UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Programa de Pós-graduação em Ciência da Computação

Auxílio a deficientes visuais utilizando Redes Neurais Convolucionais e Competição e Cooperação de Partículas

Estudos Especiais

Jefferson Antonio Ribeiro Passerini

Rio Claro
2021

Jefferson Antonio Ribeiro Passerini

Auxílio a deficientes visuais utilizando Redes Neurais Convolucionais e Competição e Cooperação de Partículas

Monografia apresentada ao Instituto de Geociências e Ciências Exatas (IGCE) da Universidade Estadual Paulista "Júlio de Mesquita Filho", como parte dos requisitos necessários para aprovação na disciplina de Estudos Especiais do Programa de Pós-graduação em Ciência da Computação
.

Universidade Estadual Paulista – UNESP
"Júlio de Mesquita Filho"
Programa de Pós-graduação em Ciência da Computação

Orientador: Prof. Dr. Fabrício Aparecido Breve

Rio Claro
2021

RESUMO

Atualmente no mundo existem 2,2 bilhões de pessoas que possuem alguma deficiência visual, e uma das dificuldades é auxiliar essa população no seu ato diário de se locomover, assim, o desenvolvimento de ferramentas que auxiliem essa população se faz necessário. Este projeto propõe a extração de características utilizando redes neurais convolucionais VGG16+VGG19, a seleção de características utilizando os algoritmos PCA e UMAP e a classificação utilizando o modelo de aprendizado semi-supervionado de competição e cooperação de partículas (PCC), a partir de um conjunto de imagens de modo a identificar se existem obstáculos no caminho. Durante a pesquisa, foram estudados os impactos dos métodos PCA e UMAP na seleção de características e o impacto na acurácia durante a classificação com o modelo PCC. Como resultado, obteve uma acurácia de 80,88% com abordagem PCC+PCA em conformidade com análise previamente publicada e 89,26% na abordagem PCC+UMAP, o que é um resultado próximo à acuraria de redes neurais convolucionais como VGG e Xception.

Palavras-chaves: Detecção de obstáculos. Seleção de características. Redes neurais convolucionais. Transferência de aprendizado. Competição e cooperação de partículas.

ABSTRACT

Currently in the world there are 2.2 billion people who have some visual impairment, and one of the difficulties is to help this population in their daily act of getting around, so the development of tools to help this population is necessary. This project proposes the extraction of features using VGG16+VGG19 convolutional neural networks, the selection of features using the PCA and UMAP algorithms and the classification using the semi-supervised learning machine model of Particle Competition and Cooperation (PCC), from the set of images to identify whether there are obstacles in the way. During the research, the impacts of PCA and UMAP methods on feature selection and the impact on accuracy during classification with the PCC model were studied. As a result, it obtained an accuracy of 80.88% with PCC+PCA approach in accordance with previously published analysis and 89.26% with PCC+UMAP approach, which is a result close to the accuracy of convolutional neural networks such as VGG and Xception.

Keywords: Obstacle Detection. Feature Selection. Convolutional Neural Network. Transfer Learning. Particle Competition and Cooperation.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Pessoas com deficiências visuais ou totalmente cegas são um aspecto importante da sociedade atual. O aumento de ocorrência de doenças relacionadas aos olhos e à redução da visão representam, cada vez mais, um desafio às instituições e governos.

Atualmente, existem no mundo pelo menos 2,2 bilhões de pessoas que possuem algum tipo de deficiência visual, e a assistência a essas pessoas é escassa. Existe desigualdade na cobertura de atendimento, assim como na qualidade dos serviços de prevenção, tratamento e reabilitação (ONU, 2019).

Segundo o IGBE (2010), no censo demográfico realizado em 2010, o Brasil registrou mais de 35 milhões de pessoas que declaravam possuir algum acometimento que prejudicava em algum grau sua visão. Destes, 506.377 eram totalmente cegos e outros 6.056.533 possuíam grande dificuldades em enxergar.

Desta população de totalmente cegos, 27,26% se encontravam na faixa acima dos 65 anos, 59,63% entre 15 e 64 anos e 13,11% em menores de 14 anos. Se se observar a população que se declarava com grande dificuldade de visão, 29,44% da população acima de 65 anos e da população entre 15 e 64 anos, encontra-se um índice de 65,65% e, na população menor de 14 anos, têm-se 4,91% (IBGE, 2010).

Quando se olha o percentual de pessoas que declaravam alguma dificuldade de visão no último censo demográfico realizado no Brasil, tem-se que a população do país neste levantamento era de 190 milhões de pessoas, ou seja, 18,75% da população possuía algum grau de deficiência visual, sendo que, destes indivíduos, 3,18% possuíam grandes dificuldades e 0,27% eram totalmente cegos (IBGE, 2010).

A deficiência visual no Brasil e no mundo afeta a sua qualidade de vida e de seus familiares e, muitas vezes, causa desequilíbrio social se essa questão não for tratada adequadamente. Esses problemas levaram pesquisadores a explorar novos caminhos em várias disciplinas, como tecnologias assistivas, psicologia cognitiva, visão computacional, processamento sensorial, reabilitação, acessibilidade incluindo interação humano-computador.

As tecnologias assistivas facilitam o acesso dos deficientes visuais a informações, promovem a segurança, apoiam sua mobilidade e geram considerável melhora na qualidade de vida, com grande impacto direto na inclusão social (MANDUCHI; COUGHLAN, 2012).

As pesquisas de tecnologias assistivas estão focadas em mobilidade, identificação de objetos, navegação, acesso a informações a artefatos impressos e interação social. Os avanços referem a reabilitação, engenharia, tecnologias vestíveis, adaptações multissensoriais, bengalas inteligentes e, principalmente, o uso de *smartphones*. Os aplicativos abrem novas perspectivas de oportunidades para melhorar a qualidade de vida das pessoas com deficiência visual (TERVEN *et al.*, 2014).

A área de reconhecimento de objetos/obstáculos cresceu nos últimos anos e se tornou uma técnica de visão computacional bem conhecida para identificação de objetos em imagens ou vídeos. Várias pesquisas visam utilizar os recursos acessíveis e poderosos dos *smartphones* para desenvolvimento de aplicações que focam, por exemplo, o reconhecimento de obstáculos à frente do usuário. A área de visão computacional também teve muitos avanços a partir da utilização de redes neurais convolucionais (JIANG *et al.*, 2019; NEHA; SHAKIB, 2021).

Existe a necessidade de desenvolvimento de modelos que permitam ao deficiente visual aproveitar os recursos disponíveis em *smartphone*, mesmo que desconectado da rede mundial, como uma ferramenta assistiva para o seu dia a dia.

1.1 JUSTIFICATIVA

Muitos avanços foram propostos na área de visão computacional e de sistemas de navegação na última década (RIZZO *et al.*, 2017; LAKDE; PRASAD, 2015). Muitos sistemas propostos estavam baseados em equipamentos caros, pesados ou não estão amplamente disponíveis à população (HOANG *et al*., 2017; RIZZO *et al.*, 2017), ou, ainda, utilizavam como requisito para seu funcionamento uma rede de dados móvel para que obtivesse acesso a algum servidor remoto (JIANG *et al*., 2019; LIN *et al*., 2017). Deste modo a bengala continua sendo o dispositivo preferido para detecção de obstáculos pois é um equipamento portável e de baixo custo.

Com o advento dos métodos de aprendizado de máquina denominados aprendizado profundo[1] (LECUN; BENGIO; HINTON, 2015; SCHIMIDHUBER, 2015), especialmente as redes neurais convolucionais (CNN)[2] (HOWARD *et al.*, 2017; CAI *et*

[1] Do inglês: *deep learning*.
[2] Do inglês: *convolutional neural networks* (CNN).

al., 2016; KRIZHEVSKY; SUTSKEVER; HINTON, 2012) foram responsáveis por grandes avanços na detecção de objetos e na classificação de imagens, sendo a classe de redes neurais comumente utilizada para esse objetivo.

Como qualquer modelo de aprendizado supervisionado, as CNNs necessitam de uma grande quantidade de amostras (imagens) rotuladas que, na sua fase de treinamento, serão expostas a milhares de características diferentes de cada uma das amostras. O processo de rotular imagens é muito custoso e demorado para ser realizado (OQUAB *et al.*, 2014).

A utilização de transferência de aprendizado[3] permite aplicar camadas que foram treinadas em grandes conjuntos de imagens otimizando o aprendizado em aplicações de CNNs, cujas bases de treinamento são escassas, reduzindo os custos de treinamento do modelo a ser empregado (OQUAB *et al.*, 2014). A técnica de transferência de aprendizado foi utilizada com sucesso em diversas situações (GOPALAKRISHNAM *et al.*, 2017; SZEGEDY *et al.*, 2016a; SIMONYAN; ZISSERMAN, 2015).

O treinamento das CNNs é a etapa que exige o maior custo computacional, mas, uma vez treinadas, conseguem estabelecer inferências de forma relativamente rápida, o que possibilita sua utilização na maioria dos *smartphones* do mercado, permitindo a captura de uma imagem através destes aparelhos e obtendo uma inferência através de modelos algoritmos de CNNs como o VGG19 (SIMONYAN; ZISSERMAN, 2015) ou o InceptionV3 (SZEGEDY *et al.*, 2016a).

Neste cenário, é possível construir um sistema assistente que, utilizado juntamente com a bengala, permita coletar imagens do caminho a ser percorrido pelo usuário através de seu *smartphone* classificando-o como um caminho limpo ou com obstáculos. A aplicação necessitaria de nenhum recurso extra como acessórios ou conexão com a internet.

A aplicação de CNNs no cenário proposto, com todas as limitações de recursos para deixar o sistema simples e barato, teria limitações se houvesse a necessidade de coletar o retorno do usuário do sistema em relação à classificação realizada para a imagem coletada. Neste caso, seria necessário o “retreinamento” da CNN no *smartphone*, o que não seria plausível de execução.

[3] Do inglês: *transfer learning*.

Para resolver esse problema, foram propostas a utilização das CNNs como um algoritmo para extração de características e a utilização de modelos de aprendizado semi-supervisionado como o algoritmo de Competição e Cooperação de Partículas (PCC) para classificação das imagens como caminho limpo ou com obstáculos (BREVE; FISCHER, 2020).

1.2 OBJETIVOS

O objetivo principal deste trabalho é propor um modelo com a utilização de *feature learning* por meio da utilização de redes neurais convolucionais e, a partir das características extraídas, realizar a classificação se o caminho a ser trilhado pelo deficiente visual está sem obstáculos ou com obstáculos. As CNNs serão utilizadas sem a camada de classificação, ficando essa tarefa a cargo do modelo de aprendizado semi-supervionado de Competição e Cooperação de Partículas (PCC) (BREVE *et al.*, 2012).

Breve e Fischer (2020) propuseram essa abordagem, cujo modelo apresentou uma acurácia média de 72,51% quando utilizado com a rede neural convolucional VGG16 de Simonyan e Zisserman (2015) para extração de características. A acurácia média foi de 71,52% quando a extração de características foi realizada pela CNN VGG19 de Szegedy *et al.* (2016a). Por fim, combinando características extraídas por ambas as arquiteturas (VGG16+VGG19), obteve-se uma acurácia média de 73,43%.

Durante o estudo de Breve e Fischer (2020), foi utilizado o algoritmo Análise de Componentes Principais (PCA)[4] (JOLLIFFE, 2002) para reduzir a dimensionalidade das características extraídas pelas CNNs VGG16 e VGG19, antes de aplicar o classificador.

Para isso durante os estudos especiais, os objetivos específicos foram:

I. Fazer um estudo bibliográfico dos conceitos a serem utilizados durante esse projeto de pesquisa.
II. Fazer um levantamento de trabalhos correlatos avaliando metodologias utilizadas e conjuntos de imagens disponíveis para futuros testes.
III. Estabelecer uma metodologia para a continuidade da pesquisa até a fase de qualificação.

[4] Do inglês*: Principal Components Analysis* (PCA).

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho foi organizado em 6 seções: a seção comporta a introdução, com a temática sobre os deficientes visuais e sua necessidade de tecnologias assistivas; a seção 2 apresenta a fundamentação teórica para o entendimento dos conceitos e abordagens utilizadas; seção 3 traz um levantamento bibliográfico de trabalhos relevantes ao contexto da pesquisa; a seção 4 contém a metodologia proposta para o desenvolvimento do modelo; a seção 5 apresenta os resultados esperados da pesquisa; e a seção 6, com o cronograma de execução desta pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção tem como objetivo apresentar os conceitos necessários para a elucidação da metodologia proposta, tais como informações sobre as deficiências visuais, aprendizado profundo (*deep learning*), principalmente redes neurais convolucionais (CNN) e redução de dimensionalidade. Outros aspectos importantes e necessários para avaliação de atributos, tais como medidas de desempenho e classificadores, também são apresentados.

2.1 CLASSIFICAÇÃO DE NÍVEIS DE DEFICIÊNCIA VISUAL

A Classificação Internacional de Doenças – versão 10 (CID 10) estabelece quatro níveis de função visual: visão normal, deficiência visual moderada, deficiência visual grave e cegueira (WHO, 2003).

Essa classificação estabelece duas escalas oftalmológicas com parâmetros para avaliar a deficiência visual: a acuidade visual (capacidade de reconhecer objeto em determinada distância) e campo visual (amplitude da área alcançada pela visão) (WHO, 2003).

A função da visão é classificada em 6 níveis, categoria 1 – deficiência visual leve, categoria 2 – deficiência visual moderada, categoria 3 – cegueira, categoria 4 – cegueira severa, categoria 5 – cegueira muito severa, categoria 6 – cegueira total e categoria 9 – não especificado (WHO, 2003).

Os termos “cegueira legal” ou “cegueira parcial” são utilizados para classificar a deficiência visual de indivíduos que apresentam uma de duas condições: visão corrigida do melhor olho de 20/400 ou menor, ou, diâmetro mais largo do campo visual com medida inferior a 20 graus de arco, ainda que o campo visual possa ser superior à medida anterior (chamado de “visão em túnel”) (OTTAIANO *et al.*, 2019).

O nível de deficiência visual é uma informação a ser considerada, pois pessoas totalmente cegas ou indivíduos com diferentes níveis de deficiência visual possuem padrões e comportamento diferentes. Baseado na literatura, pode-se verificar que existe mudança de mentalidade causada pela forma sensorial alterada referente ao mundo exterior, o que causa impacto no padrão de comportamento do indivíduo (ANDERSON *et al.*, 2003).

Bicket *et al.* (2020) demonstraram que, em pessoas com glaucoma, de acordo com a gravidade da doença, ocorrem distúrbios na forma de andar. O glaucoma está associado a alterações de vários parâmetros do passo, como o comprimento da passada. Concluem que pessoas com mais danos em seu campo visual demonstram degradação extrema da forma de andar, o que não pode ser medido apenas pelo medo de cair.

Turano *et al.* (2001) determinaram, em seu estudo, que pessoas com retinite pigmentosa (RP) tinham comportamentos diferentes de uma pessoa com visão normal durante uma caminhada. Indivíduos com essa deficiência, quando andam, fixam seu campo visual em uma área 3 vezes maior do ambiente, e 87% das fixações foram direcionadas para objetos ou para baixo no campo visual.

Geruschat *et al.* (2006) exploraram, em seu estudo, os padrões de olhar de sujeitos com visão normal e deficientes visuais durante atividade de alto risco. Foram observados 12 indivíduos com visão perfeita, 12 com deficiência resultante de glaucoma e 9 indivíduos com deficiência visual resultante de degeneração macular relacionada à idade (DMRI). As observações foram realizadas utilizando um rastreador durante a travessia de um cruzamento. Como resultado, tem-se que todos os grupos, inicialmente, tiveram o mesmo comportamento, que era de fixar sua visão nos veículos e, à medida que atravessavam a rua, mudavam o foco de sua visão; do mesmo modo, o tempo em que identificavam os obstáculos era diferente. Deste modo, conclui-se que o *status* da visão afeta a alocação da fixação do foco visual durante essa atividade.

Aspinall *et al.* (2014) relataram que pacientes com DMRI possuem o seu campo visual central particularmente afetado, o que impacta no comportamento durante as execuções de suas tarefas diárias incluindo a navegação em comparação com pessoas com visão normal.

Assim, mudanças no processo de aprendizagem, de acordo como as experiências a que as pessoas portadoras de deficiências visuais foram submetidas, definem padrões de comportamento em diferentes estágios. Uma importante diferença observa-se nas habilidades e preferências sensoriais e cognitivas entre cegos congênitos e tardios.

Pasqualotto e Proulx (2012) observaram que a cegueira, muitas vezes, resulta na reorganização neural adaptativa das modalidades restantes, produzindo desempenho auditivo mais nítido. No entanto, as modalidades não visuais podem não

ser capazes de compensar totalmente a falta de experiência visual, como no caso da cegueira congênita.

Por exemplo, a experiência visual de desenvolvimento parece ser necessária para a maturação de neurônios multissensoriais para tarefas espaciais. Além disso, a capacidade da visão de transmitir informações em paralelo pode ser considerada como o atributo principal que não pode ser totalmente compensado pelas modalidades poupadas.

