



Coordenadoria de Tecnologia da Informação
Tecnologia em Análise e Desenvolvimento de Sistemas

Avaliação de desempenho de algoritmos genéricos para reconhecimento de íris

Julia Guedes Velico

Sorocaba
Junho – 2022



Coordenadoria de Tecnologia da Informação
Tecnologia em Análise e Desenvolvimento de Sistemas

Julia Guedes Velico

Avaliação de desempenho de algoritmos genéricos para reconhecimento de íris

Trabalho de Graduação apresentado à Faculdade de
Tecnologia de Sorocaba – FATEC, como parte dos
pré-requisitos para obtenção do título de Tecnólogo
em Análise e Desenvolvimento de Sistemas

Orientador: Maria das Graças J. M. Tomazela

Sorocaba
Junho – 2022

Agradecimentos

Agradeço à minha orientadora professora doutora Maria das Graças por ter me auxiliado em cada etapa dessa pesquisa, por ter me corrigido e pelas conversas aleatórias.

Agradeço também à professora Adriane Cavichioli por ter me ensinado a usar a plataforma *Custom Vision*, sem ela, essa pesquisa nunca teria acontecido.

E por fim, agradeço à minha perseverança, por não me deixar abandonar este curso que foi meu maior desafio sendo uma pessoa de humanas tentando crescer na área de exatas, e que me mostrou que quando nos deparamos com desafios e dificuldades, é apenas nós mesmos que devemos nos apoiar e seguir em frente.

RESUMO

O avanço tecnológico vem facilitando a vida das pessoas e dando-lhes segurança em suas vidas cotidianas. Nesse sentido, a biometria vem cada vez mais ganhando espaço, seja como reconhecimento facial, impressão digital ou reconhecimento de íris. A complexa textura da íris traz informações muito úteis para se diferenciar as pessoas. Nesse contexto, o objetivo deste trabalho foi avaliar o desempenho de algoritmos de *machine learning* para o reconhecimento da íris humana. Dessa forma, realizou-se uma pesquisa experimental, utilizando-se da ferramenta *Custom Vision* para realizar a tarefa proposta de reconhecer as íris, usando um algoritmo de *machine learning*. Os experimentos foram realizados utilizando-se três domínios diferentes do algoritmo, cada um dos domínios foi avaliado por um integrante de um grupo de três alunos que também analisaram o tema, tendo como o domínio Geral[A1] o alvo dessa pesquisa. Para isso, foram usadas imagens de íris do banco de dados UBIRIS V2, que possuía imagens de 260 pessoas. Para esta pesquisa, foram utilizadas imagens de 10 pessoas com 60 imagens cada, totalizando 600 imagens no total, sendo 500 para treinamento e 100 para testes. Após selecionadas as imagens e inseridas na plataforma, foi realizado o treino de 15 minutos do domínio Geral[A1] e logo após foram inseridas as imagens de teste. Como resultado, a acurácia do modelo foi baixa, sendo de apenas 37,15%. Como o domínio Geral[A1] deve ser treinado por mais tempo, um novo treinamento foi realizado, esse realizado por um período de 1 hora, trazendo assim resultados melhores, como a acurácia de 82,12%. Os outros dois integrantes realizaram o mesmo procedimento, mas com domínios diferentes. O modelo do domínio Geral obteve acurácia de 98,8% no treinamento de 1 hora e o domínio Geral[A2] obteve acurácia de 97,32%, também no treinamento de 1 hora. Foi possível identificar que imagens em que as íris estavam obstruídas por óculos, cabelo ou um pouco tampadas, o algoritmo teve dificuldade em reconhecê-las em todos os domínios, mesmo com maior tempo de treinamento; assim como a única classe de uma pessoa idosa foi a que teve maior precisão. Após a análise dos resultados de todos os domínios, foi possível identificar que o algoritmo genérico da plataforma teve bom desempenho quando treinado por maior tempo, porém, ele usa

características das imagens como, por exemplo, a área periocular para reconhecer as imagens, e não a íris humana de fato.

Palavras-chave: Inteligência artificial; redes neurais; reconhecimento de íris; *machine learning*; *deep learning*.

Sumário

Resumo.....	4
Introdução	9
1. Fundamentação teórica.....	11
1.1. Conceitos chave.....	11
1.1.1. Inteligência artificial.....	11
1.1.2. Aprendizado de máquina	14
1.1.2.1. Medidas de desempenho dos classificadores	15
1.1.2.2. Redes Neurais	18
1.1.3. <i>Deep learning</i>	21
1.1.3.1. Redes Neurais Convolucionais	24
1.1.4. Reconhecimento de imagem.....	26
1.1.5. Reconhecimento de íris.....	27
1.2. Trabalhos relacionados	29
2. Metodologia	37
2.1. Natureza da pesquisa	37
2.2. Ferramenta utilizada	37
2.3. Dados de experimento	37
2.4. Algoritmo	38
2.5. Experimento de pesquisa	38
2.6. Critérios para avaliação da pesquisa	39
3. Descrição do experimento	40
4. Resultados obtidos.....	43
4.1. Resultados desta pesquisa.....	43
4.2. Comparação dos resultados entre os integrantes	52
5. Considerações finais.....	56
Referências	58

Lista de Figuras

Figura 1 – Representação de um neurônio artificial.....	19
Figura 2 – Rede neural típica.....	21
Figura 3 – <i>Deep learning</i> model.....	23
Figura 4 – Representação convencional da CNN.	25
Figura 5 – Configurações necessárias para criação de um projeto	40
Figura 6 – Upload de imagens.....	42
Figura 7 – Indicadores de avaliação	44
Figura 8 – Performance por classe	45
Figura 9 – Inserindo uma imagem de teste.....	46
Figura 10 – Indicadores de avaliação com maior tempo de treinamento	48
Figura 11 – Performance por classe com maior tempo de treinamento.....	49
Figura 12 – Comparação das acurácias de acordo com o tempo de treinamento do domínio Geral[A1]	51
Figura 13 – Comparação das acurácias de acordo com o tempo de treinamento do domínio Geral	53
Figura 14 - Comparação das acurácias de acordo com o tempo de treinamento do domínio Geral[A2]	54
Figura 15 – Comparação das acurácias de acordo com o tempo de treinamento do domínio	55
Figura 16 – Precisão por classe entre domínios	55

Lista de Tabelas e Quadros

Tabela 1 – Matriz de Confusão de um classificador – para K classes	15
Quadro 1– Trabalhos relacionados	35
Tabela 2 – Precisão geral e médias.....	47
Tabela 3 – Precisão geral e médias com maior tempo de treinamento do domínio Geral[A1].....	50
Tabela 4 – Precisão geral e médias com maior tempo de treinamento do domínio Geral	52
Tabela 5 – Precisão geral e médias com maior tempo de treinamento do domínio Geral[A2]	53

INTRODUÇÃO

Os dispositivos móveis e computadores estão desempenhando cada vez mais um papel importante na vida diária das pessoas, não só para comunicações, mas também para entretenimento, atividades de trabalho, atividades sociais e relacionamentos. Devido ao grande aumento do uso de smartphones e dispositivos no dia-a-dia, a quantidade de dados sensíveis que esses dispositivos armazenam também está aumentando (por exemplo, contas bancárias, e-mails pessoais, fotografias). Essa situação leva à necessidade de proteger o acesso a esses dados sensíveis, e a biometria é oferecida como mecanismo alternativo para tal proteção.

Os sistemas de reconhecimento de íris estão entre os sistemas biométricos mais precisos disponíveis. A íris é um órgão interno que a torna mais robusta a ataques quando comparado a algumas outras tecnologias biométricas, especialmente impressão digital e reconhecimento facial. Isso se deve ao fato de que é mais difícil manipular um órgão interno do que disfarçar um corpo externo como, por exemplo, o rosto e por não deixar vestígios como a impressão digital (LI e JAIN, 2015).

O reconhecimento de íris requer algoritmos especializados capazes de reconhecer a íris humana, rastrear os limites internos e externos da íris, segmentá-las e descartar as áreas que não são íris. Portanto, algoritmos de reconhecimento de íris são complexos e mais difíceis de implementar do que algoritmos genéricos de reconhecimento de imagem. Com base no contexto apresentado, foi possível identificar o seguinte problema de pesquisa:

Qual é o desempenho de algoritmos de reconhecimento de imagem genéricos quando utilizado no reconhecimento de íris?

A partir do problema proposto, o objetivo da pesquisa foi verificar o desempenho de algoritmos genéricos de reconhecimento de imagem quando são utilizados para o reconhecimento de íris.

A hipótese é que a utilização de algoritmos genéricos também pode obter bom desempenho quando utilizados no reconhecimento de íris.

Como metodologia, esta pesquisa utilizou a pesquisa experimental que, segundo Gil (2007), consiste em determinar um objetivo de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto.

Este trabalho está estruturado da seguinte forma: no Capítulo 1, é mostrada a fundamentação teórica trazendo os conceitos chave e trabalhos relacionados; o Capítulo 2 descreve a metodologia da pesquisa adotada; o Capítulo 3 representa a descrição do experimento realizado passo a passo; o Capítulo 4 mostra os resultados obtidos da pesquisa e, por fim, o Capítulo 5 apresenta as considerações finais.

1. Fundamentação Teórica

1.1. Conceitos chave

No embasamento desta pesquisa, optou-se por organizar este capítulo em duas partes. Primeiramente, apresentam-se os seguintes conceitos chave que referenciam este trabalho: inteligência artificial, aprendizado de máquina, redes neurais, *deep learning* e reconhecimento de imagem e de íris. Em segundo lugar, apresenta-se um conjunto de trabalhos relacionados a esta pesquisa.

1.1.1 Inteligência artificial

Segundo Russel e Norvig (2013), a inteligência artificial tem como objetivo criar sistemas inteligentes que possam executar tarefas de forma semelhante aos seres humanos. Ela pode ser dividida em 4 categorias:

1) pensando como um humano: Deve-se entender como funciona a mente humana para então expressá-la como um programa de computador. Se os comportamentos de entrada/saída e sincronização do programa coincidirem com os comportamentos humanos correspondentes, haverá a evidência de que alguns dos mecanismos do programa também podem estar operando nos seres humanos.

2) pensando racionalmente: refere-se às leis do pensamento, que deu origem à lógica, a chamada tradição logicista, que consistia em usar a lógica clássica para resolver qualquer problema solucionável.

3) agindo como seres humanos: refere-se a abordagem do teste de Turing, que consistia em testar uma máquina e uma pessoa com perguntas e descobrir se as respostas vieram de uma pessoa ou de um computador. Para isso, o computador precisa da capacidade de processamento de linguagem natural para permitir que ele se comunique com sucesso em um idioma natural; representação de conhecimento para armazenar o que sabe ou ouve; raciocínio automatizado para usar as informações armazenadas com a finalidade de responder a perguntas e tirar novas conclusões e, por fim, aprendizado de máquina para se adaptar a novas circunstâncias e para detectar e extrapolar padrões.

4) agindo racionalmente: é a abordagem do agente racional, aquele que age para alcançar o melhor resultado ou, quando há incerteza, o melhor resultado esperado. Para IA, foi dada ênfase a inferências corretas. Essa realização de inferências corretas é aquilo que caracteriza um agente racional pois uma das formas de agir racionalmente é raciocinar de modo lógico até a conclusão de que dada ação alcançará as metas pretendidas e, depois, agir de acordo com essa conclusão.

Segundo Sage (1990 apud Haykin, 2001), o objetivo da inteligência artificial é o desenvolvimento de paradigmas ou algoritmos que requeiram máquinas para realizar tarefas cognitivas. Um sistema de IA deve ser capaz de fazer três coisas: 1) armazenar conhecimento; 2) aplicar o conhecimento armazenado para resolver problemas e 3) adquirir novo conhecimento por meio da experiência. A IA possui três componentes fundamentais: representação, raciocínio e aprendizagem.

Há diversos paradigmas da IA conhecidos, como sistemas especialistas baseado em regras, lógica fuzzy, raciocínio baseado em casos, redes neurais, entre outros. As redes neurais serão abordadas mais detalhadamente na seção 1.1.2.2, os outros paradigmas serão abordados, resumidamente, a seguir.

Sistemas especialistas são sistemas baseados em conhecimento, construídos com regras que reproduzem o conhecimento do especialista. Em geral são utilizados para solucionar problemas em domínios específicos, como por exemplo na área médica. Um sistema especialista possui várias vantagens, podendo ser capaz de estender as facilidades de tomada de decisão para muitas pessoas; o conhecimento dos especialistas pode ser distribuído de forma que possa ser utilizado por um grande número de pessoas; reduzem o grau de dependência que as organizações mantêm quando se veem em situações críticas, inevitáveis, como, por exemplo, a falta de um especialista. Ao registrar o conhecimento de empregados nos sistemas especialistas, promove-se uma significativa redução no grau de dependência entre a empresa e a presença física do empregado. Esses sistemas podem ser utilizados em treinamentos de grupos de pessoas, de forma rápida e agradável, servindo como instrumento para coleta de informações sobre o desempenho dos treinandos, obtendo subsídios para reformulação das lições para a obtenção de melhor desempenho, além de oferecer suporte e conselhos aos usuários (MENDES, 1997).

Segundo Gomide e Gudwin (1994), a Lógica Fuzzy suporta os modos de raciocínio que são aproximados ao invés de exatos. Modelagem e controle fuzzy de sistemas são técnicas para o tratamento de informações qualitativas de forma rigorosa. Derivada do conceito de conjuntos fuzzy, a lógica fuzzy constitui a base para o desenvolvimento de métodos e algoritmos de modelagem e controle de processos, permitindo a redução da complexidade de projeto e implementação. Para Alavala (2008), lógica Fuzzy é uma lógica multivalorada que permite que valores intermediários sejam definidos entre avaliações convencionais como verdadeiro/falso, sim/não, alto/baixo e etc. Noções como muito alto ou muito rápido podem ser formuladas matematicamente e processadas por computadores, a fim de aplicar uma aparência mais humana na forma de pensar na programação de computadores. As características da lógica fuzzy são: raciocínio exato é visto como um caso linear de raciocínio aproximado; tudo é uma questão de grau e o conhecimento é interpretado como uma restrição elástica ou, equivalente, difusa em uma coleção de variáveis.

O Raciocínio Baseado em Casos (RBC) é um enfoque para a solução de problemas e para o aprendizado baseado em experiências passadas. O RBC resolve problemas ao recuperar e adaptar experiências passadas - chamadas casos -armazenadas em uma base de casos. Um novo problema é resolvido com base na adaptação de soluções de problemas similares já conhecidas, ou seja, relembrando uma situação anterior similar (WANGENHEIM et al, 2013).

