autor, interpretabilidade em ML "envolve ganhar um entendimento daquelas entradas em particular que são mais influentes em contribuir para sua variação [de um modelo]" e então entender a dependência do modelo sobre essas entradas.

Friedman (2001, p. 1219) explica que "observar modelos com argumentos de alta-dimensão é mais difícil", de modo que esse método cumpre melhor seu propósito visualizando a dependencia parcial do modelo sobre um "conjunto selecionado e pequeno de variáveis de entrada". O autor deixa claro que "dificilmente um conjunto desses gráficos pode prover um desenho compreensivo da aproximação realizada pelo modelo, mas ele pode produzir pistas que ajudam" (FRIEDMAN, 2001, p. 1219).

A ideia nessa técnica é de "observar como mudanças nas entradas afetam a previsão de saída" (KRAUSE; PERER; NG, 2016, p. 4). Um exemplo de como esse modelo se comporta pode ser visto na figura 5.

Figura 5 – Gráfico de Dependência Parcial




Fonte: <https://www.twosigma.com/articles/interpretability-methods-in-machine-learning-a-brief-survey/>. Acessado em 10 de março de 2021.

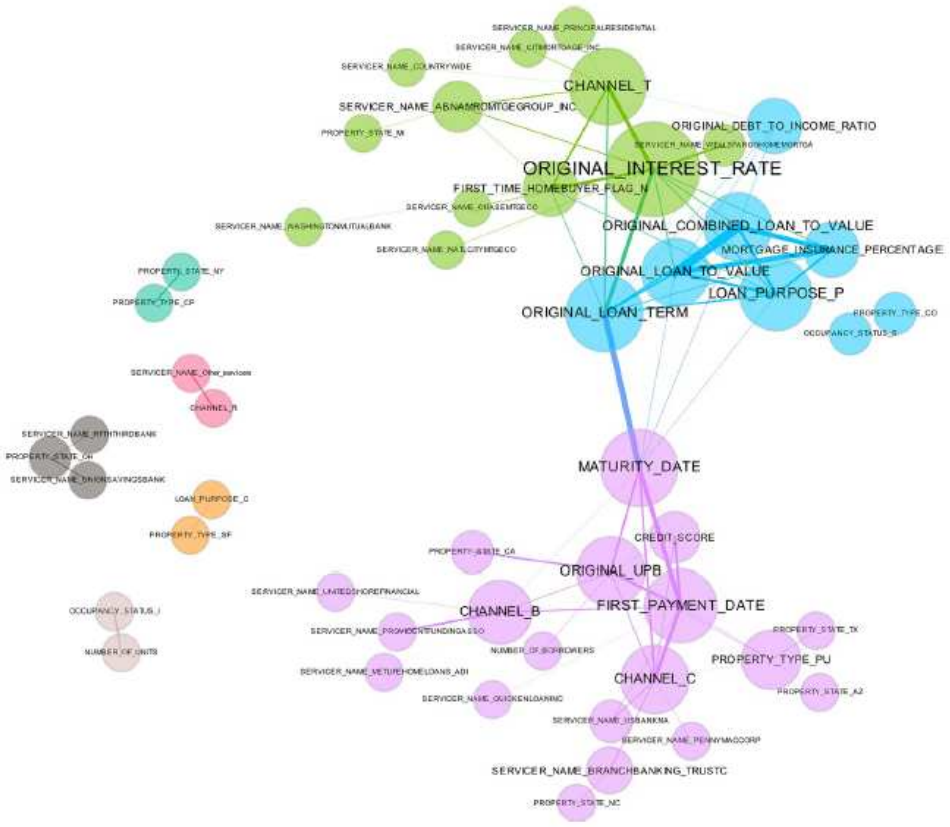
O eixo $x$ representa o valor da característica $f0$ e o eixo $y$ representa o valor previsto. A linha sólida na área sombreada mostra como a previsão média varia de acordo com as mudanças no valor de $f0$.

O segundo conjunto de técnicas apresentado por Hall e Gill (2019, p. 24) diz respeito aos grafos de correlação, ou *correlation graphs*. Essa técnica é uma representação bidimensional das relações no conjunto de dados, permitindo o entendimento das possíveis

correlações. Segundo os autores, esse tipo de técnica é uma ferramenta poderosa em "mineração de texto ou modelagem de tópicos, pois permite a visulização entre entidades e ideias". Nesse grafos, os nodos (ou vértices) são as variáveis do dataset e as arestas entre os nodos têm seu peso (visualmente expresso na espessura da aresta) definido pelo valor absoluto do seu coeficiente de correlação de Pearson (também chamado de coeficiente de correlação produto-momento, o qual mede o grau da correlação linear entre duas variáveis quantitativas).

Hall e Gill (2019, p. 24) explicam que para "simplicidade visual, pesos absolutos abaixo de um certo limite não são exibidos" e que o tamanho dos nodos é determinado pelo seu grau, o qual aumenta conforme maior número de conexões esse tiver. Cor e posição do nodo também passam por processos de determinação específicos. Segundo os autores, "o grafo de correlação permite a visulização de variáveis correlatas, identificação de variáveis irrelevantes e desvela ou verifica relações importantes que o modelo de aprendizagem de máquina deve incorporar". Um exemplo pode ser visto na figura 6.

Figura 6 – Grafo de Correlação




Fonte: Hall e Gill (2019, p. 25)

## 2.4 Humanidade, Técnica e Cultura

Os seres humanos sempre viveram em um ambiente composto tanto por objetos naturais quanto por objetos técnicos, onde esses dois elementos não se relacionam de modo que o técnico seja apenas um meio através do qual se conquista o natural, ao invés disso, esses objetos "constituem um sistema dinâmico que condiciona a experiência e existência humana" (HUI, 2016, p.1).

Sobre essa relação entre humanidade e técnica, Stiegler (1998, p. 135) afirma que a história da técnica é a história da saga humana em busca da evolução "a partir de meios que não a vida". Para o autor, essa também é a história da própria humanidade, uma vez que o processo de "corticalização" do cérebro foi paralelo ao "curso da lenta evolução das técnicas de pedra lascada", tão lenta que o autor questiona quem inventou quem e afirma que o "técnico inventa o humano e o humano inventa o técnico" (STIEGLER, 1998, p.137).

Stiegler (2009, p.2) se refere a essa relação de genesis compartilhada entre humanidade e técnica a partir do termo cunhado por Gilbert Simondon para tratar do mesmo assunto: "transdutiva", ou seja, "uma relação cujo os elementos são consituídos de modo que um não pode existir sem o outro - onde os elementos são co-constituentes", Stiegler (2009, p.163) afirma ainda que o "conhecimento humano é tecnológico em sua essência", não havendo a concepção do conhecimento empírico sem o advento de meios e métodos para a externalização da memória.

No entanto, no século XIX, o medo da substituição do homem pela máquina se solidifica em um pessimismo progressista que sabota a relação entre cultura e técnica (Cf. SIMONDON, 1958, p.15). Assim, no século XX, a cultura enquanto "base de significações, de meios de expressão, de justificações e de formas" (SIMONDON, 1958, p.14) "não comporta o desenvolvimento da realidade técnica do seu tempo" (KRITSKI; CALAZANS, 2020).

Desse modo, forma-se uma polaridade entre a nossa cultura e a cultura técnica, esse hiato entre esses dois conjuntos de valores e significados se aprofunda a partir de um humanismo fácil que ignora a realidade técnica enquanto algo repleto de esforço humano e que media a relação da humanidade com o mundo (Cf. SIMONDON, 1958, p.09).

> "Com isso, a cultura deixa de absorver valores importantíssimos, advindos da teoria da informação e da comunicação, para a estabilização das dinâmicas presentes na sociedade. Pois os seres técnicos, nos seus estados de individualização, são realidades humanas materializadas, que participam e interferem na dinâmica social." (KRITSKI; CALAZANS, 2020)

Para Simondon (1958) é somente a partir da cultura "em sua forma completa" que se pode desvelar essa humanidade materializada nos artefatos, pois existe uma comunicação que somente a cultura é capaz de realizar entre os indivíduos no mundo e na sociedade

e entre os líderes e o mundo. Enquanto "reguladora da sociedade" (SIMONDON, 1958, p.14), a cultura motiva e assegura o que é cristalizado no artefato. Uma cultura defasada de símbolos e esquemas técnicos atuais impõe uma distorção na realidade o que leva a uma comunicação falha entre líderes e mundo, onde não se pode estabelecer uma causalidade circular entre a autoridade e a realidade governada (Cf. SIMONDON, 1958, p.14).

Quando trazemos essas questões para a atualidade, nos encontramos meio a um novo tipo de objeto, o objeto digital, "cuja a natureza ainda precisa ser explicada" (HUI, 2016, p.2). Para Hui (2016, p.1), ao se referir a objetos digitais o autor recorta seu escopo àquilo que constituem os dados, tanto os que "tomam forma em uma tela" ou àqueles que "se escondem no *back-end* de um programa de computador", o autor define objetos digitais como "uma nova forma de objeto industrial que permeia todos os aspectos de nossas vidas nesses tempos mídia onipresente".

No caso dos objetos digitais, a análise matemática dos processos que permitem o funcionamento dos artefatos é apenas uma face que nos ajuda entender a existência desses objetos e que, segundo Hui (2016, p.3), acaba limitada às expectativas práticas dos engenheiros e cientistas da computação. O autor explicíta também a necessidade de uma conceituação filosófica dos objetos digitiais, pois estes fogem do alcance do entendimento construído de "Aristoteles até à filosofia moderna" acerca dos objetos naturais (HUI, 2016, p.2).

Nesse sentido, o modo de existência dos objetos digitais também precisa ser caracterizado na cultura, pois para compreender os possíveis desdobramentos do uso destas ferramentas é preciso levar em conta que "a invenção intervém quando o filtro social a deixa passar" (SIMONDON, 1958 apud FEENBERG, 2015). A absorção da tecnicidade dos objetos digitais para dentro do conjunto de significados que constroem a cultura importa porque não se trata apenas de frutos dessa relação da sociedade com o mundo, mas também de agentes que ponderam as mais diversas relações de causalidade na sociedade e no mundo a partir de seus valores intrínsecos que lhes são atribuídos em sua construção a partir do esforço humano empregado nesta. Assim, é essencial que a cultura perceba o objeto técnico e digital, aprendendo seus esquemas de funcionamento de modo que os pesos que este atribui nas mais diversas relações não seja ignorado e nem mal compreendido. O capítulo a seguir apresenta obras que remetem justamente as problemáticas que surgem desse hiato entre os objetos técnicos e digitais e a cultura.

# 3 Questões Éticas e Sociais em Produções de Inteligência Artificial

Esse capítulo apresenta obras correlatas a essa pesquisa, no sentido de que se preocupam em observar os desdobramentos sociais e éticos do uso das aplicações de inteligência artificial na vida cotidiana, suas limitações e perigos. O capítulo seguinte apresentará a ferramenta TCAV a partir de sua obra de introdução por Kim et al. (2017).

## 3.1 Armas de Destruição Matemática

Em uma análise crítica das influencias dos algoritmos de Inteligência Artificialna sociedade, Cathy O'Neil (2016) apresenta um trabalho notável ao expor as fragilidades em ferramentas munidas de algoritmos matemáticos que orientam a tomada de decisões de impacto social local e até mesmo global. O'Neil apresenta situações como a do algoritmo IMPACT, utilizado pelo governo de Washington, nos Estados Unidos, em 2009 para decidir quais professores seriam culpabilizados pelo baixo desempenho escolar no período e demitidos. O IMPACT se tratava de um processo de avaliação dos professores e seu resultado era mais relevante que a avaliação da comunidade a da administração da escola. Ou seja, professores que eram queridos pela comunidade e bem vistos pela administração da escola tiveram esses fatores suprimidos pelo resultado obtido matematicamente pelo IMPACT. Um modelo que não buscava por respostas sobre o porquê do mal desempenho das escolas, mas personificava verdades que não consideravam aquilo que ele não estava programado para assimilar em seu algoritmo.

O'Neil relata que esse tipo de aplicações matematicamente empoderadas são justamente baseadas em escolhas humanas, que ainda que eventualmente sejam dotadas das melhor intenções, ainda assim codificam preconceitos, incompreensões e enviesamento em algoritmos que cada vez se apoderam dos direcionamentos da vida cotidiana. A autora ainda se refere a esse tipo de modelos como "deuses" que agem de forma incompreensível para todos, exceto aos quais ela chama de "sumos sacerdotes": os matemáticos e os cientistas da computação. Seu livro é uma chamada para os perigos que essa concentração de saberes em nichos muito pequeno da sociedade representa em meio essa sociedade orientada à produção e consumo de dados.

## 3.2 Algoritmos de Opressão

No seu livro Algoritmos de Opressão (2018), Noble buscar iniciar um processo de entendimento e visibilização das "consequências de longo prazo de ferramentas de tomada de decisão no processo de mascarar e aprofundar desigualdades sociais". Para a Noble, o desafio para a compreensão, do que ela chama de opressão algorítmica, está em "entender que as formulações matemáticas que levam às decisões automatizadas são feitas

por humanos", onde ressalta que não há neutralidade, benignidade ou ainda objetivismo em termos como "algoritmos" e "big data", pois "as pessoas que tomam essas decisões carregam em si todo tipo de valor, muitas das quais abertamente promovem racismo, sexismo e falsas noções de meritocracia".

Noble se propõe a explorar, a partir da perspectiva de identidades negras, como alguns desses processos algoritmicamente orientados se tornaram "fundamentais na organização e classificação de informação e a que custo". Para tal, ela evidencia e argumenta que "grandes monopólios, como Google, precisam ser divididos e regulamentados", uma vez que a consolidação do poder e influência cultural desse tipo de empresa torna a competição majoritariamente impossível", ameaçando a própria democracia, segundo a autora.

A autora afirma que é necessário se perguntar "o que se perde, quem é machucado e o que deve ser esquecido" na adoção de ferramentas inteligentes de tomada de decisão, pois não é de "benefício social coletivo organizar recursos informacionais na rede a partir de processos que solidificam desigualdade e marginalização".

## 3.3 Uma Crítica Feminista às Tendências Recentes em Interação Humano-Robô

Em um recorte crítico mais restrito, a partir da epistemologia feminista, Weber produziu em 2005 um artigo ao qual ela da o subtítulo de "uma crítica feminista às tendências recentes em interação humano-robô". A autora examina algumas "(questionáveis) presunções epistemológicas, ontológicas e antropológicas" do então emergente campo da interação humano-robô.

Nesse trabalho, a crítica de Weber passa pela análise de alguns artefatos que reproduzem estruturas de relacionamento questionáveis com os sujeitos humanos da sociedade. O artefatos que Weber explora são: o robô Kismet, (BREAZEAL 2002, apud WEBER 2005), que é desenhado para ser tratado como uma criança; robôs-pet (robô-cachorro, robô-gato, etc) (Steels & Kaplan, 2001 apud Weber 2005), para serem treinados por seus donos; a Robota, uma boneca-robô-brinquedo desenvolvida por Billard que se apresenta dentro de normas bastante estereotipadas (vestimentas, corpo, comportamentos, por exemplo, sua tarefa favorita é gostar de se vestir); e ainda a robô Valerie, uma dita "androide doméstica", desenhada de como um "robô de limpeza de casa com um forte, se não ameaçador, sex appeal" (WEBER, 2005).

A crítica de Weber a esses modelos de apresentação dos artefatos e interação dos mesmos com a sociedade se divide em quatro dimensões (representação de gênero, teoria social, antropológicas e ontológicas) a partir de uma epistemologia da tecnociência feminista. Na primeira dimensão ela argumenta que por mais "claro que seja a necessidade de se criticar esses estereótipos, padrões, normas e papéis de gênero na tecnologia" não é "simplesmente suficiente revisar o design dessas tecnologias no sentido de remover suas formas estereotipadas implícitas ou explícitas" uma vez que a base conceitual teórica utilizada na construção tecnológica está apoderada e "questões de gênero ontológicas,

epistemológicas e antropológicas e a eliminação a nível social das normas, papéis e estereótipos não resolve essas questões.

Sobre a segunda dimensão, Weber justifica que a relação entre "'máquinas sociais' e a padronização da vida cotidiana deveria ser explorada a partir da perspectiva da teoria social", onde ela propõe o questionamento sobre qual papel esperar que esses novos agentes tenham na sociedade, e se essas novas relações surgem da necessidade de se suprir "deficiências de nossa vida social moderna e especialmente em uma economia neoliberal"(WEBER, 2005).

No campo antropológico e ontológico, Jutta questiona qual a conceituação de quatro aspectos: sociedade, sociabilidade, interação humana e interação humano-máquina. Segundo, a autora, o desenho desses artefatos não é reducionista apenas no que diz respeito às limitações impostas pelo estereótipo, mas um entendimento reducionista de sociabilidade em muitas abordagens de ciências sociais e comportamentais. A autora então questiona se é realmente desejável esse tipo de relação na interação humano-máquina e ainda se "nós realmente queremos educar nossas máquinas".

A autora conclui questionando para além da conceitualização da relação humano-robô, mas também os próprios papéis que se atribuem e se pretende atribuir a não só enquanto nossos "espelhos ou substitutos".

# 4 Testagem Quantitativa por meio de Vetores de Ativação de Conceitos

Em seu artigo, Kim et al. (2017) apresentam a operação de testar com CAVs, "*concept activation vectors*", ou vetores de ativação de conceito, chamada pelos pesquisadores de TCAV ( *"testing with concept activation vectors"*). Segundo os autores o TCAV "provê uma interpretação do estado interno de uma rede neural em termos de conceitos humano-amigáveis" (KIM et al., 2017) em redes de aprendizagem supervisionada utilizando os "estados internos de alta dimensão como um auxílio e não um obstáculo" (KIM et al., 2017). Os autores alegam que os CAVs podem ser facilmente usados por não-especialistas que "apenas precisam prover os exemplos"(KIM et al., 2017).