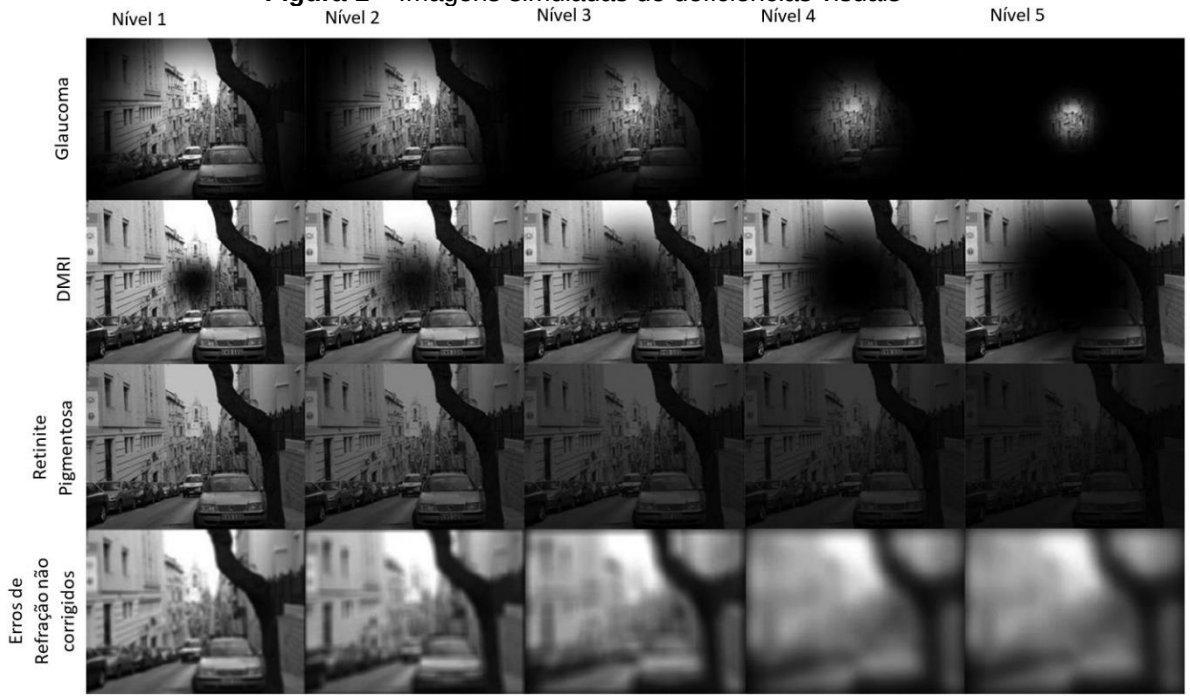
Pasqualotto e Proulx (2012) observaram que a classificação comum de cego "precoce" pode ser um equívoco, já que um ou dois anos de experiência visual pode levar ao desenvolvimento do cérebro semelhante ao de um cego tardio ou mesmo de uma pessoa com visão.

Para a cegueira congênita, as pessoas portadoras deste tipo de deficiência possuem habilidades superiores de memória auditiva (PASCHALOTTO; LAM; PROULUX, 2013).

As principais causas globais de deficiência visual e cegueira são erros de refração não corrigidos, catarata, DMRI e glaucoma (OTTAIANO *et al.*, 2019).

Hu *et al.* (2019) geraram imagens simuladas no computador e as resumiram na Figura 1.

Figura 1 – Imagens simuladas de deficiências visuais



Fonte: Adaptado de Hu *et al.* (2019).

Através da Figura 1, Hu *et al.* (2019) demonstraram como as pessoas com glaucoma no início estágio perdem seu campo visual periférico, e então uma visão tubular aparece lentamente à medida que a doença se deteriora. A segunda linha demonstra que a deficiência visual da DMRI se manifesta principalmente como perda de visão central. A retinite pigmentosa é uma doença ocular incurável, e a visão das pessoas com RP vai piorar à medida que a doença progride (a terceira linha). Erros de refração não corrigidos podem ser corrigidos pelo uso de lentes dióptricas (a última linha da Figura).

Essas imagens simuladas (Figura 1) podem fornecer orientação para o projeto e desenvolvimento de dispositivos auxiliares.

2.2 APRENDIZADO PROFUNDO

O aprendizado de máquina se tornou muito difundido nas pesquisas e tem sido incorporado a uma variedade de aplicações como: mineração de texto, detecção de *spam*, recomendação de vídeo, classificação de imagem, entre outros. O aprendizado profundo (*Deep learning*) é derivado da rede neural convencional, mas supera consideravelmente seu predecessor (LECUN; BENGIO; HINTON, 2015).

Uma rede neural convencional é limitada em sua capacidade de processar dados naturais de forma bruta, assim, esse modelo exige uma cuidadosa engenharia e uma considerável experiência de domínio do tema a ser analisada para projetar um recurso extrator que transforma os dados brutos (valores de pixel em uma imagem, por exemplo) em uma representação adequada ou um vetor de recursos no qual um classificador pode detectar padrões (LECUN; BENGIO; HINTON, 2015).

O aprendizado profundo é capaz de analisar, de forma eficaz, funções complexas e não lineares, é capaz de gerar representações de recursos distribuídos e hierárquicos e de permitir o uso eficaz de dados rotulados e não rotulados (PANG *et al.*, 2018). As técnicas de aprendizado profundo empregam transformações gráficas simultâneas a fim de construir modelos de aprendizagem multicamadas (ALZUBAIDI *et al.*, 2021).

O aprendizado de máquina é altamente dependente da integridade e da representação dos dados de entrada, sendo determinante para um desempenho adequado nos modelos propostos. Deste modo, a extração de recursos (características) tornou-se alvo de inúmeras pesquisas, visando construir tais recursos

a partir de dados brutos e, muitas vezes, para atingir seus objetivos, necessitam de um grande esforço humano (ALZUBAIDI *et al.*, 2021).

São exemplos dos esforços para extração adequada de dados dentro da visão computacional: histograma de gradientes orientados (HOG) (DALAL; TRIGGS, 2005), *Scale-Invariant Feature Transform* (SIFT) (LOWE, 1999), *Bag of Words* (BoW) (WU; HOI; YU, 2010), *Speed Up Robust Features* (SURF) (BAY; TUYTELAARS; VAN GOOL, 2006). LeCun, Bengio e Hinton (2015) afirmam que, como a extração de recursos é realizada de forma automática dentro dos algoritmos de aprendizado profundo, possibilita aos pesquisadores extraírem características discriminativas usando o menor esforço humano e conhecimento do problema.

Para Alzubaidi *et al.* (2021), os algoritmos de aprendizado profundo têm uma arquitetura de representação de dados multicamadas, cujas primeiras camadas extraem os recursos de baixo nível, enquanto as últimas camadas extraem os recursos de alto nível. Este processo simula o que ocorre nas regiões sensoriais do cérebro humano: utilizando diferentes cenas, o cérebro extrai automaticamente a representação de dados e, na saída do processo, os objetos são classificados, o que é o principal benefício da aprendizagem profunda.

As redes neurais convolucionais (CNN) são um dos mais populares modelos de redes de aprendizado profundo, sendo esse modelo que trouxe a maior visibilidade para a redes de aprendizado profundo. Isso ocorre porque, automaticamente, detectam as características significativas sem qualquer supervisão humana. As redes neurais de aprendizado profundo podem ainda ser do tipo: RN recursivas ou RN recorrentes (ALZUBAIDI *et al.*, 2021).

2.3 REDES NEURAIS CONVOLUCIONAIS

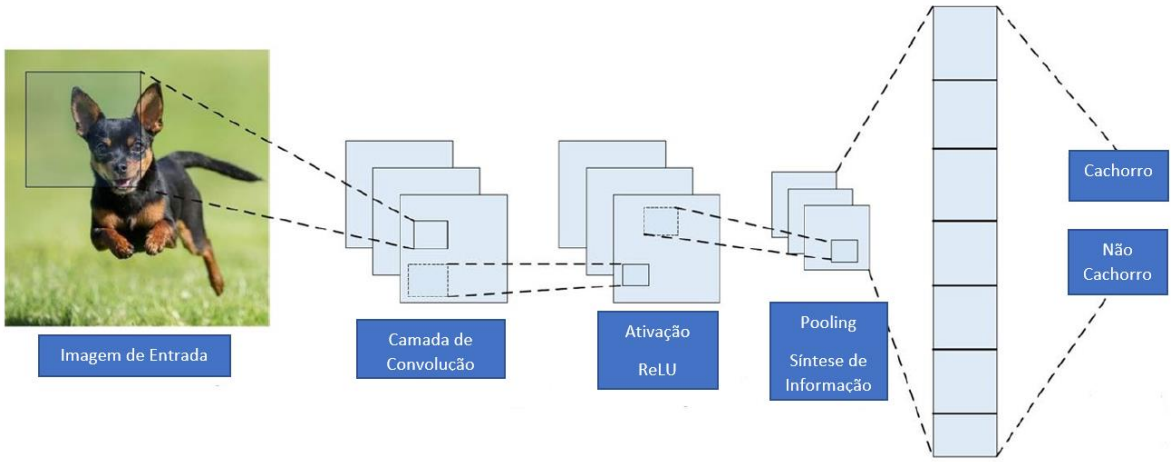
As redes neurais convolucionais são redes neurais do tipo *feedforward* e extraem as características dos dados através de convolução, sendo a sua arquitetura inspirada em percepção visual. Deste modo, cada *kernel* dentro da rede representa diferentes receptores que respondem a cada tipo de característica (LI *et al.*, 2020; ALZUBAIDI *et al.*, 2021; HAKIM; FADHIL, 2021).

Li *et al,* (2020) fazem um comparativo onde as redes neurais convolucionais apresentam vantagens em relação às redes neurais tradicionais, como: I – Conexões locais – onde cada neurônio não está conectado a todos os neurônios da camada

anterior, mas apenas a um pequeno número deles, o que é eficaz reduzindo parâmetros e acelerando a convergência; II – Peso compartilhado – um grupo de conexões pode compartilhar os mesmos pesos na rede, o que reduz ainda mais o número de parâmetros; III – Redução de dimensionalidade (*down-sampling*) – uma camada de *pooling* aproveita o princípio da correlação local da imagem para reduzir a amostragem de uma imagem, diminuindo a quantidade de dados, mas mantendo a formação.

A abordagem comum de redes neurais convolucionais é similar a uma rede neural *multi-layer perceptron (MLP),* consistindo de várias camadas de convolução precedendo camadas de subamostragem (agrupamento), e as camadas finais são do tipo totalmente conectadas (*fully-connected*), como se pode verificar na Figura 2 (ALZUBAIDI *et al.*, 2021).

Figura 2 – Arquitetura de Redes Neurais Convolucionais para classificação de imagens



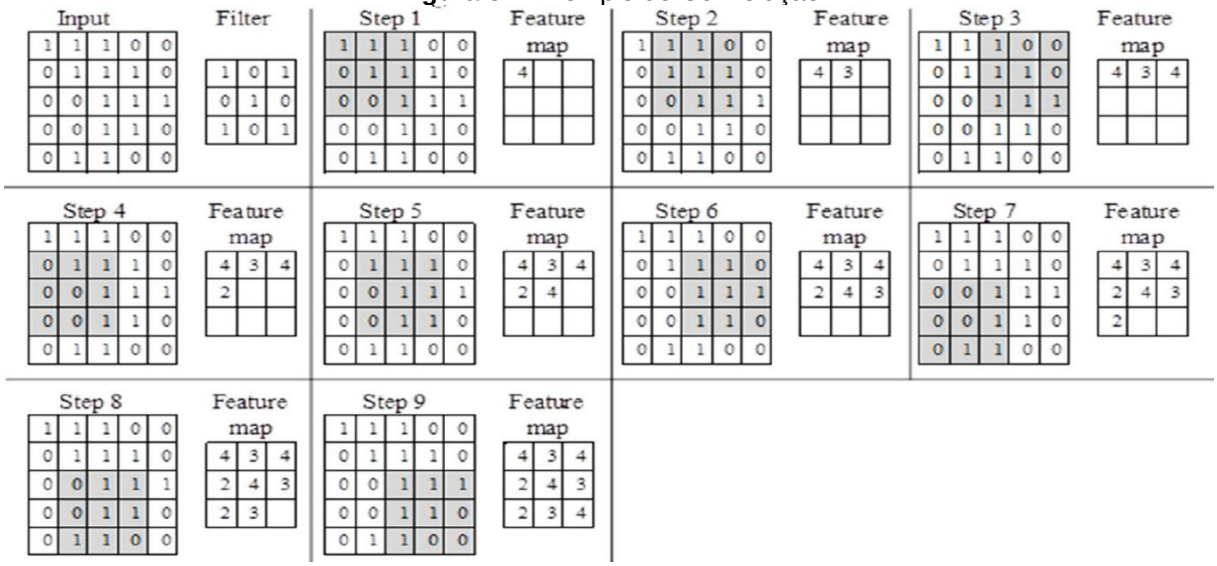
Fonte: Adaptado de Alzubaidi *et al.* (2021).

A entrada x de cada camada em um modelo de CNN é organizada em três dimensões: altura, largura e profundidade, ou $m\ x\ n\ x\ r$, onde a altura (m) é igual à largura (n), e a profundidade (r) é determinada pelo número de canais. Por exemplo, em uma imagem RGB, a profundidade é igual a três. Vários filtros (*kernels*) estão disponíveis em cada camada convolucional, os filtros são indicados por *w* e possuem três dimensões $i\ x\ j\ x\ q$, semelhantes à camada de entrada (ALZUBAIDI *et al.*, 2021; HAKIM; FADHIL, 2021).

Nos filtros, entretanto, o valor de i deve ser menor que m, o valor de j deve ser menor que o valor de n, enquanto *q* é igual ou menor que *r*; além disso, os filtros (*kernels*) são a base das conexões locais e compartilham parâmetros semelhantes de *bias* b^k e pesos w^k, de forma a gerar k mapas de características h^k, com tamanho de $(m - n - 1)$ (ALZUBAIDI *et al.*, 2021; HAKIM; FADHIL, 2021).

Hakim e Fadhil (2021) demonstram graficamente o processo de convolução (Figura 3), onde um filtro desliza por todos os elementos da imagem sendo multiplicado por cada um dos elementos e produzindo uma soma desta multiplicação, que, como resultado, irá gerar a matriz de características. No exemplo, tem-se uma imagem 5x5 realizando a convolução com um filtro 3x3.

Figura 3 – Exemplo de Convolução



Fonte: Adaptado de Hakim e Fadhil (2021).

A camada de convolução calcula um produto escalar entre sua entrada e os pesos como na Equação 1 (ALZUBAIDI *et al.*, 2021).

$$h^k = f(w^k * x + b^k) \quad 1$$

Na Equação 1, a função f representa a função de não linearidade ou função de ativação. Esta operação também é denominada de Unidade Linear Retificada ou do inglês *Rectified Linear Unit*, ou simplesmente ReLU; é a função que adiciona a não linearidade às NNs, permitindo apreender modelos não lineares (HAKIM; FADHIL, 2021). Esta operação substitui os valores negativos resultantes da convolução $w * x$

gerando o vetor de características. A função ReLU é assim denominada, pois é a mais comumente utilizada e é determinada pela Equação 2.

$$f(x)_{ReLU} = \max(0, x) \quad (2)$$

A função de ativação também pode receber outras funções como a função Sigmoide $f(x)_{sigm} = \frac{1}{1+e^{-x}}$, onde a saída é uma resposta entre $0\ a\ 1$ e a função Tanh (tangente hiperbólica) $f(x)_{tanh} = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$, onde o retorno da função é um intervalo de $-1\ a\ 1$ (ALZUBAIDI *et al.*, 2021).

O próximo passo de processamento em uma CNN é reduzir a amostragem levando a uma redução de parâmetros da rede, acelerando o processo de treinamento e permitindo o tratamento do problema de *overfitting*[5]. A camada de *pooling* é responsável por realizar a síntese da informação,através de uma função de máximo, soma ou média $(\max | \ sum \ | \ avg)$ para uma determinada instância, onde cada pontuação representa a probabilidade de uma classe específica (ALZUBAIDI *et al.*, 2021).

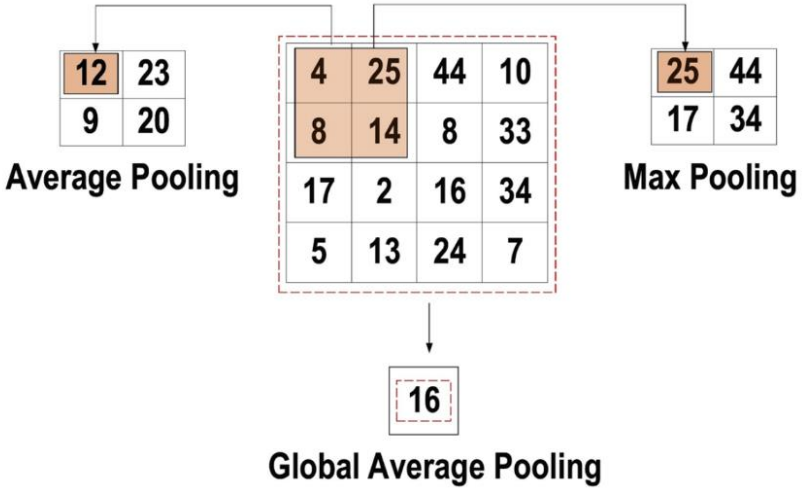
Hakim e Fadhil (2021) estabelecem que *pooling* é um dos conceitos distintivos das CNNs, sendo o seu objetivo reduzir a dimensionalidade de cada mapa de características, eliminando convoluções ruidosas e redundantes, mas ainda retendo a maioria das informações importantes.

Aluzbaidi *et al* (2021) demonstram na Figura 4 o cálculo da camada de *pooling* utilizando uma janela 2x2, onde se em o "*average pooling*", ou seja, calculado pela média dos elementos na janela indicada. Do mesmo modo tem-se o "*max pooling*" que é calculado extraindo o elemento mais relevante dentro da janela analisada e o "*global average pooling - GAP*, onde o valor calculado representa a média dos elementos do mapa de características.

Os métodos apresentados na Figura 3 são os mais utilizados, mas se podem utilizar outras funções como "*tree pooling", "gated pooling", "min pooling"* ou o *"global max pooling*".

[5] Excesso de treinamento da rede neural.

Figura 4 – Exemplo de cálculo da camada de *pooling*



Fonte: Adaptado de Alzubaidi *et al.* (2021).