Segundo Damião, Caçador e Lima (2014), pode-se definir agente como um especialista em determinado assunto, no qual ele precisa ser inteligente para que, tendo uma percepção adequada do meio em que está inserido, ele execute uma tarefa da melhor forma possível. Portanto, os agentes inteligentes são sistemas computacionais que têm como principais características a atuação de forma autônoma, a percepção do ambiente onde estão inseridos, a adaptação às mudanças, como também a capacidade de trabalhar em função dos objetivos, podendo ser atribuídas aos agentes, tarefas normalmente desempenhadas por seres humanos. Para Russel e Norvig (2013), o aprendizado em agentes inteligentes pode ser resumido como um processo de modificação de cada componente do agente, a fim de levar os componentes a um acordo mais íntimo com as informações de realimentação disponíveis, melhorando assim o desempenho global do agente.

1.1.2. Aprendizado de máquina

A aprendizagem de máquina é feita por meio de algoritmos de aprendizagem. Segundo Goodfellow, Bengio e Courville (2016), um algoritmo de aprendizado de máquina é um algoritmo capaz de aprender com os dados. Baseado no aprendizado, um algoritmo de aprendizado deve ser capaz de aprender a partir de recursos de exemplos que recebe. Para avaliar as habilidades de um algoritmo de aprendizado de máquina, deve-se projetar uma medida quantitativa de seu desempenho. Normalmente, esta medida de desempenho P é específico para a tarefa T que está sendo executada pelo sistema. Pode-se obter informações importantes medindo a taxa de erro, ou seja, a proporção de exemplos para os quais o modelo produz uma saída incorreta, assim como a precisão (saída correta). Ainda de acordo com os mesmos autores, esses algoritmos podem ser divididos em 3 categorias, sendo elas:

Algoritmos de aprendizagem não supervisionados experimentam um conjunto de dados contendo muitos recursos e, em seguida, aprendem propriedades úteis da estrutura deste conjunto de dados. Um exemplo de algoritmo de aprendizagem de máquina não supervisionado é a *Clusterização*, que consiste em particionar os registros da base de dados em subconjuntos (ou *clusters*) de maneira que elementos presentes em um cluster compartilhem um conjunto de propriedades comuns e que os diferenciem dos elementos de outros clusters.

Algoritmos de aprendizagem supervisionada experimentam um conjunto de dados contendo recursos, mas cada exemplo também está associado a um rótulo ou destino. Por exemplo, o conjunto de dados Iris contém as espécies de cada planta íris, classificadas como íris-setosa, íris-versicolor, íris-virginica¹. Um algoritmo aprendizagem supervisionada pode estudar o conjunto de dados Iris e aprender a classificar plantas de íris nessas três espécies diferentes com base em suas medições.

Algoritmos de aprendizagem por reforço interagem com um ambiente, então existe um ciclo de feedback entre o sistema de aprendizagem e suas experiências. Pode ser visto como caso particular de aprendizagem supervisionada. A principal diferença entre o aprendizado supervisionado e o aprendizado por reforço é a medida de desempenho usada

¹ Conforme disponível em: <<https://archive.ics.uci.edu/ml/datasets/Iris>>. Acesso em: 01 dez. 2021.

em cada um deles. No aprendizado supervisionado, a medida de desempenho é baseada no conjunto de respostas desejadas usando um critério de erro conhecido, enquanto no aprendizado por reforço a única informação fornecida à rede é se uma determinada saída está correta ou não. A ideia básica tem origem em estudos experimentais sobre aprendizado dos animais. Quanto maior a satisfação obtida com uma certa experiência em um animal, maiores as chances de ele aprender.

1.1.2.1. Medidas de desempenho dos classificadores

Para medir o desempenho dos modelos de classificação, pode ser utilizada uma matriz de confusão, que resume as informações sobre as classes corretas e aquelas preditas usando um sistema de classificação. Conforme apresentado na Tabela 1, os resultados são apresentados em duas dimensões: classes verdadeiras e classes preditas, para k classes distintas $\{C_1, C_2, C_k\}$. Cada elemento $M(C_i, C_j)$ da matriz, $i, j = 1, 2, \dots, K$, representa o número de exemplos da base de dados que pertencem à classe C_i , mas que foram classificados como sendo da classe C_j .

Tabela 1 - Matriz de Confusão de um Classificador - para K classes

Classes	Predita C_1	Predita C_2	...	Predita C_k
Verdadeira C_1	$M(C_1, C_1)$	$M(C_1, C_2)$...	$M(C_1, C_k)$
Verdadeira C_2	$M(C_2, C_1)$	$M(C_2, C_2)$...	$M(C_2, C_k)$
...
Verdadeira C_k	$M(C_k, C_1)$	$M(C_k, C_2)$...	$M(C_k, C_k)$

Fonte: MONARD E BARANAUSKAS, 2003, P. 102

O número de acertos, para cada classe, está localizado na diagonal principal da matriz, $M(C_i, C_i)$. Os demais elementos $M(C_i, C_j)$, em que $i \neq j$, representam erros na classificação. Para exemplificar essas medidas, pode-se usar uma matriz para um problema com apenas duas classes, denominadas como C_+ (positiva) e C_- (negativa). Nesse caso existem duas

possibilidades de acerto: Verdadeiro Positivo (VP) e Verdadeiro Negativo (VN) e duas possibilidades de erro: Falso Positivo (FP) e Falso Negativo (FN).

Assim, as entradas da matriz de confusão têm os seguintes significados:

- VP é o número de predições corretas da classe C_+ ;
- FN é o número de predições incorretas da classe C_+ ;
- FP é o número de predições incorretas da classe C_- ;
- VN é o número de predições corretas da classe C_- .
- Acurácia (AC) é a proporção do número total de predições que foram corretas:

$$AC = (VP + VN) / (VP + VN + FP + FN) \quad (1)$$

- Sensitividade, Revocação, ou Taxa de Verdadeiro Positivo (TVP) é a proporção de casos positivos que foram identificados corretamente:

$$TVP = VP / (VP + FN) \quad \dots \quad (2)$$

- Taxa de Falso Positivo (TFP) é proporção de casos negativos que foram classificados incorretamente como positivos:

$$TFP = FP / (FP + VN) \quad (3)$$

- Especificidade ou Taxa de Verdadeiro Negativo (TVN) é definida como a proporção de casos negativos que foram classificados corretamente:

$$TVN = VN / (VN + FP) \quad (4)$$

- Taxa de Falso Negativo (TFN) é a proporção de casos positivos que foram classificados incorretamente como negativos:

$$TFN = FN / (FN + VP) \quad (5)$$

- Precisão (P) é a proporção de casos positivos preditos que foram corretos:

$$P = VP / (VP + FP) \quad (6)$$

De acordo com Han, Kamber e Pei (2011) as taxas de verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo são úteis para avaliar os custos (ou riscos) e benefícios associados com o modelo de classificação.

Existem diversos métodos para avaliar o desempenho de um classificador. Nesses métodos, pode-se dividir o conjunto em grupos de treinamento e teste. O modelo será construído a partir do grupo de treinamento e será testado pelo grupo de teste.

Essa divisão do conjunto de dados em grupos de treinamento e de teste pode ser feita de várias formas. O método denominado *resubstituição*, consiste em construir o modelo e testar o seu desempenho no mesmo conjunto de dados, ou seja, o grupo de teste é igual ao grupo de treinamento. Esse método produz estimativas extremamente otimistas da precisão, pois o processo de classificação tenta maximizá-la. (MONARD e BARANAUSKAS, 2005).

Um método bastante utilizado é conhecido como *holdout* (HAN, KAMBER e PEI, 2011; WITTEN e FRANK, 2005). Nesse método o conjunto de dados é dividido em dois grupos distintos para o treinamento e teste. Uma proporção comum utilizada é a de 2/3 dos dados para treinamento e 1/3 restante para teste.

Quando não há dados suficientes para particionar em conjuntos distintos de treinamento e teste sem perder capacidade significativa de modelagem ou teste utiliza-se o método denominado validação cruzada (*cross-validation*) (WITTEN e FRANK, 2005). Nesse método o conjunto de dados é dividido em k subconjuntos mutuamente exclusivos de tamanhos aproximadamente iguais. O procedimento é executado k vezes, cada vez um subconjunto é utilizado para teste e os demais conjuntos para treinamento. A acurácia final é calculada pela média das acurácias obtidas em cada um dos subconjuntos de teste. No método denominado validação cruzada estratificada (*stratified cross-validation*) as partições são estratificadas de maneira que a distribuição das instâncias nas classes em cada partição é equivalente àquela do conjunto inicial (HAN, KAMBER e PEI, 2011).

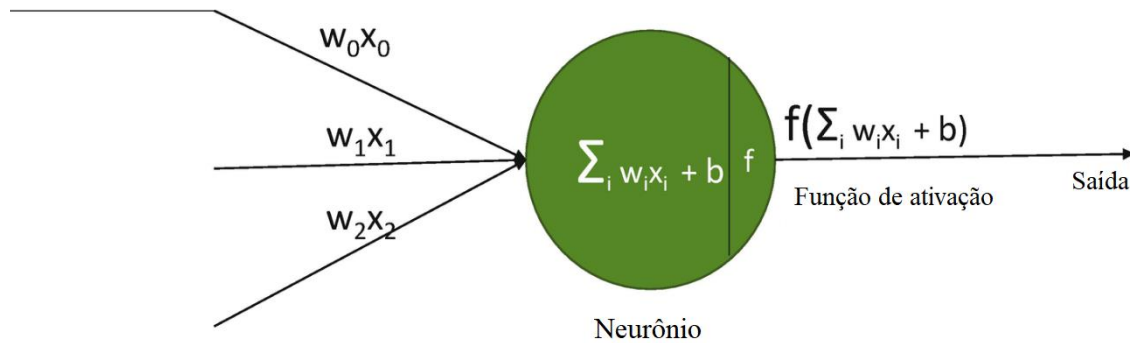
1.1.2.2. Redes Neurais

Uma rede neural artificial (RNA) é um algoritmo projetado com a finalidade de modelar a maneira como o cérebro humano realiza uma tarefa particular ou função de interesse; a rede normalmente é implementada utilizando componentes eletrônicos ou é simulada por programação em um computador. Para alcançar um bom desempenho, a rede emprega uma interligação de células computacionais chamadas de "neurônios" (HAYKIN, 2001).

De acordo com Haykin (2001), uma rede neural é um processador distribuído constituído de unidades de processamento simples, que tem a propensão natural para armazenar conhecimento experimental e torná-lo disponível para uso. Ela se assemelha ao cérebro humano nos seguintes aspectos: a) o conhecimento é adquirido pela rede a partir de seu ambiente por meio de um processo de aprendizagem; b) forças de conexão entre neurônios (conhecidas como pesos sinápticos) são utilizadas para armazenar o conhecimento adquirido.

Segundo Verdhan (2021), um neurônio recebe as entradas das camadas anteriores ou das camadas de entrada e, em seguida, faz o processamento das informações e compartilha uma saída. Os dados de entrada podem ser os dados brutos ou informações processadas de um neurônio anterior. O neurônio então combina a entrada com seu próprio estado interno e atinge um valor usando uma função de ativação. Posteriormente, uma saída é gerada usando essa função. A figura 1 representa esquematicamente o funcionamento de uma rede neural. A entrada recebida é calculada como uma soma ponderada, e um termo de viés é geralmente adicionado também. Esse é o objetivo de uma *função de propagação*. Como mostrado na figura 1, f é a função de ativação, w é o termo de peso, e b é o termo viés. Depois que os cálculos são feitos, recebe-se a saída.

Figura 1: Representação de um neurônio artificial.



Fonte: Adaptado de Verdhan, 2021.

As redes neurais aprendem a realizar tarefas semelhantes aprendendo ou sendo treinada. Isso é feito olhando para vários exemplos de pontos de dados históricos, como dados transacionais ou imagens e na maioria das vezes sem ser programado para regras específicas. Por exemplo, para distinguir entre um carro e um homem, uma rede neural começará sem compreensão prévia e conhecimento dos atributos de cada uma das classes. Em seguida, gera atributos e características de identificação a partir dos dados de treinamento. Então, aprende esses atributos e os usa mais tarde para fazer previsões e mostrar sua saída (VERDHAN, 2021).

O termo "aprender" no contexto das Redes Neurais Artificiais refere-se ao ajuste dos pesos e do viés dentro da rede para melhorar a precisão para essa rede. Uma maneira de fazer isso é reduzir o termo de erro, que é a diferença entre o valor real e o valor previsto. Para medir a taxa de erro, tem-se uma função de custo definida que é rigorosamente avaliada durante a fase de aprendizado da rede (VERDHAN, 2021).

Segundo Verdhan (2021), uma arquitetura básica da rede neural consiste em predominantemente três camadas:

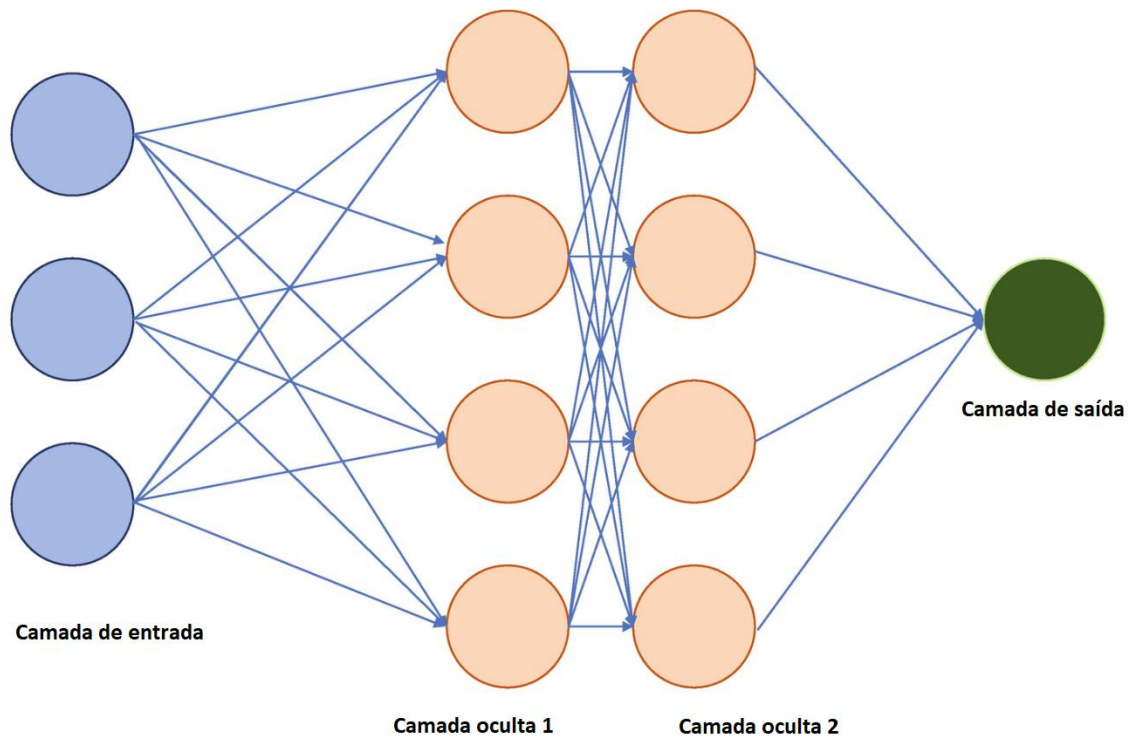
- Camada de entrada: Consiste em receber os dados de entrada. A rede recebe imagens brutas/imagens processadas na camada de entrada.
- Camadas ocultas: São o coração e a alma da rede. Todo o processamento, extração de recursos, aprendizado e treinamento são feitos nessas camadas. Camadas ocultas

quebram os dados brutos em atributos e recursos e aprendem os detalhes dos dados. Este aprendizado é usado posteriormente na camada de saída para tomar uma decisão.