Os experimentos realizados pelos pesquisadores buscaram apontar como o TCAV "permite a análise quantitativa da relação de um conceito para uma classificação" (KIM et al., 2017) em modelos de classificação de imagem já treinados (*Inception V3* e *googlenet*). Por exemplo, o quão importante foi o conceito 'capa', para a predição do classe 'super-herói', ou então o conceito 'árvores' para a classe 'bosque', etc. O artigo explica que o método foi desenvolvido com o intuito de atingir os seguintes objetivos:

> "requerisse pouco ou nenhum conhecimento referente à aprendizagem de máquina [...], que pudesse ser adaptado à qualquer conceito sem se limitar à conceitos considerados durante o período de treino [...], que funcionasse sem modificações ao modelo de aprendizagem de máquina"; e que "pudesse interpretar classes ou conjuntos inteiros de exemplos com apenas uma medida" (KIM et al., 2017)

Kim et al. (2017) reconhecem a atualidade e significância de compreender o comportamento de modelos de aprendizagem de máquina, dada a abrangência e importância de suas aplicações. Os autores afirmam que é preciso se importar não só com a qualidade e acurácia nas suas diversas aplicações, mas também garantir que esses "modelos em aprendizagem de máquina estejam alinhados com nossos valores" (KIM et al., 2017).

O desafio que o TCAV se propõe a superar é a abstração da interpretação dos estados internos do modelo para conceitos mais amigáveis à interpretaçao humana, do que por exemplo, "descrever um modelo de aprendizagem de máquina em termos das características de entrada que esse considera" (KIM et al., 2017) como acontece na atribuição de pesos de importância aos pixeis em mapas de saliência. A dificuldade se dá pois, de acordo com Kim et al. (2017), o espaço vetorial para dados de entrada e ativações neurais do estado interno de um modelo é $E_m$ enquanto o espaço vetorial que compreende o conjunto desconhecido de conceitos humano-interpretáveis é $E_h$. Nessa perspectiva, interpretação linear dá quando $g$ tal que $g : E_m \rightarrow E_h$.

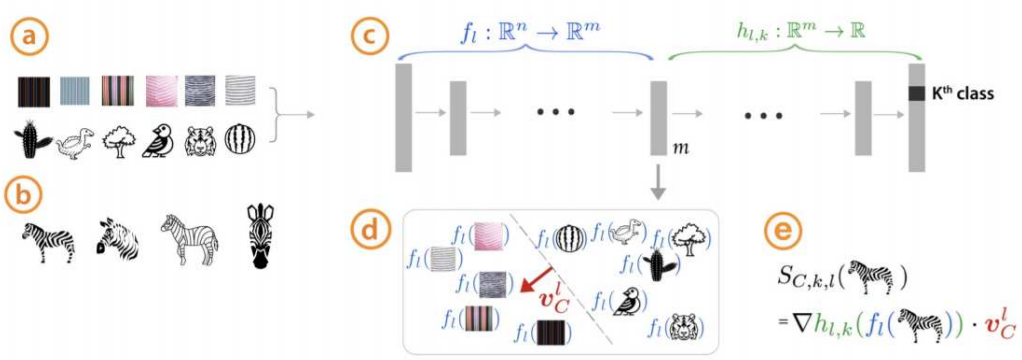
Kim et al. (2017) afirmam que um vetor de ativação de conceito (CAV) é uma

maneira de traduzir $E_m$ para $E_h$. E segundo Doshi-Velez e Kim (2017 apud KIM et al., 2017), não há necessidade de perfeição em uma função de interpretabilidade, pois "permite-se a essa falhar em explicar alguns aspectos da entrada" e assume-se a "inevitabilidade de não se cobrir todos os possíveis conceitos humanos em $E_h$.

O desenvolvimento desse trabalho busca apontar as fragilidades no método por trás do TCAV, uma vez que os conceitos em $E_h$ considerados por $g$ são definidos por um conjunto de exemplos à escolha de quem opera a ferramenta, traduzindo desse modo o conjunto de entradas e ativações neurais em $E_m$ para uma pontuação associada (chamada no texto de *TCAV Score*) aos conceitos disponíveis em $E_h$.

Para tanto, o método em TCAV busca um vetor no espaço das ativações que represente cada conceito em $E_h$, uma vez que um CAV é justamente um vetor na direção desses conceitos partindo da reta normal ao hiperplano que melhor separa os exemplos relacionados à classe dos conceitos aleatórios. Na figura 7, retirada do artigo original, o CAV é representado pelo vetor em vermelho, apontando para as características atribuídas ao conceito estudado no exemplo do artigo (zebras).

Figura 7 – Testando com Vetores de Ativação de Conceito.




Dado um conjunto de exemplos definido pelo usário (ex: listrado) e um conjunto de dados aleatórios, **a)** dados de treino classificados (zebras), **b)** uma rede neural treinada, então **c)** o TCAV pode quantificar a sensibilidade da rede ao conceito para aquela classe. CAVs são apreendidos ao se treinar o classificador linear para distinguir entre as ativações relacionadas aos exemplos e aos dados aleatórios em qualquer camada; **d)** o CAV é o vetor ortogonal ao classificador (hiperplano) em vermelho. Para a classe de interesse (zebras), **e)** o TCAV utiliza derivadas direcionais $S_{c,k,1}$ para quantificar a sensibilidade conceitual.

Fonte: Traduzido de Kim et al. (2017)

A significância dos resultados se dá a partir da quantidade de rodadas de treinamento para cada CAV, os quais são apreendidos no processo de treinar o classificador linear para discernir entre as ativações produzidas pelos exemplos e contraexemplos nas camadas da rede neural. As quantidade de rodadas de treinamento são definidas pela quantidade de conjuntos de conceitos aleatórios disponíveis. Os testes realizados pelos autores apresentam resultados para 500 conjuntos de conceitos aletórios, ou seja 500 rodadas de treinamento, mas em explicação no código fonte da ferramenta diz-se possível obter resultados significativos a partir de testes com 10 conjuntos de conceitos aleatórios.

Matematicamente a operação se dá pela expressão a seguir, onde $X_k$ se refere a todas as entradas, $k$ representa a classe e $S_{C,k,l}(x)$ é a derivada direcional de uma ativação $x$ para um exemplo de uma camada $l$, para a classe $k$ e o conceito $C$. Desse modo, o *TCAV score* mede a razão das entradas para a classe $k$ que são positivamente afetadas pelo conceito $C$.

$$TCAV_{Q_{C,k,l}} = \frac{|x \epsilon X_k : S_{C,k,l}(x) > 0)|}{|X_k|}$$

Ou seja, dado um conjunto de todos exemplos $X_k$ para uma classe $k$ conhecida por uma rede neural $Y$, observa-se a partir do *Score* $S_{C,k,l}(x)$, como os estados internos de $Y$ se comportam em termos de ativações para cada exemplo $x$ do conjunto $X_k$, dada a classe $k$ e a camada $l$. O *TCAV score* é a média dos valores ao final de todas rodadas de treinamento. Desse modo, a ferramenta quantifica a sensibilidade contextual de um conceito para todas as entradas de uma classe $k$, enquanto o CAV é o vetor no espaço da rede neural $Y$ obtido a partir do treinamento do classificador linear que divide os exemplos contendo conceitos propostos dos exemplos de conceitos aletórios.

Para verificar estatisticamente os resultado obtido, o método realiza internamente um teste de hipótese nula apresentado a seguir.

## 4.1 Teste Estatístico de Significância

Para validar o *TCAV score* obtido, é necessário que a hipótese nula de um *TCAV score* igual a 0.5 seja rejeitada, então pode-se considerar o conceito significativo à classe. Desse modo, quando o *p-value* for maior ou igual a 0.05, a hipótese do teste (de que o conceito é relevante à classe) deve ser rejeitada. A seção a seguir explica como a ferramenta TCAV funciona enquanto um "método global de perturbação" e como é possível validar os resultados obtidos a partir desse conceito.

## 4.2 Um Método Global de Perturbação

O TCAV se enquadra como uma ferramenta de pós-processamento de modelos de aprendizagem de máquina que faz uso de um método global de perturbação também conhecido como análise de sensibilidade. Isso significa que é possível utilizar uma saída $h$ gerada para uma determinada entrada $m$ em uma rede neural enquanto uma nova entrada $m_1$, buscando validar as conceituações obtidas, pois a partir dessa nova entrada, verifica-se se a nova saída $h_1$ é consistente com os resultados obtidos para a entrada inicial $m$.

As inconsistências de um método baseado em perturbação podem surgir quando a "explicação só for verdade para um ponto particular do conjunto de dados e seus vizinhos" (RIBEIRO; SINGH; GUESTRIN, 2016 apud KIM et al., 2018), o que é chamado de

explicação local, em contrapartida à explicação global a qual o TCAV se propõe, onde a explicação é consistente para todas as entradas da classe.

Para validar que "os CAVs aprendidos estão alinhados com o conceitos de interesse" (KIM et al., 2017), são apresentados dois métodos diferentes. O primeiro é a classificação de imagens em respeito a relação delas com o conceito de interesse, utilizando imagens que não foram usadas para treinar o CAV. O experimento busca verificar o quais imagens do conjunto mais se enquadram em um conceito apreendido pelo CAV.

Como o CAV codifica a direção de um conceito no espaço vetorial da camada selecionada (chamada de *bottleneck* pelo autores), a partir das ativações das novas imagens, computa-se a semelhança de cosseno, entre o conjunto de imagens de interesse para o CAV classificar as imagens enquanto mais ou menos semelhantes ao conceito. A figura 8 mostra os resultados obtidos nesse tipo de classificação para o conceito listras e classe zebra,

Figura 8 – Classificação de Imagens de Zebra Enquanto sua Sensibilidade para o Conceito Listras


Fonte: Traduzido de Kim et al. (2017)

A segunda forma proposta de evidenciar o alinhamento entre CAV e conceito, é a partir da técnica de maximização de ativação (*Deepdream* para o padrão que possui o TCAV *Score* mais alto, onde compara-se o resultado obtido a partir do Deepdream com as "noções semânticas [dos pesquisadores] sobre o conceito".

Em seguida, os pesquisadores apresentam um sumário gráfico dos insights obtidos nos seus experimentos (figura 10) e apontam um suposto enviesamento dos modelos a partir da experiência com o TCAV. O artigo também mostra o desempenho do TCAV na interpretação de um modelo médico para diagnóstico de retinopatia diabética, quando o modelo divergia dos especialistas na área. O artigo conclui assumindo o TCAV como um passo inicial na construção da interpretabilidade em redes neurais profundas e sugere que

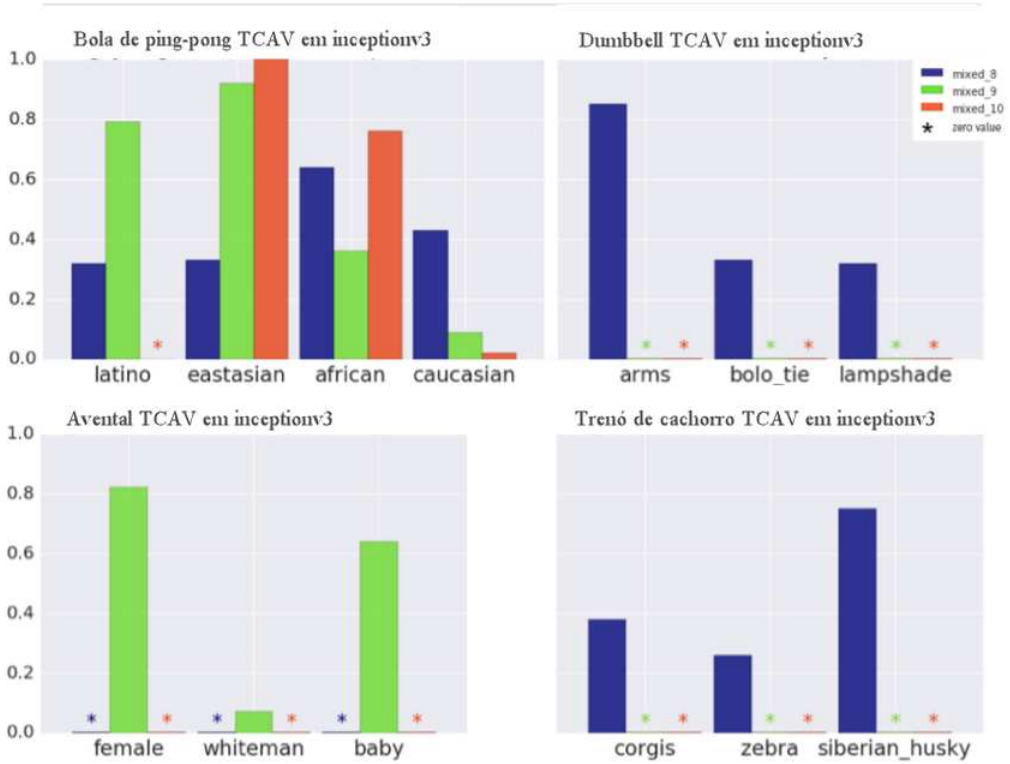
Figura 9 – Deepdream

Testando com Deepdream usando vetores conceituais (CAV) para a textura tricô, e as raças de cachorro: corgi e huskey siberiano.

Fonte: Traduzido de Kim et al. (2017)

o método pode ser usado como uma técnica útil no arcabouço de um analista.

Figura 10 – Resultados de teste de sensibilidade com o TCAV na rede neural Inceptionv3



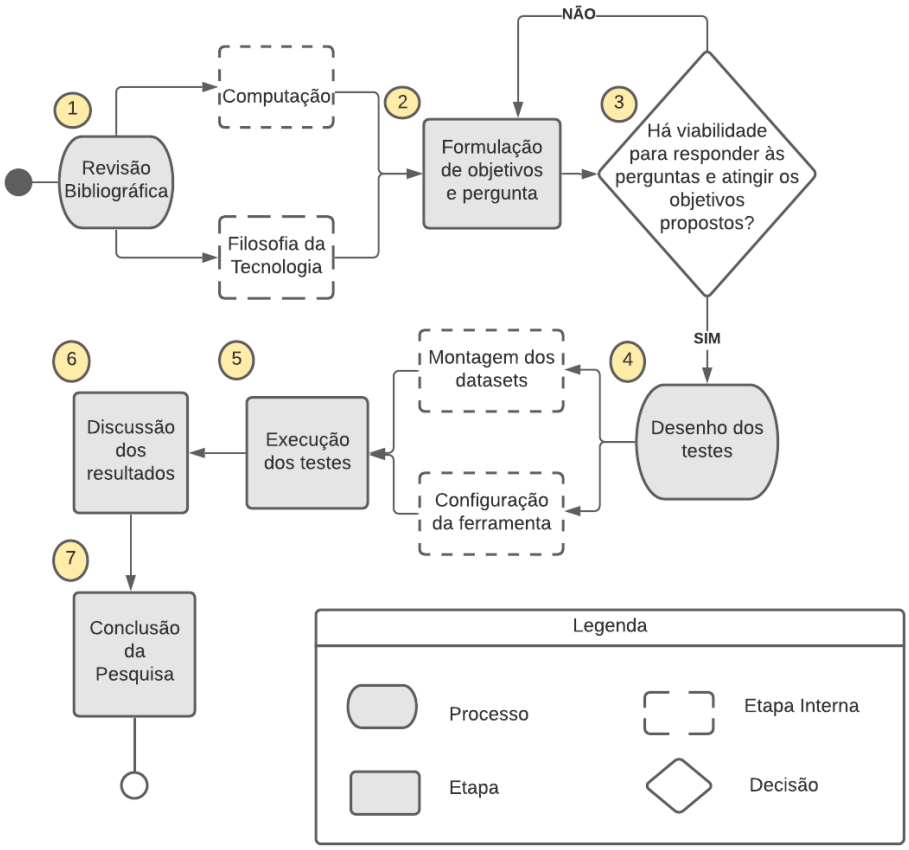

Fonte: Traduzido de Kim et al. (2017)

O próximo capítulo destina-se a apresentação do percurso metodologico empregado nessa pesquisa.

# 5 Percurso Metodológico

Esse capítulo apresenta o percurso metodológico desta pesquisa a fim de estudar a ferramenta TCAV, ilustrado na figura 11, e, a seguir, descreve cada etapa individualmente.

Figura 11 – Metodologia do estudo




Fonte: Autoria própria

1. Revisão Bibliográfica: se refere ao processo de buscar bases teóricas à consecução do estudo do objeto teste, TCAV, apresentado no capítulo 2 deste trabalho. Internamente esse processo inicial se subdivide em duas grandes áreas: computação e filosofia da tecnologia. A partir da primeira área visamos à compreensão do funcionamento computacional do objeto, enquanto na segunda, buscamos a partir de alguns conceitos da filosofia da tecnologia, ampliar nossa perspectiva sobre a natureza do objeto teste e seus possíveis efeitos sociais.
2. Formulação de objetivos e pergunta: nessa etapa, a partir da fundamentação teórica construída no passo anterior, constroem-se os objetivos e a pergunta a ser respondida por essa pesquisa (descritos na seção 1.2).

3. Etapa de verificação da viabilidade, em termos de recursos técnicos e tempo disponível para a realização da pesquisa, de atender aos objetivos e responder à pergunta proposta na etapa 2. Em caso negativo é necessário retornar à etapa anterior e reformular objetivo e pergunta a fim de que a conclusão da pesquisa seja viável.
4. Desenho dos testes possíveis: processo de análise e escolha das classes conhecidas pela rede utilizada para o estudo (*Inception 5h*) e dos conceitos a serem verificados pelo objeto teste para cada classe escolhida, expostos na seção 6.1. As etapas internas a esse processo consistem na montagem dos datasets para cada classe e conceito a partir de ferramentas de pesquisa de imagem como *Google* e *Yandex* e também configuração da ferramenta (descrita na subseção 6.1.2).
5. Etapa de execução dos testes propostos, descritos na seção 6.2.
6. Discussão dos resultados obtidos, a partir da nálise dos resultados obtidos e do discursos sobre o TCAV em Kim et al. (2017) a partir do arcabouço teórico apresentado nessa pesquisa, no capítulo 2;
7. A etapa de conclusão dessa pesquisa consiste em identificar se objetivos, apresentados na seção 1.2, foram atingidos, bem como, a partir dos resultados obtidos e apresentados no capítulo 6, se desdobrar sobre a pergunta proposta, apresentando as considerações finais dessa pesquisa sobre o objeto teste.