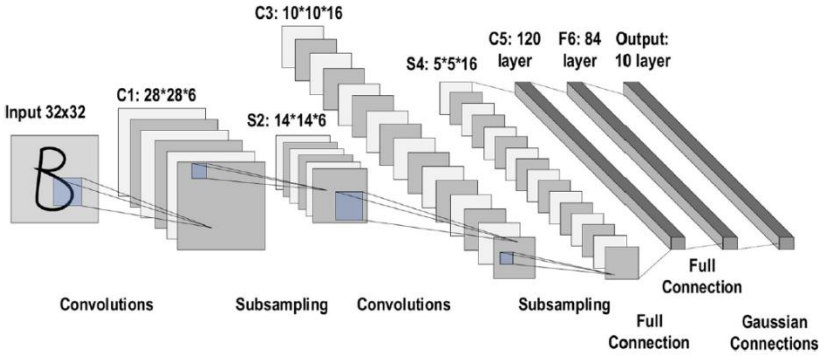
A camada totalmente conectada está normalmente localizada no final da arquitetura de uma CNN. Nesta camada, cada neurônio está conectado a todos os neurônios da camada, de modo que esta abordagem é denominada *Fully Connected (FC)*. Deste modo, é utilizada uma rede neural multicamadas como classificador, utilizando como entrada o vetor de saída da última camada, seja de *pooling*, seja convolucional (ALZUBAIDI *et al*., 2021; HAKIM; FADHIL, 2021).

2.3.1 Arquiteturas de Redes Neurais Convolucionais

Desde 1989, várias arquiteturas de CNNs foram apresentadas, sendo a arquitetura um fator crítico para o desempenho em diferentes aplicações. Entre as reformulações, incluem-se a reformulação estrutural, regularização, otimizações de parâmetros etc. Nesta seção são apresentadas as arquiteturas mais utilizadas.

A LeNet proposta por Lecun *et al.* (1989) iniciou a história das redes neurais convolucionais. No seu trabalho, o autor propunha a identificação de dígitos escritos a mão. Lecun *et al.* (1995) apresentam as atualizações de seu trabalho e, neste artigo, pode-se verificar a arquitetura básica de uma rede neural convolucional (Figura 5).

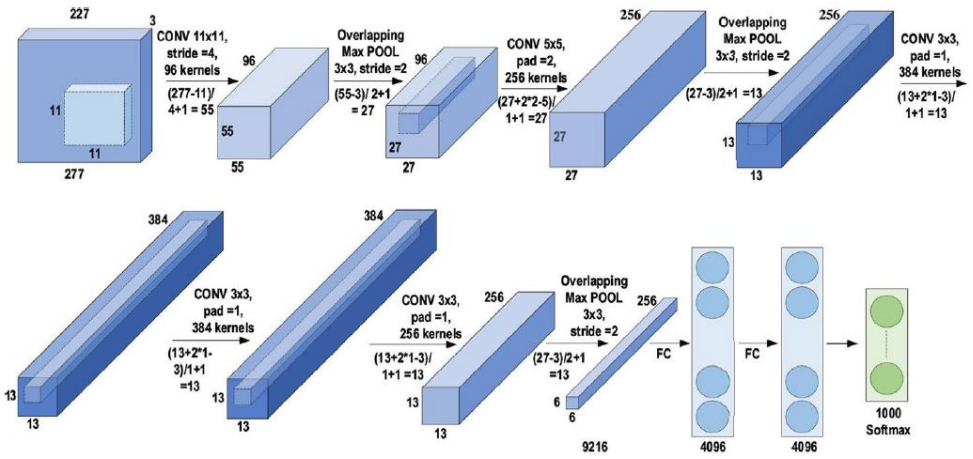
Figura 5 – Arquitetura da CNN – LeNet.



Fonte: Adaptado de Alzubaidi *et al.* (2021).

Krizhevesky, SutSkever e Hinton (2012) propuseram a AlexNet, que incrementou os resultados no campo de reconhecimento e classificação de imagens, pois melhorou a capacidade de aprendizagem das CNNs através do aumento da profundidade da rede e da implementação de várias estratégias de otimização como a aplicada da função de ativação (ReLU) durante o processo de convolução. Em testes realizados com base ImageNet, obteve-se uma taxa de erro de 15,3%. Na Figura 6 está representada a arquitetura da AlexNet.

Figura 6 – Arquitetura da CNN – AlexNet.



Fonte: Adaptado de Alzubaidi (2021).

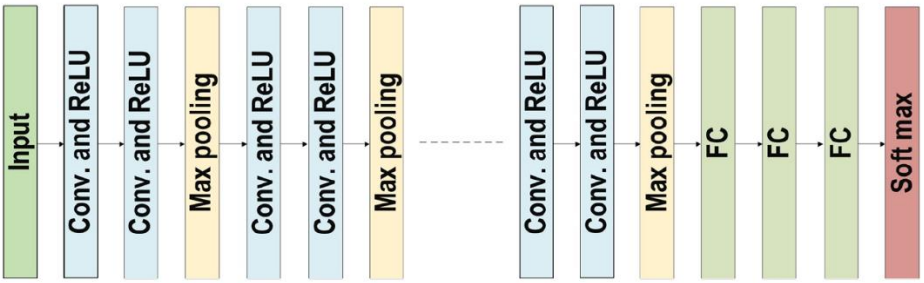
A CNN Network-in-Network (NIN) foi proposta por Lin, Chen e Yan (2014) e introduziu novos conceitos na arquitetura das CNNs com a camada denominada “mlpConv”. Essa camada adiciona dois filtros 1x1 que proporcionam uma adição de não linearidade extra na rede, permitindo aumentar sua profundidade após ser

regularizada por *dropout.* O modelo foi testado nas bases CIFAR-10, CIFAR-100 e MINST com taxas de erro de 10,41%, 35,68% e 0,45% respectivamente.

Zeiler e Fergus (2014) propuseram a ZefNet, que apresentou o conceito de visualização de camadas. Essa proposta surgiu a partir da necessidade da compreensão do mecanismo de aprendizagem das CNNs, por meio do monitoramento da ativação do neurônio, conseguindo visualizar quantitativamente a rede, o que permitiu melhorar a arquitetura da CNN. A ZefNet em testes com a base ImageNet (RUSSAKOVSKY *et al.*, 2015) obteve uma taxa de erro de 11,7%.

Simonyan e Zisserman (2015) apresentaram a rede neural convolucional denominada *Virtual Geometry Group* (VGG), que foi desenhada para ser eficaz no reconhecimento de imagem. Esta rede introduziu uma camada de filtros 3x3 que produzia a mesma influência na rede de filtros maiores e, ao se inserir um filtro 1x1 entre as camadas convolucionais, consegue –se regular a complexidade da rede, além de aumentar o número de camadas na rede (Figura 7). Este modelo foi testado utilizando a base ImageNet (RUSSAKOVSKY *et al.*, 2015) e obteve uma taxa de erro de 7,3%.

Figura 7 – Arquitetura da CNN – VGG.



Fonte: Adaptado de Alzubaidi (2021).

A GoogleLeNet (Inception-V1) proposta por Szegedy *et al.* (2015) apresentou o conceito de blocos que combina transformações convolucionais em múltiplas escalas (5x5, 3x3 e 1x1) e concatenações gerando uma camada 1x1 ao final de cada bloco. Utilizou conexões esparsas para superar o problema de informações redundantes e diminuiu o custo ao negligenciar os canais irrelevantes, este processo leva, ocasionalmente, à perda de informações importantes. Demonstraram sua CNN com 22 camadas e utilizaram a base ImageNet para testes com taxa de erro de 6,7%.

Em 2015, Srivastava, Greff e Schmidhuber (2015) propuseram a CNN denominada como “Highway”, que contém uma rede que permite que os valores se propaguem pelas camadas através de múltiplos caminhos, o que possibilita redes mais profundas e minimização do problema de custo em relação ao aprendizado nestes tipos de arquiteturas. A Highway foi testada na base CIFAR-10 e obteve uma taxa de erro de 7,76%.

He *et al.* (2016) desenvolveram a ResNet, cujo objetivo foi desenvolver uma rede ultra profunda. Para isso, utilizaram os conceitos empregados na CNN Highway empregando múltiplos caminhos (*Skip-Links*) dentro da arquitetura de sua rede. Os autores construíram redes de 34 a 1202 camadas, sendo o mais comum a ResNet50, que compreendia 49 camadas convolucionais e 1 totalmente conectada. Na ILSVRC 2015, uma rede ResNet de 152 camadas obteve o 1º lugar, atingindo uma taxa de erro de 3,57% utilizando a conjunto de dados ImageNet (RUSSAKOVSKY *et al.*, 2015).

A CNN Inception-V3 foi apresentada por Szegedy *et al.* (2016a). O conceito do Inception-V3 era minimizar o custo computacional sem efeito na generalização da rede. Para isso, foram utilizados filtros de tamanho pequeno como 1x5 e 1x7, além de uma camada 1x1 antes de filtros maiores. Esse processo recebeu o nome de fatoração. A Inception-V3 obteve uma taxa de erro de 3,5% na classificação da base ImageNet (RUSSAKOVSKY *et al.*, 2015).

Szegedy *et al.* (2016b) demonstraram a Intercepton-V4, que é uma rede com mais módulos de iniciação do que a Intercep-V3, e sua estrutura é mais simplificada, alcançou em testes com a base ImageNet a taxa de erro de 3,08%. Foi apresentada também a Inception-Resnet, que é uma rede Inception-V4 utilizando conexões residuais (*Residual Links* – que foram introduzidos neste trabalho), o que permitia largura e profundidade ampliadas. Essa rede pode ser treinada de forma mais rápida e possui o custo computacional de uma rede Inception-V3, mas, em testes com a base ImageNet (RUSSAKOVSKY *et al.*, 2015), obteve uma taxa de erro um pouco maior que 3,52%.

A CNN DenseNet proposta por Huang *et al.* (2017) segue na direção do que foi proposto anteriormente nas CNN Resnet e Highways. Essa CNN é estruturada em blocos de camadas conectadas e possui uma arquitetura estreita, o que torna a DenseNet custosa computacionalmente. Em testes realizados com as bases CIFAR-

10, CIFAR-100 e ImageNet (RUSSAKOVSKY *et al.*, 2015), obteve, como taxa de erro na classificação, 3,46%, 17,18% e 3,54% respectivamente.

A rede convolucional WideResNet, proposta por Zagoruyko e Komodakis (2016), visava resolver o problema da baixa contribuição que certos blocos de recursos ou transformações forneciam às redes residuais profundas. Para isso, os autores propuseram diminuir a profundidade da rede e aumentar sua largura; utilizaram, também, o conceito de *Drop-out* para eliminar alguns neurônios da rede durante o processamento. A WideResNet em testes com a base CIFAR-10 obteve 3,89% de taxa de erro, enquanto, com a base CIFAR-100, os testes demonstraram uma taxa de erro de 18,85%.

Chollet (2017) apresentou a CNN Xception tendo a rede Inception como base e modificou sua estrutura inicial diminuindo a profundidade e aumentando sua largura. Este processo é denominado pelo autor como convolução separável em profundidade. Em testes com a base ImageNet o Xception, supera ligeiramente o Inception-V3. O Xception possui o mesmo número de parâmetros do Inception-V3, cujos ganhos de desempenho não foram obtidos através de aumento de capacidade, mas com o uso mais eficiente destes parâmetros dentro do modelo.

Wang *et al.* (2017) apresentaram a CNN denominada *Residual Attention Network*. Nesse trabalho, os autores propuseram um bloco na rede a que chamaram de “bloco de atenção”. Este bloco visa habilitar a rede para apreender as características do objeto. Como estratégia, divide-se em dois ramos denominados de máscara (*mask*) e tronco *(trunk),* adotando um aprendizado de cima para baixo e de baixo para cima respectivamente. Essa abordagem permite o empilhamento de centenas de blocos na rede. Em testes, atingiu, com a base CIFAR-10, uma taxa de erro de 3,90% e, com a base CIFAR-100, apresentou uma taxa de erro de 20,45%. Os testes ainda demonstraram que o modelo é robusto em imagens ruidosas.

Hu, Shen e Sun (2018) apresentaram a rede denominada “Squeeze-and-excitation networks”. Os autores afirmam que as redes convolucionais são construídas sobre a operação de convolução que extrai recursos por fusão de informações espaciais e de canais. A fim de aumentar o poder de representação de uma rede, é importante melhorar a codificação espacial. Propuseram um novo bloco de arquitetura de CNN denominado “Squeeze-and-Excitation - SE”, cuja função é ser um mecanismo de reconhecimento de conteúdo que pondera de forma adaptativa, gerando um único parâmetro para cada canal fornecendo uma escala linear de relevância.

Em CNNs como VGG (SIMONYAN; ZISSERMAN, 2015) ou ResNet (He *et al.*, 2016), as imagens de entrada são codificadas através de convoluções em imagens de baixa resolução, mas, em tarefas de visão sensíveis, a posição e estimativa de pose humana são necessárias representações de alta resolução. Sun *et al.* (2020) apresentam a *High-Resolution Net* (HRNet) e Cheng *et al.* (2020) apresentam a HigherHRNet, que são redes que mantêm, no primeiro estágio, a alta resolução da imagem e adicionam, gradualmente, sub-redes de alta para baixa resolução em paralelo, formando múltiplas sub-redes variando a resolução espacial. A vantagem é a obtenção de uma representação mais precisa no domínio espacial e no domínio semântico para predição de pose humana.

Na Tabela 1 estão resumidas as arquiteturas de redes neurais convolucionais apresentadas nesta seção.

Tabela 1 – Visualização de Arquiteturas de CNNs

Model	Inovação	Camadas	Conjunto de Dados	Taxa de erro (%)	Tamanho da entrada	Ano
AlexNet	Dropout e ReLU	8	ImageNet	15,3	227x 227x3	2012
Network-in-Network (NIN)	Nova Camada “mlpconv”, GAP	3	CIFAR-10 CIFAR-100 MNIST	10,41 35,68 0,45	32x32x3	2013
ZefNet	Visualização de camadas intermediárias	8	ImageNet	11,7	224x224x3	2014
VGG	Maior profundidade e filtro pequeno	16, 19	ImageNet	7,3	224x224x3	2014
GoogleLeNet	Maior profundidade, conceito de bloco, tamanhos diferentes de filtros, conceito de concatenação	22	ImageNet	6,7	224x224x3	2015
Inception-V3	Melhor representação dos recursos, filtro pequeno	48	ImageNet	3,5	229x229x3	2015
Highway	Conceito de multi-percurso	19, 32	CIFAR-10	7,76	32x32x3	2015
Incepton-V4	Transformada dividida e conceito de integração	70	ImageNet	3,08	229x229x3	2016
ResNet	Robusto contra overfitting devido a simetria baseado no mapeamento “*skip-links”*	152	ImageNet	3,57	224x224x3	2016
Inception-ResNet-V2	Introduz o conceito de “*residual links*”	164	ImageNet	3,52	229x229x3	2016
WideResnet	Diminuiu a profundidade e aumento a largura	28	CIFAR-10 CIFAR-100	3,89 18,85	32x32x3	2016

Xception	Convoluções separáveis em profundidade	71	ImageNet	0,055	229x229x3	2017
Residual attention neural network	Apresentou a técnica de atenção	452	CIFAR-10 CIFAR-100	3,90 20,40	40x40x3	2017
Squeeze-and-excitation networks	Interdependências modeladas entre canais	152	ImageNet	2,25	229x229x3 224x224x3 320x320x3	2017
DenseNet	Blocos de camadas e camadas conectadas umas com as outras	201	CIFAR-10 CIFAR-100 ImageNet	3,46 17,18 3,54	224x224x3	2017
HRNetV2	Representações de alta resolução	-	COCO MPII Human Pose		224x224x3 256x256x3 384x288x3	2020

Fonte: Adaptado de Alzubaidi *et al* (2021).

2.3 TRANSFERÊNCIA DE APRENDIZADO (*TRANSFER LEARNING*)

As redes neurais convolucionais, usualmente, precisam de grandes conjuntos de dados para que o treinamento seja efetivo, porém, quando os dados para treinamento são escassos, é possível transferir as representações aprendidas em outras tarefas de reconhecimento onde grandes conjuntos de dados estão disponíveis para a arquitetura da rede.

Weiss, Khoshgoftaar e Wang (2016) afirmam que a principal motivação para a transferência de aprendizado é a necessidade da criação de grandes bases de dados para treinamento de modelos que consigam realizar a predição de forma correta e que, por serem correlacionados ao domínio do modelo treinado, esses dados são, muitas vezes, difíceis e caros de se obter.

Tan *et al.* (2018) afirmaram que dados de treinamento são, geralmente, insuficientes dentro dos modelos de *Deep Learning*, principalmente em domínios de aprendizado especiais, e citam, como exemplo, como pode ser cara e complexa a criação de um conjunto de dados em grande escala e de alta qualidade em uma aplicação para bioinformática que, muitas vezes, envolve pacientes e ensaios clínicos. Frisa, ainda, que esse conjunto de dados pode ficar desatualizado facilmente.

A transferência de aprendizagem relaxa a hipótese de que os dados de treinamento devem ser independentes e distribuídos de forma idêntica com os dados de teste. Com essa técnica, não existe essa necessidade, e o modelo no domínio destino não precisa ser treinado do zero, reduzindo, significativamente, a demanda

por grandes conjuntos de treinamento e o tempo necessário para treinamento do domínio destino. A transferência de aprendizado é importante técnica para resolver os problemas de dados de treinamento insuficientes (TAN *et al.*, 2018; WEISS; KHOSHGOFTAAR; WANG, 2016).

Tan *et al.* (2018) classificam a transferência de aprendizado em quatro abordagens:

I. O aprendizado baseado em instâncias: refere-se a uma estratégia de ajuste de pesos da instância de domínio origem para o domínio destino; assim, instâncias parciais do domínio origem serão suplementos ao treinamento do domínio destino. Parte do pressuposto que, embora haja diferenças entre os domínios, instâncias parciais do domínio origem podem ser utilizadas pelo domínio destino como pesos apropriados.

II. Aprendizado por mapeamento: refere-se ao mapeamento dos domínios origem e destino em um novo espaço de dados. Parte do princípio de que, embora haja diferenças entre dois domínios, eles podem ser mais um conjunto em um novo espaço de dados.