- Camada de saída: A camada de decisão é a etapa final da rede. Ela aceita as saídas das camadas ocultas anteriores e, em seguida, faz um julgamento sobre a classificação final.

A explicação acima pode ser vista na figura 2 que representa uma arquitetura típica de uma rede neural básica.

Figura 2: Rede neural típica.



Fonte: Adaptado de Verdhan, 2021.

1.1.3. *Deep Learning*

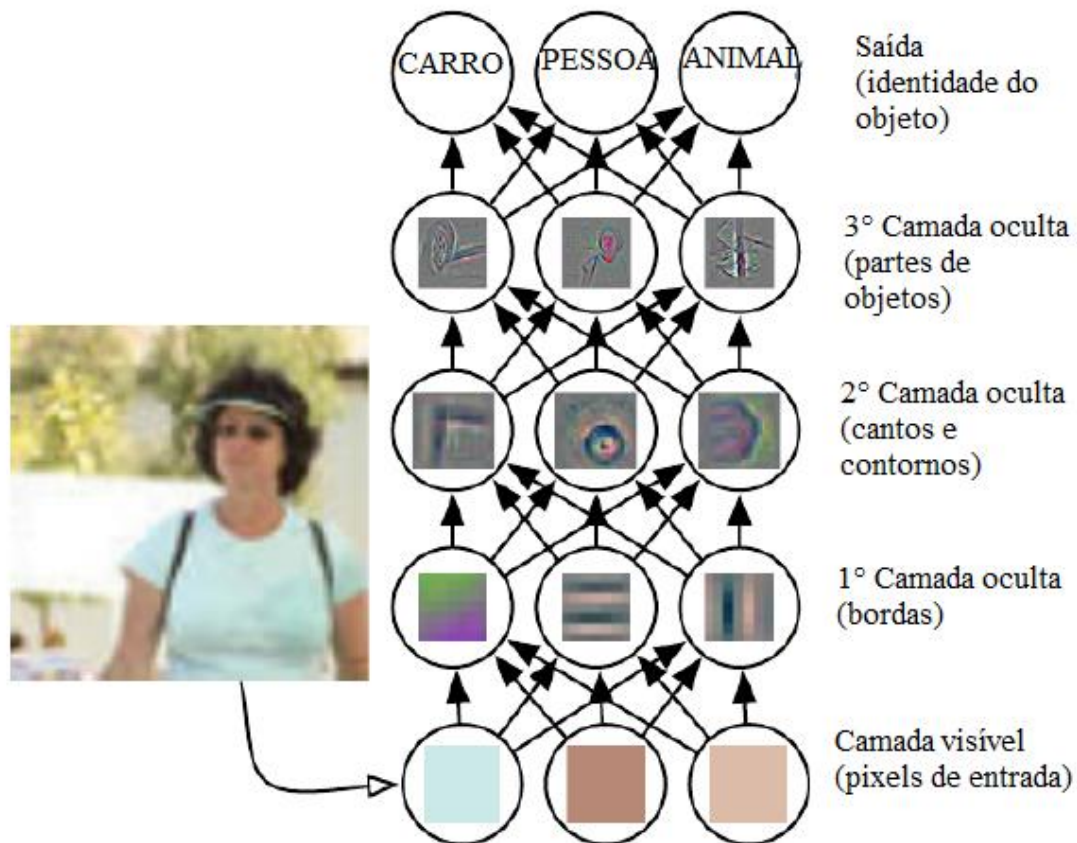
O *deep learning* permite que modelos computacionais que são compostos por várias camadas de processamento aprendam representações de dados com vários níveis de abstração. Esses métodos melhoraram o reconhecimento de voz, reconhecimento de objetos, detecção de objetos e entre outros. O *deep learning* descobre uma estrutura intrincada em grandes conjuntos de dados usando o algoritmo de retro propagação para indicar como uma máquina deve alterar seus parâmetros internos que são usados para calcular a representação em cada camada a partir da representação da camada anterior. Redes convolucionais profundas trouxeram avanços no processamento de imagens, vídeo, fala e áudio, enquanto as

redes recorrentes iluminaram dados sequenciais, como texto e fala (LECUN; BENGIO; HINTON, 2015).

De acordo com Ponti e Costa (2017), os métodos de aprendizagem profunda procuram usar um conjunto de dados de exemplos e um método para descobrir modelos como regras e parâmetros para orientar a aprendizagem de modelos com base nesses exemplos. No final do processo de aprendizagem, existe uma função que pode receber dados brutos como entrada e fornece uma representação suficiente do problema como saída.

Para Goodfellow, Bengio e Courville (2016), uma das principais dificuldades em muitas aplicações de inteligência artificial é que muitos dos fatores de variação influenciam cada pedaço de dados recebidos. Por exemplo, os pixels individuais em uma imagem de um carro vermelho podem ser muito semelhantes com a cor preta à noite e a forma da silhueta do carro pode depender do ângulo de visão. A maioria das aplicações exige que os fatores de variação separem e descartem aqueles com os quais são importantes. O *deep learning* resolve este problema central na aprendizagem de representação, introduzindo representações que são expressas em termos de outras representações mais simples, permitindo que o computador crie conceitos complexos a partir de conceitos mais simples. A Figura 3 mostra como um sistema de *deep learning* pode representar o conceito da imagem de uma pessoa combinando conceitos mais simples, como cantos e contornos, que, por sua vez, são definidos em termos de arestas.

Figura 3: Deep learning model.



Fonte: Adaptado de Goodfellow, Bengio e Courville, 2016. Pág. 6.

A entrada é apresentada pela camada visível, que possui esse nome pois contém as variáveis que são possíveis de se observar. Em seguida, há uma série de camadas ocultas cada vez mais abstratas da imagem. Essas camadas são chamadas de "ocultas" porque seus valores não são fornecidos nos dados, em vez disso, o modelo deve determinar quais conceitos são úteis para explicar as relações nos dados observados. Dados os pixels, a primeira camada pode facilmente identificar bordas, comparando o brilho dos pixels vizinhos. Dada a primeira camada oculta, é possível ter uma descrição das bordas, assim, a segunda camada oculta pode facilmente pesquisar cantos e contornos estendidos, que são reconhecíveis como coleções de arestas. Dada a segunda camada oculta em termos de cantos e contornos, a terceira camada oculta pode detectar partes inteiras de objetos específicos, encontrando coleções específicas de contornos e cantos. Por último, a descrição da imagem

em termos das partes do objeto que ela contém pode ser usada para reconhecer os objetos presentes na imagem (GOODFELLOW; BENGIO; COURVILLE, 2016).

1.1.3.1. Redes Neurais Convolucionais

Segundo Verdhhan (2021), convolução é extrair características importantes para classificação de imagens, detecção de objetos e assim por diante. As características serão bordas, curvas, quedas de cor, linhas, entre outros. Uma vez que o processo tenha sido bem treinado, ele aprenderá esses atributos em um ponto significativo da imagem. Então ele pode detectá-lo mais tarde em qualquer parte da imagem. Uma rede neural convolucional (CNN) é capaz de extrair todas essas características usando *deep learning* e a convolução ajuda a extrair os atributos significativos da imagem.

Para Saha (2018), a arquitetura de uma CNN realiza um melhor ajuste ao conjunto de dados da imagem devido à redução no número de parâmetros envolvidos e reutilização de pesos. A rede pode ser treinada para entender melhor a sofisticação da imagem. O papel da CNN é reduzir as imagens a uma forma mais fácil de processar, sem perder recursos essenciais para obter uma boa previsão. Elas não precisam ser limitadas a apenas uma camada convolucional. A primeira camada é responsável por capturar os recursos de baixo nível, como bordas, cor e orientação de gradiente. Com camadas adicionadas, a arquitetura se adapta aos recursos de alto nível também, se tornando uma rede que tem o entendimento completo de imagens no conjunto de dados

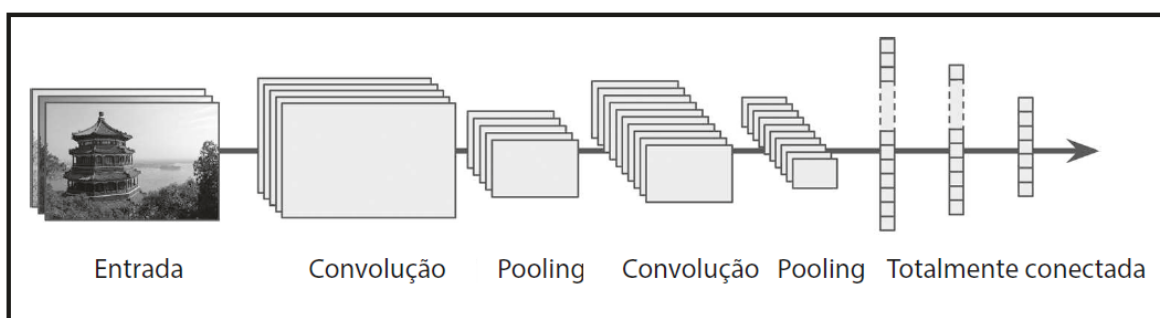
Semelhante à camada convolucional, a camada de pooling é responsável por reduzir o tamanho espacial do recurso convolvido. Isso diminui o poder computacional necessário para processar os dados por meio da redução da dimensionalidade. Além disso, a camada de pooling também pode ser usada para extrair características dominantes que são invariantes rotacionais e posicionais, mantendo assim o processo de treinamento efetivo da rede. Existem dois tipos de Pooling: o Max Pooling (Pooling máximo) e o Average Pooling (Pooling médio). O Max Pooling retorna o valor máximo da parte da imagem coberta pelo Kernel (camada de convolução). Atua também como um supressor de ruído, descartando totalmente as ativações ruidosas e também executa a eliminação de ruído junto com a redução de dimensionalidade. Por outro lado, o Average Pooling simplesmente executa a redução da

dimensionalidade como um mecanismo de supressão de ruído. O Average Pooling retorna a média de todos os valores da parte da imagem coberta pelo Kernel (SAHA, 2018).

De acordo com Géron (2019), as arquiteturas CNN típicas empilham algumas camadas convolucionais, depois uma camada pooling e mais outras camadas convolucionais e de pooling e assim por diante. A imagem fica cada vez menor à medida que avança pela rede, mas também fica mais profunda (com mais mapas de características) devido às camadas convolucionais.

A Figura 4 representa uma rede neural convolucional típica que facilita a visualização e entendimento sobre seu funcionamento.

Figura 4: Representação convencional da arquitetura de uma CNN.



Fonte: Géron, 2019. Pág 381.

Segundo Géron (2019), existem várias arquiteturas de CNNs disponíveis que têm sido fundamentais na construção de algoritmos que alimentam e devem alimentar a IA como um todo em um futuro previsível. Algumas delas são:

LeNet: É uma das arquiteturas CNN mais conhecidas e foi criada por Yann LeCun em 1998 e amplamente utilizada para reconhecimento de dígitos manuscritos (MNIST).

AlexNet: Foi desenvolvida por Alex Krizhevsky, Ilya Sutskever e Geoffrey Hinton, e foi a primeira a empilhar camadas convolucionais diretamente umas sobre as outras em vez de empilhar uma camada pooling no topo de cada camada convolucional.

GoogLeNet: foi desenvolvida por Christian Szegedy et al. da Google Research, tendo sua taxa de erro abaixo de 7%. Esse desempenho foi dado pelo fato de que sua rede era muito mais profunda que outras CNNs. O GoogLeNet tem 10 vezes menos parâmetros que a AlexNet.

ResNet: desenvolvida por Kaiming He et al., é uma CNN extremamente profunda composta por 152 camadas. A chave para treinar uma rede tão profunda é utilizar conexões skip (chamadas também de shortcuts connections): o sinal fornecido a uma camada também é adicionado à saída de uma camada localizada um pouco acima da pilha; isso faz com que a ResNet apresente uma taxa de erro abaixo de 3,6%.

1.1.4. Reconhecimento de Imagem

O reconhecimento de imagem é um dos métodos mais ativamente buscados nas áreas no campo das ciências da imagem e engenharia. O motivo é evidente: a capacidade de substituir as capacidades visuais humanas por uma máquina é muito importante e possui diversas utilidades. A ideia principal é inspecionar uma imagem processando os dados como pixels e padrões de imagens. Com o reconhecimento de imagens, é possível reduzir a carga de trabalho e melhorar a precisão de tomada de decisões por operadores humanos em diversos campos, incluindo o militar e defesa, sistemas de engenharia biomédica, monitoramento de saúde, cirurgia, sistemas de transporte inteligentes, manufatura, entretenimento, e sistemas de segurança. O reconhecimento de imagem é um campo multidisciplinar, que requer contribuições de diversas tecnologias e experiência em sensores, sistemas de imagem, algoritmos de processamento de sinal/imagem, hardware e software e sistemas de empacotamento/integração (JAVIDI, 2002).

Para Douro (2012), a tarefa de reconhecimento de imagens implica em várias formas de associação entre elas, dependendo do interesse da aplicação que podem ser:

- **Comparação de uma imagem com parte de outra imagem:** essa abordagem pode ser encontrada em situações em que se procura algum determinado objeto em uma imagem.

Essa abordagem geralmente infere sobre questionamentos como: a imagem x existe “dentro” da imagem X .