O capítulo a seguir descreve o desenvolvimento da metodologia proposta acima no que diz respeito à execução das etapas 4 a 6.

# 6 Desenvolvimento

Este capítulo se divide em três seções e descreve as etapas 4 a 6 da metodologia empregada nesta pesquisa e descrita no capítulo 5. A seção a seguir apresenta os testes propostos para os fins da investigação acerca do funcionamento da ferramenta. -

## 6.1 Desenho dos Testes

Essa seção descreve as questões que motivaram os testes elaborados para a verificação experimental da ferramenta TCAV, cujos resultados obtidos estão descritos a seguir, na subseção 6.2.

Nas instruções[1] para a execução da ferramenta é explicitado que resultados significativos são obtidos a partir de 10 a 20 rodadas de treinamento. O teste 1 se dedica a entender qual o impacto das diferenças de cenário de teste no comportamento da ferramenta. As diferenças no cenário dos teste são: o modelo utilizado e o número de rodadas de treinamento para o *TCAV Score*. Em Kim et al. (2017, p. 11) utiliza-se o modelo *Inception V3* e 500 rodadas de treinamento e esta pesquisa utiliza o modelo *Inception 5h* e 10 rodadas de treinamento.

Para verificar quantitativamente essa diferença de cenários, o teste 1 dessa pesquisa reproduz o teste realizados em Kim et al. (2017, p.11) para verificar a sensibilidade da predição da classe “zebra” para os conceitos “listrado”, “ziguezagueado”, e “pontilhado” sob as novas condições de cenário. O conjunto de dados para os conceitos e a classe foram obtidos junto ao código fonte da ferramenta.

Os testes 2 e 3 verificam a condição do conjunto de dados para os conceitos “listrado”, “ziguezagueado”, e “pontilhado”, utilizados no teste 1, quando empregados na verificação de sensibilidade da predição de outras classes. Propõe-se então a verificação da sensibilidade da predição das classes ”dálmata” e “girafa” para estes conceitos supracitados, utilizando os mesmos conjuntos de dados empregados no teste 1. A ideia desses dois cenários de teste é averiguar empiricamente se a construação desses conjuntos de dados inicialmente utilizados para a predição da classe ”zebra” (apresentado em Kim et al. (2017, p. 11) e na figura **??**), seriam significativos também para outras classes. A hipótese é de quem a classe “dálmata” irá ativar o conceito “pontilhado” e a classe “girafa” irá ativar o conceito “ziguezagueado”.

Os testes de 4 a 7 verificam a sensibilidade da predição da classe “abelha” para diferentes conceitos. A proposta desses testes é verificar se o TCAV é realmente capaz de nos ajudar a entender como o modelo *Inception 5h* prediz a classe “abelha”, uma vez que o modelo reduz à uma única palavra (“abelha”) algo que na materialidade se manifesta

[1] Disponível em https://github.com/tensorflow/tcav

em cerca de 30 mil espécies diferentes, de acordo com Bradbear et al. (2009, p.5). Os teste 4 e 5 (figura 27 e figura 28) verificam a sensibilidade dos conceitos "listrado", "flores" e "asas de abelha". No teste de número 4, o conjunto de dados para o conceito "listrado" é o mesmo utilizado no teste 1, utilizado inicialmente para predição da classe "zebra". No segundo teste para a classe "abelha", o teste de número 5, esse conjunto de dados para o conceito "listrado", é incrementado com imagens mais diversas que também representam o conceito, permitindo-nos verificar se ao aumentar diversidade do conjunto de dados aumenta-se também o grau de sensibilidade do conceito para a predição da classe.

Os teste 6 e 7 se aproximam da questão central para os testes com a classe "abelha" (figura 29), verificando qual a sensibilidade do modelo para diferentes espécies de abelhas na predição da classe. No teste 6, considera-se enquanto conceitos as espécies de abelha: "abelha-europeia" (*Apis mellifera*), "abelha carpinteiro tropical" (*Xylocopa latipes*) e "abelha do suor" (*Halictidae*). As espécies foram escolhidas pela visível diferença física entre cada, de modo que o conjunto de dados para cada espécie-conceito pudesse ser montado por um não especialista, exemplos de cada espécie podem ser vistos nas figuras 12, 13 e 14. O conjunto de dados de entrada para a classe foi composto pela união dos conjunto dos conceitos.

Figura 12 – Imagem exemplo para o conceito "abelha-europeia"

Fonte: <http://www.henryhartley.com/?p=19191>

Figura 13 – Imagem exemplo para o conceito “abelha carpinteiro tropical”

Fonte: <www.NatureLoveYou.sg>

Figura 14 – Imagem exemplo para o conceito “abelha do suor”

Fonte: <https://nature.mdc.mo.gov/discover-nature/field-guide/halictid-bees-sweat-bees>

O teste 7 verifica os mesmos conceitos do teste 6 com adição do conceito “vespa-mandarina” (ilustrada na figura 15, que não é uma espécie de abelha. O objetivo desse teste é verificar se a predição da classe abelha é sensível a um conceito que não diz respeito as abelhas em si, mas a outra espécie de inseto. Almeja-se com os resultados desse teste (figura 30) analisar a relação que o TCAV ilustra quantitativamente entre classe e conceito quando ambos não necessariamente se relacionam na realidade.

Figura 15 – Imagem exemplo para o conceito “vespa mandarina”

Fonte: <https://www.theinsectcollector.com/acatalog/info-34187.html>

Os testes 8 a 10 verificam a sensibilidade da predição da classe “urso-polar” para os mesmos conceitos, “pelagem”, “pelagem branca”, “face de urso polar” e “polo ártico”, mas utilizando conjuntos distintos para representar a classe. As imagens utilizadas no conjunto de dados do conceito “face de urso polar” foram todas recortadas para que o enquadramento ficasse como demonstrado na figura 19.

Figura 16 – Imagem exemplo para o conceito “pelagem branca”

Fonte: <https://www.publicdomainpictures.net/pictures/210000/velka/white-fur-background-1481782017pXZ.jpg>

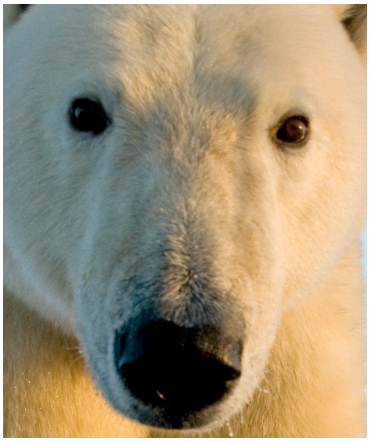
Figura 17 – Imagem exemplo para o conceito “pelagem”

Fonte: <https://wallpapercave.com/wp/wp3239003.jpg>

Figura 18 – Imagem exemplo para o conceito “polo ártico”

Fonte: <https://www.flickr.com/photos/gsfc/5937599688/>

Figura 19 – Imagem exemplo para o conceito “face de urso polar”



Fonte: Adaptado de
<https://www.hakaimagazine.com/wp-content/uploads/header-polar-bear-conflict.jpg>

O teste 8 (figura 31) verifica à sensibilidade da predição da classe “urso polar” para esses conceitos quando o conjunto de dados de entrada para a classe constitui-se de imagens que exibem um ou mais ursos polares, com a maior parte do corpo visível na imagem, em seu habitat natural ou não.

Figura 20 – Exemplo de enquadramento das imagens utulizadas como entrada para a classe “urso polar” para o teste 8

Fonte: Adaptado de <https://arctickingdom.com/wp-content/uploads/2020/03/DSC_5688.jpg>

O conjunto de dados de entrada para a classe no teste 9 (figura 32) utiliza apenas imagens da face de ursos polares recortadas para enquadrar apenas olhos e focinho. O teste 10 (figura 33) utiliza a união do conjunto de dados de entrada utilizados no teste 8 e no teste 9.

Figura 21 – Exemplo de enquadramento das imagens utilizadas como entrada para a classe “urso polar” para o teste 9

Fonte: <https://kids.sandiegozoo.org/sites/default/files/2017-07/polar-bear-face.jpg>

O teste 11 verifica a classe “urso polar” para os conceitos “lobo branco” e “animais peludos brancos” utilizando a união do conjunto de dados de entrada utilizados no teste 8 e

no teste 9. Esse teste busca verificar se a ferramenta pode tender a considerar significativos conceitos visualmente parecidos à classe mas que não se referem a ela na realidade.

Figura 22 – Imagem exemplo para o conceito “animais peludos brancos”

Fonte: <https://besthqwallpapers.com/Uploads/22-6-2019/96840/thumb2-white-fluffy-puppy-pomeranian-spitz-small-white-dog-pets-puppies.jpg>

Figura 23 – Imagem exemplo para o conceito “lobo branco”

Fonte: <http://absfreepic.com/free-photos/download/the-white-wolf-in-forest-2594x1727_84269.html>

Os testes 12 a 18 reproduzem os cenários de testes propostos nos teste 5 a 11, se preocupando em equalizar o número de imagens de cada conceito, buscando verificar se o tamanho do cojunto dados pode influenciar resultados onde conceitos com conjuntos maiores seriam beneficiados com um TCAV *Score* maior. As figuras 35 à 40 ilustram os resultados obtidos.

A subseção a seguir apresenta a configuração necessária para a instalação da ferramenta e execução dos testes propostos.

### 6.1.1 Configuração da Ferramenta

O código fonte para a ferramenta TCAV está disponibilizado em <https://github.com/tensorflow/tcav> e no momento de instalação da ferramenta e execução dos testes os requerimentos para executar a ferramenta eram:

```
1 matplotlib==2.2.4
2 Pillow==6.2.0
3 scikit-learn==0.20.3
4 scipy==1.2.1
5 tensorflow==1.15.2
6 numpy==1.16.0
7 protobuf==3.10.0
```

1. *Matplotlib* é uma biblioteca para criar visualizações (estáticas, animadas e/ou iterativas) no *Python*.
2. *Pillow* é uma biblioteca de processamento de imagens que amplia o suporte à abertura e gravação de diversos formatos de imagem distintos.
3. A *scikit-learn* é uma biblioteca de aprendizado de máquina de código aberto para a linguagem de programação *Python*.
4. A *SciPy* é uma biblioteca de código aberto em linguagem Python que foi feita para matemáticos, cientistas e engenheiros.
5. *Tensorflow* é uma biblioteca de código aberto para aprendizado de máquina, a qual permite ao desenvolvedor criar e treinar redes neurais.
6. *NumPy* é um pacote para a linguagem Python que suporta operações matemáticas com vetores e matrizes multidimensionais.
7. *Protocol buffers* ou *protobuf* é um método de serialização de dados estruturados.

Dada a determinação do *tensorflow* à versão 1.15.2, os testes aconteceram em um ambiente virtual *Python* com a versão 3.7, pois versões mais recentes do *Python* não eram compatíveis com a versão do *tensorflow* designada nos requerimentos.

A subseção a seguir aborda o processo de montagem dos *data sets* para cada cenário de teste proposto.

### 6.1.2 Montagem dos Conjuntos de Dados

Os conjuntos de dados para classe e conceitos foram utilizados como organizados abaixo. Os três primeiros testes foram realizados com os conjuntos de dados fornecidos enquanto o exemplo para o teste com a classe zebra (disponível para download junto ao código fonte da ferramenta). O quarto teste utiliza ainda o conjunto de dados de entrada para o conceito “listrado” disponível para download como exemplo para o teste com a

classe zebra. Os testes seguinte tiveram seus conjuntos de dados de entrada produzidos a partir de busca por palavra chave nas plataformas de pesquisa Google e Yandex.

1. Teste de sensibilidade da predição da classe 'zebra' para os conceitos:
    - Listras (50 imagens)
    - Pontilhado (50 imagens)
    - Ziguezague (50 imagens)
2. Teste de sensibilidade da predição da classe 'girafa' para os conceitos:
    - Listras (50 imagens)
    - Pontilhado (50 imagens)
    - Ziguezague (50 imagens)
3. Teste de sensibilidade da predição da classe 'dálmata' para os conceitos:
    - Listras (50 imagens)
    - Pontilhado (50 imagens)
    - Ziguezague (50 imagens)
4. Teste de sensibilidade da predição da classe 'abelha' para os conceitos:
    - Listras (50 imagens)
    - Flores (306 imagens)
    - Asas de Abelha (58 imagens)
5. Teste de sensibilidade da predição da classe 'abelha' para os conceitos:
    - Listras 2 (150 imagens)
    - Flores (306 imagens)
    - Asas de Abelha (58 imagens)
6. Teste de sensibilidade da predição da classe 'abelha' para os conceitos:
    - Abelha-europeia (*Apis mellifera*) (159 imagens)
    - Abelha carpinteiro tropical (*Xylocopa latipes*) (80 imagens)
    - Abelha do suor (*Lasioglossum spp*) (77 imagens)
7. Teste de sensibilidade da predição da classe 'abelha' para os conceitos:
    - Abelha-europeia (*Apis mellifera*) (159 imagens)
    - Abelha carpinteiro tropical (*Xylocopa latipes*) (80 imagens)
    - Abelha do suor (*Lasioglossum spp*) (77 imagens)
    - Vespa-mandarina (257 imagens)
8. Teste de sensibilidade da predição da classe 'urso polar' (utilizando imagens de urso-polar inteiro e cenário como conjunto de dados de entrada para a classe) para os conceitos:
    - Pelagem (333 imagens)
    - Pelagem Branca (85 imagens)
    - Face de Urso Polar (67 imagens)
    - Polo Ártico (225 imagens)
9. Teste de sensibilidade da predição da classe 'urso polar' (utilizando imagens de rosto

de urso-polar como conjunto de dados de entrada para a classe) para os conceitos:
   - Pelagem (333 imagens)
   - Pelagem Branca (85 imagens)
   - Face de Urso Polar (67 imagens)
   - Polo Ártico (225 imagens)

10. Teste de sensibilidade da predição da classe 'urso polar' (utilizando imagens dos conjuntos de entrada para a classe 'urso polar' dos testes 8 e 9 combinados como conjunto de dados de entrada para a classe) para os conceitos:
   - Pelagem (333 imagens)
   - Pelagem Branca (85 imagens)
   - Face de Urso Polar (67 imagens)
   - Polo Ártico (225 imagens)

11. Teste de sensibilidade da predição da classe 'urso polar' (utilizando imagens dos conjuntos de entrada para a classe 'urso polar' dos testes 8 e 9 combinados como conjunto de dados de entrada para a classe) para os conceitos:
   - Animais brancos peludos (313 imagens)
   - Lobo branco (99 imagens)

12. Teste de sensibilidade da predição da classe 'abelha' para os conceitos:
   - Listras 2 (50 imagens)
   - Flores (50 imagens)
   - Asas de Abelha (50 imagens)

13. Teste de sensibilidade da predição da classe 'abelha' para os conceitos:
   - Abelha-europeia (*Apis mellifera*) (77 imagens)
   - Abelha carpinteiro tropical (*Xylocopa latipes*) (77 imagens)
   - Abelha do suor (*Lasioglossum spp*) (77 imagens)

14. Teste de sensibilidade da predição da classe 'abelha' para os conceitos:
   - Abelha-europeia (*Apis mellifera*) (77 imagens)
   - Abelha carpinteiro tropical (*Xylocopa latipes*) (77 imagens)
   - Abelha do suor (*Lasioglossum spp*) (77 imagens)
   - Vespa-mandarina (77 imagens)

15. Teste de sensibilidade da predição da classe 'urso polar' (utilizando imagens de urso inteiro e cenário como conjunto de dados de entrada para a classe) para os conceitos:
   - Pelagem (67 imagens)
   - Pelagem Branca (67 imagens)
   - Face de Urso Polar (67 imagens)
   - Polo Ártico (67 imagens)

16. Teste de sensibilidade da predição da classe 'urso polar' (utilizando imagens de rosto de urso como conjunto de dados de entrada para a classe) para os conceitos:
   - Pelagem (67 imagens)

- Pelagem Branca (67 imagens)
- Face de Urso Polar (67 imagens)
- Polo Ártico (67 imagens)

17. Teste de sensibilidade da predição da classe ‘urso polar’ (utilizando imagens dos conjuntos de entrada para a classe ’urso polar’ dos testes 8 e 9 combinados como conjunto de dados de entrada para a classe) para os conceitos:
    - Pelagem (67 imagens)
    - Pelagem Branca (67 imagens)
    - Face de Urso Polar (67 imagens)
    - Polo Ártico (67 imagens)
18. Teste de sensibilidade da predição da classe ‘urso polar’ (utilizando imagens dos conjuntos de entrada para a classe ’urso polar’ dos testes 8 e 9 combinados como conjunto de dados de entrada para a classe) para os conceitos:
    - Animais brancos peludos (99 imagens)
    - Lobo branco (99 imagens)

Além dos conjuntos de dados para classe e conceitos, também foram montados 20 conjuntos de dados com conceitos aleatórios, cada um com cerca de 45 imagens. A próxima seção discorre sobre a execução dos testes propostos e apresenta os resultados impingidos pela ferramenta.

## 6.2 Relatório de Execução dos Testes

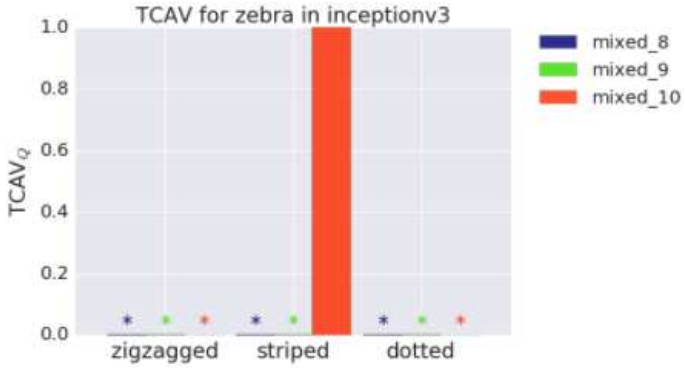
Os testes realizados para esta pesquisa apresentam os resultados de *TCAV Scores* submetidos a 10 rodadas de treinamento para o modelo *Inception 5h*.