III. Aprendizado baseado em rede: refere-se à reutilização parcial da rede que, treinada no domínio origem, incluindo sua estrutura e conexões, faz parte da rede neural profunda utilizada no domínio destino. Parte do pressuposto de que a rede neural é semelhante ao mecanismo do cérebro humano e é um processo iterativo e contínuo de abstração.

IV. Aprendizado baseado no adversário: refere-se à introdução da tecnologia adversária para encontrar representações transferíveis que são aplicáveis ao domínio de origem e ao domínio de destino. Baseia-se no pressuposto de que, para uma transferência de aprendizado eficaz, uma boa representação de dados deve ser gerada para ser discriminativa para a tarefa de aprendizado principal entre os domínios.

Alzubaidi *et al.* (2021) descrevem que modelos de CNN como a AlexNet (KRIZHEVESKY; SUTSKEVER; HINTON, 2012), GoogleLeNet (SZEGEDY *et al.*, 2015) e ResNet (HE *et al.*, 2016) foram treinados em grandes conjuntos de dados como a ImageNet (RUSSAKOVSKY *et al.*, 2015) para a finalidade de reconhecimento de imagens. Assim, esses modelos podem ser empregados para reconhecer uma tarefa diferente sem a necessidade de serem treinados a partir do zero, e um modelo

pré-treinado pode auxiliar na generalização e até na velocidade de convergência de uma rede. Enfatizam, também, que a generalização adequada de uma CNN evita problemas de *overfitting* esse processo de generalização requer grandes volumes de informações rotuladas que são custosas para ser obtidas.

Cook, Feuz e Krishnan (2013) e Cao *et al.* (2013) discutem, em seus trabalhos, a utilização de tipos de dados de origens diferentes do objetivo destino durante a utilização de transferência de aprendizado e que isso é uma questão essencial relacionada a esse processo. Cao *et al.* (2013) obtiveram bons resultados utilizando transferência de aprendizado de bases diferentes na identificação de pedestres. Outro exemplo de utilização de bases fora do domínio é demonstrado no trabalho de Alzubaidi *et al.* (2020), que utilizou uma rede treinada com uma base de animais para classificar uma base médica de úlcera do pé diabético, onde obteve um resultado melhor com a transferência de aprendizado do que treinando o modelo do zero.

Oquab *et al.* (2014) reutilizaram camadas de redes convolucionais treinadas a partir do conjunto de dados ImageNet (RUSSAKOVSKY *et al.*, 2015) para processar a representação intermediária de imagens para classificar objetos no conjunto de dados Pascal VOC (EVERINGHAM *et al.*, 2010), superando os resultados obtidos anteriormente com métodos do estado da arte. Gopalakrishnam *et al,* (2017) treinaram CNNs com o conjunto de dados ImageNet para detectar rachaduras em imagens da superfície de vias pavimentadas.

Shin *et al.* (2016) aplicaram com sucesso a transferência de aprendizado de imagens comuns que compõem o conjunto ImageNet (RUSSAKOVSKY *et al.*, 2015) para classificação de imagens de tomografia computadorizada. Saleh *et al.* (2017) utilizaram CNNs para detectar o caminho livre para deficientes visuais através de segmentação de imagens em pixel. Eles adicionaram algumas camadas de convolução para as necessidades específicas do trabalho.

Monteiro *et al.* (2017) utilizaram um conjunto de dados de vídeo coletados a partir do ponto de visão de um cão-guia para treinar uma CNN para identificar as atividades que ocorrem em torno da câmera do usuário. Eles realizaram simulações com as redes AlexNet (KRIZHEVESKY; SUTSKEVER; HINTON, 2012) (*fully-trained* e *fine-tuned*) e GoogLeNet (SZEGEDY *et al.*, 2015) (*fully-trained* e *fine-tuned*) e com pré-treinamento realizado no conjunto ILSVRC 2012 ImageNet (RUSSAKOVSKY *et al.*, 2015).

2.4 REDUÇÃO DE DIMENSIONALIDADE

Ciências como a biologia, a química ou a astronomia observaram, nas últimas décadas, uma explosão de dados disponíveis obtidos através de instrumentos cada vez mais complexos que relatam centenas ou milhares de medições para um único experimento, enquanto os métodos estatísticos enfrentam tarefas desafiadoras ao lidar com dados de alta dimensionalidade.

Os procedimentos de redução de dimensionalidade são métodos matemáticos para diminuir o número de variáveis disponíveis antes da aplicação de algum processamento ao conjunto de dados. Esta tarefa pode ser conduzida de duas maneiras: a primeira somente selecionando as variáveis mais relevantes do conjunto de dados original e é denominada seleção de características (*feature selection*), ou explorando a redundância dos dados de entrada e encontrando um conjunto menor de novas variáveis resultantes da combinação dos dados de entrada originais. Esse processo é denominado redução de dimensionalidade (*dimensionality reduction*).

O principal algoritmo de redução de dimensionalidade linear é o de Análise de Componentes Principais (PCA) (JOLLIFFE, 2002), sendo o mais amplamente utilizado. O PCA faz uma projeção linear dos dados em uma direção que preserva a variação máxima (ou, equivalentemente, minimiza o erro de reconstrução) dos dados originais. Se os dados estiverem em um formato bidimensional, o PCA é capaz de preservar todos os dados, no entanto o mapeamento linear nem sempre é suficiente para “explicar” todos os dados (NISKANEN; SILVÉN, 2003).

Tenenbaum, Silva e Langford (2000) propuseram um modelo de redução de dimensionalidade não linear denominado Isomap, que mede a distância entre dois pontos distantes no conjunto de dados (distância global) e tenta obter um conjunto de baixa dimensionalidade através deste processo. Este modelo gera um grafo com os *K* vizinhos próximos, calcula as distâncias entre os vizinhos para que, a partir disso, consiga encontrar o menor caminho utilizando um algoritmo de Dijasktra ou Floyd-Warshall’s.

Sobre o caminho mais curto encontrado, reduz-se a dimensionalidade utilizando o algoritmo de escala multidimensional (MDS)[6], que é um método baseado

[6] Do inglês: *Multidimensional Scaling* (MDS).

em matriz de distâncias em um espaço euclidiano de baixa dimensionalidade (AGARWAL; PHILIPS; VENKATASUBRAMANIAN, 2010). Este modelo tem problemas quando o menor caminho encontrado não representa adequadamente os dados, pois ficou muito pequeno ou por problemas de ruídos nos dados.

Roweis e Saul (2000) propuseram uma abordagem local em contrapartida ao Isomap. Este algoritmo foi denominado *Locally Linear Embedding* (LLE) e elimina a necessidade de estimar as distâncias de pares de pontos de dados amplamente separados. O Isomap é otimizado para preservar as distâncias globais entre os pares de vértices, enquanto o LLE analisa as simetrias locais.

Saxena, Gupta e Mukerjee (2004) propõem o modelo K_{LL} *Isomaps* que tenta resolver o problema do algoritmo de Isomap quando este constrói o grafo utilizando os *K* vizinhos próximos de forma global propondo reconstruir o grafo utilizando combinação linear local entre os vértices.

Maaten e Hinton (2008) apesentaram uma nova técnica de redução de dimensionalidade denominada *t-Distributed Stochastic Neighbor Embedding* (T-SNE), que é base no modelo *Embedding Stochastic Neighbor* (SNE) de Hinton e Roweis (2002). Ele é baseado em dois estágios: primeiro constrói uma distribuição de probabilidade sobre pares de objetos de alta dimensão, de tal forma que a objetos semelhantes são atribuídas probabilidades mais altas e a pontos diferentes uma probabilidade mais baixa. Em uma segunda etapa, a técnica estabelece uma nova distribuição de probabilidade, agora, em dados de baixa dimensionalidade e minimiza a divergência entre as distribuições utilizando a entropia de Kullback-Leibler. O algoritmo T-SNE demonstrou-se mais eficiente em separar dados de alta dimensionalidade que outros modelos, mas pesa seu alto custo computacional na ordem de $O(n^2)$.

McInnes, Healy e Melville (2020) apresentaram a técnica de redução de dimensionalidade denominada *Uniform Manifold Approximation and Projection* (UMAP). Esta técnica funciona de forma semelhante ao T-SNE, pois o UMAP também constrói um grafo para reorganizar os dados em um espaço de baixa dimensionalidade.

Para construir o grafo de alta dimensionalidade inicial utiliza o que se chama de “*fuzzy simplicial*”, onde se tem um grafo ponderado com os pesos das arestas representando a probabilidade de que dois pontos estejam conectados.

Para determinar a conectividade, o UMAP estende um raio para fora de cada ponto, conectando pontos quando esses raios se sobrepõem. Essa escolha é crítica: se for muito pequena levará a aglomerados locais e, se for muito grande, conectará tudo. Para resolver o desafio, o algoritmo escolhe um raio localmente com base na distância até o enésimo vizinho mais próximo de cada ponto.

Assim, o UMAP torna o grafo difuso, diminuindo a probabilidade de conexão conforme o raio aumenta. Por fim, ao estipular que cada ponto deve ser conectado a pelo menos a seu vizinho mais próximo, garante que a estrutura local seja preservada em equilíbrio com a estrutura global. O UMAP produz uma saída semelhante ao T-SNE, mas com um custo computacional baixo.

2.5 MODELO DE COMPETIÇÃO E COOPERAÇÃO DE PARTÍCULAS

O modelo de Competição e Cooperação de Partículas[7] (PCC) é um método de aprendizado de máquina semi-supervisionado baseado em grafos e inspirado na natureza. Neste modelo, times de partículas caminham pela rede cooperando entre si e competindo com outros times, tentando possuir o maior número de vértices possíveis (BREVE *et al.*, 2012).

Os times são as classes dentro do contexto do aprendizado de máquina, e as partículas que representam a mesma classe caminham cooperativamente para espalhar o seu rótulo pelos vértices. Ao mesmo tempo, partículas de classes diferentes competem para definir os limites de cada classe (BREVE *et al.*, 2012).

A rede é representada por um grafo que é gerado a partir dos dados de entrada. Cada elemento se torna um vértice no grafo, e arestas são criadas entre cada vértice e seus *k*-vizinhos mais próximos, onde a distância entre vértices é determinada por alguma medida, geralmente a distância euclidiana.

Para cada vértice que corresponde a um elemento rotulado, uma partícula é gerada, cuja posição inicial é o mesmo vértice, denominado “nó inicial” da partícula. Conforme as partículas mudam de posição, as distâncias entre o seu nó atual e o seu nó inicial são registradas. As partículas geradas por elementos de uma mesma classe atuam como uma equipe.

[7] Do inglês: *Particle Competition and Cooperation* (PCC).

Cada vértice do grafo tem um vetor onde cada elemento representa o nível de dominação de uma equipe sobre esse nó. A soma deste vetor é sempre constante. Conforme o sistema executa, as partículas percorrem o grafo e aumentam o nível de dominância de sua equipe sobre o nó, ao mesmo tempo em que eles reduzem os níveis de dominação de outras equipes, sempre mantendo a soma constante.

Além disso, cada partícula tem um nível de força, que aumenta quando ela visita um nó dominado por sua equipe e diminui quando visita um nó dominado por outra equipe. Esta força é importante porque a mudança que uma partícula causa em um nó é proporcional à força que possui no momento. Este mecanismo garante que uma partícula seja mais forte quando está em sua vizinhança, protegendo-a, e é mais fraca quando está tentando invadir territórios de outras equipes.

As partículas escolhem o próximo nó a ser visitado com base em uma de duas regras. Em cada iteração, elas escolhem, aleatoriamente, uma das regras com probabilidades pré-definidas. As duas regras são descritas a seguir:

- Regra Aleatória: a partícula escolhe aleatoriamente, com igual probabilidade, qualquer nó vizinho para visitar.
- Regra Gulosa: a partícula escolhe aleatoriamente qualquer nó vizinho para visitar, mas com probabilidades proporcionais ao nível de domínio que sua equipe tem em cada vizinho e inversamente proporcional à distância do vizinho em relação ao seu nó inicial.

Portanto, a regra gulosa é útil para manter as partículas em seu próprio território, ou seja, um comportamento defensivo. Por outro lado, na regra aleatória, as partículas são mais propensas a ir para nós não dominados e distantes, assumindo um comportamento de abordagem exploratória.

Observa-se que uma partícula só permanece no nó escolhido se for capaz de dominar esse nó, caso contrário, é expulsa e volta ao nó anterior até a próxima iteração. Esta regra é usada para evitar que uma partícula saia do seu território e perca toda a sua força. Também favorece a formação de bordas suaves nas regiões de cada classe, pois uma partícula não pode dominar um determinado nó sem antes dominar os nós no seu caminho. No fim das iterações, cada nó é rotulado com a classe da equipe que o dominou.

O PCC já foi estendido para lidar com saídas *fuzzy* onde o rótulo determina o nível de pertencimento à classe para cada nó (BREVE; ZHAO. 2013), para ser mais robusto a ruídos (BREVE; ZHAO; QUILES, 2015a), para lidar com o conceito de *“drift”,*

ou seja, dados ocultos ou situações em que os dados são influenciados por fatores externos ao conjunto de dados estudado (BREVE; ZHAO, 2012), para realizar aprendizagem ativa (BREVE, 2013), para trabalhar com segmentação de imagens (BREVE; ZHAO; QUILES, 2015b), entre outros. O modelo tem sido aplicado a dados de diferentes domínios, incluindo engenharia de *software*, bioinformática e diagnóstico médico.

2.6 MÉTRICAS DE DESEMPENHO

As métricas de avaliação adotadas nas tarefas de aprendizado profundo desempenham um papel fundamental na obtenção dos resultados da otimização de um classificador. Assim, uma seleção de métricas de avaliação adequadas é uma chave importante para discriminar e obter o classificador ideal (HOSSIN; SULAIMAN, 2015). As métricas são aplicadas para confirmar uma hipótese. Na área de processamento de imagens ou reconhecimento de padrões, essas métricas são extraídas a partir de uma rotulagem dos dados investigados.

Para problemas de classificação binária, a avaliação de uma melhor (ótima) solução para uma determinada classificação pode ser definida com base na matriz de confusão. A partir desta matriz, rótulos são atribuídos: verdadeiro positivo (vp), falso positivo (fp), verdadeiro negativo (vn) e falso negativo (fn), onde:

- Verdadeiro positivo (vp): se uma instância é positiva e é classificada como positiva;
- Verdadeiro negativo (vn): se uma instância é negativa e é classificada como negativa;
- Falso positivo (fp): se uma instância é negativa e é classificada como positiva;
- Falso negativo (fn): se uma instância é positiva e é classificada como negativa.

A partir da matriz de confusão, várias métricas comumente usadas podem ser geradas, conforme descrito na Tabela 2 (HOSSIN; SULAIMAN, 2015).

Tabela 2 – Métricas (Matriz de Confusão)

Métrica	Fórmula	Foco de Avaliação
Acurácia (acc)	$\frac{vp + vn}{vp + fp + vn + fn}$	Em geral, a métrica de acurácia mede a proporção de previsões corretas sobre o total de instâncias avaliadas.
Taxa de Erro (err)	$\frac{fp + fn}{vp + fp + vn + fn}$	Mede a proporção de previsões incorretas sobre o número de instâncias avaliadas.
Sensibilidade (sn)	$\frac{vp}{vp + fn}$	Utilizada para medir a fração de padrões positivos que são classificados corretamente.
Especificidade (sp)	$\frac{vn}{vn + fp}$	Utilizada para medir a fração de padrões negativos que são classificados corretamente.
Precisão (p)	$\frac{vp}{vp + fp}$	Utilizada para medir a relação entre as previsões positivas realizadas corretamente e todas as previsões positivas (incluindo as falsas).
Recall (r)	$\frac{vp}{vp + vn}$	Utilizado para medir a fração de padrões positivos corretamente classificados.
F_1 *Score* (fm)	$2 * \frac{p * r}{p + r}$	Representa a média harmônica entre os valores de precisão e *recall*.

Fonte: Adaptado de Hossin e Sulaiman (2015).

Hossin e Sulaiman (2015) afirmam que a precisão é a métrica de avaliação mais utilizada para problemas de classificação binária ou de múltiplas classes. Uma métrica complementar à acurácia é a taxa de erro que avalia a solução produzida por sua porcentagem de previsões corretas, sendo que uma das grandes vantagens de ambas é a facilidade de serem calculadas, entendidas e aplicáveis a diversos problemas diferentes. Os autores ainda afirmam que a métrica F_1 *Score* é um bom discriminador e obteve melhor desempenho do que a acurácia na otimização de problemas de classificação binária.

Uma métrica que é amplamente utilizada na área de inteligência artificial, pois representa com maior precisão a distinção entre duas classes, é a área sob a curva ROC (*Area Under Curve* – AUC), que reflete o desempenho geral de classificação de um modelo. A análise da curva ROC é uma representação gráfica entre a sensibilidade (sn) e a especificidade (sp) sendo determinada pela Equação 3 em um problema de classificação com duas classes:

$$AUC = \frac{S_p - n_p(n_n + 1)/2}{n_p n_n} \quad 3$$

onde, S_p representa a soma de todas as amostras classificadas como positivas, enquanto n_p e n_n denotam o número de amostras positivas e negativas respectivamente.

Embora o desempenho da AUC tenha sido excelente para processos de avaliação e discriminação, o custo computacional de AUC é alto, especialmente para discriminar um volume de soluções geradas para problemas multiclasse (ALZUBAIDI *et al.*, 2021).

3 TRABALHOS RELACIONADOS

As tecnologias assistivas ou formas de fornecer acesso à população com deficiência visual têm atraído considerável atenção em todo o mundo devido ao seu notável significado social. Na última década, uma variedade de tecnologias vem sendo desenvolvida para pessoas com deficiências visuais (BHOWMICK; HAZARIKA, 2017). Nesta seção, foram resumidas diversas abordagens encontradas durante o levantamento de trabalhos relacionados ao auxílio a deficientes visuais.