- **Comparação de duas imagens:** Essa abordagem espera responder qual o grau de semelhança entre duas imagens, podendo ser total no caso de duas imagens serem idênticas. Esse tipo de abordagem é base para as comparações efetuadas em bancos de imagens e com essa resposta pode-se ter conhecimento sobre qual classe uma imagem pertence.

- **Comparação de uma imagem com um banco de imagens:** Para esse tipo de abordagem, uma imagem é comparada com outras imagens, uma a uma. É uma abordagem que é efetuada visando obter uma medida de comparação um para-muitos e a atribuição da imagem sem classificação a uma classe já conhecida.

1.1.5. Reconhecimento de Íris

Para Minaee e Abdolrashidi (2019), os sistemas de reconhecimento de íris são amplamente usados para aplicativos de segurança, uma vez que contêm um rico conjunto de recursos e não mudam significativamente com o tempo. Eles também são virtualmente impossíveis de se falsificar. Desta forma, tem havido vários trabalhos propondo diferentes abordagens para o reconhecimento de íris. Muitas das abordagens tradicionais seguem duas etapas: abordagem de aprendizado de máquina, na qual um conjunto de recursos feitos à mão são derivados de imagens da íris e no segundo passo, um classificador é usado para reconhecer as imagens da íris. Os algoritmos para reconhecimento de íris alcançam altas taxas de precisão, porém, eles envolvem muito pré-processamento (incluindo a segmentação da íris) e usando algumas características que podem não ser ideais para íris em diferentes conjuntos de dados (coletados sob diferentes raios, ambientes e condições).

Para Delgado (2013), o processo baseado na identificação da íris pode ser dividido nas seguintes etapas:

- **Captura da imagem:** Corresponde a adquirir a imagem do olho. A qualidade da imagem é essencial para a performance das outras. É recomendado que a captura seja com luz infravermelha, com o intuito de revelar até mesmo detalhes que não podem ser vistos apenas com luz visível.

- **Segmentação e localização:** Nesta etapa a pupila, a íris e as pálpebras são localizadas e segmentados, ou seja, essa etapa determinará a região em que a íris se encontra.
- **Normalização:** Após a segmentação da íris, é preciso compensar as variações de distância entre a câmera e o usuário, assim como as alterações do tamanho da pupila causadas por variações de luminosidade.
- **Representação:** Já com a íris normalizada, deve-se extrair as características que serão usadas para serem representadas.
- **Reconhecimento:** Etapa final que corresponde ao método que será usado para distinguir entre duas ou mais representações da íris e, logo após, determinar se correspondem à mesma pessoa ou não.

Para Abiyev e Altunkaya (2008), no processamento da íris, ela é detectada e extraída de uma imagem do olho e normalizada. A imagem normalizada após o realce é representada por uma matriz que descreve os valores da escala de cinza da imagem da íris. Esta matriz torna os dados de treinamento definidos para a rede neural. O sistema de reconhecimento de íris inclui dois modos de operação: modo de treinamento e modo online. Na primeira fase, o treinamento do sistema de reconhecimento é realizado usando valores da escala de cinza das imagens da íris, e então a rede neural é treinada com todas as imagens da íris. Após o treinamento, no modo online, a rede neural realiza a classificação e reconhece os padrões que pertencem à íris de uma determinada pessoa.

A imagem do olho contém não apenas a região da íris, mas também algumas partes inúteis, como a pupila, pálpebras e esclera. Por esse motivo, na primeira etapa, a segmentação deve ser feita para localizar e extrair a região da íris da imagem do olho. A localização da íris é a detecção da área da íris entre a pupila e a esclera. Portanto, precisa-se detectar os limites superior e inferior da íris e determinar seus círculos internos e externos. Para isso, vários algoritmos foram desenvolvidos para localização da íris. Alguns deles são baseados na transformação de Hough (Hough transform). Um algoritmo de detecção de borda circular com transformação de Hough (técnica comumente usada para a detecção de curvas regulares, como linhas, círculos e elipses) é aplicado para detectar o interior e os limites externos da íris. A transformação circular de Hough é então empregada para deduzir o raio e as

coordenadas do centro das regiões da pupila e da íris. Assim, a partir do canto superior esquerdo da íris, a transformação de Hough circular é aplicada. Este algoritmo é usado para cada círculo interno e externo separadamente (ABIYEV e ALTUNKAYA, 2008).

1.2. Trabalhos relacionados

Wu e Li (2015) apresentaram um estudo sobre reconhecimento de imagem baseado em *deep learning* para comparar o desempenho da classificação de caracteres escritos à mão e dígitos manuscritos MNIST. Usando o modelo CNNA (*Convolution Neural Network Algorithm*- Algoritmo de rede neural convolucioal), os autores escolheram o banco de dados MNIST² e um banco de dados de caracteres manuscritos de palavras reais para comparar o desempenho do *deep learning*. O banco de dados MNIST conteve 60.000 amostras de treinamento e 10.000 amostras de teste. Os caracteres escritos à mão tiveram 18760 amostras de treinamento e 3240 amostras de teste. Incluídos 10 números, 26 letras maiúsculas em inglês, 26 letras inglesas minúsculas e 5 caracteres chineses, um total de 67 caracteres diferentes. Esses caracteres tendo sido escritos por 500 pessoas com uma caligrafia diferente. Para observar como o número de kernels de convolução afetaram o desempenho geral, foi escolhido três CNNs diferentes. Usando os dados do banco de dados MNIST, o primeiro CNN teve acurácia de 99,25%, o segundo teve 99,05% e o terceiro teve 99,28% de acurácia. Usando o banco de dados de caracteres manuscritos de palavras reais, o primeiro CNN obteve acurácia de 92,91%, o segundo teve 91,98% e o terceiro teve 88,72%. As acurácias obtidas no banco de dados MNIST são superiores ao banco de dados de caracteres manuscritos de palavra real, logo, os autores afirmaram que isso ocorre porque o volume de amostras de treinamento no primeiro banco de dados é maior do que o último.

A pesquisa de Rodrigues (2018) objetivou realizar o reconhecimento automático de caracteres em placas de licenciamento automotivo usando *deep learning* e redes neurais convolucionais, com a finalidade de demonstrar a viabilidade da resolução do problema de reconhecimento das placas. Para isso, foi necessário criar uma base de dados com placas de licenciamento veiculares brasileiras e desenvolver uma CNN (rede neural convolucional)

² grande banco de dados de dígitos manuscritos comumente usados para treinar vários sistemas de processamento de imagem.

para processar classificação de dígitos de placas veiculares. Como primeiro passo, decidiu-se que o banco de imagens de placas seria obtido de forma manual, ou seja, um banco de imagens próprio seria criado e para isso foi escolhida a UFPB, Campus I, para a tomada das fotos. No processo, foi solicitada a autorização para fotografar os veículos nos estacionamentos do Campus I. Para o recorte das placas foi utilizada uma ferramenta online chamada Supervisely. Após a obtenção dos recortes das placas, as imagens foram convertidas para escala de cinza e submetidas a um processo de equalização de histograma. Para esta tarefa, um script desenvolvido pelo autor, na linguagem Python, utilizou uma biblioteca de manipulação de imagens de código aberto, a OpenCV. O trabalho obteve como resultado uma taxa média de acerto de caracteres de 89,24% e de placas inteiras de 49,22%. Uma normalização do banco foi necessária para algumas placas que levavam a casos isolados como, por exemplo, placas de carros de aluguel que por serem vermelhas levavam a poucos casos com caracteres de cor branca, foram removidas do banco de teste e treinamento. Placas com alto grau de rotação também foram retiradas do banco de treinamento e teste, visto que a baixa ocorrência interferiu no treinamento da rede. Outros problemas como sombra parcial em demasiado ou alto grau de luminosidade natural foram fatores que interferiram bastante no treinamento e aumento da precisão nas inferências.

A pesquisa de Basso (2019) teve como objetivo entender como algoritmos de Inteligência Artificial podem auxiliar na identificação das expressões faciais em um ambiente empresarial e gerar um relatório sobre o estado aparente de seus colaboradores. Nesta pesquisa usou-se Inteligência artificial, redes neurais artificiais, reconhecimento facial e inteligência emocional, além de tecnologias como Phyton, Anaconda Navigator, Open Source e Keras 5. O experimento consistiu na aplicação de um algoritmo de reconhecimento de expressões e emoções faciais treinados a partir de Redes Neurais. Foram utilizadas as 7 emoções básicas raiva, nojo, medo, feliz, triste, surpresa e neutro. Utilizou-se uma Rede Neural Convolutacional supervisionada que passou por um processo de aprendizagem/treinamento entradas e saídas definidas. A rede neural foi treinada utilizando uma base de dados adquirida pelo *Kaggle* com 35888 imagens de diversas pessoas com diferentes idades e gênero, com ordenação aleatória. A avaliação da pesquisa foi realizada com três pessoas que trabalham em desenvolvimento de sistemas, que se voluntariaram para captura dos rostos em seu dia-a-dia no ambiente de trabalho. Suas faces eram capturadas pela

webcam e com base na expressão daquele momento, a aplicação armazenava o resultado obtido. Esse processo era realizado no decorrer do dia, gravando-se 2 segundos de vídeo a cada 1 hora, logo, foram geradas 6 mil fotos para validar a rede neural. A autora concluiu que quando seria acionado a aplicação para captura dos rostos e das expressões, os colaboradores tentaram forçar algumas expressões, reduzindo assim a acurácia dos resultados, pois o ideal era validar as expressões naturais de cada colaborador em seu ambiente de trabalho, por isso as expressões de assustado e triste resultaram em muitos falsos positivos. Entretanto, a acurácia do modelo se torna mais efetiva com as expressões de feliz, neutra e raiva, pois são nessas expressões que o índice de acerto é maior nos três casos de teste, atingindo assim o objetivo da pesquisa, sendo possível a captura das expressões faciais em tempo real pela aplicação e a avaliação das expressões e emoções expressas pelos participantes.

Na pesquisa de Traore, Kamsu-Foguem e Tangara (2018), os autores apresentaram uma abordagem baseada em CNN para reconhecimento de imagem para classificar se as imagens microscópicas contêm um patógeno da cólera ou da malária, cientificamente denominado *Vibrio cholerae* e *Plasmodium falciparum*. Foi necessário treinar a CNN de sete camadas, utilizar um banco de dados com várias imagens, conseguidas pelo Google imagens com uma palavra-chave sobre diferentes nomes de patógenos epidêmicos, 80 imagens de testes nomeadas numericamente e incrementalmente e usar a ferramenta TensorFlow. Após o estudo, os autores chegaram à conclusão de que a arquitetura proposta da CNN fornece a melhor classificação de resultados, alcançando a precisão de classificação de 94%, com 200 imagens de cólera *Vibrio* e 200 imagens de *Plasmodium falciparum*.

A pesquisa de Alboy (2019) teve como objetivo o estudo de métodos de reconhecimento de objetos utilizando técnicas de *deep learning* para o desenvolvimento de um classificador para ambientes internos, utilizando como ambiente de testes o Departamento de Ciência da Computação da Ufscar e classificando não só uma série de objetos, mas também seus estados, como aberto e fechado. A pesquisa visou impactar diretamente na vida de pessoas com deficiências visuais, aumentando a percepção do ambiente que as cerca. Na pesquisa, o autor fez o uso de CNN's de diferentes métodos e usou também o Kinect para obter imagens nos formatos RGB e com profundidade. O processo teve início com a aquisição da imagem. Ela ocorre com uso do dispositivo Kinect e obtém a

imagem no formato RGB e com profundidade na proporção de 640 x 480. Nesta pesquisa, foram utilizadas as CNNs AlexNet e GoogLeNet. Para a realização do treinamento das CNNs foi construída inicialmente uma base composta por três categorias de imagens que são portas gavetas e fogões. A base foi feita com imagens retiradas do ImageNet, que é uma base com imagens de uso gratuitos para fins científicos, e conta com 300 imagens em cada categoria, totalizando 900 imagens nesta base inicial. Foram utilizados 25% do total da base para serem utilizadas como testes e 75% para o treinamento. Os parâmetros analisados para a obtenção dos resultados nos experimentos parciais foram o tempo de treinamento e a acurácia da rede. Como resultado, o autor chegou à conclusão de que a CNN AlexNet e GoogLeNet apresentaram 79.09% e 81% no valor de sua acurácia. Estes são bem superior ao desempenho do SVM para a mesma base. A diferença nos resultados entre os métodos de CNN e SVM se dá pela forma que os dados são utilizados por cada um, pois as CNN's utilizam as imagens de forma a extrair características de uma forma mais direta do que o SVM que necessita de um algoritmo de extração antes de ser utilizado.

O estudo de Özyurt (2019), teve como objetivo comparar diversas arquiteturas de reconhecimento de imagens. O estudo consistiu em duas pesquisas com intuito de analisar as arquiteturas Alexnet, VGG16, VGG19, GoogleNet, ResNet e SqueezeNet. O segundo estudo consistindo no uso do algoritmo ReliefF com os mesmos recursos. Estes foram obtidos da última camada dessas arquiteturas CNNs, fornecidos ao classificador SVM para medir o desempenho da classificação. Os recursos obtidos de um total de seis arquiteturas foram combinados em um único vetor, totalizando 6.000 recursos. A seguir, o algoritmo ReliefF foi aplicado, em seguida, 1000 recursos com a maior capacidade de predição foram divididos em 100. O estudo experimental foi compilado usando o software MATLAB (R2018b). O autor chegou à conclusão de que o melhor desempenho foi exibido na arquitetura ResNet, na qual 80% do treinamento foi realizada em todas as arquiteturas, com uma taxa de precisão de 96,9%. O desempenho da arquitetura ResNet foi seguido pelo SqueezeNet com 95,24% de precisão, que, apesar de uma redução de 50× no tamanho do modelo em comparação com AlexNet (93,81%), teve melhor em comparação com a arquitetura AlexNet. Além disso, embora o número de camadas de convolução na arquitetura VGG-19 (93,10%) é mais do que VGG-16 (94,05%), taxas de desempenho mais baixas foram obtidas. O GoogleNet apresentou 94,29% de precisão. No segundo estudo, o algoritmo Relief obteve uma taxa de

precisão de 98,76% e 99,29%, obtido nas taxas de treinamento de 50% e 80%, pelo método CNN-Relief-SVM.