Para Kim et al. (2017, p. 11), os resultados iniciais demonstram que depois de 500 rodadas de treinamento, dentre os conceitos testados, o único conceito relevante para a predição da classe “zebra” pela rede *Inception V3* é o conceito “listrado” (“*striped*”). As autoras não trazem o detalhamento dos resultados em seu artigo, mas é possível concluir a partir da exposição dos resultados, adaptada na figura **??**, que o conceito “listrado” obtém um *TCAV Score* igual a um, o valor máximo possível.

Já no teste realizado por esta pesquisa, exposto na figura 25, utilizando a rede treinada *Inception 5h*, após 10 rodadas de treinamento, a predição da classe “zebra” se mostrou sensível ao conceito “listrado” e também ao conceito “ziguezagueado”. Com valores de *TCAV Score* expostos na tabela 1.

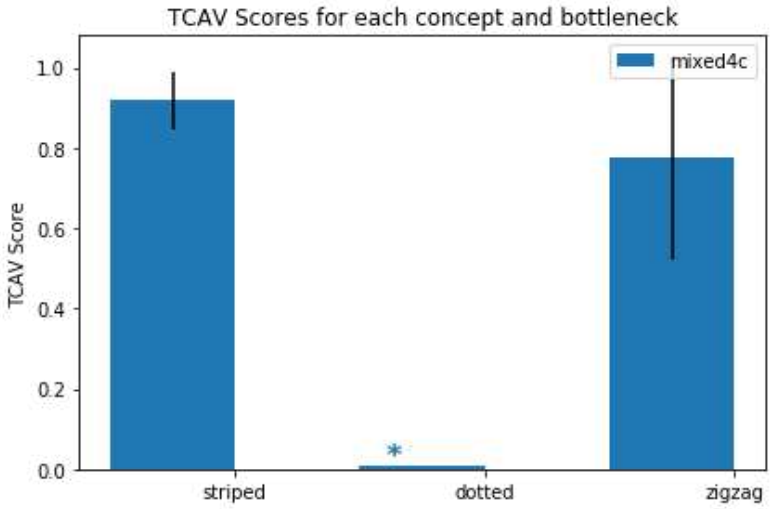
Essa pesquisa não traz os resultados para o teste de número 2 proposto na subseção 6.1, uma vez que a classe “girafa” não era uma classe conhecida pela rede *Inception 5h*, fato que não foi notado no processo de desenho dos teste. Por limitações de tempo, não foi possível encontrar dentro do conjunto de classes conhecidas pela rede alguma que pudessemos verificar na realidade ser especificamente sensível ao o conceito “ziguezagueado”.

Figura 24 – *TCAV Score* para a classe Zebra na rede *Inception V3*




Fonte: Adaptado de Kim et al. (2017)
A legenda da imagem original de Kim et al. (2017) se refere às camadas da rede *Inception V3* onde se mediu a sensibilidade da predição da classe zebra para os conceitos "ziguezagueado" (*zigzagged*), "listrado" (*striped*), , e "pontilhado" (*dotted*).

Figura 25 – Teste 1: TCAV Score para classe Zebra no modelo Inception 5h




Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 1
   target = 'zebra'
   concepts = ["striped", "dotted", "zigzag"]
```

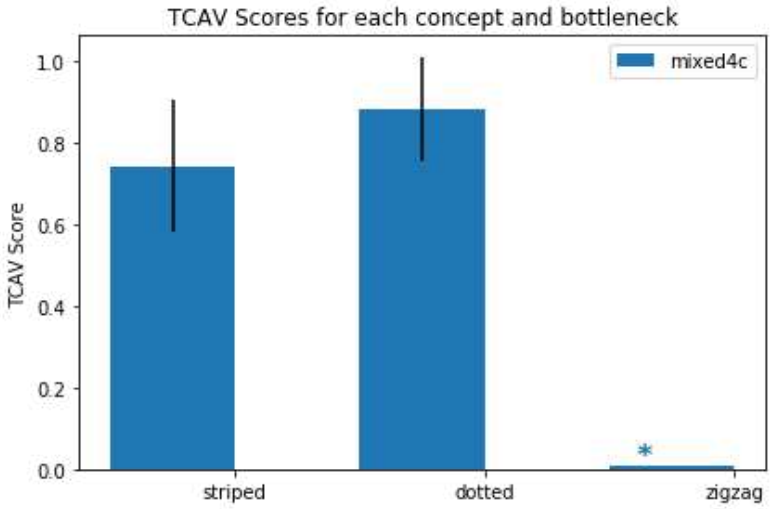
```
Class = zebra
  Concept = striped
    Bottleneck = mixed4c. TCAV Score = 0.92 (+- 0.07), random was
        0.52 (+- 0.28). p-val = 0.000 (significant)
  Concept = dotted
    Bottleneck = mixed4c. TCAV Score = 0.57 (+- 0.19), random was
        0.52 (+- 0.28). p-val = 0.592 (not significant)
  Concept = zigzag
    Bottleneck = mixed4c. TCAV Score = 0.77 (+- 0.25), random was
        0.52 (+- 0.28). p-val = 0.009 (significant)
{'mixed4c': {'bn_vals': [0.9184210526315789, 0.01,
   0.7736842105263158], 'bn_stds': [0.07202069570212638, 0,
   0.25410753062259606], 'significant': [True, False, True]}}
```

| **Classe Zebra** | | | |
|---|---|---|---|
| **Conceito** | **TCAV Score** | **p-value** | **Resultado** |
| Listrado | 0.92 | 0.00 | Significativo |
| Pontilhado | 0.57 | 0.592 | Não Significativo |
| Ziguezagueado | 0.77 | 0.009 | Significativo |

Tabela 1 – Tabela para os resultados do teste 1 exibidos na figura 25

Para o teste de número 3, os resultados expostos na figura 26 demostram que a predição da classe "Dálmata" se mostrou sensível aos conceitos "listrados" ("*striped*") e "pontilhado" ("*dotted*"). Com valores expostos na tabela 2.

Figura 26 – Teste 3: TCAV Score para classe Dalmata a no modelo Inception 5h




Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 3
target = 'dalmatian'
concepts = ["striped", "dotted", "zigzag"]
```

```
Class = dalmatian
  Concept = striped
    Bottleneck = mixed4c. TCAV Score = 0.74 (+- 0.16), random was
        0.53 (+- 0.27). p-val = 0.016 (significant)
  Concept = dotted
    Bottleneck = mixed4c. TCAV Score = 0.88 (+- 0.13), random was
        0.53 (+- 0.27). p-val = 0.000 (significant)
  Concept = zigzag
    Bottleneck = mixed4c. TCAV Score = 0.69 (+- 0.22), random was
        0.53 (+- 0.27). p-val = 0.067 (not significant)
{'mixed4c': {'bn_vals': [0.743, 0.883, 0.01], 'bn_stds':
   [0.16056462873248267, 0.12938701635017325, 0], 'significant': [
   True, True, False]}}
```

| **Classe Dálmata** | | | |
|---|---|---|---|
| **Conceito** | **TCAV Score** | **p-value** | **Resultado** |
| Listrado | 0.74 | 0.016 | Significativo |
| Pontilhado | 0.88 | 0.000 | Significativo |
| Ziguezagueado | 0.69 | 0.067 | Não Significativo |

Tabela 2 – Tabela para os resultados do teste 3 exibidos na figura 26

Para os testes 4 e 5, verificou-se a sensibilidade da predição da classe abelha para os conceitos "listrado" ("*striped*"), "flores" ("*flowers*") e "asas [de abelha]" ("*wings*"). No teste 4, com resultados expostos na figura 27, o conceito "listrado" é representado exatamente pelo mesmo conjunto de dados utilizado nos testes 1 e 2, enquanto no teste 5, na figura 28, o conjunto de dados de entrada para o conceito "listrado" (nesse caso, chamado de"*striped_2*") é formado pelo conjunto original (utilizado nos testes anteriores) e adição de imagens de listras que representam o conceito de forma mais abrangente.

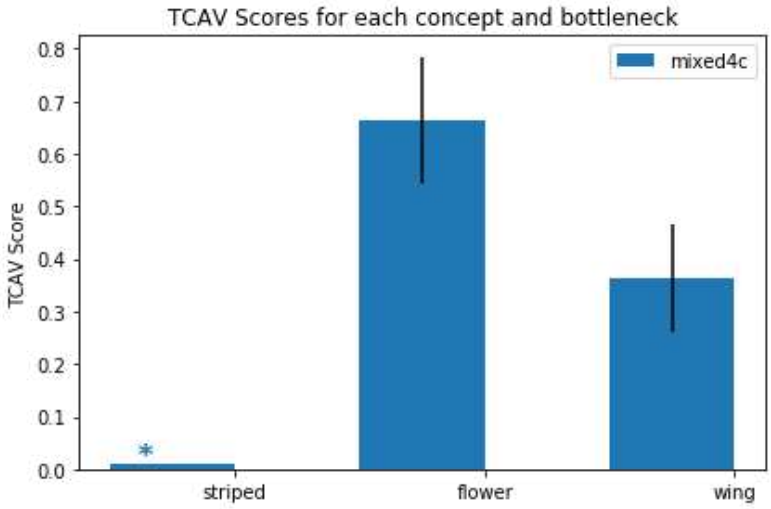
Busca-se melhor representar o conceito no que diz respeito à listras de abelhas e entender se dessa forma o conceito se torna mais relevante ou não à predição da classe abelha. Os resultados apontam um aumento de 0.03 ao TCAV *Score* total em relação ao teste 4, demonstrado na figura 27. Em ambos os testes os resultado mostram que a predição da classe abelha é sensível aos conceitos "flores" e "asas". Com valores expostos na tabela 3.

| **Classe Abelha** | | | |
|---|---|---|---|
| **Conceito** | **TCAV Score** | **p-value** | **Resultado** |
| Listrado | 0.43 | 0.203 | Não Significativo |
| Listrado 2 | 0.47 | 0.584 | Não Significativo |
| Flores | 0.67 | 0.001 | Significativo |
| Asas | 0.36 | 0.007 | Significativo |

Tabela 3 – Tabela de resultados dos testes 4 e 5 exibidos na figura 27 e 28

.

Figura 27 – Teste 4: TCAV Score para classe Abelha no modelo Inception 5h

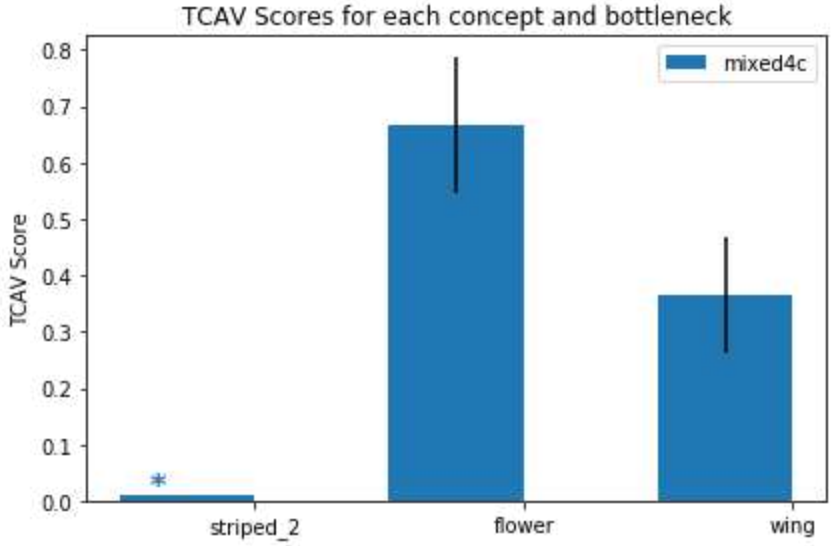



Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 4
target = 'bee'
concepts = ["striped", "flower", "wing"]
```

```
Class = bee
  Concept = striped
    Bottleneck = mixed4c. TCAV Score = 0.43 (+- 0.14), random was
        0.50 (+- 0.15). p-val = 0.203 (not significant)
  Concept = flower
    Bottleneck = mixed4c. TCAV Score = 0.67 (+- 0.12), random was
        0.50 (+- 0.15). p-val = 0.001 (significant)
  Concept = wing
    Bottleneck = mixed4c. TCAV Score = 0.36 (+- 0.10), random was
        0.50 (+- 0.15). p-val = 0.007 (significant)
{'mixed4c': {'bn_vals': [0.01, 0.665, 0.364], 'bn_stds': [0,
   0.1205197079319395, 0.10258654882585728], 'significant': [False
   , True, True]}}
```

Figura 28 – Teste 5: TCAV Score para classe Abelha a no modelo Inception 5h




Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
#Teste 5
target = 'bee'
concepts = ["striped_2", "flower", "wing"]
```

```
Class = bee
  Concept = striped_2
    Bottleneck = mixed4c. TCAV Score = 0.47 (+- 0.19), random was
        0.50 (+- 0.15). p-val = 0.584 (not significant)
  Concept = flower
    Bottleneck = mixed4c. TCAV Score = 0.67 (+- 0.12), random was
        0.50 (+- 0.15). p-val = 0.001 (significant)
  Concept = wing
    Bottleneck = mixed4c. TCAV Score = 0.36 (+- 0.10), random was
        0.50 (+- 0.15). p-val = 0.007 (significant)
{'mixed4c': {'bn_vals': [0.01, 0.665, 0.364], 'bn_stds': [0,
   0.1205197079319395, 0.10258654882585728], 'significant': [False
   , True, True]}}
```

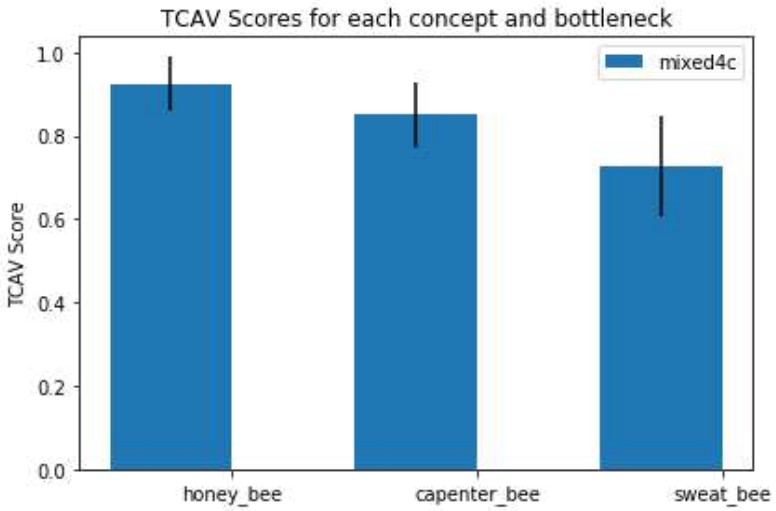
Para o teste 6, na figura 29, os resultados obtidos mostram que a predição da classe abelha se mostrou sensível aos conceitos "abelha europeia" (*"honey_bee"*), [abelha] "carpenteiro tropical" (*"carpenter_bee"*) e "abelha do suor" (*"sweat_bee"*). Na figura 30, apresentam-se os resultados para o sétimo teste, onde para além dos conceitos testados no teste 6, acrescenta-se a verificação de sensibilidade para o conceito "vespas mandarinas" (*"hornet"*). Os valores de TCAV *Score* obtidos nos testes 6 e 7 estão expostos na tabela 4.

| Classe Abelha | | | |
|---|---|---|---|
| **Conceito** | **TCAV Score** | **p-value** | **Resultado** |
| Abelha Europeia | 0.92 | 0.000 | Significativo |
| Carpenteiro Tropical | 0.85 | 0.000 | Significativo |
| Abelha do Suor | 0.73 | 0.000 | Significativo |
| Vespa Mandarina | 0.78 | 0.000 | Significativo |

Tabela 4 – Tabela para os resultados dos testes 6 e 7 exibidos respectivamente nas figuras 29 e 30

.