3.1 TECNOLOGIAS ASSISTIVAS

A utilização de bengalas no auxílio dos deficientes visuais é crítica, pois reduz o risco de colisão auxiliando a caminhar de forma mais confiante. No geral, essas tecnologias são caracterizadas por sensores e dispositivos que retornam algum estímulo de orientação para o usuário montados sobre uma bengala comum. Nestes tipos de projetos, são comumente utilizadas câmeras RGB, sensores ultrassônicos e lasers. Outro formato de dispositivo auxiliar para pessoas com deficiência visual são óculos adaptados de sensores como se verifica nas bengalas, mas também são encontradas pesquisas que utilizam cintos e outros tipos de acessórios que possam ser vestidos pelo usuário.

Kumar *et al.* (2014) desenvolveram uma bengala ultrassônica para auxiliar os cegos a navegar. Este dispositivo está equipado com transmissores e receptores ultrassônicos possibilitando que o usuário identifique, através de avisos sonoros, obstáculos aéreos e terrestres a uma distância de 1,5 m. Gupta *et al.* (2015) também utilizaram sensores ultrassônicos em uma bengala comum e adicionaram um módulo de geolocalização (GPS) que permite que os deficientes visuais caminhem ao ar livre utilizando a rede de satélites. O retorno da identificação de obstáculos foi gerado através de áudio e a distância útil do sensor utilizado foi de 0,5 a 2 metros.

Sadi *et al*. (2014) incorporaram um sensor ultrassônico e um microcontrolador em um par de óculos para desenvolver um dispositivo para auxiliar pessoas com deficiências visuais durante sua caminhada. A região de detecção do sensor ultrassônico cobre uma distância de 3 m e um ângulo de 60º. Informação processada, que corresponde à distância do obstáculo, é enviada para usuários por meio de sinais

de áudio. Em experimentos de validação realizados no laboratório, o dispositivo demonstrou uma precisão na detecção de obstáculos acima de 93%.

Yasuno *et al.* (2016) compararam 3 sensores diferentes: um sensor ultrassônico, um sensor infravermelho e um sensor a laser, empregando várias métricas para a comparação, como precisão, tamanho, peso. Foi selecionado o sensor ultrassônico para o desenvolvimento de um par de óculos para auxiliar deficientes visuais em sua caminhada. O sensor ultrassônico foi acoplado a um microcontrolador que processava as informações e gerava um retorno através de avisos sonoros e vibração. Os testes foram realizados por pessoas com deficiências que comprovaram a validade do protótipo.

Diversos outros pesquisadores construíram protótipos de bengalas ou óculos utilizando sensores ultrassónicos. Romadhon e Husein (2020), Ansari *et al.* (2020) também desenvolveram uma bengala utilizando sensores ultrassônicos e Arduino.

Além de sensores ultrassônicos, também são utilizadas câmeras RGB e laser. Ye *et al.* (2016) utilizaram uma câmera 3D montada em uma bengala para identificar os obstáculos. O retorno ao usuário foi realizado através de mensagens de áudio. O protótipo foi validado com auxílio de pessoas portadoras de deficiência visual, e o estudo demonstrou que o protótipo proposto foi capaz de identificar obstáculos com uma acurácia acima de 90%.

Dang *et al.* (2016) propôs uma bengala utilizando uma unidade de laser, uma câmera RGB e uma unidade de medição inercial (UMI) como sensores para classificar o tipo de obstáculo e estimar a distância do usuário. A UMI é um dispositivo eletrônico que mede a taxa angular do usuário em relação ao solo para determinar a posição em relação ao movimento de busca com a bengala. O sensor UMI rastreia a posição da faixa de laser onde são coletadas imagens dessa faixa. O retorno para o usuário é realizado por um sinal sonoro não falado. O desempenho da bengala é facilmente influenciado por iluminações fortes, limitando o escopo da aplicação e, em condições normais, detectou objetos a 180 cm com uma acurácia acima de 90%.

Majeed e Baadel (2016) integraram, em uma bengala, uma câmera RGB com uma lente de 270º que permite capturar uma grande área do ambiente; permitiu às pessoas com deficiência visual reconhecer através do dispositivo os rostos de outras pessoas identificando-as quando seus rostos estivessem gravados na base de imagem do dispositivo (cartão SD), o protótipo também possui sensores ultrassônicos que permitem ao usuário desviar de obstáculos a uma distância máxima de 10 metros.

Os sensores infravermelhos também são muito populares para a coleta de informações para bengalas inteligentes. Esses sensores captam informações do ambiente em outra faixa do espectro eletromagnético fora da luz visível permitindo a detecção de obstáculos no escuro, assim como medir sua distância.

Scherlen *et al.* (2007) combinaram um sensor infravermelho, um sensor de brilho e um sensor de água para desenvolver uma bengala inteligente, capaz de reconhecer objetos e os materiais de que eram compostos. Os autores descrevem que foi possível detectar 4 tipos de materiais metal(aço), vidro, papelão e plástico, além disso o dispositivo detectava poças de água e faixas de pedestres.

Buchs *et al.* (2017) montaram uma bengala com dois sensores infravermelhos com uma faixa de detecção de 1,5 m de modo que um dos sensores foi direcionado diretamente para frente e o outro a 42º para cima. Esta disposição de sensores permite capturar obstáculos elevados. Durante o estudo, efetuaram testes com pessoas com deficiência visual que levaram aproximadamente 5 minutos para dominar o dispositivo. Os resultados demonstraram que os usuários estudados conseguiram detectar mais obstáculos utilizando o aparelho do que com uma bengala comum. Também propuseram que, se for adicionada uma câmera RGB, pode-se melhorar o desempenho do modelo.

Everding *et al.* (2016) utilizaram dois sensores DVS (*Dynamic Vision Sensor*) em um par de óculos simulando o posicionamento das retinas nos olhos humanos. O modelo demonstrou resultado satisfatório quando analisado com objetos estáticos, em relação a objetos em movimento não foi determinado em testes. Wang *et al.* (2013) utilizaram câmeras RGB em óculos para determinar a localização de portas e identificação de informações em placas (pictográficas) de forma a auxiliar o deficiente visual através do retorno de áudio.

Fan *et al.* (2014) utilizaram uma câmera RGB-D e um sensor ultrassônico para adquirir, dinamicamente, um panorama do ambiente à frente da bengala proposta e poder detectar obstáculos. A câmera RGB-D é capaz de obter vídeos que possuem sincronização com imagens coloridas e profundidade. Utilizaram, também, como recurso de navegação externa, um módulo GPS. Nos testes, foi possível estabelecer uma navegação segura para pessoas com deficiência visual, mas os dados de imagem coletados pela câmera RGB-D não foram processados.

Takizawa *et al*. (2015) também usaram uma câmera RGB-D em sua unidade de detecção, e eles chamaram essa bengala de “bengala Kinect”. Com o uso da

câmera RGB-D, a bengala Kinect pôde reconhecer diferentes tipos de obstáculos internos, incluindo cadeiras, escada e chão. Duas pessoas vendadas foram convidadas para testar o desempenho do dispositivo proposto, e obtiveram resultados de tempo médio de percurso utilizando a “bengala Kinect” significativamente mais curto do que o de uma bengala comum.

Câmeras RGB-D podem adquirir imagens em cores e com informações de distância; também são amplamente utilizadas em pesquisas para auxílio a deficientes visuais. Neto *et al.* (2016) utilizaram um sensor Kinect na altura da cabeça do usuário para criar, por meio de um capacete, um protótipo para identificação de pessoas. Neste modelo foi possível fazer a identificação em diferentes situações de iluminação, e o retorno era gerado por áudio. Stoll *et al.* (2015) utilizaram uma estrutura parecida para capturar as imagens do sensor RGB-D e converter as imagens geradas em som, que seria alterado de acordo com a distância do obstáculo. Em testes com 21 jovens adultos consideraram o sistema promissor para uso interno, mas ineficiente para uso externo.

Poggi e Mattoccia (2016) construíram um dispositivo vestível composto por óculos com um sensor RGB-D customizado, uma FPGA[8] para processamento, uma luva com três pequenos motores para um retorno tátil ao usuário, uma pequena bateria, um boné com um fone de ouvido para retorno de mensagens sintetizadas e um *smartphone*. Os autores utilizaram técnicas de aprendizagem profunda (*deep-learning*) para categorizar os obstáculos encontrados através de palavras para o usuário. Poggi *et al*. (2015) apresentaram um sistema similar para reconhecimento de faixa de pedestres.

Hoang *et al*. (2017) apresentaram um sistema de detecção de obstáculos em tempo real que utiliza o sensor Kinect para capturar o ambiente e, se o sistema detectar obstáculos, ele fornece um retorno tátil e de áudio ao usuário. Este sistema também necessita de uma mochila com um *laptop* como central de processamento.

Rizzo *et al*. (2017) uniram os sinais recebidos de uma câmera de profundidade (*stereo camera*) e sensores infravermelhos em um sensor que denominou “Fusion”. A imagem gerada pelo sensor foi processada por uma rede neural convolucional para detecção de obstáculos.

[8] *Field-programmable gate array*.

Jiang *et al*. (2019) propuseram um sistema vestível baseado em uma câmera para detecção de profundidade (*stereo vision*) e. para processamento da identificação de obstáculos. as imagens eram enviadas para um servidor na internet.

Outros sensores também são utilizados em bengalas eletrônicas. Kassim *et al.* (2016) utilizaram uma unidade de Rádio Frequência para identificação de *transponders* (RFID) para, através destes *transponders*, mapear a navegação interna de uma pessoa com deficiência visual. Durante a navegação, a antena posicionada na ponta da bengala faz a leitura dos *transponders* e, através de uma assistência por voz, faz o retorno da navegação ao usuário. Pisa *et al.* (2016) acoplaram à ponta de uma bengala uma unidade de radar de curto alcance, o que permitiu detectar obstáculos, assim como aferir a sua distância em relação ao usuário.

Niu *et al.* (2017) propuseram uma tecnologia de assistiva por redes neurais convolucionais que permite identificar portas e maçanetas em tempo real. O modelo utiliza imagens estereoscópicas e necessita de uma unidade de processamento gráfico.

Kumar e Meher (2015) apresentaram um sistema de detecção de objetos utilizando Redes Neurais Convolucionais (CNN) e Redes Neurais Recorrentes (RNR)[9], com que realizaram reconhecimento de objetos em ambientes internos e suas cores, fornecendo ao usuário como retorno informação via áudio; como resultado obtiveram uma acurácia de 94,27% com base própria de imagens e 43,50% em base de imagens de terceiros.

Islam e Sadi (2018) utilizaram CNNs para identificar buracos no caminho, cm que obtiveram excelentes resultados. Eles utilizaram um conjunto de dados com imagens com buracos pelo caminho e outro com imagens de caminhos sem buracos. O conjunto de imagens de caminhos sem buracos foi construído com imagens tiradas de um ângulo mais amplo, portanto é possível que a CNN tenha aprendido essa diferença em relação ao outro conjunto, o que, evidentemente, não faz parte do problema a ser resolvido e que pode ter facilitado o trabalho.

Berriel *et al.* (2017a) apresentaram, em seu trabalho, uma rede neural convolucional para identificação de faixas de pedestre em imagens de satélite com acurácia média de 96,9%. Barriel *et al.* (2017b) expandiram o escopo do trabalho aplicando ao modelo imagens obtidas a partir do solo, experimento em que obtiveram

[9] Do inglês: *Recurrent Neural Network* (RNN).

uma acurácia 96,51% na identificação. Os trabalhos demonstraram a validade do modelo proposto na identificação de faixa de pedestres e produziram um grande conjunto de dados de imagens classificadas para esse fim.

Malek *et al.* (2017) propuseram um método para descrever o ambiente para uma pessoa com deficiência visual em tempo real. Para isso, utilizaram as técnicas de *Local Binary Pattern (LBP), Histogram of Oriented Gradient (HOG)* e *Bag of Words* (BoW) (WU; HOI; YU, 2010) para extrair as características de uma imagem capturada por uma câmera. Esse conjunto de características os autores utilizaram em um método de aprendizagem profunda (*Auto Enconder Neural Network - AE*) para gerar um novo vetor de características denominado de “fusion vector”, classificado através de algoritmo de regressão logística (multi-rótulos) determinando os objetos do ambiente.

Yang *et al.* (2018) apresentaram um modelo de segmentação semântica que tem por objetivo diminuir a carga de processamento para identificação de obstáculos para pessoas com deficiências visuais. O modelo separa a imagem em regiões distintas utilizando redes neurais convolucionais e obteve uma acurácia de 95% na análise pixel a pixel.

3.2 TECNOLOGIAS ASSISTIVAS BASEADAS EM *SMARTPHONES*

O avanço dos *smartphone*s criou uma era de pesquisa em diferentes campos. Atualmente, os *smartphone*s se tornaram comuns para quase todos os tipos de pessoas. Assim, os assistentes de caminhada desenvolvidos com base em *smartphone*s são flexíveis e fáceis de usar. Nessas abordagens, diferentes tipos de câmera ou sensor de *smartphone* são usados para capturar dados do ambiente do mundo real, e os processadores de *smartphone* são usados para processar os dados e gerar sinais de alerta na detecção de obstáculos ou objetos.

Vera, Zentena e Sales (2014) utilizaram uma câmera RGB e um ponteiro laser para desenvolver uma bengala virtual. No dispositivo, a câmera de um *smartphone* captura a reflexão do feixe de laser a partir do qual é calculada a distância do obstáculo para o usuário do sistema. Através de uma vibração gerada pelo celular, o usuário recebe os alertas de obstáculos, a intensidade da vibração determina a distância em relação ao mesmo. Experimentos realizados demonstraram que a respostas do usuário ao obstáculo acontecem de forma mais rápida do que se

estivesse utilizando uma bengala comum. O protótipo apresentou dificuldade em detectar buracos e obstáculos de pequeno tamanho.

Niitsu *et al*. (2014) colocaram quatro sensores em uma bengala: um sensor ultrassônico, um sensor infravermelho, uma bússola e um acelerômetro triaxial. O retorno ao usuário é realizado através de um fone de ouvido. Com sua matriz de sensores, este dispositivo obteve 100% de detecção para obstáculos grandes, ou cruzando, ou aproximando de outras pessoas, e 95% de precisão quando exposto a obstáculos pequenos. Os dados foram processados em um *smartphone*.

Krishnan *et al.* (2016) desenvolveram uma bengala utilizando sensores ultrassônicos e uma câmera RGB. Através do dispositivo utilizando um *smartphone*, o usuário é guiado utilizando mapas o gps, enquanto o sensor ultrassônico detecta obstáculos e os identifica através de imagens de uma microcâmera a partir de uma base de imagens própria utilizando o método SURF (BAY; TUYTELAARS; VAN GOOL, 2006). O trabalho não apresenta resultados claros.

Tapu *et al*. (2013) apresentaram um sistema de classificação e detecção de obstáculos em tempo real, utilizando captura de vídeo a partir da câmera de um *smartphone*. Eles construíram um *framework* que inclui rastreamento, estimativa de movimento e técnicas de agrupamento. O modelo proposto classificava os objetos identificados em 4 classes carros, bicicletas, pessoas e outros obstáculos.

Tapu *et al.* (2017a) apresentaram um novo trabalho onde modificaram a forma de extração de características e utilizaram *Support Vetor Machines* (SVM) como classificador incrementando a precisão na classificação das classes de objetos. A utilização SVM depende de conexão com a internet. No mesmo ano, Tapu *et al*. (2017b) introduziram um *framework* mais complexo denominado *"Deep-See"*, baseado em CNNs para detectar, rastrear e identificar objetos em ambientes internos e externos. O sistema apresentado por Tapu *et al.* (2017b) tem o inconveniente de necessitar de um computador pessoal (*laptop*) carregado em uma mochila para ser usado como central de processamento.

Saffoury *et al*. (2016) propuseram um sistema que associava um *smartphone* e um apontador *laser*, que criava uma triangulação de *lasers* utilizada para construir um algoritmo para evitar colisão. Obtiveram 5% de falsos alarmes e uma sensibilidade de 90% para obstáculos de 1 cm de largura. Os autores também forneciam retorno ao usuário através de áudio.

Alghamdi *et al.* (2013) apresentaram uma técnica para auxiliar deficientes visuais na navegação interna ou externa baseada na tecnologia RFID que cobre, aproximadamente, 0,5 metros de distância. O sistema é composto por um *smartphone* com um leitor de RFID ligado por *bluetooth* e um fone de ouvido. O equipamento faz a busca por rede sem fio pelas *tags* que identificam os locais gerando um alerta ao utilizador. A taxa de detecção com sucesso foi de 93,5% e falsos positivos de 1%; neste modelo não existe alerta em relação a obstáculos.

Nakajima e Haruyama (2013) propuseram um sistema de mobilidade interna baseado na identificação de luzes de *led* através de um sensor acoplado a um *smartphone*. O sistema também possui um sensor geomagnético. Cada lâmpada é identificada com um código através do qual o sistema recebe a informação da localização e obtem uma rota para o destino desejado. Os testes demonstraram uma precisão de 1 a 2 metros.

Tanveer *et al.* (2015) introduziram uma ferramenta de auxílio à locomoção, baseada em um *smartphone* Android; conectados ao aparelho, existem 3 sensores ultrassônicos (1 em uma luva, e dois em um par de óculos simulando cada retina do usuário) responsáveis por detectar possíveis obstáculos no caminho. A comunicação entre eles é realizada por *bluetooth*. A navegação ocorre por meio de sistema GPS em uma aplicação baseada no Google Maps, sendo a taxa de erro geral obtida de 5% para caminhos baseados em concreto ou ladrilhos.

Cheraghi *et al*. (2017) desenvolveram um sistema chamado *GuideBeacon*, que pode ser usado para orientação de pessoas com deficiência visual para auxiliá-las na navegação em ambientes internos. No aplicativo, é informado o destino desejado e, através de uma rede de *beacons* instalados no ambiente interno, é realizada a leitura por *bluetooth* destes dispositivos, e a orientação é realizada por sinal de voz.

Tepelea *et al.* (2017) propuseram um sistema auxiliar para deficientes visuais com múltiplas funções, como navegação interna e externa, reconhecimento de caracteres e realizar ligações através do *smartphone*. O sistema é composto por um *smartphone* e utiliza seus instrumentos e sensores ultrassônicos externos para navegação, com a comunicação realizada por *bluetooth*. O sistema é portátil e de baixo custo, permite a leitura de textos, possui fácil configuração de chamadas telefônicas, no entanto o sistema não foi avaliado para navegação interna e externa.