Na pesquisa de Yu, Jia e Xu (2017), os autores propuseram uma arquitetura CNN com 11 camadas para aumentar a capacidade discriminativa para a classificação de imagem hiper espectral. Cada imagem de teste foi girada nos ângulos de 90°, 180°, 270°, criando assim 4 variações da imagem, assim, ampliando os dados de treinamento em 8 vezes. Os autores usaram 3 conjuntos de dados, sendo eles Indian, Salinas e PaviaU. Os dois primeiros conjuntos de dados foram coletados em áreas naturais, e o terceiro em uma área urbana. Os resultados obtidos usando os dados de Indian foram 99,56% de acurácia; Salinas obteve 98,36% e PaviaU obteve 69,42%. Os autores chegaram à conclusão de que sua CNN proposta teve melhor desempenho que os algoritmos SVM, RAW, MOR e Garbor, obtendo uma maior acurácia nos testes.

No estudo de Jmour, Zayen e Abdelkrim (2018), os autores estudaram uma abordagem de aprendizagem baseada no treinamento de CNNs para um tráfego de sistema de classificação de sinais. Foi usada a técnica de transferência de aprendizagem chamada “Técnica de ajuste fino”, de reutilização de camadas treinadas no conjunto de dados ImageNet usando a CNN pré treinada AlexNet. A ideia principal do estudo foi projetar um método que reutilizasse uma parte das 4 camadas de treinamento da AlexNet. Quatro tipos de placas de trânsito foram usados: sinais de parar, sinais de não parar, sinais de luz vermelha e luz verde. O conjunto de dados continha mais de 360 imagens no total, dividido em diferentes classes. Para evitar o uso de dados de teste no treinamento, foram deixadas 180 imagens do conjunto de treinamento para validação e 180 imagens de teste apresentando os quatro tipos de sinais. Os autores obtiveram uma precisão máxima de 0,8620 para 6 períodos de treinamento usando a arquitetura otimizada AlexNet e aproveitando a extração de recursos aprendida no conjunto de dados ImageNet. O modelo proposto obteve 93,33% de precisão no conjunto de teste.

Pritt e Chern (2017), propuseram um sistema baseado em *deep learnig* e CNN para reconhecimento de imagens multiespectrais de satélites. O sistema consiste em um conjunto de CNNs com redes neurais de pós-processamento que combinam as previsões das CNNs com metadados de satélite. No banco de dados fMoW (*Funcional Map of the World*), divulgada pela IARPA (*Intelligence Advanced Research Projects Agency*), há um conjunto

de dados com um milhão de imagens em 63 classes. As imagens que deveriam ser classificadas incluíram imagens únicas, bem como sequências de imagens que compreendem um evento como inundação de estradas, atividade de construção, ou deposição de detritos. O sistema foi implementado em Python usando o Keras e Bibliotecas de *deep learning* do TensorFlow. Os dados do fMoW para treinamento da rede foram divididos em dois subconjuntos: um subconjunto de treinamento com 363 mil imagens, ou 87% dos dados, e um subconjunto de validação menor com 53 mil imagens, assim como 11000 mil imagens de falsa detecção fornecidas pela IARPA. Os autores chegaram à conclusão que, quando avaliadas nas 53 mil imagens do fMoW, a precisão total do sistema é de 83%; quinze classes tiveram precisão de 95% ou mais. A maior dificuldade do sistema envolveu diferenciar estaleiros de portos, confundindo-se 56% das vezes.

O estudo de Sun, Xue, Zhang e Yen (2019) objetivou desenvolver uma nova CNN, nomeada EvoCNN e comparar seu desempenho a outras redes neurais. Nesse estudo foram usadas nove classificações de imagem amplamente utilizadas de conjuntos de dados de referência para examinar o desempenho do método proposto. Os bancos de dados usados foram o Fashion, Rectangle, Rectangle Images (RI), o Convex Sets (CS), MNIST Basic (MB), MNIST with Background Images (MBI), Random Background (MRB), Rotated Digits (MRD), e RD plus Background Images (MRDBI). Os bancos de dados foram classificados em três categorias diferentes. A primeira categoria incluiu apenas moda para reconhecer 10 objetos (por exemplo, calças, casacos e assim por diante) tendo 50.000 imagens de treinamento e 10.000 imagens de teste. A segunda foi composta pelas variantes MNIST, incluindo o MB, MBI, MRB, MRD e os benchmarks MRDBI para classificar 10 dígitos escritos à mão (0-9). A terceira categoria foi para reconhecer formas de objetos (retângulo ou não para os bancos Rectangle e RI, e convexo ou não para o banco Convex). Essa categoria continha 12.000 e 8.000 imagens de treinamento, respectivamente e todas elas incluíram 50.000 imagens de teste. Segundo os autores, o desempenho médio de EvoCNN foi ainda melhor do que o melhor desempenho de oito concorrentes, e apenas um pouco pior do que o melhor do GoogleNet e VGG16. Usando os dados MRB e MBI, as menores taxas de erro de classificação em comparação às outras foram de 6,08% e 11,5%, e as taxas médias de erro do EvoCNN foram 3,59% e 4,62%, respectivamente.

O quadro 1 apresenta as principais características dos trabalhos apresentados.

Quadro 1: Trabalhos relacionados

Autor/Ano	Objetivo	Ferramentas/Algoritmos	Base de dados
Wu, Li (2015)	Comparar o desempenho da classificação de caracteres escritos à mão e dígitos manuscritos MNIST.	CNNA	MNIST
Yu, Jia, Xu (2017)	Desenvolver uma CNN para aumentar a capacidade de classificação de imagens hiper espectrais.	SVM, RAW, MOR, Garbor	Indian, Salinas, PaviaU
Pritt, Chern (2017)	Reconhecer imagens multiespectrais de satélites.	Keras, TensorFlow	fMoW
Jmour, Zayen, Abdelkrim (2018)	Reconhecer quatro tipos de sinais de trânsito.	ImageNet, AlexNet	-
Traore, Kamsu-Foguem, Tangara (2018)	Desenvolver uma CNN de reconhecimento de imagem para classificar imagens microscópicas com cólera ou malária.	TensorFlow, CNN	Google imagens
Rodrigues (2018)	Realizar o reconhecimento automático de caracteres em placas de licenciamento automotivo.	Supervisely, OpenCV	Obtido de forma manual
Alboy (2019)	Desenvolver um classificador de reconhecimento para ambientes internos para deficientes visuais.	kinect, ImageNet, AlexNet, GoogLenet, SVM	Obtido de forma manual.
Özyurt (2019)	Comparar diversas arquiteturas de reconhecimento de imagens.	MATLAB (R2018b), ReliefF, Alexnet, VGG16, VGG19, GoogleNet, ResNet e SqueezeNet	-

Basso (2019)	Estudar algoritmos de Inteligência Artificial na identificação de expressões faciais em um ambiente empresarial.	Webcam, Anaconda Navigator, Open Source e Keras, Kaggle	Obtido de forma manual.
Sun, Xue, Zhang, Yen (2019)	Comparar a CNN desenvolvida com demais concorrentes.	GoogleNet, VGG16	Fashion, Rectangle, RI, CS, MB, MBI, MRB, MRD, MRDBI.

Fonte: Elaborado pelo autor.

2. Metodologia

2.1. Natureza da Pesquisa

Para o desenvolvimento dessa pesquisa, foi utilizada a pesquisa experimental que, segundo Gil (2007), consiste em determinar um objetivo de estudo, selecionar as variáveis que seriam capazes de influenciá-lo, definir as formas de controle e de observação dos efeitos que a variável produz no objeto.

Para esta pesquisa, foram referenciadas três variáveis de análise:

1. Imagens utilizadas para treino.
2. Imagens utilizadas para teste.
3. Domínio do algoritmo.

2.2. Ferramenta Utilizada

A ferramenta utilizada para o desenvolvimento dessa pesquisa foi:

- **Custom Vision**³: a visão personalizada é um serviço de reconhecimento de imagem da Microsoft Azure que permite criar, implantar e aprimorar os próprios modelos identificadores de imagem. Um identificador de imagem aplica rótulos (que representam classificações ou objetos) a imagens, de acordo com as características visuais detectadas. É possível identificar tanto imagens quanto objetos, usando um algoritmo de *machine learning*.

2.3. Dados de Experimento

Para o treinamento e testes dos algoritmos foi usada a base de dados chamada UBIRISV2⁴ que consiste em um banco de dados para reconhecimento de íris. As imagens

³ Conforme disponível em: <<https://docs.microsoft.com/pt-br/azure/cognitive-services/custom-vision-service/overview>>. Acesso em: 23 mar. 2022.

⁴ Conforme disponível em: <<http://iris.di.ubi.pt/ubiris2.html>>. Acesso em: 25 out. 2021.

foram adquiridas por meio de fotos capturadas em dois dias, resultando num total de 260 pastas de imagens com cada uma contendo cerca de 30 e 60 imagens cada. Os voluntários eram, na sua grande maioria, caucasianos latinos (cerca de 90%) e também negros (8%) e asiáticos (2%). Cerca de 60% dos voluntários realizaram as duas sessões de imagem, enquanto 40% realizaram apenas uma, sendo na primeira ou na segunda aquisição. A base original contém imagens de íris de 260 indivíduos, separadas por imagens do olho esquerdo e olho direito, que totalizaram 11.102 imagens. As imagens originais possuem tamanho 400 x 300 pixels, estão no formato .tiff e resolução de 72 dpi, mas que foram convertidas para .png para a realização dessa pesquisa.

2.4. Algoritmo

O algoritmo utilizado foi um algoritmo de *machine learning* feito para analisar imagens. Ele recebe dados de entrada e então treina com esses dados e calcula a própria precisão, usando a própria entrada e se testando com essas mesmas imagens.

2.5. Experimento de pesquisa

A pesquisa inicialmente proposta foi realizar um experimento em grupo com alunos da Fatec Sorocaba, Fatec Jahu e Fatec Presidente Prudente, além de alunos da faculdade HELMO situada na Bélgica. A proposta inicial para a realização deste projeto foi usar as imagens disponibilizadas do banco UBIRISV2 para realizar um treinamento de um algoritmo e depois testar sua acurácia. Foi decidido que os alunos da Fatec se comunicariam com os alunos da Bélgica para encontrar resultados no reconhecimento de íris, porém, a comunicação foi escassa, resultando na falha de alinhamento entre os integrantes da Fatec e da HELMO. Portanto, os alunos da HELMO usaram ferramentas diferentes das propostas no início da pesquisa, então os alunos da Fatec Sorocaba decidiram comparar seus resultados entre si, usando ferramentas de sua escolha.

A pesquisa de fato realizada consistiu em acessar a plataforma Custom Vision da Microsoft Azure e usar a base de dados disponibilizada para treinar o algoritmo da plataforma em diferentes domínios disponíveis. Cada um dos integrantes escolheu um

domínio entre: Geral, feito pelo integrante 1, Geral[A1] e Geral[A2], feito pelo integrante 2; o domínio usado nesta pesquisa foi o Geral[A1].

Cada integrante executou separadamente o experimento com o domínio escolhido, utilizando o mesmo conjunto de imagens, tanto para o treinamento, como para o teste. Ao final do experimento foi realizada uma comparação do desempenho dos domínios utilizado pelos integrantes.

Ressalta-se que, como o objetivo desta pesquisa foi usar um algoritmo genérico para reconhecimento de imagens, as etapas descritas em 1.1.5. citadas por Delgado (2013) não foram utilizadas.

2.6. Critérios para avaliação da pesquisa

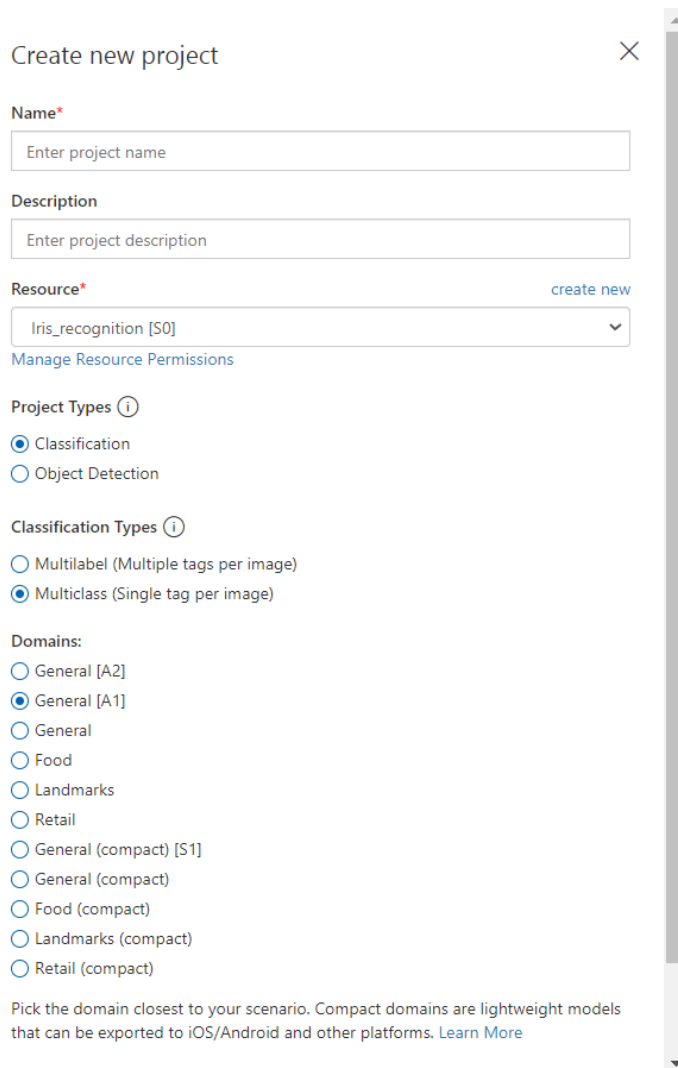
Dentre todas as 260 classes de imagens na base de dados, foram selecionadas 10 classes com 60 imagens para a pesquisa. Das 10 classes selecionadas, 50 imagens foram usadas para treinamento e 10 imagens foram usadas para teste. O critério usado para a avaliação foi selecionar três tipos de classe:

- Usando óculos: quatro classes tendo 30 imagens de pessoas usando óculos e 30 imagens sem óculos, usando apenas as imagens sem óculos para teste. O objetivo desse critério foi analisar se o algoritmo é capaz de reconhecer a qual classe pertence a imagem da pessoa sem óculos baseada nas imagens de teste em que ela usava óculos.
- Olho um pouco tampado: duas classes em que as imagens dos olhos estavam tampadas por cabelo ou os olhos estavam um pouco fechados foram usados para analisar se o algoritmo é capaz de reconhecer a qual classe pertence as imagens com olhos tampados.
- Olhos normais: quatro classes em que as imagens dos olhos estavam completamente nítidas. Esse critério teve como objetivo comparar o desempenho do algoritmo usando imagens sem obstrução da íris.