Figura 29 – Teste 6: TCAV Score para classe Abelha a no modelo Inception 5h

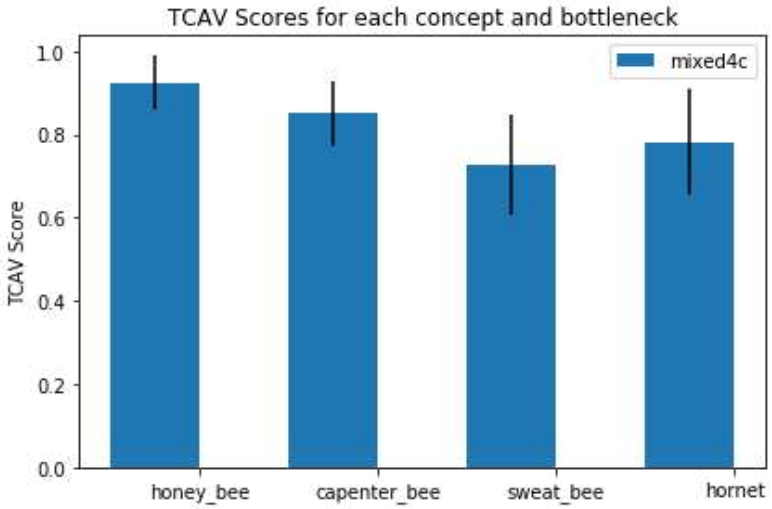



Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
#Teste 6
target = 'bee'
concepts = ["honey_bee", "capenter_bee", "sweat_bee"]
```

```
Class = bee
  Concept = honey_bee
    Bottleneck = mixed4c. TCAV Score = 0.92 (+- 0.06), random was
        0.50 (+- 0.15). p-val = 0.000 (significant)
  Concept = capenter_bee
    Bottleneck = mixed4c. TCAV Score = 0.85 (+- 0.08), random was
        0.50 (+- 0.15). p-val = 0.000 (significant)
  Concept = sweat_bee
    Bottleneck = mixed4c. TCAV Score = 0.73 (+- 0.12), random was
        0.50 (+- 0.15). p-val = 0.000 (significant)
{'mixed4c': {'bn_vals': [0.9239999999999998, 0.85,
   0.7270000000000001], 'bn_stds': [0.06406246951218786,
   0.07974960814950753, 0.12091732712891069], 'significant': [True
   , True, True]}}
```

Figura 30 – Teste 7: TCAV Score para classe Abelha a no modelo Inception 5h




Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 7
target = 'bee'
concepts = ["honey_bee", "capenter_bee", "sweat_bee", "hornet"]
```

```
Class = bee
  Concept = honey_bee
    Bottleneck = mixed4c. TCAV Score = 0.92 (+- 0.06), random was
        0.50 (+- 0.15). p-val = 0.000 (significant)
  Concept = capenter_bee
    Bottleneck = mixed4c. TCAV Score = 0.85 (+- 0.08), random was
        0.50 (+- 0.15). p-val = 0.000 (significant)
  Concept = sweat_bee
    Bottleneck = mixed4c. TCAV Score = 0.73 (+- 0.12), random was
        0.50 (+- 0.15). p-val = 0.000 (significant)
  Concept = hornet
    Bottleneck = mixed4c. TCAV Score = 0.78 (+- 0.13), random was
        0.50 (+- 0.15). p-val = 0.000 (significant)
{'mixed4c': {'bn_vals': [0.9239999999999998, 0.85,
   0.7270000000000001, 0.7809999999999999], 'bn_stds':
   [0.06406246951218786, 0.07974960814950753, 0.12091732712891069,
    0.1271573827978541], 'significant': [True, True, True, True]}}
```

Os testes de 8 a 10 verificam a sensibilidade da predição da classe urso polar ("*ice_bear*") para os conceitos "pelagem" ("*fur*"), "pelagem branca" ("*white_fur*"), "rosto de urso polar" ("*polar_bear_face*") e "polo ártico" ("*arctic_pole*").

Os resultados para o teste 8, figura 31, foram obtidos quando o conjunto de dados de entrada para a classe é composto majoritariamente de imagens de urso de corpo

inteiro. Na figura 32, os resultados do nono teste mostram que a predição da classe "urso polar", quando utilizadas imagens de entrada para a classe compostas majoritariamente de imagens da face de ursos polares. Nota-se que a sensibilidade para todos os conceitos diminuíram. Em todos os testes, todos os conceitos são classificadas como significativos à predição da classe.

O décimo teste, figura 33, realiza a mesma verificação considerando agora o conjunto de entrada para a classe enquanto a união dos conjuntos de dados de entrada para classe utilizados nos testes 8 e 9. Todos os conceitos foram considerados significativos à classe pela ferramenta e os valores obtidos para TCAV *Score* estão expostos na tabela 5.

O teste 11 (figura 34, busca identificar se animais com uma predominância da característica "pelagem branca" (identificados código como "*white_fluffy_animals*" e "*white wolf*") poderiam ser caracterizados como relevantes è predição da classe "urso polar". Ambos os conceitos não foram considerados significativos. A tabela 6 apresenta os resultados obtidos.

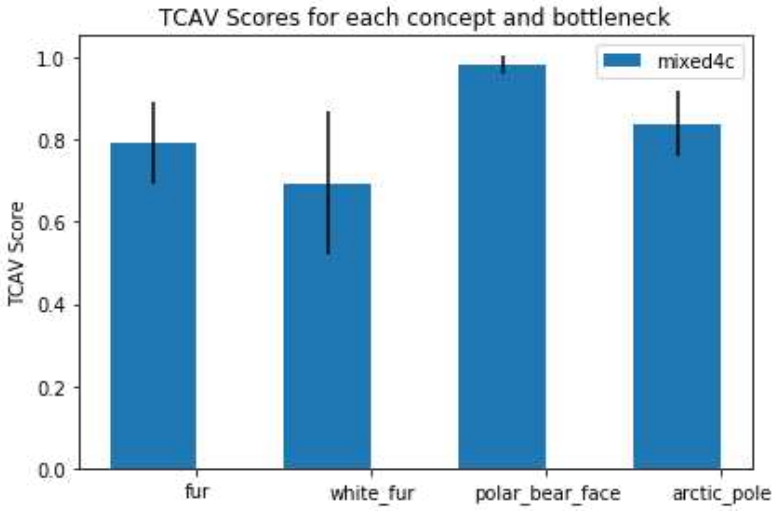
| **Urso Polar** | | | | | | |
|---|---|---|---|---|---|---|
| Imagens do conjunto de entrada focadas em: | Corpo de Urso e Cenário (A) | | Rosto de Urso (B) | | A$U$B | |
| Conceito | TCAV Score | p-value | TCAV Score | p-value | TCAV Score | p-value |
| Pelagem | 0.79 | 0.000 | 0.76 | 0.000 | 0.79 | 0.000 |
| Pelagem Branca | 0.69 | 0.014 | 0.67 | 0.016 | 0.75 | 0.001 |
| Rosto de Urso Polar | 0.98 | 0.000 | 0.95 | 0.000 | 0.98 | 0.000 |
| Polo Ártico | 0.84 | 0.000 | 0.82 | 0.000 | 0.84 | 0.000 |

Tabela 5 – Tabela de resultados dos testes 8, 9 e 10 exibidos na figura 31, 32 e 33.

| **Urso Polar** | | | |
|---|---|---|---|
| **Conceito** | **Tcav Score** | **p-value** | **Resultado** |
| Animais Brancos Peludos | 0.58 | 0.235 | Não Significativo |
| Lobo Branco | 0.63 | 0.062 | Não Significativo |

Tabela 6 – Tabela de resultados do teste 11 exibidos na figura 34.

Figura 31 – Teste 8: TCAV Score para classe Urso a no modelo Inception 5h

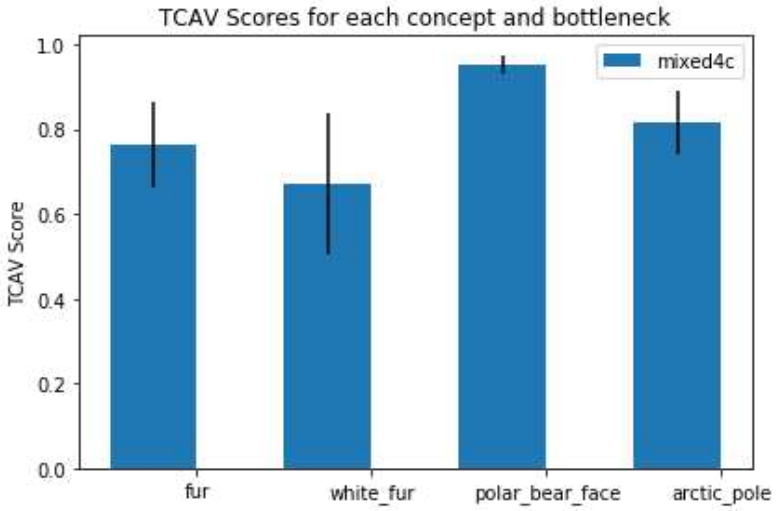



Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 8
target = 'polar_bear_body'  # as "ice bear"
concepts = ["fur", "white_fur", "polar_bear_face", "arctic_pole"]
```

```
Class = ice bear
  Concept = fur
    Bottleneck = mixed4c. TCAV Score = 0.79 (+- 0.10), random was
        0.51 (+- 0.22). p-val = 0.000 (significant)
  Concept = white_fur
    Bottleneck = mixed4c. TCAV Score = 0.69 (+- 0.17), random was
        0.51 (+- 0.22). p-val = 0.014 (significant)
  Concept = polar_bear_face
    Bottleneck = mixed4c. TCAV Score = 0.98 (+- 0.02), random was
        0.51 (+- 0.22). p-val = 0.000 (significant)
  Concept = arctic_pole
    Bottleneck = mixed4c. TCAV Score = 0.84 (+- 0.08), random was
        0.51 (+- 0.22). p-val = 0.000 (significant)
{'mixed4c': {'bn_vals': [0.792, 0.694, 0.9809999999999999,
   0.8389999999999999], 'bn_stds': [0.09897474425326896,
   0.17419529270333342, 0.021189620100417084,
   0.07917701686727026], 'significant': [True, True, True, True]}}
```

Figura 32 – Teste 9: TCAV Score para classe Urso no modelo Inception 5h

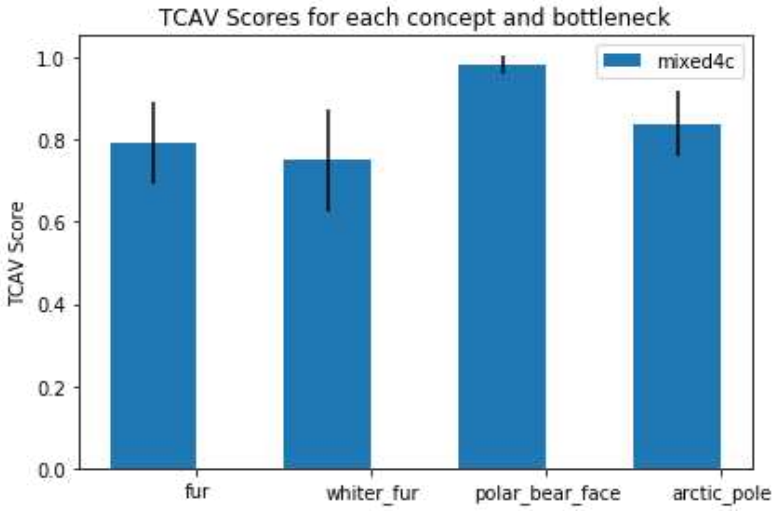



Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 9
target = 'polar_bear_face' # como "ice bear"
concepts = ["fur", "white_fur", "polar_bear_face", "arctic_pole"]
```

```
Class = ice bear
  Concept = fur
    Bottleneck = mixed4c. TCAV Score = 0.76 (+- 0.10), random was
        0.50 (+- 0.21). p-val = 0.000 (significant)
  Concept = white_fur
    Bottleneck = mixed4c. TCAV Score = 0.67 (+- 0.17), random was
        0.50 (+- 0.21). p-val = 0.016 (significant)
  Concept = polar_bear_face
    Bottleneck = mixed4c. TCAV Score = 0.95 (+- 0.02), random was
        0.50 (+- 0.21). p-val = 0.000 (significant)
  Concept = arctic_pole
    Bottleneck = mixed4c. TCAV Score = 0.82 (+- 0.07), random was
        0.50 (+- 0.21). p-val = 0.000 (significant)
{'mixed4c': {'bn_vals': [0.764, 0.67, 0.9510000000000002, 0.817],
    'bn_stds': [0.09971960689854327, 0.16751119365582706,
   0.021189620100417073, 0.07362744053679987], 'significant': [
   True, True, True, True]}}
```

Figura 33 – Teste 10: TCAV Score para classe Urso no modelo Inception 5h

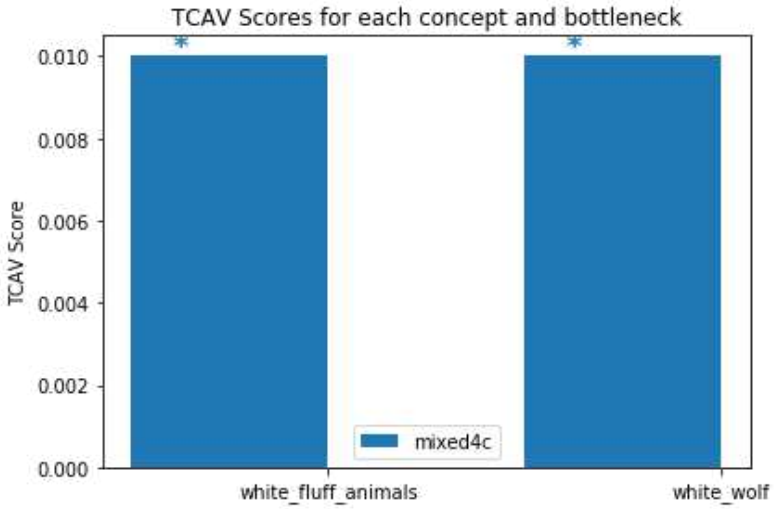



Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 10
target = 'polar_bear'  as "ice bear"
concepts = ["fur", "whiter_fur", "polar_bear_face", "arctic_pole"
    ]
```

```
Class = ice bear
  Concept = fur
    Bottleneck = mixed4c. TCAV Score = 0.79 (+- 0.10), random was
        0.51 (+- 0.22). p-val = 0.000 (significant)
  Concept = whiter_fur
    Bottleneck = mixed4c. TCAV Score = 0.75 (+- 0.13), random was
        0.51 (+- 0.22). p-val = 0.001 (significant)
  Concept = polar_bear_face
    Bottleneck = mixed4c. TCAV Score = 0.98 (+- 0.02), random was
        0.51 (+- 0.22). p-val = 0.000 (significant)
  Concept = arctic_pole
    Bottleneck = mixed4c. TCAV Score = 0.84 (+- 0.08), random was
        0.51 (+- 0.22). p-val = 0.000 (significant)
{'mixed4c': {'bn_vals': [0.792, 0.7500000000000001,
   0.9809999999999999, 0.8389999999999999], 'bn_stds':
   [0.09897474425326896, 0.12529964086141665,
   0.021189620100417084, 0.07917701686727026], 'significant': [
   True, True, True, True]}}
```

Figura 34 – Teste 11: TCAV Score para classe Urso no modelo Inception 5h




Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 11
target = 'polar_bear'  as 'ice_bear'
concepts = ["white_fluff_animals", "white_wolf"]
```

```
Class = ice bear
  Concept = white_fluff_animals
    Bottleneck = mixed4c. TCAV Score = 0.58 (+- 0.08), random was
        0.51 (+- 0.20). p-val = 0.235 (not significant)
  Concept = white_wolf
    Bottleneck = mixed4c. TCAV Score = 0.63 (+- 0.15), random was
        0.51 (+- 0.20). p-val = 0.062 (not significant)
{'mixed4c': {'bn_vals': [0.01, 0.01], 'bn_stds': [0, 0], '
   significant': [False, False]}}
```

Os testes de 12 à 17 verificam os mesmos cenários de teste descritos para os teste de 4 à 11, no entanto, o número de imagens para cada conceito é o nivelado por baixo, de modo que todos os conceitos tenham o mesmo número de imagens que o conceito com menor número de imagens.

A tabela 7, se refere aos resultados obtidos para o teste 12 (figura 35).

A tabela 8, se refere aos resultados obtidos para o teste 13 (figura 36).

A tabela 9, se refere aos resultados obtidos para o teste 13, 14 e 15 (figuras 37, figuras 38 e figuras 39). Todos os conceitos foram considerados significativos à classe pela ferramenta.

A tabela 10, se refere aos resultados obtidos para o teste 17 (figura 34). Esse foi o único cenário de teste que trouxe uma divergência na classificação de um conceito

previamente dito "não significativo", uma vez que quando verificada a sensibilidade da classe "urso polar" utilizando conjuntos de conceitos com o mesmo número de imagens, o conceito "animais brancos peludos", que teve seu número de imagens reduzido para se igualar ao tamanho em número de imagens do conceito "lobo branco", foi considerado "significativo".

| **Classe Abelha** | | | |
|---|---|---|---|
| **Conceito Normalizado** | **TCAV Score** | **p-value** | **Resultado** |
| Listrado 2 | 0.58 | 0.117 | Não Significativo |
| Flores | 0.67 | 0.001 | Significativo |
| Asas | 0.36 | 0.007 | Significativo |

Tabela 7 – Tabela de resultados dos testes 12 exibido na figura 35.

| **Classe Abelha** | | | |
|---|---|---|---|
| **Conceito Normalizado** | **TCAV Score** | **p-value** | **Resultado** |
| Abelha Europeia | 0.95 | 0.000 | Significativo |
| Carpenteiro Tropical | 0.78 | 0.000 | Significativo |
| Abelha do Suor | 0.75 | 0.000 | Significativo |
| Vespa Mandarina | 0.79 | 0.000 | Significativo |

Tabela 8 – Tabela de resultados do teste 13 exibido na figura 36.

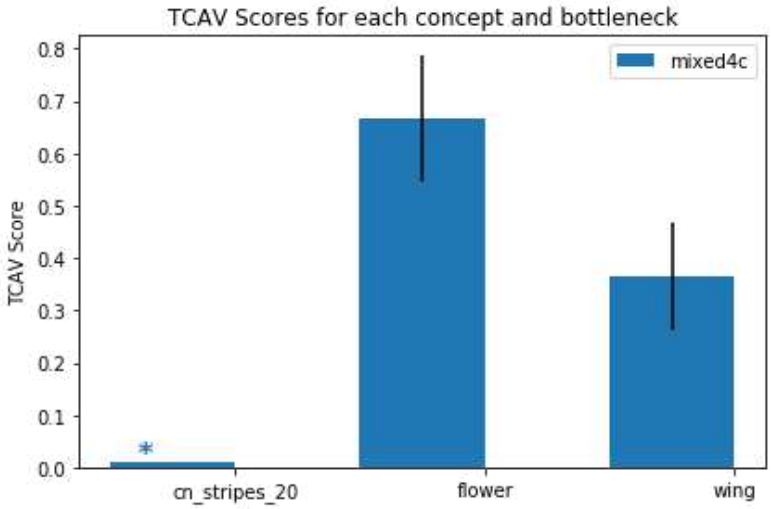
| **Classe Urso Polar** | | | | | | |
|---|---|---|---|---|---|---|
| Imagens do conjunto de entrada focadas em: | Corpo de Urso e Cenário (A) | | Rosto de Urso (B) | | A$\cup$B | |
| Conceito Normalizado | TCAV Score | p-value | TCAV Score | p-value | TCAV Score | p-value |
| Pelagem | 0.75 | 0.001 | 0.72 | 0.002 | 0.75 | 0.001 |
| Pelagem Branca | 0.80 | 0.000 | 0.77 | 0.000 | 0.80 | 0.000 |
| Rosto de Urso Polar | 0.88 | 0.000 | 0.85 | 0.000 | 0.88 | 0.000 |
| Polo Ártico | 0.88 | 0.000 | 0.85 | 0.000 | 0.88 | 0.000 |

Tabela 9 – Tabela de resultados dos testes 14, 15 e 16 exibidos nas figuras 37, 38 e 39.

| **Urso Polar** | | | |
|---|---|---|---|
| **Conceito** | **Tcav Score** | **p-value** | **Resultado** |
| Animais Brancos Peludos | 0.68 | 0.006 | Significativo |
| Lobo Branco | 0.63 | 0.062 | Não Significativo |

Tabela 10 – Tabela de resultados dos testes 17 exibidos na figura 40.