Kumar *et al.* (2017) apresentaram uma aplicação *mobile* integrada a um dispositivo Arduino e sensores ultrassônicos; permite a pessoas com deficiências visuais detectar obstáculos no caminho. Para navegação, a aplicação utiliza recursos do Google Maps. Quando o sensor ultrassônico encontra algum obstáculo, a aplicação verifica se é uma pessoa e se esta consta do banco de dados da aplicação fazendo, assim, a identificação através de uma rede neural.

Lin *et al*. (2017) propuseram um sistema guia baseado em um *smartphone* utilizando redes neurais convolucionais (CNN) para detecção de objetos. Para utilização de CNN para detecção de objetos, eles enviaram as imagens para processamento em um servidor com uma placa gráfica. O sistema em modo desconectado fornece apenas reconhecimento de rostos e escadas.

Parikh *et al.* (2018) apresentaram um modelo de reconhecimento de objetos visuais baseado em redes neurais convolucionais (CNN), de forma a orientar uma pessoa com deficiência visual em ambiente externo. O sistema está baseado em um *smartphone* com sistema Android e necessita de conectividade com a internet para enviar os *frames* de imagem capturados para um servidor que processará enviando o retorno ao usuário pelo *smartphone* através de voz. Como resultado, o sistema conseguiu realizar o reconhecimento de 11 objetos diferentes e foi capaz de guiar uma pessoa de forma mais eficaz do que com a bengala tradicional.

Fusco e Coughlan (2020) apresentaram um aplicativo para *smartphone* que permite a navegação em ambientes internos. Os autores frisam que a navegação interna é complexa, pois não pode depender de geolocalização, que é prejudicada nestes ambientes. O aplicativo utiliza um mapa 2D e pontos identificadores previamente mapeados em cada ambiente e, através da câmera do aparelho, faz a identificação da localização do usuário e sua orientação de navegação pelo mapa 2D utilizando para isso retorno de áudio. Como resultado dos testes realizados com usuários utilizando o sistema, obteve-se uma acurácia de 95% com margem de erro na localização da posição do usuário de 1,4 metros na sua média e o tempo para o posicionamento inicial em uma posição desconhecida no mapa foi de 14 a 30 segundos.

Neha e Shakib (2021) apresentaram uma plataforma móvel base em um *smartphone* Android para auxiliar pessoas com deficiências visuais em sua navegação através da detecção de obstáculos. Para isso, o sistema utiliza a extração de quadros de imagens em tempo real e técnicas como transformada de Hough e a biblioteca

OpenCV. O modelo verifica as linhas para determinar se o caminho é seguro e faz o retorno ao usuário.

Akkapusit e Ko (2021) demonstraram uma aplicação *mobile* para sistemas Android que faz o reconhecimento de objetos de interface em sistemas (tela) para auxiliar o deficiente visual no trabalho de clicar nos ícones. Para isso, utiliza a câmera do *smartphone* filmando a tela que deseja ser manipulada e, através de redes neurais convolucionais (CNN), processa os *frames* de imagem e fornece o retorno ao utilizador. Para testes, foi realizado um estudo com 20 participantes, 16 dos quais deram opiniões positivas ao protótipo. Os autores ainda relatam dificuldades em utilizar abordagens de detecção de objetos mais avançadas como *Faster*-RCNN, pois ainda não são aplicáveis a celulares.

3.3 CONSIDERAÇÕES PARCIAIS

Muitos trabalhos foram elaborados, propondo a utilização de tecnologias para a assistência ao deficiente visual. Foram abordados trabalhos que estavam focados apenas em detecção de obstáculos, enquanto outros estudos abordavam a navegação que podia ser interna, externa ou em ambos os ambientes. Ainda foram descritos trabalhos que visam apenas à identificação de locais, detecção específica de buracos ou faixas de pedestres, identificação de pessoas através de seus rostos ou a identificação de objetos.

Na Tabela 3, pode-se verificar, resumidamente, os trabalhos (em ordem de data decrescente) por domínio de atuação principal que foi identificado.

Tabela 3 - Trabalhos Analisados por área de domínio

Domínio	Trabalhos Pesquisados
Detecção de Obstáculos	Neha e Shakib (2021); Ansari *et al.* (2020); Romadhone *et al.* (2020); Jiang *et al.* (2019); Parikh *et al.* (2018); Yang *et al.* (2018); Buchs *et al.* (2017); Rizzo *et al.* (2017); Hoang *et al.* (2017); Lin *et al.* (2017); Tepelea *et al.* (2017); Dang *et al.* (2016); Everding *et al.* (2016); Krishnan *et al.* (2016); Saffory *et al.* (2016); Pisa *et al.* (2016); Poggi e Mattoccia (2016); Yasuno *et al.* (2016); Ye *et al.* (2016); Gupta *et al.* (2015); Kumar e Meher (2015); Takizawa *et al.* (2015); Tanveer *et al.* (2015); Kumar *et al.* (2014); Niitsu *et al.* (2014); Sadi *et al*. (2014); Vera *et al* .(2014); Tapu *et al.* (2017a); Tapu *et al.* (2017b); Tapu *et al.* (2013); Scherlen *et al.* (2007).
Detecção de Buracos	Islam e Said (2018).
Identificação de Locais	Alghamdi *et al* (2013).
Identificação de faixas de pedestres	Berriel *et al.* (2017a); Berriel *et al.* (2017b); Poggi *et al.* (2015).

Navegação Interna	Fusco e Coughlan (2020); Cheraghi *et al.* (2017); Kassim *et al.* (2016); Stoll *et al.* (2015); Nakajima e Haruyama (2013).
Navegação Externa	Krishnan *et al.* (2016); Gupta *et al.* (2015); Fan *et al.* (2014).
Identificação de Pessoas	Kumar *et al.* (2017); Majeed e Baadel (2016); Neto *et al.* (2016).
Identificação de Portas e Maçanetas	Niu *et al.* (2017); Wang *et al.* (2013).
Auxílio a utilização de interfaces de dispositivos inteligentes	Akkapusit; Ko (2021).
Descrição de ambientes	Malek *et al.* (2017).

Fonte: O autor (2021).

Encontraram-se os mais diversos meios de coleta e processamento dos dados para os modelos propostos no auxílio aos deficientes visuais. Na Tabela 4, podem-se visualizar, resumidamente, os recursos de processamento utilizados por autor em seus modelos. Observam-se diferentes abordagens com trabalhos menos complexos envolvendo microcontroladores e sensores ultrassônicos, até projetos que necessitam de um *smartphone* com conexão à internet para envio de imagens a um servidor.

Tabela 4 - Trabalhos analisados - unidades de processamento

Processamento	Trabalhos Pesquisados
Computador	Jiang *et al.* (2019); Islam e Sadi (2018); Yang *et al.* (2018); Berriel *et al.* (2017a); Berriel *et al.* (2017b); Hoang *et al.* (2017); Malek *et al.* (2017); Niu *et al.* (2017); Rizzo *et al.* (2017); Dang *et al.* (2016); Neto *et al.* (2016); Kumar e Meher (2015); Takizawa *et al.* (2015); Stoll *et al.* (2015); Wang *et al.* (2013); Scherlen *et al.* (2007).
Smartphone	Akkapusit e Ko (2021); Neha e Shakib (2021); Fusco e Coughlan (2020); Cheraghi *et al.* (2017); Kumar *et al.* (2017); Krishnan *et al.* (2016); Saffory *et al.* (2016); Tanveer *et al.* (2015); Niitsu *et al.* (2014); Vera *et al.* (2014); Alghamdi *et al.* (2013); Nakajima e Haruyama (2013).
Computador + *Smartphone*	Parikh *et al.* (2018); Lin *et al.* (2017); Tapu *et al.* (2017a); Tapu *et al.* (2017b); Majeed e Baadel (2016); Tepelea *et al.* (2017); Tapu *et al.* (2013).
Computador + Arduino	Kassim *et al.* (2016).
Raspberry Pi ou Arduino	Ansari *et al.* (2020); Romadhome e Husein (2020); Yassuno *et al.* (2016); Gupta *et al.* (2015); Fan *et al.* (2014); Kumar *et al.* (2014); Sadi *et al.* (2014).
FPGA	Poggi e Mattoccia (2016) - Odroid U3; Ye *et al.* (2016) – Gumstix Overo AirStorm; Poggi *et al.* (2015) - Odroid U3.
Outros Dispositivos	Pisa *et al.* (2016) - Radar; Buchs *et al.* (2017) – sensor IR industrial.
Não foi definido	Everding *et al* (2016).

Fonte: O autor (2021).

Na Tabela 5 estão descritos os sensores que foram utilizados em cada um dos trabalhos estudados durante este levantamento. Os sensores utilizados vão

desde sensores de baixíssimo custo, como os ultrassônicos, aos extremamente custosos, como um sensor radar.

Na Tabela 5, os sensores denominados como IMU (*inertial measurement unit*) são sensores de localização e posicionamento, como bússola digital e giroscópio, podendo ser industriais como proposto por Dang *et al.* (2016) e Ye *et al.* (2016) ou simplesmente leituras destes sensores a partir de um *smartphone*. Os sensores demonstrados estavam, na maioria dos trabalhos, montados em bengalas ou óculos.

Tabela 5 - Trabalhos analisados - sensores

	Câmeras/Imagem				**Sensores**					
Trabalho	RGB	RGBD	Stereo	IR	Ultrassônico	Laser	Rfid	GPS	IMU	Radar
Akkapusit e Ko (2021)	X									
Alghamdi *et al.* (2013)							X			
Ansari *et al.* (2020)					X					
Berriel *et al.* (2017a)	X									
Berriel *et al.* (2017b)	X									
Buchs *et al.* (2017)				X						
Cheraghi *et al.* (2017)							X			
Dang *et al.* (2016)	X					X			X	
Everding *et al.* (2016)			X							
Fan *et al.* (2014)		X			X			X		
Fusco e Coughlan (2020)	X									
Gupta *et al.* (2015)					X			X		
Hoang *et al.* (2017)		X								
Islam e Sadi (2018)	X									
Jiang *et al.* (2019)			X			X				
Kassim *et al.* (2016)							X		X	
Krishnan *et al.* (2016)	X				X			X		
Kumar e Meher (2015)	X									
Kumar *et al.* (2014)					X					
Kumar *et al.* (2017)	X									
Lin *et al.* (2017)	X									
Majeed e Baadel (2016)	X				X					
Malek *et al.* (2017)	X									
Nakajima e Haruyama (2013)							X			

Neha e Shakib (2021)	X									
Neto *et al.* (2016)		X								
Niitsu *et a.l* (2014)				X	X				X	
Niu *et al.* (2017)			X							
Parikh *et al.* (2018)	X									
Pisa *et al.* (2016)										X
Poggi e Mattoccia (2016)		X								
Poggi *et al.* (2015)		X								
Rizzo *et al.* (2017)			X	X						
Romadhome e Husein (2020)					X			X		
Sadi *et al.* (2014)					X					
Saffory *et al.* (2016)	X					X				
Scherlen *et al.* (2007)				X						
Stoll *et al.* (2015)		X								
Takizawa *et al.* (2015)		X								
Tanveer *et al.* (2015)					X			X	X	
Tapu *et al.* (2013)	X									
Tapu *et al.* (2017a)	X									
Tapu *et al.* (2017b)	X									
Tepelea *et al.* (2017)	X				X			X	X	
Vera *et al.* (2014)	X					X				
Wang *et al.* (2013)	X									
Yang *et al.* (2018)	X									
Yassuno *et al.* (2016)					X					
Ye *et al.* (2016)		X							X	

Fonte: O autor (2021).

Muitos trabalhos analisados não possuem descrito nenhum método de aprendizado ou extração de características baseado em imagens. Esses trabalhos utilizam-se de medições a partir de sensores ultrassônicos como proposto por exemplo por: Romadhone e Husein (2020), Ansari *et al.* (2020), Yasuno *et al.* (2016), Gupta *et al.* (2015), Fan *et al.* (2014), Kumar *et al.* (2014), Sadi *et al.* (2014). Em outros trabalhos, a medição foi realizada através de sensores infravermelhos: Buchs *et al.* (2017), Scherlen *et al.* (2007), Niitsu *et al.* (2014).

Também foram observados trabalhos que, para verificarem a distância, utilizaram sensores como Tags RFID, Beacons BLE como: Alghamdi *et al.* (2013), Cheraghi *et al.* (2017), Kassim *et al.* (2016), ou ainda sensor de radar como utilizado por Pisa *et al.* (2016).

Tepelea *et al.* (2017) utilizaram medições de diversos sensores em seu trabalho, como GPS e sensores ultrassônicos.

Na Tabela 6 estão elencados os trabalhos que utilizaram algum método de extração de características ou classificação baseada em imagens, sem discriminação em relação a como foram aferidos os resultados.

Tabela 6 - Trabalhos Analisados – Métodos de aprendizado

Autor	Área de Domínio	Métodos Utilizados
Akkupusit e Ko (2021)	Auxílio à utilização de interfaces de dispositivos inteligentes	MobileNet V2 (SANDLER *et al*, 2019)
Berriel *et al.* (2017a)	Identificação de faixas de pedestres	CNN - VGG (SIMONYAN; ZISSERMAN, 2015)
Berriel *et al.* (2017b)	Identificação de faixas de pedestres	CNN – VGG (SIMONYAN; ZISSERMAN, 2015)
Fusco e Coughlan (2020)	Navegação Interna através de detecção de símbolos	FastAdaBoost (SCHAPIRE, 2003), *Local Binary Patterns* (LBP) (OJALA; PIETIKAINEN; HARWOOD, 1994), *Histogram of Oriented Gradients* (HoG) (DALAL; TRIGGS, 2005), *Support Vector Machines* (SVM) (CORTES; VAPNIK, 1995)
Islam e Sadi (2018)	Detecção de buracos	CNN – AlexNet (KRIZHEVESKY; SUTSKEVER; HINTON, 2012)
Jiang *et al.* (2019)	Detecção de obstáculos	CNN – Resnet (HE *et al.*, 2016)
Krishnan *et al.* (2016)	Detecção de obstáculos e navegação	*Speeded Up Robust Features* - SURF (BAY; TUYTELAARS; VAN GOOL, 2006)
Kumar e Meher (2015)	Detecção e identificação de objetos	*Convolutional-Recursive deep learning* (SOCHER *et al.*, 2012)
Kumar *et al.* (2017)	Reconhecimento facial e navegação	Rede Neural Artificial – RNA (YONG; CHEN; WAN, 2013)
Lin *et al.* (2017)	Detecção de obstáculos	*Processo desconectado: Histogram of Oriented Gradients* (HoG) (DALAL; TRIGGS, 2005) e Haar Cascades (VIOLA; JONES, 2001) e *processo conectado:* F-RCNN (REN *et al.*, 2015) e YOLO (REDMON *et al.*, 2016)
Majeed e Baadel (2016)	Reconhecimento facial	Haar Cascades (VIOLA; JONES, 2001) e Fisherfaces (BELHUMEUR; HESPANHA; KRIEGMAN, 1997)
Malek *et al.* (2017)	Descrição de ambientes	Extração de características: *Local Binary Pattern - LBP* (OJALA; PIETIKAINEN; HARWOOD, 1994), *Histogram of Oriented Gradients* - HoG (DALAL;

		TRIGGS, 2005), Bag *of Words* - BoW (WU; HOI; YU, 2010); Aprendizado de características: Auto-Encoder neural network- AE (BALDI, 2012); Classificação: Regressão Logistica
Neha e Shakib (2021)	Detecção de obstáculos	Transformada de Hough e *Region of Interest* – ROI (FERNANDES; OLIVEIRA, 2008)
Neto *et al.* (2016)	Reconhecimento facial	*Histogram of Oriented Gradients* (HoG) (DALAL; TRIGGS, 2005), Principal Components Analysis - PCA (JOLLIFFE, 2002); KNN (BISHOP, 2006)
Niu *et al.* (2017)	Identificação de portas e maçanetas	CNN - YOLOv2 (REDMON; FARHADI, 2016);
Parikh *et al.* (2018)	Detecção de obstáculos	CNN - InceptionV3 (SZEGEDY *et al.*, 2016a)
Poggi e Mattoccia (2016)	Detecção de obstáculos	Ransac framework (CHOI; KIM; YU, 2009), Filtro de Kalman (KALMAN, 1960), CNN – LeNet (LECUN *et al*, 1989)
Poggi *et al.* (2015)	Identificação de faixas de pedestres	Algoritmo SGM (HIRSHMULLER, 2008), Ransac framework (CHOI; KIM; YU, 2009), Filtro de Kalman (KALMAN, 1960), CNN – LeNet (LECUN *et al.*, 1989)
Rizzo *et al.* (2017)	Detecção de obstáculos	CNN
Tapu *et al.* (2013)	Detecção e reconhecimento de objetos	*Scale-Invariant Feature Transform* (SIFT) (LOWE, 1999)*; Speeded Up Robust Features* - SURF (BAY; TUYTELAARS; VAN GOOL, 2006); *Bag of Visual Words* - BoVW (CSURKA *et al.*, 2004); Histogram *of Oriented Gradients* -HoG (DALAL; TRIGGS, 2005)
Tapu *et al.* (2017a)	Detecção e reconhecimento de objetos	*Bag of Visual Words* - BoVW (CSURKA *et al.*, 2004); representação de imagem – VLAD (JEGOU; DOUZE; SCHMID, 2011); *Histogram of Oriented Gradients* -HoG (DALAL; TRIGGS, 2005); *Support Vector Machines* (SVM) (CORTES; VAPNIK, 1995)
Tapu *et al.* (2017b)	Detecção e reconhecimento de objetos	CNN - YOLO (REDMON *et al.*, 2016)
Yang *et al.* (2018)	Detecção de obstáculos	CNN SegNet (BADRINARAYANAN; KENDAL; CIPOLLA, 2017)
Ye *et al.* (2016)	Detecção de obstáculos	Gaussian Mixture Model – GMM (BISHOP, 2006)

Fonte: O autor (2021).