3. Descrição do experimento

Para usar a plataforma do Custom Vision, foi primeiro necessário criar uma conta na Microsoft Azure com um método de pagamento, o que possibilitou a realização desta pesquisa. Ao fazer login na ferramenta, foi criado um projeto especificando as configurações necessárias para a criação de um projeto conforme a figura 5.

Figura 5: Configurações necessárias para criação de um projeto.



The screenshot shows the 'Create new project' form in the Microsoft Custom Vision interface. The form includes the following fields and options:

- Name***: A text input field with the placeholder 'Enter project name'.
- Description**: A text input field with the placeholder 'Enter project description'.
- Resource***: A dropdown menu showing 'Iris_recognition [S0]' with a 'create new' link to the right. Below the dropdown is a link 'Manage Resource Permissions'.
- Project Types**: Two radio button options: 'Classification' (selected) and 'Object Detection'.
- Classification Types**: Two radio button options: 'Multilabel (Multiple tags per image)' and 'Multiclass (Single tag per image)' (selected).
- Domains**: A list of radio button options: 'General [A2]', 'General [A1]' (selected), 'General', 'Food', 'Landmarks', 'Retail', 'General (compact) [S1]', 'General (compact)', 'Food (compact)', 'Landmarks (compact)', and 'Retail (compact)'. Below this list is a note: 'Pick the domain closest to your scenario. Compact domains are lightweight models that can be exported to iOS/Android and other platforms. [Learn More](#)'.

Fonte: elaborado pelo autor.

É necessário adicionar um nome e um grupo de recursos (*Resource*), que pode ser criado pelo próprio autor do projeto ou usar um grupo de recursos do Azure. Nesta pesquisa, foi criado um grupo chamado *Iris_recognition* [S0], S0 sendo o fator padrão de usuários pagos da Azure. Entre os tipos de projetos possíveis, foi utilizado o tipo de classificação para classificação de imagem, mas a plataforma também permite criar projetos para reconhecimento de objetos. Em tipos de classificação, há duas opções: Multirótulo ou Multiclasse. A classificação multirótulo aplica qualquer número de classe a uma imagem (zero ou mais), enquanto a classificação multiclasse agrupa as imagens em categorias únicas (cada imagem enviada será classificada de acordo com a classe mais provável).

Por último, a plataforma disponibiliza diversos métodos para aplicar em um projeto. Cada domínio otimiza o classificador do algoritmo para tipos específicos de imagem, como comida, pontos de referência, varejo e geral. O domínio Geral é usado para classificações que não se encaixam nos outros domínios; ele varia entre:

- **Geral:** Otimizado para uma ampla gama de tarefas de classificação de imagens.
- **Geral[A1]:** Otimizado para maior precisão com o tempo de inferência comparável com o domínio Geral. É recomendado para conjuntos de dados maiores ou para cenários de usuário mais difíceis, requerendo mais tempo de treinamento.
- **Geral:[A2]:** Otimizado para maior precisão com tempo de inferência mais rápido do que os domínios Geral[A1] e Geral. É recomendado para a maioria dos conjuntos de dados. Esse domínio requer menos tempo de treinamento do que domínios Geral e Geral[A1].

O domínio utilizado nessa pesquisa foi o Geral[A1], que se sobressai em comparação ao Geral mas é inferior ao Geral[A2].

Ao criar o projeto, foram adicionadas as imagens de treinamento separada por classes conforme a figura 6.

Figura 6: Upload de imagens.

The screenshot shows a web interface titled "Image upload" with a close button (X) in the top right corner. Below the title is a progress bar with three stages: "Add Tags" (indicated by a blue circle), "Uploading" (indicated by a grey circle), and "Summary" (indicated by a grey circle). Below the progress bar is a grid of 18 small images of human eyes, arranged in three rows of six. Below the grid, the text "50 images will be added..." is displayed. Underneath this is a text input field with the placeholder "Add some tags to this batch of images...". Below the input field is a section labeled "My Tags" containing a single tag "Classe 1" in a box. At the bottom right of the interface is a blue button labeled "Upload 50 files".

Fonte: elaborada pelo autor.

Foram adicionadas 50 imagens de cada classe e adicionada a Tag com o nome da classe à qual as imagens pertencem. As outras 10 imagens da classe foram separadas para teste. Após adicionar as 10 classes, totalizando 500 imagens para treinamento do modelo, foi selecionado o tipo de treino desejado. A ferramenta permite dois tipos de treinamento, sendo eles o *Quik training*, um treino rápido de 10 minutos e *Advanced Training*, no qual pode-se treinar o modelo por mais tempo, tendo 1 hora como tempo mínimo de treino e 96 horas como tempo máximo. A própria ferramenta informa sobre o tempo de treinamento, dizendo que quanto mais tempo o modelo for treinado, melhor serão os resultados. O classificador de desempenho usado no modelo foi o *Cross Validation*.

4. Resultados obtidos

4.1. Resultados desta pesquisa

O treinamento escolhido primeiramente foi o *Quik training*, sendo o modelo treinado por apenas 10 minutos.

São apresentados dois tipos de resultados para avaliar a precisão do modelo, sendo elas:

- **Precisão:** indica a fração de classificações identificadas que estão correta por classe. Por exemplo, se o modelo identificou 100 imagens na classe como gatos e 99 delas são realmente de gatos, a precisão da classe é de 99%.
- **Recall:** a recuperação indica a fração de classificações reais que foram corretamente identificadas. Por exemplo, se há de fato 100 imagens de pássaros e o modelo identifica 80 como pássaros, a recuperação é de 80%.

Como o domínio Geral[A1] requer mais tempo de treinamento (conforme descrito no item 3. Descrições do experimento), o modelo apresentou alta precisão, mas baixo Recall. Como mostra a figura 7, o Recall foi de apenas 44% e a precisão foi de 100%.

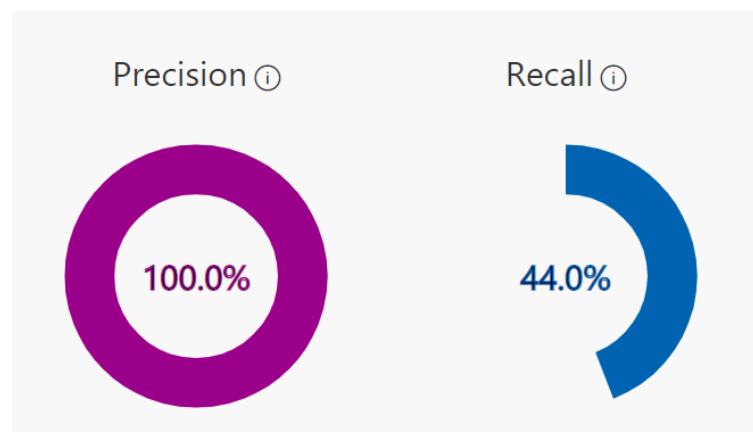
Figura 7: Indicadores de avaliação.

Iteration 1

Finished training on **16/03/2022 21:57:59** using **General [A1]** domain

Iteration id: **8ea13898-c314-403c-86c6-03bf43147167**

Classification type: **Multiclass (Single tag per image)**



Fonte: elaborada pelo autor.

O modelo também apresentou a performance para cada Tag, ou seja, para cada classe conforme a figura 8.

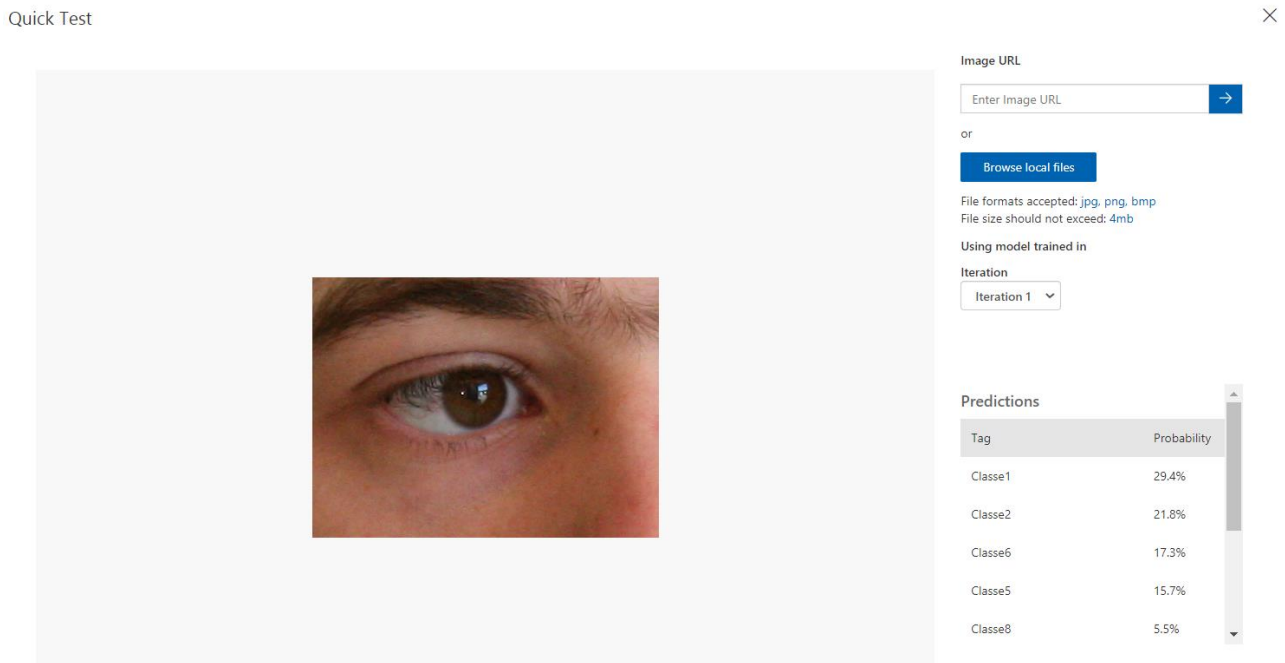
Figura 8: Performance por classe.

Tag	Precision	Recall
Classe9	100.0%	40.0%
Classe8	100.0%	80.0%
Classe7	100.0%	100.0%
Classe6	100.0%	30.0%
Classe5	100.0%	30.0%
Classe4	100.0%	40.0%
Classe3	100.0%	30.0%
Classe2	100.0%	70.0%
Classe10	100.0%	20.0%
Classe1	0.0%	0.0%

Fonte: elaborada pelo autor.

Para testar o modelo criado, foram usadas 10 imagens de cada classe, tendo sido manualmente e individualmente colocadas no modelo, totalizando 100 imagens para teste com o intuito de validar o modelo criado. A figura 9 representa uma imagem inserida no modelo, mostrando que é possível adicionar apenas uma imagem de teste por vez, tornando o processo de teste mais demorado. Ao inserir a imagem de teste, o modelo mostra uma porcentagem representando a predição da imagem inserida pertencer a uma determinada classe.

Figura 9: inserindo uma imagem de teste.



Fonte: elaborada pelo autor.

Ao inserir todas as imagens de teste no modelo e obter sua predição, os dados foram inseridos em uma tabela, usando apenas a predição correta das imagens, ou seja, apenas a probabilidade de a imagem de teste pertencer a sua própria classe. Após inserir todos os dados em porcentagem na tabela, foi tirada uma média para cada classe e, logo após, uma média geral para o modelo conforme mostra a tabela 2.

Tabela 2: Precisão geral e médias.

	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Classe 7	Classe 8	Classe 9	Classe 10
Imagem 1	29,4	51	35,8	13,5	43,3	46,9	88,3	88,2	24,6	19,4
Imagem 2	27,6	49,4	35,6	18,3	34,2	55,2	93,5	61,6	24,4	57,9
Imagem 3	12,6	31,1	12,2	18,3	18,2	37,7	94,9	60,2	36,1	44,3
Imagem 4	22,4	33,2	16,8	14,7	29,1	51,1	89,8	85,4	32,9	16,7
Imagem 5	16,6	42,3	6,1	21,7	36,9	32,4	89,5	77,2	30,6	27
Imagem 6	16,4	23,6	7,2	22,2	35,6	15,4	94,4	31	25,5	11,4
Imagem 7	19,4	38,4	7,6	8,2	27,1	55,8	96,1	25,2	23,3	32,6
Imagem 8	29,5	56,8	28,9	11	28,4	11,8	93,5	76,7	28,2	48,2
Imagem 9	7,5	58,9	7,4	44,9	30,8	27,2	63,8	63,4	29,9	27,2
Imagem 10	16,6	24,3	20,9	43,4	26,2	26	57,4	35,5	30,6	8,5
Médias	19,8	40,9	17,85	21,62	30,98	35,95	86,12	60,44	28,61	29,32
Acurácia	37,15									

Fonte: elaborada pelo autor.

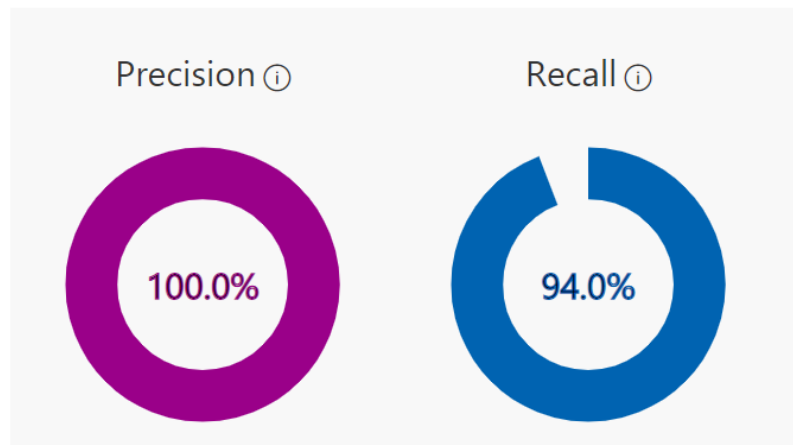
Como o tempo treinamento do modelo foi de apenas 10 minutos e o domínio Geral[A1] requer mais tempo de treinamento, o modelo apresentou uma baixa acurácia para cada classe e para o modelo em geral. Portanto, um novo treinamento foi realizado usando as mesmas imagens para treino e para teste, mas levando 1 hora de duração.

Conforme as figuras 10 e 11, pode-se ver que o tempo de treinamento influenciou nos indicadores de avaliação e na performance por classe.

Figura 10: indicadores de avaliação com maior tempo de treino.

Iteration 2

Finished training on **17/03/2022 22:12:10** using **General [A1]** domain
Iteration id: **a6f3f5a2-0cdb-424f-a72b-fff3d4283793**
Classification type: **Multiclass (Single tag per image)**



Fonte: elaborada pelo autor.