Figura 35 – Teste 12: TCAV Score para classe Abelha no modelo Inception 5h

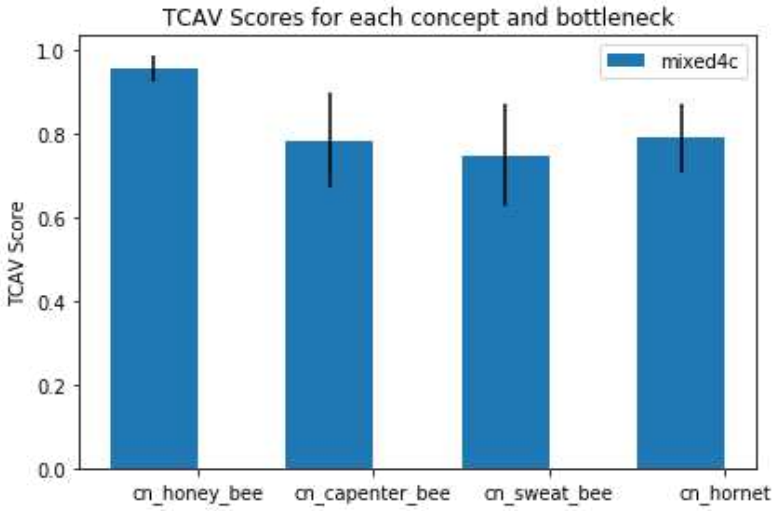



Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 12
target = 'bee'
concepts = ["cn_stripes_20", "flower", "wing"]
```

```
Class = bee
  Concept = cn_stripes_20
    Bottleneck = mixed4c. TCAV Score = 0.58 (+- 0.21), random was
        0.50 (+- 0.15). p-val = 0.117 (not significant)
  Concept = flower
    Bottleneck = mixed4c. TCAV Score = 0.67 (+- 0.12), random was
        0.50 (+- 0.15). p-val = 0.001 (significant)
  Concept = wing
    Bottleneck = mixed4c. TCAV Score = 0.36 (+- 0.10), random was
        0.50 (+- 0.15). p-val = 0.007 (significant)
{'mixed4c': {'bn_vals': [0.01, 0.665, 0.364], 'bn_stds': [0,
   0.1205197079319395, 0.10258654882585728], 'significant': [False
   , True, True]}}
```

Figura 36 – Teste 13: TCAV Score para classe Abelha no modelo Inception 5h

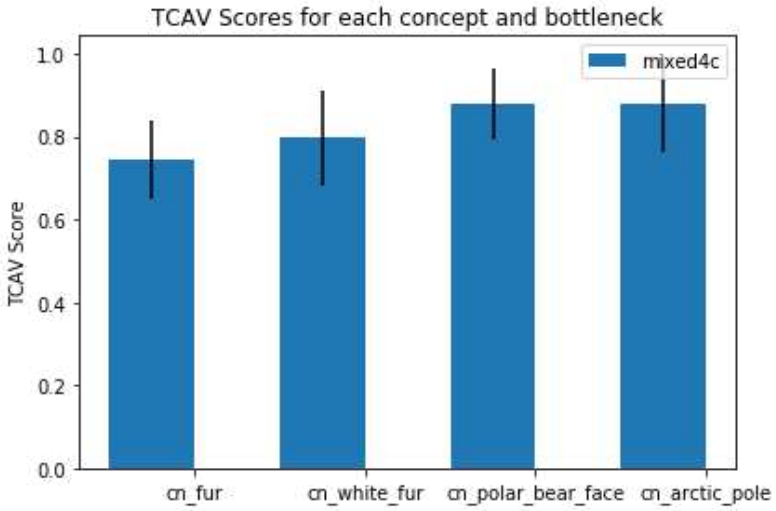



Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 13
target = 'bee'
concepts = ["cn_honey_bee", "cn_capenter_bee", "cn_sweat_bee", "
    cn_hornet"]
```

```
Class = bee
  Concept = cn_honey_bee
    Bottleneck = mixed4c. TCAV Score = 0.95 (+- 0.03), random was
        0.50 (+- 0.15). p-val = 0.000 (significant)
  Concept = cn_capenter_bee
    Bottleneck = mixed4c. TCAV Score = 0.78 (+- 0.11), random was
        0.50 (+- 0.15). p-val = 0.000 (significant)
  Concept = cn_sweat_bee
    Bottleneck = mixed4c. TCAV Score = 0.75 (+- 0.12), random was
        0.50 (+- 0.15). p-val = 0.000 (significant)
  Concept = cn_hornet
    Bottleneck = mixed4c. TCAV Score = 0.79 (+- 0.08), random was
        0.50 (+- 0.15). p-val = 0.000 (significant)
{'mixed4c': {'bn_vals': [0.954, 0.782, 0.748, 0.789], 'bn_stds':
    [0.029732137494637007, 0.11285388783732708,
    0.12383860464330178, 0.08263776376451629], 'significant': [True
    , True, True, True]}}
```

Figura 37 – Teste 14: TCAV Score para classe Abelha no modelo Inception 5h

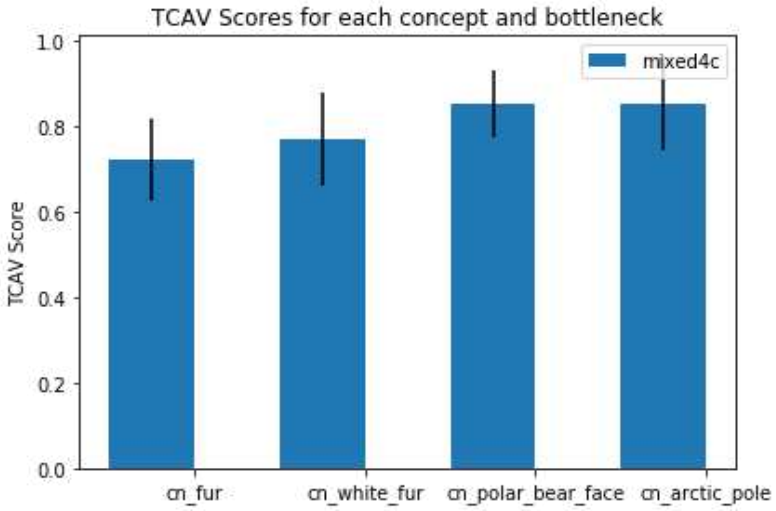



Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 14
target = 'polar_bear_body'  as ice bear
concepts = ["cn_fur", "cn_white_fur", "cn_polar_bear_face", "
    cn_arctic_pole"]
```

```
Class = ice bear
  Concept = cn_fur
    Bottleneck = mixed4c. TCAV Score = 0.75 (+- 0.10), random was
        0.51 (+- 0.22). p-val = 0.001 (significant)
  Concept = cn_white_fur
    Bottleneck = mixed4c. TCAV Score = 0.80 (+- 0.11), random was
        0.51 (+- 0.22). p-val = 0.000 (significant)
  Concept = cn_polar_bear_face
    Bottleneck = mixed4c. TCAV Score = 0.88 (+- 0.09), random was
        0.51 (+- 0.22). p-val = 0.000 (significant)
  Concept = cn_arctic_pole
    Bottleneck = mixed4c. TCAV Score = 0.88 (+- 0.12), random was
        0.51 (+- 0.22). p-val = 0.000 (significant)
{'mixed4c': {'bn_vals': [0.7460000000000001, 0.797,
   0.8779999999999999, 0.8780000000000001], 'bn_stds':
   [0.09520504188329523, 0.11489560478973945, 0.08506468127254693,
    0.116], 'significant': [True, True, True, True]}}
```

Figura 38 – Teste 15: TCAV Score para classe Abelha no modelo Inception 5h

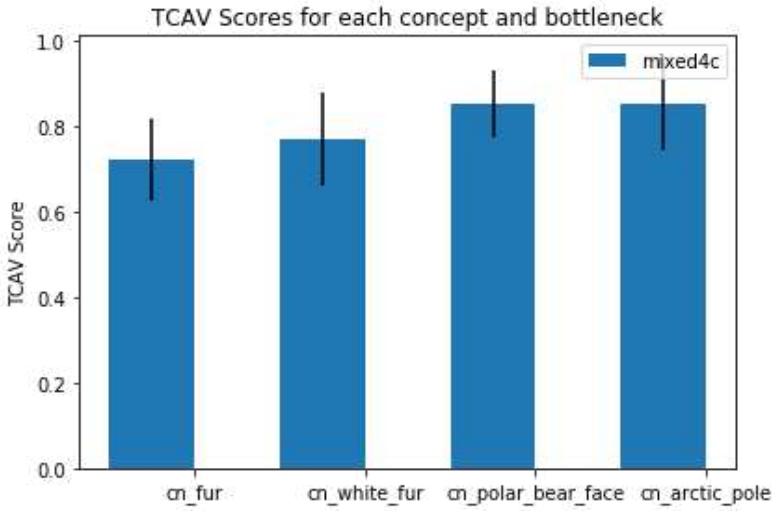



Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 15
target = 'polar_bear_body'  as ice bear
concepts = ["cn_fur", "cn_white_fur", "cn_polar_bear_face", "
    cn_arctic_pole"]
```

```
Class = ice bear
  Concept = cn_fur
    Bottleneck = mixed4c. TCAV Score = 0.72 (+- 0.10), random was
        0.50 (+- 0.21). p-val = 0.002 (significant)
  Concept = cn_white_fur
    Bottleneck = mixed4c. TCAV Score = 0.77 (+- 0.11), random was
        0.50 (+- 0.21). p-val = 0.000 (significant)
  Concept = cn_polar_bear_face
    Bottleneck = mixed4c. TCAV Score = 0.85 (+- 0.08), random was
        0.50 (+- 0.21). p-val = 0.000 (significant)
  Concept = cn_arctic_pole
    Bottleneck = mixed4c. TCAV Score = 0.85 (+- 0.11), random was
        0.50 (+- 0.21). p-val = 0.000 (significant)
{'mixed4c': {'bn_vals': [0.721, 0.7719999999999999,
   0.8530000000000001, 0.8540000000000001], 'bn_stds':
   [0.09512623192369178, 0.10916043239196153, 0.08050465825031494,
    0.1101998185116473], 'significant': [True, True, True, True]}}
```

Figura 39 – Teste 16: TCAV Score para classe Abelha no modelo Inception 5h

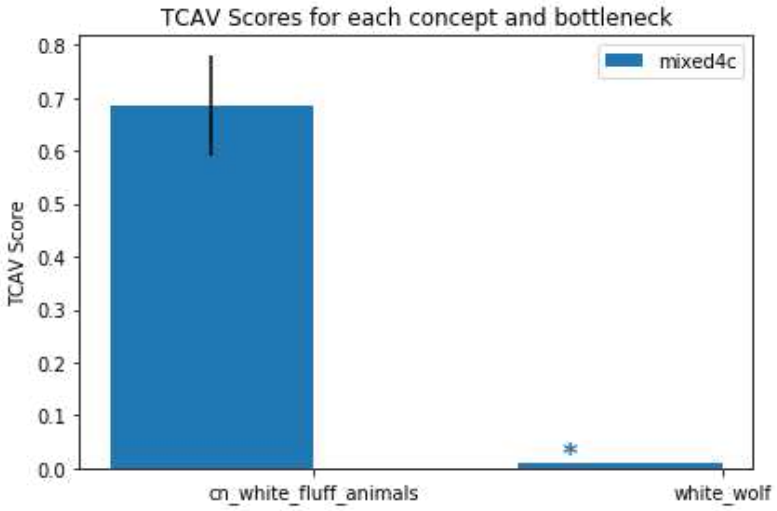



Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 16
target = 'polar_bear'
concepts = ["cn_fur", "cn_white_fur", "cn_polar_bear_face", "
    cn_arctic_pole"]
```

```
Class = ice bear
  Concept = cn_fur
    Bottleneck = mixed4c. TCAV Score = 0.75 (+- 0.10), random was
        0.51 (+- 0.22). p-val = 0.001 (significant)
  Concept = cn_white_fur
    Bottleneck = mixed4c. TCAV Score = 0.80 (+- 0.11), random was
        0.51 (+- 0.22). p-val = 0.000 (significant)
  Concept = cn_polar_bear_face
    Bottleneck = mixed4c. TCAV Score = 0.88 (+- 0.09), random was
        0.51 (+- 0.22). p-val = 0.000 (significant)
  Concept = cn_arctic_pole
    Bottleneck = mixed4c. TCAV Score = 0.88 (+- 0.12), random was
        0.51 (+- 0.22). p-val = 0.000 (significant)
{'mixed4c': {'bn_vals': [0.7460000000000001, 0.797,
   0.8779999999999999, 0.8780000000000001], 'bn_stds':
   [0.09520504188329523, 0.11489560478973945, 0.08506468127254693,
    0.116], 'significant': [True, True, True, True]}}
```

Figura 40 – Teste 18: TCAV Score para classe Urso no modelo Inception 5h




Fonte: Autoria própria.
Resultados obtidos a partir de 10 rodadas de treinamento.

```
# Teste 17
target = 'polar_bear'  as 'ice_bear'
concepts = ["cn_white_fluff_animals", "white_wolf"]
```