Os autores determinaram os resultados de seus trabalhos utilizando vários métodos, como em testes de utilização por voluntários em que os autores anotavam a percepção do usuário, ou tempo em um determinado percurso ou, ainda, o número de detecções corretas no percurso e a distância média de detecção pelos sensores.

Akkapusit e Ko (2021) aferiram os resultados do seu projeto coletando a percepção de pessoas que utilizaram seu protótipo; Buchs *et al.* (2017) observaram a detecção de obstáculos em percurso durante testes de usabilidade por voluntários; Kassim *et al.* (2016) verificaram que, em testes voluntários utilizando o dispositivo proposto em seu trabalho, realizavam um percurso 33 segundos mais rápido do que utilizando uma bengala comum.

Gupta *et al.* (2015) utilizaram, em seu projeto, sensores ultrassônicos, e aferiram que o protótipo durante testes conseguiu identificar obstáculos de 4 a 200 centímetros de distância. Dang *et al.* (2016) aferiram em testes detecção de obstáculos a 180 centímetros de distância com uma acurácia acima de 90%, geralmente utilizando sensores ultrassônicos ou de infravermelho. Outros autores também determinaram a eficácia de seus protótipos deste modo como: Romadhone e Husein (2020), Tepelea *et al.* (2017), Tanveer *et al.* (2015), Takizawa *et al.* (2015), Stoll *et al.* (2015), Kumar *et al.* (2014), entre outros.

Na Tabela 7, podem-se verificar os resultados dos trabalhos que utilizaram imagens e algum algoritmo de aprendizado ou extração de características para a detecção de objetos ou reconhecimento de pessoas.

Observa-se que trabalhos como os de Akkupusit e Ko (2021), Fusco e Coughlan (2020), Neha e Shakib (2021) apresentaram seus resultados baseados na percepção dos voluntários a partir da utilização do dispositivo proposto. No artigo de Krishnan *et al.* (2016) não foi possível identificar a acurácia de detecção.

Tabela 7 - Trabalhos Analisados – Conjunto de Dados e Resultados

Autor	Conjunto de Dados	Resultados
Akkupusit e Ko (2021)	EgoGesture Dataset + Base Própria para modelos de botões de interface (368 imagens)	Testes com 20 voluntários; 16 deram opiniões positivas
Berriel *et al.* (2017a)	Google Street View / Google Maps (245.768 imagens – 74047 imagens com faixa de pedestre e 171721 sem faixas de pedestres)	Acurácia: 96,9%
Berriel *et al.* (2017b)	IARA Dataset / GOPRO Dataset (Bases Próprias)	Acurácia: 96,51%
Fusco e Coughlan (2020)	Não utilizaaram	Realizaram testes de percurso que demonstraram a validade da proposta apresentada
Islam e Sadi (2018)	KITTI Road Dataset (289 imagens) / Pothole Detection Dataset (90 imagens)	Acurácia: 97,12%
Jiang *et al.* (2019)	Base Própria Imagens (200 imagens estereoscópicas)	Precisão de 76,6%
Krishnan *et al.* (2016)	Base Própria	Não há informações sobre acurácia
Kumar e Meher (2015)	Composição de Base Própria (330 imagens para treinamento e 455 imagens para testes) + MIT Indoor Dataset	Acurácia de 94,27% para base própria Acurácia de 43,5% para base de terceiros
Kumar *et al.* (2017)	Base Própria	Acurácia: 90% na identificação de pessoas

Lin *et al* (2017)	Base Própria – 1710 imagens	Acurácia: 60%
Majeed e Baadel (2016)	Base Própria	Acurácia acima de 90%
Malek *et al.* (2017)	Base Própria (130 imagens – 61 treinamento e 70 testes)	Acurácia de 85,4% a 90,06% em testes de laboratório
Neha e Shakib (2021)	Não utilizou	Testes de navegação
Neto *et al.* (2016)	Base Própria	Acurácia 94,26%
Niu *et al.* (2017)	Base Própria	Introduziram a base de imagens de mãos e maçanetas e o modelo precisa de ajustes para a correta detecção
Parikh *et al*. (2018)	Base Própria composta por 35 mil imagens de diversas fontes	Acurácia de 96,39% em testes de laboratório
Poggi e Mattoccia (2016)	Base Própria	Acurácia de 97,93% para cenas urbanas e naturais
Poggi *et al.* (2015)	2500 imagens de celular de cenários urbanos	Acurácia de 88% a 94% no reconhecimento de faixas de pedestres dependendo da categoria
Rizzo *et al.* (2017)	Base Própria	Demonstração de processo de fusão de sensores
Tapu *et al.* (2013)	4500 imagens extraídas do Pascal Dataset	Acurácia por tipo do objeto: Cars – 95%; Bikes – 87%; People – 94%; Obstacles – 90%;
Tapu *et al.* (2017a)	4500 imagens extraídas do Pascal Dataset + 1200 imagens base própria	Acurácia: Cars – 95,8%; Bikes – 90,7%; People – 95,4%; Obstacles – 93,9%;
Tapu *et al.* (2017b)	ImageNet (treinamento) e VOT2016 (testes)	Acurácia: Vehicle: 94%; Bikes: 91%; Pedestrian: 95%; Static Obstrution: 90%
Yang *et al.* (2018)	ADE20k Dataset; Pascal Dataset e COCO Dataset	Acurácia média de 88% e, se considerada a classificação pixel-a-pixel, a acurácia é de 95,3%
Ye *et al.* (2016)	Base Própria	Acurácia de detecção acima de 86,7% dependendo da classe do objeto.

Fonte: O autor (2021).

Em relação aos demais trabalhos estudados, podem-se verificar diversas abordagens diferentes em relação à metodologia aplicada, diferentes objetivos e diversas bases de dados para treinamento e testes dos modelos; foi possível observar acurácias de 60% a 97% na detecção de objetos ou indivíduos.

Uma característica muito comum nos trabalhos estudados é a utilização de conjuntos de dados próprios para treinamento e testes dos modelos nas pesquisas. Isto pode demonstrar a dificuldade de encontrar dados tabulados para aferir resultados de detecção quando aplicados a modelos de auxílio a deficientes visuais.

4 METODOLOGIA

Nesta seção, é apresentada uma descrição detalhada do método proposto para extração de características das imagens a partir de redes neurais convolucionais, redução de dimensionalidade e classificador utilizado.

4.1 ESTRUTURAÇÃO DO MÉTODO E JUSTIFICATIVAS

O método foi estruturado em quatro etapas. A primeira etapa constitui a extração de características através da utilização de redes neurais convolucionais (CNN); para isso, foi utilizado o método de *transfer learning* com parâmetros pré-treinados no conjunto de dados ImageNet.

A ascensão dos métodos de aprendizagem profunda (LECUN; BENGIO; HINTON, 2015; SCHIMIDHUBER, 2015), especialmente as redes neurais convolucionais (CNNs) (HOWARD *et al.*, 2017; CAI *et al.*, 2016; KRIZHEVSKY; SUTSKEVER; HINTON, 2012), foi responsável por muitos avanços no reconhecimento de objetos e classificação de imagens.

No entanto, o processo de aprendizagem de uma CNN requer uma grande quantidade de amostras de imagens rotuladas para que a rede possa estimar milhões de parâmetros. A tarefa de rotular imagens é custosa e demorada, assim impedindo a aplicação de CNNs em problemas com dados de treinamento limitados (OQUAB *et al.*, 2014).

Oquab *et al*. (2014) demostraram que representações de imagens aprendidas com CNNs em conjuntos de dados rotulados em grande escala podem ser eficientemente transferidos para outras tarefas de reconhecimento visual com um limitado conjunto de imagens para treinamento. Eles reutilizaram camadas treinadas em um grande conjunto de dados para calcular a representação de imagem de nível médio para imagens em outro conjunto de dados, levando a uma classificação significativamente melhorada de resultados.

Esta técnica, conhecida como aprendizagem por transferência, foi aplicada com sucesso em diferentes cenários (MONTEIRO *et al.*, 2017; SHIN *et al.*, 2016; CAO *et al.*, 2013).

Na primeira etapa do modelo, serão adotadas as redes neurais convolucionais VGG16 e VGG19 (SIMONYAN; ZISSERMAN, 2015) como extratores de

características, eliminando suas camadas de classificação (*Fully Connected).* As camadas de convolução são utilizadas com pesos pré-treinados com o conjunto de imagens ImageNet (RUSSAKOVSKY *et al.*, 2015). A última camada convolucional de ambas as arquiteturas (VGG16 e VGG19) geram matrizes de tamanho ($7 \; x \; 7 \; x \; 512$); deste modo, sem a aplicação de uma cada de *pooling,* há um total de 25.088 características extraídas da imagem.

Alternativamente, poderia ser utilizada a camada de *pooling* (médio global), o que acarretaria uma saída de 512 características, mas, como demonstrado por Breve e Fischer (2020), a aplicação desta camada não representou vantagem na construção do modelo com as CNNs VGG16 e VGG19.

Ao final do processo de extração com as redes convolucionais VGG16 e VGG19, obtêm-se duas matrizes de 25.088 características cada, concatenadas em uma matriz final de 50.176 características.

A segunda etapa constituirá da seleção das características a partir da matriz com 50.176 características extraídas da imagem por meio das redes convolucionais VGG16 e VGG19, deste modo, realizando uma redução de dimensionalidade antes da montagem do grafo.

Para essa tarefa, serão utilizados, alternativamente, os métodos de *Principal Components Analysis* (PCA) (JOLLIFFE, 2002) e *Uniform Manifold Approximation and Projection* (UMAP) (MCINNES; HEALY; MELVILLE, 2020).

Niskanen e Silvén (2013) afirmam que o PCA é o método de redução de dimensionalidade mais amplamente utilizado, capaz de preservar todos os dados, mas nem sempre é suficiente para “explicar” todas as características do conjunto. Em seu estudo, Breve e Fischer (2020) escolheram o PCA como redutor de dimensionalidade e verificaram que, para se obterem melhores resultados na classificação, não são necessários mais do que 20 componentes principais.

UMAP é um método de redução de dimensionalidade adequado para grandes conjuntos de dados, possui funcionamento similar ao T-SNE (MAATEN; HINTON, 2008), mas com um custo computacional mais baixo. Kobak e Linderman (2021) e Becht *et al.* (2018) verificaram que o UMAP possui tempos de execução mais rápidos que o T-SNE e alta reprodutibilidade de dados.

Nesta segunda fase do modelo proposto, a matriz com 50.176 características será submetida a cada um dos modelos de redução de dimensionalidade, gerando matriz com p componentes variando de 1 a 25 elementos. Cada uma das saídas

geradas será utilizada para a construção do grafo e posterior classificação pelo modelo de competição e cooperação de partículas.

Para se analisar a diferença dos componentes gerados pelos algoritmos PCA e UMAP, foi realizado um teste, de onde foram extraídos 5 componentes principais a partir da matriz de 50.176 características gerada pelas CNNs VGG16 e VGG19. A partir do resultado do teste, foram gerados gráficos de dispersão com a visualização das amostras no plano, de modo que seja possível visualizar a separação dos conjuntos proporcionados pela seleção de características (Figura 8).

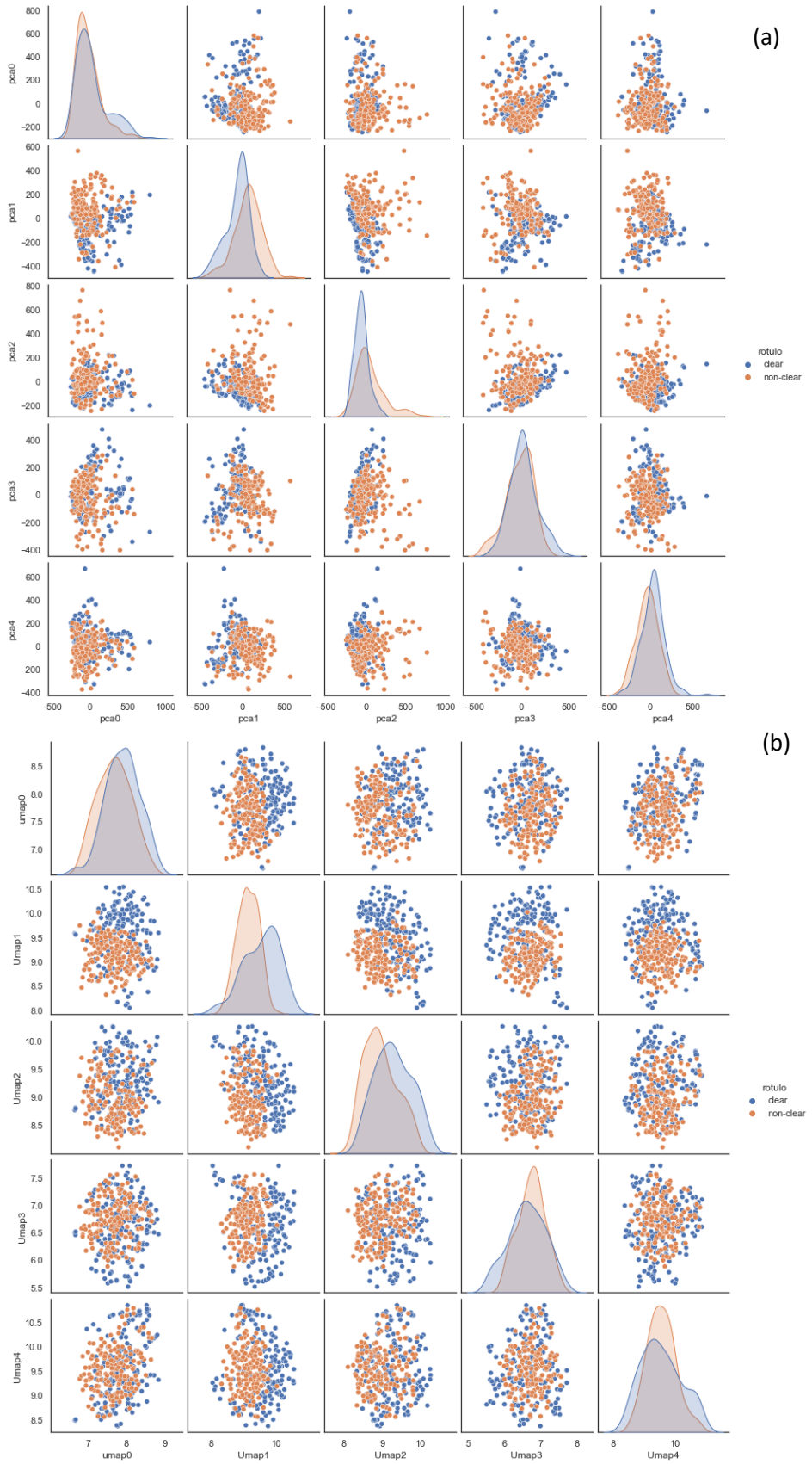
Na Figura 8(a), pode-se verificar o comportamento dos componentes quando se utiliza o algoritmo PCA para seleção de características. Na Figura 8(b), observa-se o comportamento para o algoritmo UMAP. Nesta análise, pode-se visualizar uma maior separabilidade dos pontos quando a seleção de características foi realizada com o algoritmo UMAP, o que deve permitir maior acurácia na classificação do conjunto de dados.

A terceira fase é constituída pela construção da rede complexa (grafo) necessária para o classificador. Para esta etapa, serão utilizados os componentes principais extraídos da fase de seleção de características (UMAP, PCA) e, então, é construído um grafo não orientado e sem pesos, onde cada vértice representa uma imagem e as arestas conectam os vértices através dos seus *k*-vizinhos mais próximos. O algoritmo *K-nearest neighbors* utiliza a distância euclidiana como métrica para a seleção dos vizinhos próximos. Nesta fase, são geradas redes com variação no número k de vizinhos próximos de 1 a 25 elementos para cada um dos conjuntos de dados com p componentes gerados pelos redutores de dimensionalidade PCA e UMP.

Na quarta fase, o grafo é alimentado para o classificador de Competição e Cooperação de Partículas (PCC) (BREVE *et al.*, 2012). O PCC possui complexidade computacional de $O(n)$, onde n se refere à quantidade de imagens. Breve e Fischer (2020) afirmaram, em seu trabalho, que este classificador é adequado para execução rápida em *smartphone*s e já foi estendido para realizar aprendizagem indutiva.