Figura 11: Performance por classe com maior tempo de treinamento.

Tag	Precision	^	Recall
Classe 9	100.0%		80.0%
Classe 8	100.0%		100.0%
Classe 7	100.0%		100.0%
Classe 6	100.0%		100.0%
Classe 5	100.0%		100.0%
Classe 4	100.0%		90.0%
Classe 3	100.0%		90.0%
Classe 2	100.0%		90.0%
Classe 10	100.0%		100.0%
Classe 1	100.0%		90.0%

Fonte: elaborada pelo autor.

O modelo também apresentou médias gerais e para cada classe com um desempenho bem maior com relação ao treinamento de 10 minutos como pode ser visto na tabela 3.

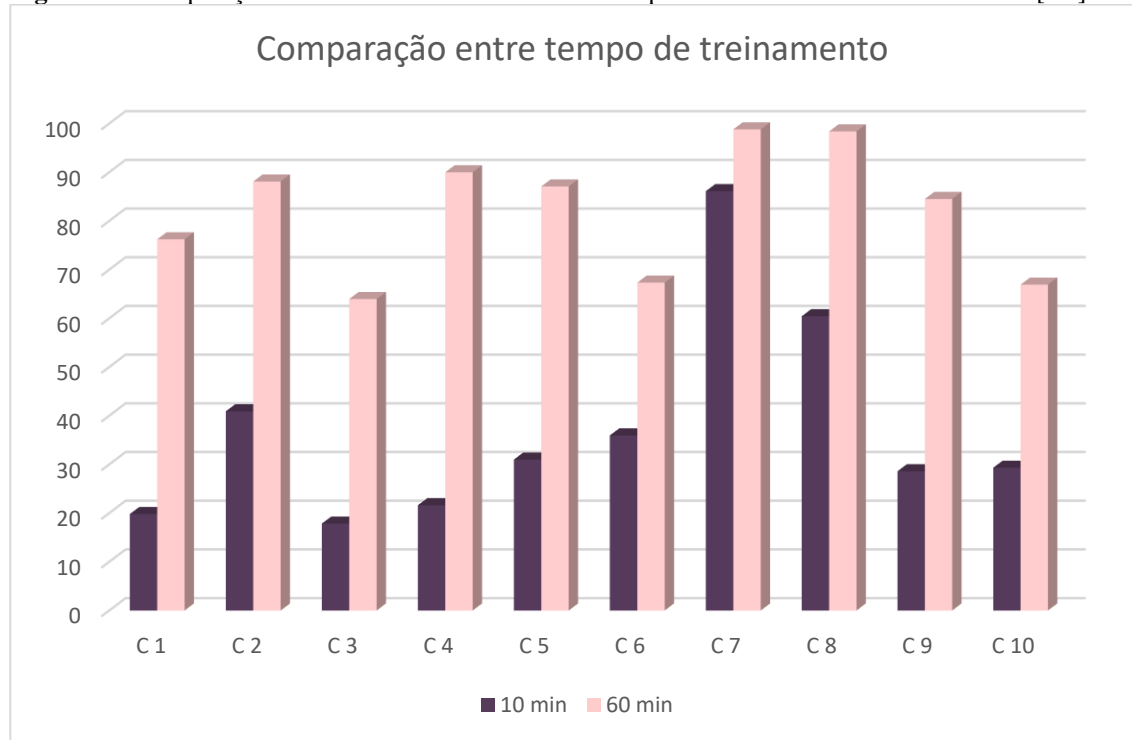
Tabela 3: Precisão geral e médias com maior tempo de treinamento do domínio Geral[A1].

	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Classe 7	Classe 8	Classe 9	Classe 10
Imagem 1	64,4	95,4	86,4	93,4	99,5	84,4	97,8	99,8	90,1	70,3
Imagem 2	98,6	96,4	87,9	96,4	97,9	87,2	99,4	99,7	94,8	93,8
Imagem 3	86,7	67,8	84,6	93,6	65,1	56,3	99,8	99,6	95,4	71,7
Imagem 4	93,2	77,2	81,9	82,3	90,9	72,6	98,9	99,9	97,2	1,2
Imagem 5	50,9	95,1	32,7	88	96,4	65,4	99,2	99,5	86,9	76,5
Imagem 6	92,4	77,4	44	85,5	60,6	87,5	99,8	97,1	84,7	79,5
Imagem 7	96,2	84,2	26,2	91,6	98,6	23,1	99,4	95,3	75	85,4
Imagem 8	97,1	98,5	93,6	86,1	84,1	6,7	99,7	99,4	69,6	94,2
Imagem 9	24,3	98,4	30,5	86,4	95,7	98,9	97,2	96,2	78,3	95,8
Imagem 10	58,5	90,5	71,8	96,5	82,1	91,2	96,8	97,3	73,1	0,8
Médias	76,23	88,09	63,96	89,98	87,09	67,33	98,8	98,38	84,51	66,92
Acurácia	82,12									

Fonte: elaborada pelo autor.

Para uma melhor compreensão de como o tempo de treinamento afetou drasticamente a acurácia do modelo, foi gerado um gráfico para visualização do desempenho do modelo que pode ser visto na figura 12. Os elementos do gráfico mostram valores de 0 a 100 que representam a acurácia do modelo, e as classes são representadas pelas abreviaturas, como por exemplo: C 1.

Figura 12: Comparação das acurácias de acordo com o tempo de treinamento do domínio Geral[A1].



Fonte: elaborado pelo autor.

Como pode ser visto analisando o gráfico, as classes em que o algoritmo mais teve dificuldade de encontrar a qual classe pertencia a imagem de teste foram as classes 3, 6 e 10. A classe 3 continha imagens de olhos com e sem óculos e foram usadas apenas imagens sem óculos para teste, já as classes 6 e 10 possuíam imagens com olhos tampados, no caso da classe 6, com cabelo na frente dos olhos e a classe 10 os olhos estavam um pouco fechados. Ambas as classes usaram as imagens dos olhos mais tampados como teste.

A classe com maior porcentagem de acerto foi a classe 7, na qual os olhos pertenciam a uma pessoa idosa.

Com base nos resultados apresentados, é possível chegar à conclusão de que o algoritmo do *Custom Vision* no domínio Geral[A1] obtém maior acurácia do modelo quando é treinado por mais tempo. Apesar de ter apresentado ótimos resultados, o algoritmo de *machine learning* usado na ferramenta não é próprio para reconhecimento de íris, e sim para reconhecimento de imagem e de objeto, levando em consideração que a maior precisão por classe pertencente a uma pessoa idosa, então o algoritmo se baseou na imagem inteira ao invés de apenas a íris.

4.2. Comparação dos resultados entre os integrantes

Após cada integrante realizar seus próprios experimentos, os resultados foram compartilhados para comparação dos modelos. Os integrantes criaram um modelo utilizando as mesmas imagens de treino e teste para comparação. O integrante 1⁵ utilizou o domínio Geral com tempo de treinamento de 1 hora e obteve os resultados representados pela tabela 4.

Tabela 4: Precisão geral e médias com maior tempo de treinamento do domínio Geral.

	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Classe 7	Classe 8	Classe 9	Classe 10
Imagem 1	99,5	100,0	99,9	99,9	89,4	63,6	99,9	99,9	99,9	99,8
Imagem 2	99,9	100,0	99,8	99,9	99,9	98,3	99,9	99,9	99,9	99,9
Imagem 3	95,8	98,3	99,2	98,9	97,0	98,1	99,9	99,9	99,9	96,4
Imagem 4	99,8	99,9	47,1	98,3	84,8	97,8	99,9	99,9	99,9	0,0
Imagem 5	60,5	100,0	92,5	76,2	67,5	83,9	99,9	99,9	96,6	95,1
Imagem 6	99,9	99,9	91,2	8,5	90,7	99,9	99,9	96,5	98,5	44,6
Imagem 7	99,9	99,9	7,9	99,8	99,2	9,1	99,9	98,1	99,6	75,6
Imagem 8	99,9	99,9	83,1	96,7	98,8	40,0	99,9	99,9	83,9	99,4
Imagem 9	68,5	100,0	2,9	92,6	50,4	99,9	82,1	99,8	99,8	97,8
Imagem 10	10,5	99,9	90,4	97,7	99,8	99,9	99,1	99,3	98,8	0,0
Medias	99,65	99,90	90,80	98,00	93,85	97,95	99,90	99,90	99,70	95,75
Acurácia	98,9									

Fonte: elaborado pelo integrante 1.

Logo após, o integrante 1 gerou um gráfico representando a comparação do tempo de 15 minutos e 1 hora pelo domínio Geral como pode ser visto na figura 13.

⁵ Lucas Shinji Yamane

Figura 13: Comparação das acurácias de acordo com o tempo de treinamento do domínio Geral.



Fonte: elaborado pelo integrante 1.

O integrante 2⁶ fez o mesmo processo, mas usando o domínio Geral[A2] e obteve os resultados representados pela tabela 5.

Tabela 5: Precisão geral e médias com maior tempo de treinamento do domínio Geral[A2].

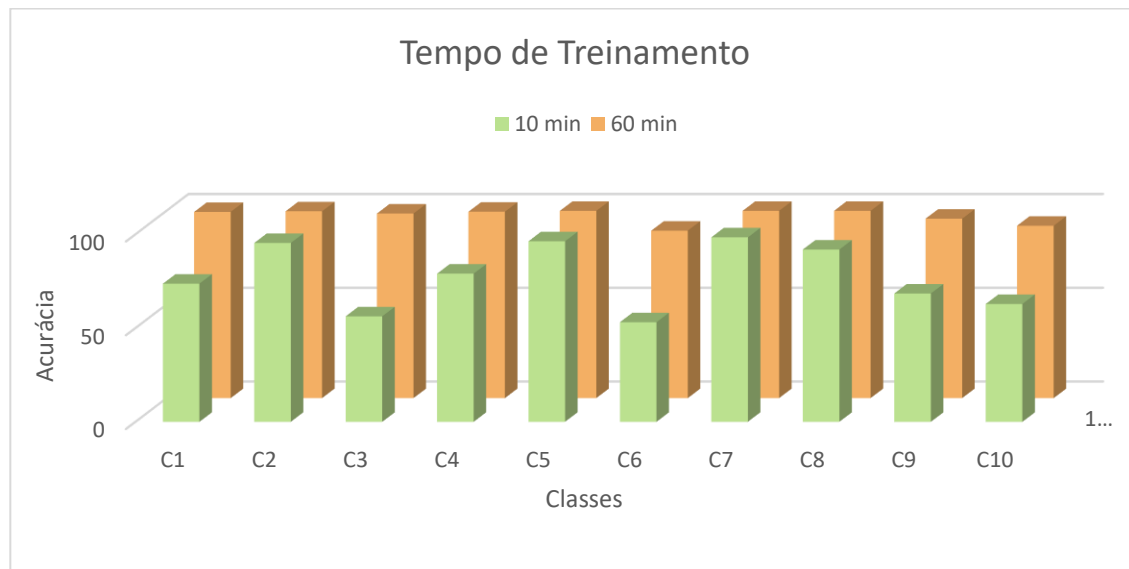
	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Classe 7	Classe 8	Classe 9	Classe 10
Imagem 1	99,4	99,9	99,9	99,9	99,9	99,4	99,9	99,9	99,9	99,7
Imagem 2	99,9	99,9	99,9	99,9	99,8	96,8	99,9	99,9	99,9	99,9
Imagem 3	99,8	99,5	99,3	99,9	99,9	96,6	99,8	99,9	99,8	99,6
Imagem 4	99,9	98,2	98,9	99,9	99,7	99,9	99,9	99,9	99,6	98,6
Imagem 5	98,4	99,9	99,8	99,9	99,9	86,6	99,9	99,9	99,9	99,6
Imagem 6	99,9	99,5	94,4	94,8	99,9	98,9	99,8	99,9	58,8	99,9
Imagem 7	99,7	99,9	99,2	99,9	99,8	96,9	99,9	99,9	99,9	99,8
Imagem 8	99,9	99,8	99,9	99,9	99,9	17,8	99,8	99,9	99,9	99,9
Imagem 9	99,5	99,9	95,5	99,9	99,9	99,9	99,9	99,9	99,5	99,9
Imagem 10	96,5	99,8	97,6	99,8	99,6	99,9	99,9	99,7	99,5	22,3
Médias	99,29	99,63	98,44	99,38	99,83	89,27	99,87	99,88	95,67	91,92
Acurácia	97,32									

Fonte: elaborada pelo integrante 2.

⁶ Larissa Cristina Jarduli Leite

Logo após, o integrante 2 gerou um gráfico representando a comparação do tempo de 15 minutos e 1 hora pelo domínio Geral[A2] como pode ser visto na figura 14.

Figura 14: Comparação das acurácias de acordo com o tempo de treinamento do domínio Geral[A2].



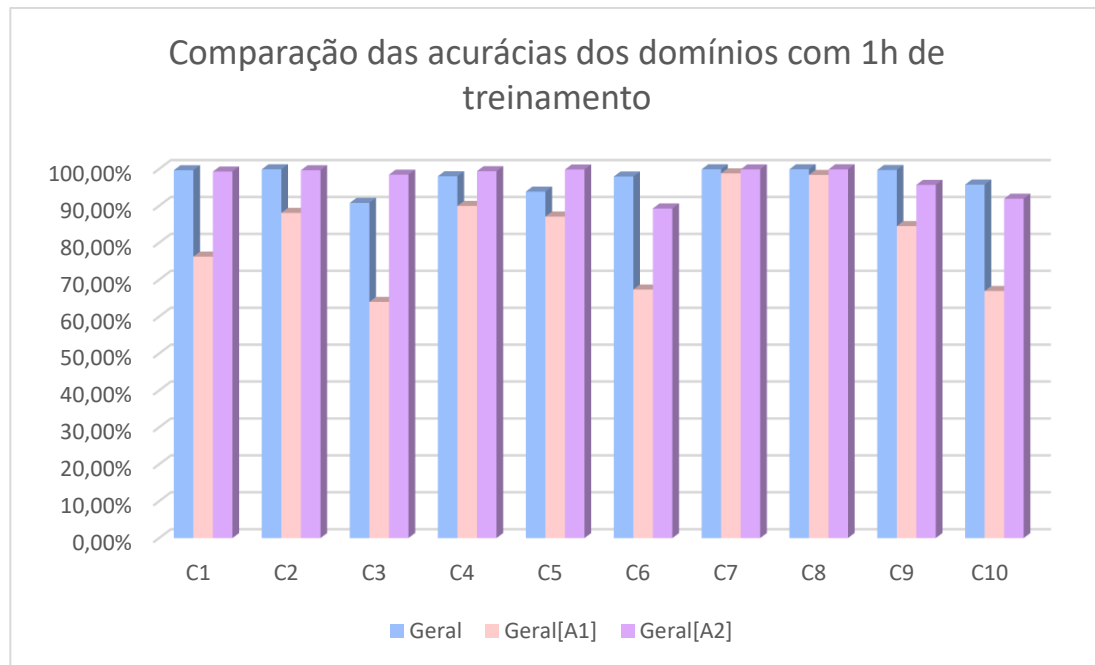
Fonte: elaborado pelo integrante 2.