```
Class = ice bear
  Concept = cn_white_fluff_animals
    Bottleneck = mixed4c. TCAV Score = 0.68 (+- 0.09), random was
        0.51 (+- 0.20). p-val = 0.006 (significant)
  Concept = white_wolf
    Bottleneck = mixed4c. TCAV Score = 0.63 (+- 0.15), random was
        0.51 (+- 0.20). p-val = 0.062 (not significant)
{'mixed4c': {'bn_vals': [0.6835051546391753, 0.01], 'bn_stds':
   [0.09494056181313644, 0], 'significant': [True, False]}}
```

A seção seguinte inicia a análise dos resultados, inaugarando a discussão que se prolonga até a conclusão da presente pesquisa e que visa atender os objetivos propostos pela mesma.

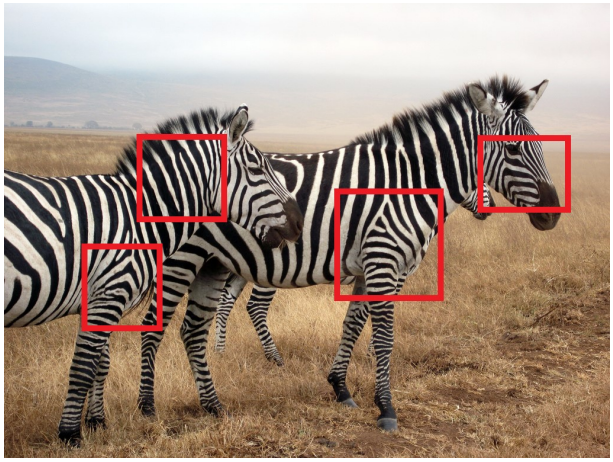
## 6.3 Discussão dos Resultados

O teste 1 serve ao propósito de nos ajudar a enxergar de que maneira as diferenças entre modelo e número de rodadas de treinamento da ferramenta TCAV impactam nos resultados para um mesmo cenários de teste, nesse caso, o de predição da classe "zebra". Não é possível fazer alguma análises acerca das diferenças de arquitetura entre os modelos utilizados em cada caso (*Inception* 5h, no caso dessa pesquisa, e *Inception* V3 em Kim et al.

(2017, p.11)), uma vez que a documentação para o modelo *Inception* 5h não foi divulgada pelos desenvolvedores do *Google*. Em pesquisa sobre o modelo *Inception* 5h em fórums de desenvolvedores na internet, encontramos alguns membros dessas comunidades sugerindo que se trate de uma versão mais simples do modelo *Inception* V3 ou que equivalha à primeira geração do mesmo (*Inception* V1). Por se tratar de fontes não oficiais, essa pesquisa suspende a análise nesse sentido e busca compreender as diferenças entre os resultados obtidos num nível mais alto de abstração.

O resultado de teste 1 nos permite sugerir que quanto maior o número de rodadas de treinamento, conceitos visualmente mais proeminentes nas imagens de dada classe tendem ser classificados como "significativo", enquanto conceitos que existem de forma mais sutil passam a ter menos relevância. O que nos leva à essa reflexão, é que enquanto os resultados para Kim et al. (2017, p.11) demonstram que apenas o conceito "listrado" é relevante para a predição da classse "zebra" (após 500 rodadas de treinamento), o teste 1, após 10 rodadas de treinamento, traz como "significativo" ambos conceitos "listrado" e "ziguezague". Embasa-se essa premissa na ideia de que o conceito "ziguezague" não é irrelevante à classe "zebras". Vejamos, o significado de "ziguezague" é "linha ou série de linhas quebradas, flexuosas, que formam alternadamente ângulos agudos e obtusos, salientes e reentrantes"[2] o que pode ser facilmente observado em uma zebra comum (figura 41).

Figura 41 – Exemplo do conceito "zigue-zague" em imagem exemplo para a classe "Zebra"



Fonte: Adaptado de:
https://memoria.ebc.com.br/sites/_portalebc2014/files/atoms_image/deu_zebra.jpg
A imagem apresenta em destaque vermelho os pontos na padronagem de zebras comuns que conferem com a definição do conceito zigue-zague.

Essa primeira análise possibilitada a partir dos resultados do teste 1, nos ajuda a entender que o número elevado de rodadas de treinamento não necessariamente produz um resultado mais apurado que um teste com menos rodadas de treinamento. Se o TCAV se

[2]Definição de *Oxford Languages*

propõe a "quantificar o grau no qual um conceito definido pelo usuário é importante para o resultado de uma classificação" (KIM et al., 2017), e se com 10 rodadas de treinamento a ferramenta consegue identificar relação entre conceito e classe para um conceito menos evidente (zigue-zagueado) e com 500 rodadas de treinamento a ferramenta não consegue enxergar essa relação, entendemos então que a ferramente está propensa a ignorar conceitos em menor evidência na realidade, mas que isso não necessariamente significa a classe não seja sensível ao conceito. O que no caso da classificação de uma zebra não se torna um problema de desdobramentos sociais grave, mas que se pensarmos na utilização da ferramente em outros casos pode representar a invisibilização de conceitos velados pela normatividade.

O teste 3, em contrapartida, nos permite visualizar como poucas rodadas de treinamento podem permitir que conceitos não verificados pela percepção humana sobre a classe na realidade sejam possivelmente significativos para o modelo a classificar corretamente. Nesse caso, a ferramenta considerou que a classe "dalmáta" além de ser sensível ao conceito "pontilhado", o que era um resultado esperado, também considerou a predição da classe sensível ao conceito "listrado". Esse resultado nos ajuda a levantar algumas questões, como por exemplo, "existe um número de rodadas de treinamento que seja ideal?" , no sentido de que, como no teste de Kim et al. (2017, p.11) verificamos a exclusão de um conceito que pode ser verificado na classe (ainda que de forma sutil) e agora, no teste 3 verificamos que um conceito de relação alheia à percepção humana quando obervando a classe foi considerado significativo para sua predição na rede neural.

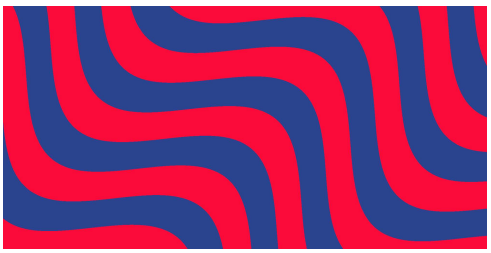
Outra questão que surge é "o que se considera enquanto "listrado" na percepção humana é o mesmo que para uma rede neural no processo de tomada de decisão uma vez que os estados internos da mesma não nos são conhecidos?", talvez em algum nível no processo de classificação de "dalmátas" o conceito "listrado" seja relevante para a predição da classe, mas não no nível de abstração conceitual que um ser humano olhando para a imagem verificaria essa relação. Essa pesquisa entende como problemático quantificar o grau da relevância de conceitos em um processo de tomada de decisões que não se pode ainda verificar como funciona internamente.

Os testes 4 e 5 exploram a predição da classe "abelha" para os conceitos "listrado", "flores" e "asas [de abelha]" A primeira análise possível a partir dos resultados desses dois testes é a de entender como a variedade de representação no conjunto dos conceitos disponibilizados para a ferramenta impacta o valor final do TCAV *Score*. No teste 4, o conceito "listrado" é representado pelo mesmo conjunto utilizado nos testes anteriores, que foi disponibilizado por Kim et al. (2017) junto ao código fonte da ferramenta enquanto exemplo para o teste com a classe "zebra" (apresentado em 24). Já no teste 5, esse mesmo conjunto de dados foi atualizado com a adição de imagens de listras mais diversas e também algumas imagens mais próximas em cor e textura do como se verifica a partir da percepção humana as listras em algumas espécies de abelhas. Nos dois casos, o conceito não foi

classificado enquanto significativo para a predição da classe, no entanto, nota-se que houve sim um aumento no TCAV *Score* para o conjunto onde havia maior representatividade.

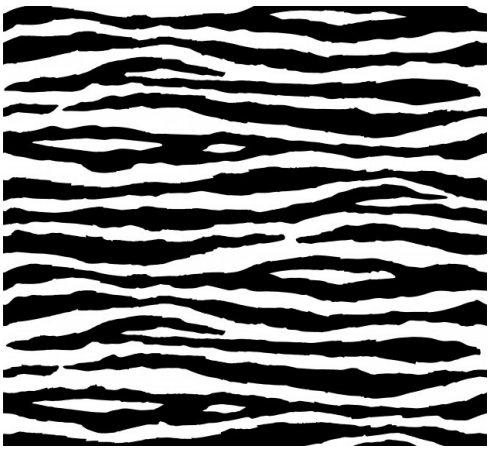
Esse cenário expõe mais uma fragilidade na construção da relação entre classe e conceito pela ferramente TCAV, uma vez que o que é disponibilizado para representar o conceito para a ferramenta depende do que quem está formulando esse conjunto o entende como e pode também levar a resultados inconsistentes com a realidade. Por exemplo, as figuras 42 e 43 todas representam o conceito listrado, agora imaginemos que o conjunto de imagens para o conceito pode ser formado apenas por um tipo das listras abaixo, o que impactaria como esse conceito seria classificado para alguma classe que se relaciona com algum outro tipo de listra. Desse modo, os resultados obtidos pelo TCAV se mostram facilmente manipuláveis.

Figura 42 – Exemplo 1 de possível representação do conceito “listrado”



Fonte: <https://andreashackel.de/assets/2017-10-03-stripes-shader-1/final.png>

Figura 43 – Exemplo 2 de possível representação do conceito “listrado”



Fonte: <https://img.freepik.com/free-vector/zebra-stripes-black-white-pattern-seamless_10083-198.jpg?size=626&ext=jpg>

As imagens ofertadas para representar a classe, também são uma escolha de quem utiliza a ferramenta, no caso do teste para a classe “abelhas”, foram utilizadas imagens de todas as 3 espécies de abelhas analisadas enquanto conceitos no teste 6 e 7, desse modo, algumas continham listras (como por exemplo, as imagens para a abelha-europeia) e outras

não (como as imagens para a abelha carpinteiro-tropical). É possível supor que caso essa pesquisa se valesse apenas de imagens de espécies de abelhas com listras, talvez o conceito "listrado" fosse considerado significativo, uma vez que o resultado se pauta em um teste de hipótese nula que verifica justamente a probabilidade de não se verificar o efeito nos dados examinados.

Uma vez que a classe abelha conhecida pelo modelo *Inception* 5h não específica à qual espécie a classe se refere, o teste 6 verifica qual entre 3 espécies de abelhas é mais relevante para a classificação e se todas são significativas. O resultado positivo para todos os conceitos testados, nos levou ao teste 7, que verifica se a espécie de inseto "vespa mandarina", seria considerada um conceito significativo na classificação da classe abelha. O resultado obtido não só foi positivo, como obteve um TCAV *Score* maior do que a espécie "abelha do suor", sendo considerado mais relevante para a classificaçao da classe "abelha".

O resultado do teste 7 evidencia que a ferramenta TCAV pode fazer relações entre classe e conceito que no entendimento humano são equivocadas. Isso porque a partir da percepção humana vespas mandarinas e abelhas em geral são animais distintos, inclusive, vespas mandarinas são predadores naturais das abelhas. Esse resultado nos leva a questionar se vespas são realmente relevantes na hora do modelo classificar a imagem de uma abelha ou se existe apenas uma conveniência visual que leva o TCAV em seu processo de ativação de vetores a considerar que vespas mandarinas são relevantes. E se não são realmente relevantes para a classificação, sendo apenas o caso de um falso positivo, então qualquer uma das outras conceituações são passíveis do mesmo questionamento. Se torna ainda mais complexa a questão, uma vez que não é possível validar o método nem destrinchar o processo de tomada de decisão de modelo.

Os testes de 8 a 10 trazem para verificação a predição da classe "urso polar" para 4 conceitos ("pelagem", "pelagem branca" , "rosto de urso polar" e "polo ártico"), onde todos foram considerados significativos. O teste 8 verifica a predição da classe para esses conceitos com imagens de ursos polares onde aparece o corpo inteiro do animal e o fundo do cenário. O teste 9 verifica a predição da classe para esses conceitos com imagens de ursos polares onde aparece o rosto do animal. E o teste 10 verifica a predição da classe para esses conceitos com da união do conjunto de imagens de ursos polares utilizados nos dois testes anteriores. A ideia é entender o que esses resultados para os mesmos conceitos nos dizem sobre a influência do conjunto de dados de entrada para a classe no resultado da verificação para os dados conceitos.

O conceito "pelagem" tem um resultado menor no teste 9, onde as imagens da classe são apenas de rosto de urso. Para os testes 8 e 10 o valor se matém constante. O conceito "pelagem branca" oscila seu valor, diminuindo um pouco no teste 8 (onde temos imagens com mais cenário) e aumentando consideravelmente no teste 10 (onde o conjunto de imagens de rosto de urso é adicionado ao conjunto de urso e cenário). O conceito "rosto

de urso polar" ironicamente obtém seu menor resultado no teste 9, onde o conjunto de dados de entrada para a classe é composto de imagens de rosto de urso polar. O conceito "polo ártico" obtém seu menor resultado no teste 9, onde o enfoque das imagens é no rosto dos ursos polares.

O teste 11 verifica a sensibilidade da classe "urso polar" para os conceitos "animais brancos peludos" e "lobo branco", os dois conceitos foram caracterizados enquanto não significativos à classe.

Os testes 12 à 17 verificam os testes anteriores (4 à 11) utilizando a mesma quantidade de imagens para todos os conceitos. No teste 12, os conceitos "asas de abelha" e "flores" tiveram seus conjuntos de dados normalizados para com o tamanho do conjunto "listrado". O resultado nos trouxe uma pequena variação no TCAV *Score* do conceito "listrado", o qual se mostrou um pouco maior nessa rodada de testes, mas permanece classificado enquanto não significativo.

No teste 12 e 13, os conceitos "abelha européia", "abelha carpinteira" e "vespa mandarinas" tiveram seus conjuntos de dados reduzidos para se equivaler ao tamanho do conjunto de dados do conceito "abelha do suor". Todos os conceitos se mantiveram classificados enquanto significativos à predição da classe. A alteração no valor do TCAV *Score* mais significativa, foi para o conceito "abelha carpinteira" que diminuiu em 0,08.

Nos testes 15, 16 e 17, os conceitos foram reduzidos ao número de imagens para o conceito "rosto de urso polar". Todas as classificações se mantiveram como significativas à predição da classe. O conceito "pelagem branca" e "face de urso" foram os que obtiveram maior variação.

O teste 17 verifica a sensibilidade da classe "urso polar" para os conceitos apresentados no teste 11, com a alteração de tamanho de conjunto de imagens do conceito "animais brancos peludos", de modo que tivesse o mesmo número de imagens que o conceito "lobo branco". Nesse cenário, diferentemente do teste original onde nenhum dos conceitos verificados foi considerado significativo, agora o conceito foi considerado significativo para a previsão da classe.

Os testes realizados por essa pesquisa com conceitos com mesmo número de imagens no conjunto de dados não nos permitiu observar nenhuma tendência que seguisse a um padrão. Não é possível afirmar com clareza se e como o número de imagens para um dado conceito influencia no seu TCAV *Score* e *p-value*.

A seguir aparesentamos em duas tabelas os cenários de testes propostos (classe e conceitos verificados e, ao lado de cada conceito entre parenteses o tamanho do conjunto de imagens que o representa), os resultados observados ("sim", se significativo e "não", se não significativo de acordo com o método na ferramenta TCAV) e um breve comentário sobre a análise proposta por essa pesquisa possibilitada a partir de cada teste. Um olhar mais aprofundado acerca do contexto e conteúdo desses resultados será apresentado na conclusão dessa pesquisa, no capítulo seguinte.

No próximo capítulo, essa pesquisa aproxima-se de seu fim e constrói sua conclusão a partir de conexão das percepções acerca dos resultados analisados com os objetivos propostos no ínicio desse trabalho, sob a luz das motivações humanísticas que suscitaram essa pesquisa em seu processo embrionário.

Tabela 11 – Resultados Comentados dos Testes de 1 a 11

| | **Classe** | **Hipótese e Classificação** | | **Sobre a Classificação** |
|---|---|---|---|---|
| 1 | Zebra | Listras (50);<br>Pontilhado (50);<br>Ziguezague (50); | Sim;<br>Não;<br>Sim; | Com mais rodadas de treinamento,<br>apenas conceitos visualmente mais<br>proeminentes nas imagens da classe<br>tendem ser considerados "significativo". |
| 2 | Girafa | | - | Teste não realizado, pois o modelo Inception<br>5h não conhecia a classe<br>escolhida para o teste. |
| 3 | Dálmata | | Sim;<br>Sim;<br>Não; | Poucas rodadas de treinamento podem permitir<br>que conceitos não verificados pela percepção<br>humana sobre a aparência visual da classe na<br>realidade sejam possivelmente significativos<br>para o modelo a classificar corretamente. |
| 4 | Abelha | Listras (50);<br>Flores (306);<br>Asas de Abelha (58); | Não;<br>Sim;<br>Sim; | É possível supor que caso essa pesquisa se valesse apenas<br>de imagens de espécies de abelhas com listras,<br>talvez o conceito "listrado" fosse considerado significativo,<br>uma vez que o conjunto Listras 2 (representado de forma mais abrangente)<br>obteve maior TCAV Score do que o conjunto Listras. |
| 5 | Abelha | Listras 2 (150);<br>Flores (306);<br>Asas de Abelha (58); | Não;<br>Sim;<br>Sim; | |
| 6 | Abelha | Abelha-europeia (159);<br>Abelha carpinteiro tropical (80);<br>Abelha do suor (77); | Sim;<br>Sim;<br>Sim; | O resultado positivo para todos os conceitos<br>testados nos levou ao teste 7. |
| 7 | Abelha | Abelha-europeia (159);<br>Abelha carpinteiro tropical (80);<br>Abelha do suor (77);<br>Vespa-mandarina (257); | Sim;<br>Sim;<br>Sim;<br>Sim; | Vespas são realmente relevantes para modelo<br>classificar uma abelha corretamente?<br>Ou existe apenas uma conveniência visual que<br>leva o TCAV a considerar que vespas<br>mandarinas são relevantes a classe? |
| 8 | Urso Polar (urso inteiro e cenário) | Pelagem (333);<br>Pelagem Branca (85);<br>Face de Urso Polar (67);<br>Cenário Ártico (225); | Sim;<br>Sim;<br>Sim;<br>Sim; | Quais cenários podemos identificar a partir da<br>verificação de diferentes dados de entrada<br>para a mesma classe e mesmos conceitos?<br>O conceito "rosto de urso polar" obtém seu<br>menor resultado no teste 9, onde o conjunto<br>de dados de entrada para a classe é composto<br>de imagens de rosto de urso polar. O conceito "polo<br>ártico" obtém seu menor resultado no teste 9,<br>onde o enfoque das imagens é no rosto dos ursos polares. |
| 9 | Urso Polar (rosto) | | | |
| 10 | Urso Polar (urso inteiro com cenário e rosto de urso) | | | |
| 11 | Urso Polar (urso inteiro e cenário e rosto de urso) | Animais Brancos Peludos (313);<br>Lobo Branco (99; | Não;<br>Não; | Ver teste 18 na tabela **??**. |

Tabela 12 – Resultados Comentados

| | **Classe** | **Hipótese e Classificação** | | **Sobre a Classificação** |
|---|---|---|---|---|
| 12 | Abelha | Listras II (50);<br>Flores (50);<br>Asas de Abelha (50); | Não;<br>Sim;<br>Sim; | Os conceitos tiveram seus conjuntos de dados reduzidos para se equivaler ao tamanho do conjunto de dados do conceito "abelha do suor". Todos os conceitos se mantiveram classificados enquanto significativos à predição da classe. A alteração no valor do TCAV Score mais significativa foi para o conceito "abelha carpinteira" que diminuiu em 0,08. |
| 13 | Abelha Abelha | Abelha-europeia (77);<br>Abelha carpinteiro tropical (77);<br>Abelha do suor (77);<br>Vespa-mandarina (77); | Sim;<br>Sim;<br>Sim;<br>Sim; | |
| 14 | Urso Polar (urso inteiro e cenário) | Pelagem (67);<br>Pelagem Branca (67);<br>Face de Urso Polar (67);<br>Cenário Ártico (67); | Sim;<br>Sim;<br>Sim;<br>Sim; | Nos testes 15, 16 e 17, os conceitos foram reduzidos ao número de imagens para o conceito ~"rosto de urso polar". O conceito "pelagem branca" e "face de urso" foram os que obtiveram maior variação no TCAV Score. |
| 15 | Urso Polar (rosto de urso) | | | |
| 16 | Urso Polar (urso inteiro e cenário e rosto de urso) | | | |
| 17 | Urso Polar (urso inteiro e cenário e rosto de urso) | Animais Brancos Peludos (99);<br>Lobo Branco (99); | Sim;<br>Não; | Nesse cenário, diferentemente do teste original onde nenhum dos conceitos verificados foi considerado significativo, agora o conceito "animais brancos peludos" que teve seu tamanho reduzido) foi considerado significativo para a previsão da classe. |