Portanto, foi possível comparar as acurácias dos domínios quando usado os mesmos dados para treinamento e teste e tempo de 1 hora de treinamento, e elas foram:

- **Geral:** 98,8%
- **Geral[A1]:** 82,12%
- **Geral[A2]:** 97,32

A figura 15 representa a comparação das acurácias entre todos os domínios estudados usando o treinamento de 1 hora para melhor compreensão.

Figura 15: Comparação das acurácias dos domínios estudados.



Fonte: elaborada pelo autor.

Ao observar as acurácias dos modelos, foi possível identificar que quanto maior o tempo de treinamento, maior são as acurácias e que em todos os domínios, as classes 3, 6 e 10 tiveram as menores precisões no treinamento de 1 hora. A figura 16 apresenta essas classes e suas respectivas acurácias indicando a dificuldade dos modelos de cada domínio em reconhecer imagens da classe 3,6 e 9.

Figura 16: precisões por classe entre domínios.



Classe 3

Geral: 90,80%
Geral[A1]: 63,93%
Geral[A2]: 98,44%



Classe 6

Geral: 97,95%
Geral[A1]: 67,33%
Geral[A2]: 89,27%



Classe 10

Geral: 95,75%
Geral[A1]: 66,92%
Geral[A2]: 91,92%

Fonte: elaborada pelo autor.

5. Considerações finais

Esta pesquisa teve o seguinte problema de pesquisa: “Qual é o desempenho de algoritmos de reconhecimento de imagem genéricos quando utilizado no reconhecimento de íris?”. Ao fim de todo o experimento, foi possível chegar às seguintes considerações finais:

A plataforma *Custom Vision* utiliza um algoritmo de *machine learning* para classificação de imagens e cada domínio tem sua especificidade e requisitos de tempo de treinamento. Após a análise dos resultados, foi possível perceber que o domínio estudado nessa pesquisa, o Geral[A1], requer mais tempo de treinamento que os outros domínios gerais e, mesmo tendo sido treinado por 1 hora, ainda obteve a menor acurácia dentre todos os domínios Gerais estudados.

Apesar da baixa acurácia do domínio Geral[A1], os demais domínios gerais obtiveram ótimas acurácias apesar de serem algoritmos genéricos não especializados em reconhecimento de íris, mas os modelos apresentaram “dificuldade” em reconhecer classes em que as imagens dos olhos estavam obstruídas de alguma forma, sendo por óculos, por ter cabelo ou pelos olhos estarem um pouco fechados. A classe que mais obteve sucesso em ser reconhecida foi a classe que continha imagens dos olhos de uma pessoa idosa, mostrando que o algoritmo utiliza características diferentes nas imagens para poder reconhecê-las, como por exemplo a área periocular.

O tempo de treinamento foi um fator decisivo na comparação das acurácias, pois em todos os domínios estudados, a acurácia foi menor quando testada por menor tempo. Pelos resultados obtidos, foi possível perceber que quanto maior o tempo de treinamento, maior é a acurácia obtida pelo modelo.

Portanto, conclui-se que os algoritmos de reconhecimento genéricos podem exibir bons resultados para reconhecimento de íris, mas procuram por características semelhantes nas imagens para reconhecer a qual classe a imagem de teste pertence, e não procura pela íris de fato, podendo, assim, falhar em alguns casos no reconhecimento de íris propriamente dito, mas não se descarta a eficiência dos algoritmos genéricos, uma vez que obtiveram bom desempenho na classificação.

Conclui-se também que o tempo de treinamento afeta na acurácia geral do modelo, sendo recomendado maior tempo de treinamento para verificar sua acurácia.

Pode-se destacar ainda que o domínio Geral[A1], por ser recomendado para cenários de usuário mais difíceis, requer maior tempo de treinamento, pois quando treinado apenas pelo treinamento rápido da plataforma, os resultados foram lastimáveis, podendo-se descartar completamente o uso deste domínio para reconhecimento de íris, a menos que o tempo de treinamento seja maior. Mesmo aumentando o tempo de treinamento, o Geral[A1] obteve baixo desempenho comparado aos demais domínios, mais baixo até mesmo que o Geral; portanto, o modelo Geral[A1] só deve ser utilizado para reconhecimento de íris quando treinado pelo maior tempo possível.

Por fim, mediante os resultados obtidos na presente pesquisa, é possível afirmar que a hipótese foi parcialmente confirmada e que os objetivos foram alcançados.

Referências⁷

- ABIYEV, Rahib H.; ALTUNKAYA, Koray. Personal Iris Recognition Using Neural Network. **International Journal Of Security And Its Applications**. [S. L.]. Maio 2008.
- ALAVALA, Chennakesava R.; GUDWIN, Ricardo Ribeiro. **Fuzzy Logic and Neural Networks: Basic Concepts and Applications**. [S. L.]: New Age Publications (Academic), 2008. 276 p.
- ALBOY, Renan Galeane. **Técnicas de reconhecimento de imagem para incorporação em ferramentas de auxílio a deficientes visuais**. 2019. 83 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de São Carlos, São Carlos, 2019. Disponível em: <https://repositorio.ufscar.br/bitstream/handle/ufscar/11810/Disserta%C3%A7%C3%A3o-Final.pdf?sequence=4>. Acesso em: 16 ago. 2021.
- BASSO, Bianca Cavalcante. **Inteligência artificial para reconhecimento de emoções em um ambiente empresarial Indaiatuba 2019**. 2019. 59 f. TCC (Graduação) - Curso de Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Centro Estadual de Educação Tecnológica Paula Souza Faculdade de Tecnologia de Indaiatuba, Indaiatuba, 2019.
- DAMIÃO, Mateus Araujo; CAÇADOR, Rodrigo Menezes Costa; LIMA, Sérgio Muinhos Barroso. Princípios e aspectos sobre agentes inteligentes. **Revista Eletrônica da Faculdade Metodista Granbery**, Juiz de Fora, v. 17, jun. 2014. Disponível em: <http://re.granbery.edu.br/artigos/NTIw.pdf>. Acesso em: 21 set. 2021. Disponível em: <https://ieeexplore.ieee.org/document/8712430>. Acesso em: 26 ago. 2021.
- DELGADO, Edmundo Daniel Hoyle. **Reconhecimento biométrico usando informação da íris e de características perioculares**. 2013. 88 f. Tese (Doutorado) - Curso de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013.
- DOURO, Rômulo Ferreira. **Reconhecimento de Formas em Imagens Através da Associação de Pontos e Avaliação Multicritério de Arestas**. 2012. 126 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal do Espírito Santo, Vitória, 2012. Disponível em: http://repositorio.ufes.br/bitstream/10/4259/1/tese_5456_.pdf. Acesso em: 14 out. 2021.
- GARCÍA-ORDÁS, María Teresa; BENÍTEZ-ANDRADES, José Alberto; GARCÍA-RODRÍGUEZ, Isaías; BENAVIDES, Carmen; ALAIZ-MORETÓN, Héctor. Detecting Respiratory Pathologies Using Convolutional Neural Networks and Variational

⁷ Referências elaboradas com uso da ferramenta MORE da UFSC. MORE: Mecanismo online para referências, versão 2.0. Florianópolis: UFSC. Disponível em: < <http://www.more.ufsc.br/>

Autoencoders for Unbalancing Data. **Sensors**, [S.L.], v. 20, n. 4, p. 1214, 22 fev. 2020. MDPI AG. <http://dx.doi.org/10.3390/s20041214>. Acesso em: 08 out. 2021.

GÉRON, Aurélien. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow**: conceitos, ferramentas e técnicas para a construção de sistemas inteligentes. Rio de Janeiro: Alta Books, 2019. 577 p.

GIL, A. C. Métodos e técnicas de pesquisa social. 5. ed. São Paulo: Atlas, 2007.

GOMIDE, F. A. C.; GUDWIN, R. R.: **Modelagem, Controle, Sistemas E Lógica Fuzzy**. 1994. Universidade Estadual de Campinas (UNICAMP). Disponível em: <https://www.dca.fee.unicamp.br/~gudwin/ftp/publications/RevSBA94.pdf>. Acesso em: 15 set. 2021.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. [S. L.]: The Mit Press, 2016. 775 p.

HAN, J.; KAMBER, M., PEI, J. **Data mining**: concepts and techniques. 2. ed. São Francisco: Morgan Kaufmann, 2011. 770 p.

HAYKIN, Simon. **Redes Neurais**: princípios e prática. 2. ed. [S. L.]: Bookman, 2001. 898 p.

JAVIDI, Bahram. **Image Recognition and Classification**: algorithms, systems, and applications. Nova York: Marcel Dekker, 2002. 493 p.

JMOUR, Nadia; ZAYEN, Sehla; ABDELKRIM, Afef. International conference on advanced systems and electric technologies (IC_ASET), 2018, Hammamet. **Convolutional neural networks for image classification**. Tunis: Ieee Xplore, 2018. 6 p. Disponível em: <https://ieeexplore.ieee.org/document/8379889>. Acesso em: 23 ago. 2021.

LI, Stan Z.; JAIN, Anil K. **Encyclopedia of Biometrics**. 2. ed. [S. L.]: Springer, 2015. 1651 p.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **Nature**. [S. L.]. 27 maio 2015. Disponível em: <https://www.nature.com/articles/nature14539#citeas>. Acesso em: 20 ago. 2021.

M. Pritt; G. Chern. "Satellite Image Classification with Deep Learning," 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Washington, DC, USA, 2017 pp. 1-7. doi: 10.1109/AIPR.2017.8457969. Disponível em: <https://doi.ieeecomputersociety.org/10.1109/AIPR.2017.8457969>. Acesso em: 25 ago. 2021.

MENDES, Raquel Dias. **Inteligência artificial: sistemas especialistas no gerenciamento da informação**. Ciência da Informação, [S.L.], v. 26, n. 1, p. 39-45, jan. 1997.

FapUNIFESP (SciELO). Disponível em: <https://doi.org/10.1590/S0100-19651997000100006>. Acesso em: 16 set. 2021.

MINAEE, Shervin; ABDOLRASHIDI, Amirali. **DeepIris: Iris Recognition Using A Deep Learning Approach**. New York: Cornell University, 2019. 4 p.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: REZENDE, S. O. (Org.). **Sistemas inteligentes: fundamentos e aplicações**. Barueri: Manole, 2003. p. 89-114.

ÖZYURT, Fatih. Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures. **The Journal Of Supercomputing**, [S.L.], v. 76, n. 11, p. 8413-8431, 14 dez. 2019. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/s11227-019-03106-y>. Disponível em: <https://link.springer.com/article/10.1007%2Fs11227-019-03106-y>. Acesso em: 18 ago. 2021.

PONTI, Moacir A.; COSTA, E Gabriel B. Paranhos da. **Tópicos em Gerenciamento de Dados e Informações**. [S. L.]: Sbc, 2017. Disponível em: <https://sol.sbc.org.br/livros/index.php/sbc/catalog/book/31>. Acesso em: 20 ago. 2021.

RODRIGUES, Diego Alves. **Deep learning e redes neurais convolucionais: reconhecimento automático de caracteres em placas de licenciamento automotivo**. 2018. 37 f. Monografia (Especialização) - Curso de Ciência da Computação, Centro de Informática, Centro de Informática Universidade Federal da Paraíba, João Pessoa, 2018. Disponível em: <https://repositorio.ufpb.br/jspui/bitstream/123456789/15606/1/DAR20052019.pdf>. Acesso em: 15 ago. 2021.

RUSSELL, Stuart; NORVIG, Peter. **Inteligência Artificial**. 3. ed. [S. L.]: Elsevier, 2013. 1016 p.

SAHA, Sumit. **A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way**. [S.L.], 15 dec. 2018. Medium: @_sumitsaha_. Disponível em: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. Acesso em: 27 set. 2021.

TRAORE, Boukaye Boubacar; KAMSU-FOGUEM, Bernard; TANGARA, Fana. **Deep convolution neural network for image recognition**. 2018. 13 f. Curso de Tecnologia, B Université Des Sciences, Des Techniques Et Des Technologies de Bamako (Usttb), Faculté Des Sciences Et Techniques Colline de Badalabougou, Toulouse, 2018. Disponível em: https://oatao.univ-toulouse.fr/21329/1/Traore_21329.pdf. Acesso em: 15 ago. 2021.

VERDHAN, Vaibhav. **Computer Vision Using Deep Learning: Neural Network Architectures with Python and Keras**. [S. L.]: Apress, 2021. 308 p.

VON WANGENHEIM, Christiane Gresse; VON WANGENHEIM, Aldo; RATEKE, Thiago. **Raciocínio Baseado em Casos**. 2. ed. [S. L.]: Bookess, 2013. 76 p. Disponível em: https://www.researchgate.net/publication/262374659_Raciocinio_Baseado_em_Casos_-_2_ed_Revisada_e_Atualizada. Acesso em: 21 set. 2021.

WITTEN, I. H.; FRANK, E. **Data mining**: practical machine learning tools and techniques. 2nd ed. San Francisco: Morgan Kaufmann, 2005.

WU, Meiyin; CHEN, Li. **Image Recognition Based on Deep Learning**. 2015. 5 f. Monografia (Especialização) - Curso de Science And Technology, College Of Computer Science And Technology, Wuhan University Of Science And Technology Key Laboratory Of Intelligent Information Processing And Real-Time Industrial System, Wuhan, 2015. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7382560/>. Acesso em: 11 ago. 2021.

Y. Sun; B. Xue; M. Zhang; G. Yen. "**Evolving Deep Convolutional Neural Networks for Image Classification**," in *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 394-407, April 2020, doi: 10.1109/TEVC.2019.2916183.

YU, Shiqi; JIA, Sen; XU, Chunyan. Convolutional neural networks for hyperspectral image classification. **Neurocomputing**. [S. L.], p. 88-98. 5 jan. 2017. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0925231216310104?via%3Dihub>. Acesso em: 19 ago. 2021.