# 7 Conclusão

Neste trabalho foi apresentada a ferramenta TCAV a partir de uma análise do artigo de apresentação do método por Kim et al. (2017) e da verificação empírica da mesma. Essa pesquisa se propôs a investigar quais os limites conceituais da relação construída para classe e conceito pelo método na ferramenta TCAV. Para tanto essa pesquisa se valeu de objetivos específicos que nos ajudam a enxergar os caminhos para uma resposta a essa investigação.

O primeiro objetivo específico dessa pesquisa almejava verificar a influência que tipos de cenários distintos podem provocar no funcionamento da ferramenta. Os resultados dos testes realizados (apresentados no capítulo 6) nos permitiram observar alguns comportamentos da ferramenta no processo de formulação das relações entre classe e conceito, de modo que é possível dizer que o uso do TCAV pode produzir resultados que se enquadram nos seguintes casos:

1. Divergentes para a mesma classe e conceito – como verificado a partir dos resultados dos testes para a classe "urso polar" e o conceito "animais peludos brancos" (os resultados para esses testes estão expostos nas figuras 34 e 40 e tabelas 6 e 10);
2. Fruto de conveniência visual – no sentido de garantir um grau de sensibilidade maior a um conceito visualmente parecido com a classe, mas que não necessariamente é verdadeiramente relevante à predição dessa ou se relaciona com a classe na realidade, como verificado nos testes que quantificam a sensibilidade da predição da classe "abelha" ao conceito "vespas mandarinas" e outras espécies de abelhas (resultados expostos nas figuras 30 e 37 e tabelas 4 e 7);
3. Limitados – de modo que os resultados obtidos a partir do uso da ferramenta TCAV são limitados ao domínio de conhecimento sobre classe e conceito da pessoa que a utiliza. O valores produzidos pelo TCAV dependem do conteúdo escolhido para representar classe e conceito, por exemplo, nos testes para a classe "abelha" os resultados obtidos estão limitados ao meu próprio conhecimento sobre abelhas. Assim, qualquer não-especialista de domínio que utilize a ferramenta vai produzir resultados limitados ao seu entendimento individual sobre as classes e conceitos que se pretendem testar.
4. Incompletos – pela definição de incompletude de Doshi-Velez e Kim (2017, p. 3) apresentada na sessão 2.3 sobre Interpretabilidade, a ferramenta TCAV produz resultados que por si só também requerem uma explicação de seu processo interno, uma vez que os resultados produzidos pela ferramenta podem estar sendo derivados de conjunto que podem conter viéses não considerados a priori;
5. Facilmente manipuláveis – uma vez que a ferramenta baseia todo o seu funcionamento em conjuntos de imagem disponibilizados por quem à utiliza e que não

necessariamente representam verdadeiramente classe e/ou conceito;

O processo de verificação empírica da ferramenta, também nos permitiu observar algumas questões acerca do discurso sobre a ferramenta em Kim et al. (2017), como proposto no segundo objetivo específico dessa pesquisa. Os autores afirmam que o desenvolvimento do TCAV foi visado a partir de alguns objetivos, o primeiro deles referente à "acessibilidade". De acordo com os autores, esse objetivo determinava que a utilização do TCAV fosse possível por pessoas com "pouco ou nenhum" conhecimento em aprendizagem de máquina. Sobre esse objetivo, a presente pesquisa discorda da escolha de palavras utilizadas para descrever a realidade da ferramenta, inclusive, o primeiro desafio que se impôs ao desenvolvimento dessa pesquisa foi o tempo dedicado à compreensão do construto computacional.

Os requerimentos técnicos para a instalação da ferramenta se mostraram uma barreira à sua utilização de forma "acessível", de modo que o tempo gasto nesse primeiro contato com o TCAV superou as expectativas iniciais e subverteu a ideação inicial dessa pesquisa de percorrer uma proposta de análise mais profunda sobre o método pos trás da ferramenta e seus desdobramentos. Para realizar os testes propostos foi primeiro necessário conseguir utilizar a ferramenta, o que não foi trivial e despendeu inúmeras horas para a correta instalação e versionamentos de todos os requerimentos para a execução do código.

Uma vez instalada a ferramenta, a primeira máquina utilizada para tanto não detinha capacidade computacional necessária de processamento em paralelo para realizar testes significativos[1], sendo capaz apenas de 3 rodadas de treinamento para cada conceito. Assim, o tempo dedicado à realização dessa pesquisa precisou novamente ser empregado no processo de instalação da ferramenta em uma segunda máquina com maior capacidade de processamento e que ainda assim não era capaz de realizar mais do que 10 rodadas de treinamento.

Desse modo, ainda que o código fonte da ferramenta esteja disponível para livre utilização e apresente breve instruções de uso para cada parte do código executável, a pessoa sem ou com pouco conhecimento sobre aprendizagem de máquina será talvez capaz de a instalar e utilizar, no entanto estará completamente alienada pelos seus resultados, uma vez que não será capaz de compreender minimamente o funcionamento do TCAV. Assim, essa pesquisa considera problemática a apropriação do termo "acessibilidade" para descrever a ferramenta.

Esse aspecto de potencial alheamento que se caracteriza na ferramenta a partir do questionamento da premissa de que ela é acessível àqueles com pouco ou nenhum conhecimento em aprendizagem de máquina também se verifica no sentido oposto, uma vez que quando utilizada por quem a conhece e entende o seu funcionamento, os resultados impingidos na ferramenta se tornam facilmente manipuláveis e possivelmente alienantes.

Essa possibilidade de utilização alienada e alienante da ferramenta toca também

[1] De acordo com os desenvolvedores, em comentário no código fonte da ferramenta, é possível obter resultados relevantes a partir de 10 rodadas de treinamento.

um segundo objetivo definido em Kim et al. (2017), a capacidade do método de produzir quantificações globais, ou seja, capaz de produzir uma relação classe e conceito baseada em todos os exemplos providos e não apenas em entradas específicas. A problemática nesse sentido é que a quantificação global além de ser um método de alta aproximação, ou seja pouco específico, ele acontece a partir dos dados providos por quem utiliza a ferramenta e esses dados estão limitados ao conhecimento dessa pessoa sobre a hipótese em teste. Assim a construção dessa quantificação global para a relação classe e conceito pode ser um instrumento que reforça o potencial de alheamento na ferramenta, especialmente quando essa é completamente desprovida de de estratégias contra uso malicioso.

As conclusões possibilitadas até aqui nos direcionam no sentido de verificar a partir da análise despendida nessa pesquisa se a ferramenta realmente funciona como um método para interpretação dos estados internos de uma rede neural, como é apresentada pelos autores. A definição de interpretabilidade em RN apresentada na seção 2.3 é "dar um sentido humano à" (DOSHI-VELEZ; KIM, 2017, p.3), e é especialmente necessária quando existe um viés intangível no conjunto de saída modelo que caracteriza uma incerteza quantificável ou uma incerteza por "incompletude".

Ou seja, segundo as autoras, é necessário colocar em termos humanos o esquema interno de uma rede neural quando as saídas do modelo são incertas pelos valores que a produzem ou quando carregam em si questões (éticas, de segurança, etc) que apontam para um "buraco na formalização do problema". No entanto, fica evidente a partir dos cenários identificados acima nessa conclusão que as quantificações providas pela ferramenta TCAV podem por si só serem incompletas, uma vez que, ainda que se teste uma hipótese que verifica todos os conceitos possíveis, conhecidos e desconhecidos, para predição de uma classe, o que poderia talvez gerar um entendimento do esquema interno da RN, ainda assim os resultados que se apresentam carregam em si os viéses que se imprimem no conjunto de dados no momento do teste.

Ademais, matematicamente, o cálculo do TCAV Score que quantifica a relação de importância de um conceito para uma dada classe, nada mais é do que o cálculo da média dos valores de TCAV Score obtidos para cada entrada do conjunto, reforçando o caráter de "alta aproximação" de uma quantificação global, ao mesmo tempo que se distancia do sentido pretendido à ferramenta (o de interpretar), porque não da um sentido humano ao estado interno da RN, mas se limita a buscar um sentido computacional nas percepções humanas conceitualizáveis sobre aquela classe.

Além das limitaçõe citadas acima, é necessário reconhecer também que não é possível verificar na realidade se os resultados produzidos pelo uso ferramenta TCAV são fidedignos ao real esquema interno do modelo no processo de classifição, porque não é possível comparar os resultados da ferramenta com esse real estado interno. Desse modo essa pesquisa conclui que os resultados apresentados pelo TCAV podem identificar pistas sobre o quão relevante um conceito é para a predição de uma classe, no entanto, é uma

extrapolação afirmar que a ferramenta é capaz de produzir resultados que levem a correta interpretação do estado interno de uma RN.

Durante o percurso metodológico dessa pesquisa, o que se tornou mais relevante na observação do funcionamento da ferramenta é como o algoritmo em si passa a ocupar um papel secundário quando buscamos observar o que tem mais relevância nos resultados, isso porque o que determina os resultados é na realidade o conjunto de dados utilizado pelas operações do método no TCAV. E esse método, em última instância, funciona como um modelo de classificação linear (que discerne entre os conceitos relevantes e os exemplos aletórios) que, como previamente citado na subseção 2.1.1 sobre conjuntos de dados, nada mais é do que o "conjunto acumulado de relações descobertas" (SELBST, 2017, p.677), e que no caso específico da ferramenta TCAV, as relações que podem ser descobertas já são previamente recortadas por aquilo que é conhecido e escolhido por quem utiliza a ferramenta. Desse modo, é imprescindivel reconhecer tudo aquilo que o TCAV não pode observar, seus dados obscuros ou ("*dark data*"), como sendo tão relevante quanto ou até mais ao processo de classificação do modelo observado.

Assumindo que todos os dados que não são observados na operação da ferramenta podem ser criticamente relevantes à classificação observada, se trata de um caminho perigoso na direção da superinteligência se passamos a acreditar que podemos interpretar o estado interno de uma rede neural a partir de uma ferramenta potencialmente alienante. A realidade é que a ferramenta apenas quantifica nossos próprios vieses instaurados na hipótese em teste. Se a ferramente TCAV passa a ter um uso mais massificado pode se tornar comôdo assumir meias-verdades acerca do funcionamento interno de RN, pois é o tipo de informação que pode nos manter com a sensação de controle e entendimento, especialmente quando a ferramenta é fruto da empresa dona da maior ferramenta de pesquisa on-line do mundo atual, o que torna ainda mais fácil vestir e comercializar meias-verdades enquanto verdades absolutas.

## Referências

ANDERSON, J. A. *An introduction to neural networks*. [S.l.]: MIT press, 1995. Citado na página 20.

ASSIS, R. et al. Uso de redes neurais artificiais para o desenvolvimento de modelos de previsão da condição de pavimentos de aeroportos. In: . [S.l.: s.n.], 2016. Citado na página 20.

BILLARD, A. Robota: Clever toy and educational tool. *Robotics and Autonomous Systems*, Elsevier, v. 42, n. 3-4, p. 259–269, 2003. Citado na página 32.

BOSTROM, N. *Superintelligence: Paths, Dangers, Strategies*. [S.l.]: Oxford University Press, 2014. Citado na página 13.

BRADBEAR, N. et al. Bees and their role in forest livelihoods: a guide to the services provided by bees and the sustainable harvesting, processing and marketing of their products. *Non-wood Forest Products*, Food and Agriculture Organization of the United Nations (FAO), n. 19, 2009. Citado na página 42.

BREAZEAL, C. Designing sociable robots, mit press. *Cambridge, MA*, 2002. Citado na página 32.

BREIMAN, L. et al. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, Institute of Mathematical Statistics, v. 16, n. 3, p. 199–231, 2001. Citado na página 26.

BUCHANAN, B. G. A (very) brief history of artificial intelligence. *Ai Magazine*, v. 26, n. 4, p. 53–53, 2005. Citado na página 13.

CASTELVECCHI, D. Can we open the black box of ai? *Nature News*, v. 538, n. 7623, p. 20, 2016. Citado na página 22.

COOPER, A. et al. *The inmates are running the asylum:[Why high-tech products drive us crazy and how to restore the sanity]*. [S.l.]: Sams Indianapolis, 2004. Citado na página 25.

CORTIZ, D. *Ano XII - Nº 1 - Inteligência Artificial: equidade, justiça e consequências*. CGI.BR / NIC.BR, 2020. Disponível em: <https://www.cetic.br/pt/publicacao/ano-xii-n-1-inteligencia-artificial-equidade-justica-e-consequencias/>. Citado na página 17.

DOSHI-VELEZ, F.; KIM, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017. Citado 6 vezes nas páginas 13, 24, 25, 35, 79 e 81.

FEENBERG, A. Simondon and constructivism: a recursive contribution to the theory of concretization. *Scientiae Studia*, SciELO Brasil, v. 13, n. 2, p. 263–281, 2015. Citado na página 30.

FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, JSTOR, p. 1189–1232, 2001. Citado 2 vezes nas páginas 26 e 27.

GIEST, S.; SAMUELS, A. 'for good measure': data gaps in a big data world. *Policy Sciences*, Springer, p. 1–11, 2020. Citado na página 17.

GOODMAN, B.; FLAXMAN, S. European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, v. 38, n. 3, p. 50–57, 2017. Citado na página 24.

GUIDOTTI, R. et al. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*, 2018. Citado na página 24.

HALL, P.; GILL, N. *An introduction to machine learning interpretability*. [S.l.]: O'Reilly Media, Incorporated, 2019. Citado 3 vezes nas páginas 26, 27 e 28.

HAND, D. J. *Dark Data: Why What You Don't Know Matters*. Princeton University Press, 2020. ISBN 069118237X,9780691182377. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=9c572880f80fe5d8c4be4c97f6efdf3d>. Citado 2 vezes nas páginas 15 e 18.

HAYKIN, S. S. et al. *Neural networks and learning machines*. [S.l.]: Pearson education Upper Saddle River, 2009. v. 3. Citado 2 vezes nas páginas 21 e 22.

HILTON, D. J. Conversational processes and causal explanation. *Psychological Bulletin*, American Psychological Association, v. 107, n. 1, p. 65, 1990. Citado na página 25.

HUI, Y. *On the existence of digital objects*. [S.l.]: U of Minnesota Press, 2016. v. 48. Citado 2 vezes nas páginas 29 e 30.

KIM, B. et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279*, 2017. Citado 18 vezes nas páginas , 13, 15, 31, 34, 35, 36, 37, 38, 40, 41, 51, 52, 72, 73, 79, 80 e 81.

KRAUSE, J.; PERER, A.; NG, K. Interacting with predictions: Visual inspection of black-box machine learning models. In: *Proceedings of the 2016 CHI conference on human factors in computing systems*. [S.l.: s.n.], 2016. p. 5686–5697. Citado na página 27.

KRITSKI, P. M. B.; CALAZANS, V. F. B. Gilbert simondon: a técnica como pensamento e objeto. In: OLIVEIRA, J. (Ed.). *Filosofia da Tecnologia. Seus autores e seus problemas*. Caxias do Sul - RS: Educs, 2020. cap. 10, p. 266–290. Citado na página 29.

KUBAT, M.; HAYKIN, S. Neural networks: a comprehensive foundation by simon haykin, macmillan, 1994, isbn 0-02-352781-7. *The Knowledge Engineering Review*, Cambridge University Press, v. 13, n. 4, p. 24, 1999. Citado na página 20.

LIPTON, Z. C. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, ACM New York, NY, USA, v. 16, n. 3, p. 31–57, 2018. Citado na página 24.

MÁRQUEZ, G. *Cem anos de solidão*. Record, 2019. ISBN 9788501116550. Disponível em: <https://books.google.com.br/books?id=MAqQDwAAQBAJ>. Citado na página .

MARTIN, T. Interpretable machine learning. 2019. Citado na página 24.

MILLER, T.; HOWE, P.; SONENBERG, L. Explainable ai: Beware of inmates running the asylum or: How i learnt to stop worrying and love the social and behavioural sciences. *arXiv preprint arXiv:1712.00547*, 2017. Citado 2 vezes nas páginas 14 e 25.

MINSKY, M. Steps toward artificial intelligence. *Proceedings of the IRE*, IEEE, v. 49, n. 1, p. 8–30, 1961. Citado na página 16.

MONTAVON, G.; SAMEK, W.; MÜLLER, K.-R. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, Elsevier, v. 73, p. 1–15, 2018. Citado na página 22.

NOBLE, S. U. *Algorithms of oppression: How search engines reinforce racism*. [S.l.]: nyu Press, 2018. Citado na página 31.

PALUSZEK, M.; THOMAS, S. *MATLAB machine learning*. [S.l.]: Apress, 2016. Citado 2 vezes nas páginas 16 e 18.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. Why should i trust you?: Explaining the predictions of any classifier. In: ACM. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. [S.l.], 2016. p. 1135–1144. Citado na página 36.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3. ed. [S.l.]: Prentice Hall, 2010. Citado na página 16.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural networks*, Elsevier, v. 61, p. 85–117, 2015. Citado na página 21.

SELBST, A. D. Disparate impact in big data policing. *Ga. L. Rev.*, HeinOnline, v. 52, p. 109, 2017. Citado 2 vezes nas páginas 17 e 82.

SHARMA, S.; SHARMA, S.; ATHAIYA, A. Activation functions in neural networks. *International Journal of Engineering Applied Sciences and Technology*, IJEAST, v. 4, n. 12, p. 7, 2020. Citado 2 vezes nas páginas 20 e 21.

SHWARTZ-ZIV, R.; TISHBY, N. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017. Citado na página 22.

SIMONDON, G. Du mode d'existence des objets techniques. 1958. Citado 2 vezes nas páginas 29 e 30.

STENROOS, O. et al. Object detection from images using convolutional neural networks. 2017. Citado 5 vezes nas páginas 18, 19, 21, 23 e 24.

STIEGLER, B. *Technics and Time, 1, trans. by R. Beardsworth and G. Collins*. [S.l.]: Stanford: Stanford University Press, 1998. Citado na página 29.

STIEGLER, B. *Technics and time 2 (trans: Barker, S.)*. [S.l.]: Stanford, CA: Stanford University Press, 2009. Citado na página 29.

Torralba, A.; Efros, A. A. Unbiased look at dataset bias. In: *CVPR 2011*. [S.l.: s.n.], 2011. p. 1521–1528. Citado na página 17.

VARGAS, A. C. G.; PAES, A.; VASCONCELOS, C. N. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In: SN. *Proceedings of the xxix conference on graphics, patterns and images*. [S.l.], 2016. v. 1, n. 4. Citado na página 23.

WEBER, J. Helpless machines and true loving care givers: a feminist critique of recent trends in human-robot interaction. *Journal of Information, Communication and Ethics in Society*, Emerald Group Publishing Limited, v. 3, n. 4, p. 209–218, 2005. Citado 2 vezes nas páginas 32 e 33.

WERBOS, P. J. et al. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, v. 78, n. 10, p. 1550–1560, 1990. Citado na página 22.

WINSTON, P. H. Artificial intelligence 3rd edition. *Addison-Wesley, Reading, MA*, v. 34, p. 167–339, 1992. Citado na página 16.