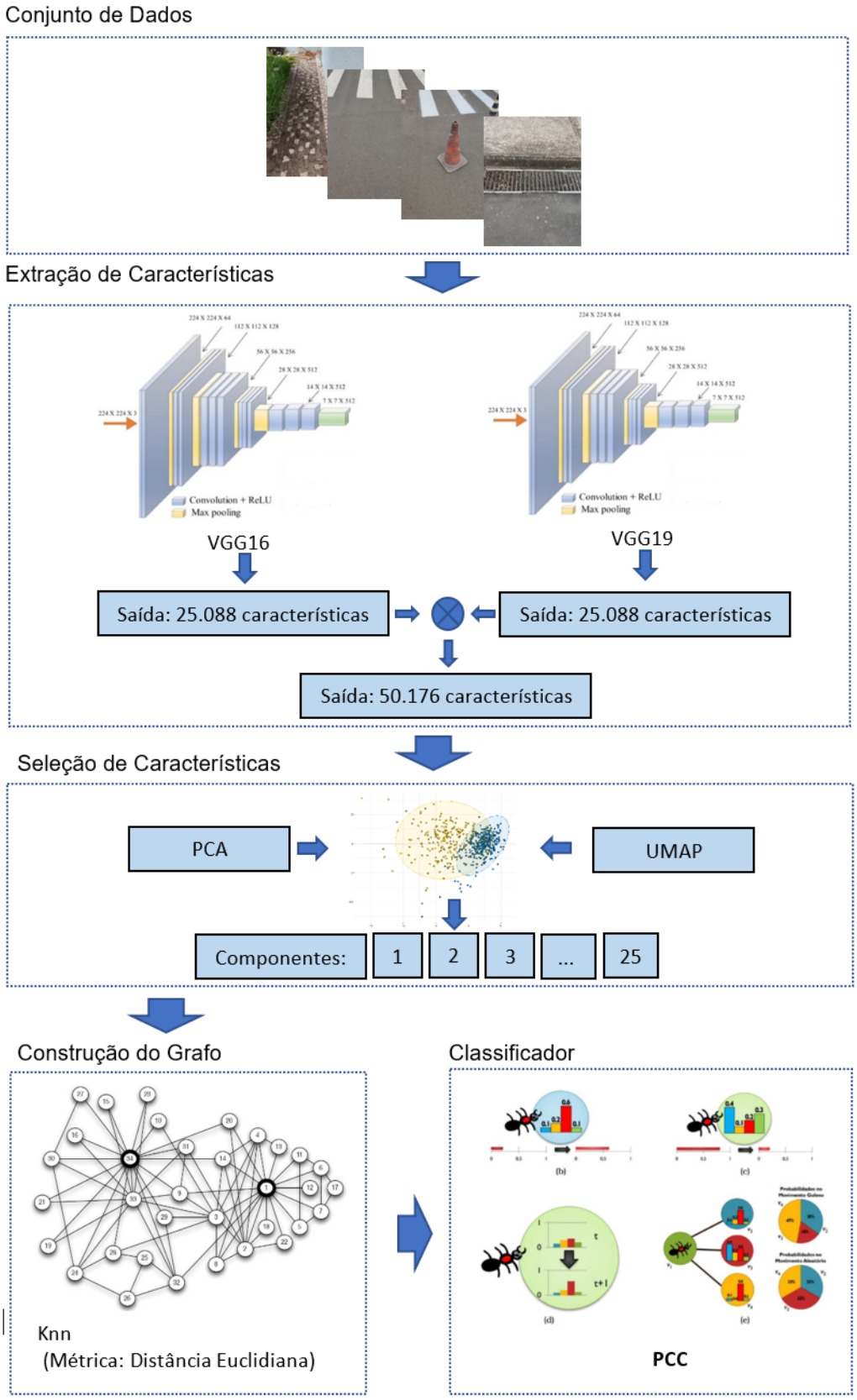
Pode-se verificar o método proposto de forma resumida na Figura 9.

Figura 8 – Análise de Componentes Principais - (a) PCA (b) UMAP.



Fonte: O autor (2021).

Figura 9 – Ilustração das etapas do método proposto.



Fonte: O autor (2021).

4.2 CONFIGURAÇÃO PARA TESTES

As redes convolucionais VGG16 e VGG19 são empregadas como extratores de características utilizando pesos pré-treinados com o conjunto de dados ImageNet, sem camada de *pooling* e com a última cada convolucional gerando uma matriz no formato $(7\ x\ 7\ x\ 512\ )$ com 25.088 características para cada arquitetura. Ambas as matrizes são concatenadas, gerando uma única matriz com 50.176 características.

Para a seleção de características, os algoritmos PCA e UMAP foram utilizados. O PCA foi utilizado a partir da biblioteca *Numpy* para a linguagem Python, com configurações padrão e apenas variando o número de componentes $p = \{1, \ldots, 25\}$ para a saída do modelo. O modelo de redução de dimensionalidade UMAP é fornecido pelos autores e está implementado para Python, onde também foi realizada a variação do número de componentes $p = \{1, \ldots, 25\}$ para a saída do modelo; da configuração de execução padrão foi alterado apenas o número de vizinhos que o algoritmo utiliza para a construção do grafo de 15 para 20 vizinhos próximos.

Na construção do grafo a partir da saída do processo de seleção de características utilizando o método dos k vizinhos próximos, também foi realizada a variação de $k = \{1, \ldots, 25\}$ para cada uma das configurações de valores de p de cada um dos redutores de dimensionalidade aplicados. O algoritmo knn utiliza como métrica a distância euclidiana.

O classificador PCC foi configurado com o redutor de processamento como $\Delta_v = 0.1$, fator de sorteio de movimento guloso ou aleatório como $P_{grd} = 0.5$. Para o controle de parada do classificador, o número máximo de iterações foi configurado em $max_{Ite} = 1.000.000$, número máximo de iterações sem melhorar a média de dominância dos vértices como $es_{chk} = 2.000$. Como é um classificador semi-supervisionado, foram fornecidos ao modelo 20% das imagens como rótulos para o processamento.

A extração de características foi realizada uma vez para cada uma das arquiteturas de CNNs (VGG16 e VGG19), e cada mudança no número de componentes p do processo de seleção de características é novamente processada. Para cada combinação de $p\ e\ k$, são realizados 50 testes de classificação com o

conjunto de dados no classificador PCC; a cada execução do classificador, os rótulos são novamente sorteados. A Tabela 8 resume o processo de testes realizado.

Tabela 8 - Modelagem dos testes realizados

P	K	Seleção de características	Classificador PCC
1	$1..25$	PCA	50 execuções com sorteio de rótulos a cada execução de k
⋮	⋮	⋮	⋮
25	$1..25$	PCA	50 execuções com sorteio de rótulos a cada execução de k
1	$1..25$	UMAP	50 execuções com sorteio de rótulos a cada execução de k
⋮	⋮	⋮	⋮
25	$1..25$	UMAP	50 execuções com sorteio de rótulos a cada execução de k

Fonte: O autor (2021).

Para verificar o resultado dos testes, foi utilizada a acurácia (acc) aferida a cada execução do classificador, e extraída a média e desvio padrão de 50 execuções para cada combinação de p e k.

4.3 CONJUNTO DE DADOS

Foram realizados testes com um conjunto de dados que possui 342 imagens divididas em duas classes, 175 que demonstram um caminho livre e 167 um caminho com obstáculos. Essas imagens foram obtidas com a câmera de um *smartphone* e redimensionadas para 750x1000 *pixels*. O *smartphone* foi colocado na altura do peito do usuário e inclinado aproximadamente de 30º a 60º em relação ao solo, para que pudesse capturar alguns metros à frente do caminho a ser percorrido, estando além do alcance de uma bengala.

Breve e Fischer (2020) apresentaram o conjunto de imagens proposto para testes. Na Figura 10 podem-se visualizar exemplos de imagens existentes.

Embora não seja grande, o conjunto de dados cobre áreas internas e externas, com diferentes tipos de piso, seco ou molhado, diferentes situações de iluminação e diferentes tipos de obstáculos como cones, buracos, animais entre outros. Realizados os testes, foi verificado o impacto na acurácia pelo modelo proposto no trabalho.

Figura 10 – Imagens extraídas do conjunto de dados proposto para testes.

(a) Caminho Limpo

(b) Caminho com obstáculos

Fonte: Adaptado de Breve e Fischer (2020).

4.4 MATERIAIS

Os testes foram realizados em um *notebook* com processador Intel I7-9750H de 2,6GHz com 32GB de memória RAM, com sistema operacional Windows 10 Home de 64bits. Os modelos foram desenvolvidos na linguagem Python, versão 3.8.12, executados na plataforma Anaconda onde foi utilizada a IDE[10] *Spyder* na versão 4.1.4.

[10] IDE: Ambiente de Desenvolvimento Integrado.

5 RESULTADOS

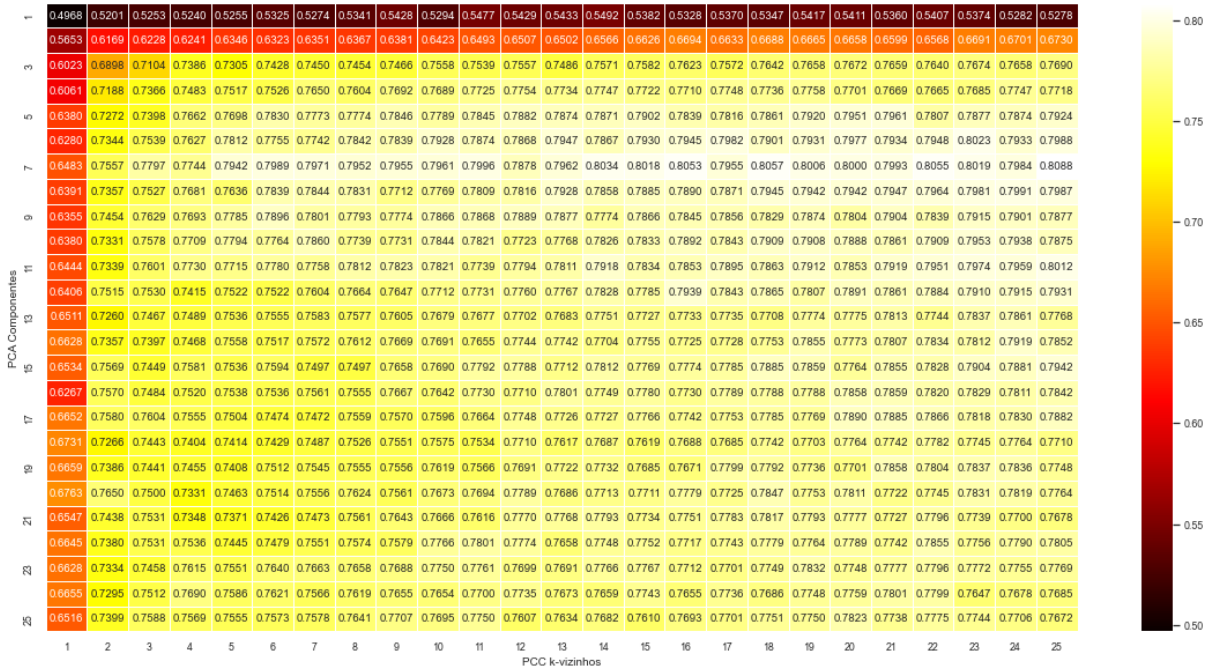
As técnicas propostas foram executadas com o conjunto de imagens, e os resultados estão expostos nas próximas subseções, analisando a classificação do PCC a partir da seleção de características por PCA, UMAP e finalizando com um comparativo entre as duas propostas e resultados publicados.

5.1 RESULTADOS PCC+PCA

Na Figura 11, pode-se verificar toda a variação de acurácia em p e k durante a classificação executada com o modelo de competição e cooperação de partículas PCC a partir do conjunto de dados selecionado através de PCA. Deve-se enfatizar que, para cada combinação de p e k, a acurácia é uma média de 50 execuções do classificador.

Para o conjunto de dados selecionado através de PCA, observa-se que, com $p = 7$, o classificador atingiu sua maior acurácia em $k = 25$. Mas também fica evidenciado no mapa de calor que, com 7 componentes, o classificador obteve seu melhor desempenho em praticamente todas as faixas de k.

Figura 11 – PCC+PCA Componentes - Mapa de calor de acurácia.



Fonte: O autor (2021).

Na Tabela *9* estão demonstrados os melhores desempenhos para a combinação de PCC+PCA, onde fica evidente que com $p = 7$ se obtiveram os melhores desempenhos do classificador.

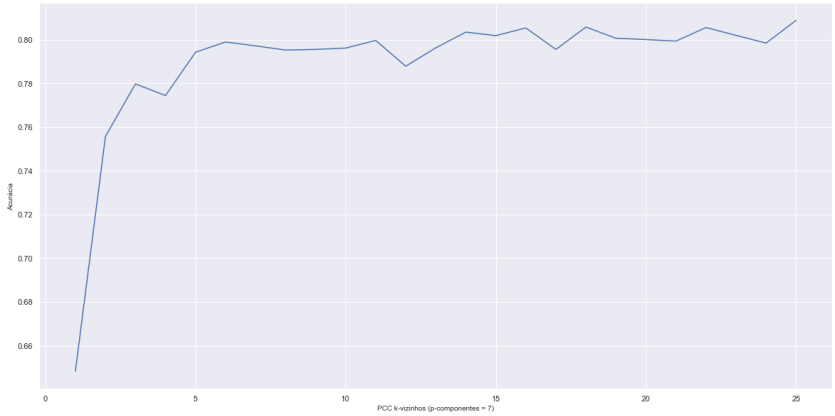
Tabela 9 - PCC+PCA Componentes – melhores acurácias

Posição	PCA (p)	k	Acurácia (acc)	Tempo (s)
1	7	25	80,88% ±2,36%	3,66s ±1,09s
2	7	18	80,57% ±2,93%	5,02s ±1,51s
3	7	22	80,55% ±2,63%	4,57s ±1,29s
4	7	16	80,53% ±2,09%	5,19s ±1,42s
5	7	14	80,34% ±2,21%	6,27s ±1,82s
6	6	23	80,23% ±2,70%	3,64s ±0,89s
7	7	23	80,19% ±3,14%	4,22s ±1,38s
8	7	15	80,18% ±2,70%	5,51s ±1,26s
9	11	25	80,12% ±2,51%	3,54s ±0,93s
10	7	19	80,06% ±3,64%	4,53s ±1,26s
11	7	20	80,00% ±3,07%	5,29s ±1,65s
12	7	11	79,96% ±2,14%	7,05s ±1,95s
13	7	21	79,93% ±3,45%	4,29s ±1,13s
14	8	24	79,91% ±2,09%	3,70s ±0,86s
15	7	6	79,89% ±2,18%	12,07s ±3,20s

Fonte: O autor (2021).

Na Figura 12, observa-se a curva de acurácia com $p = 7$ componentes principais variando no número de k vizinhos no classificador. O ponto de melhor desempenho está em $k = 25$.

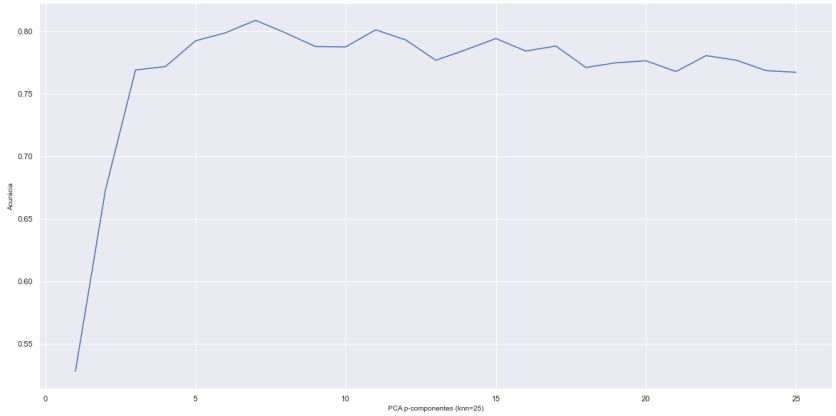
Figura 12 – Variação da acurácia em k com componentes $p = 7$ (PCC+PCA).



Fonte: O autor (2021),

Na Figura 13, observa-se a curva de acurácia com $k = 25$ vizinhos próximos, variando no número de p componentes principais na seleção de características com PCA. O ponto de melhor desempenho está em $p = 7$ a partir de onde decai a acurácia.

Figura 13 – Variação da acurácia em p com vizinhos $k = 25$ (PCC+PCA).



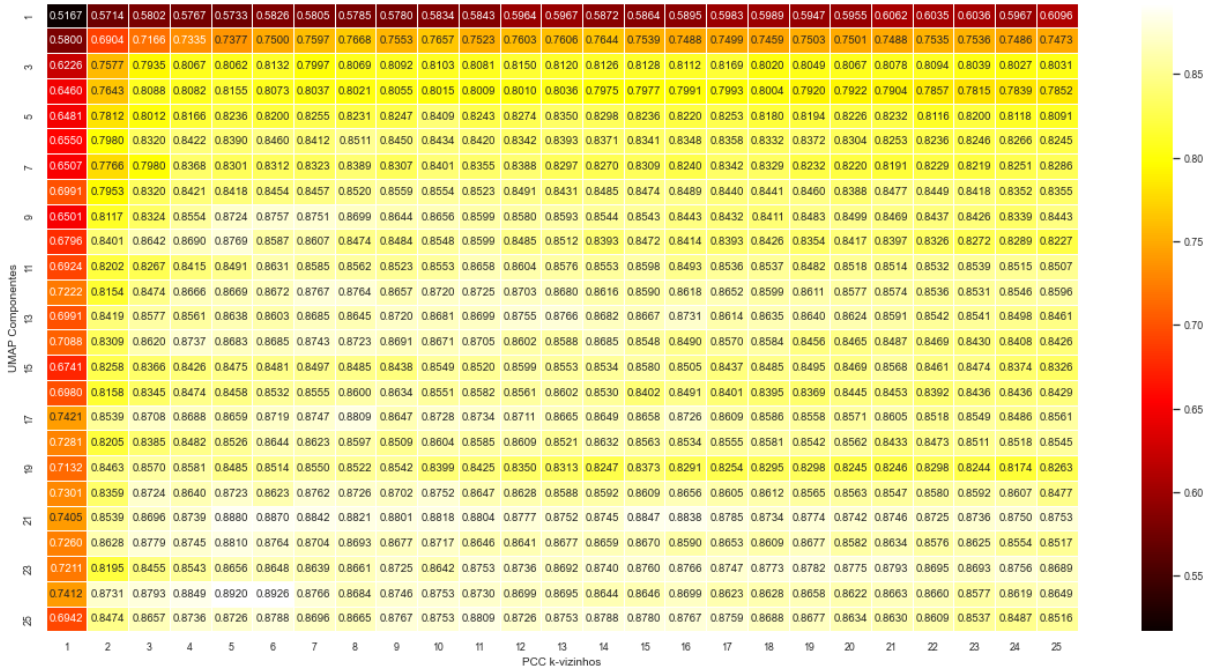
Fonte: O autor (2021).

5.2 RESULTADOS PCC+UMAP

Na Figura 14, pode-se verificar toda a variação de acurácia em p e k durante a classificação executada com o modelo de competição e cooperação de partículas PCC a partir do conjunto de dados selecionado através de UMAP. Deve-se enfatizar que, para cada combinação de p e k, a acurácia é uma média de 50 execuções do classificador.

Para o conjunto de dados selecionado através de UMAP, observa-se que, com $p = 24$, o classificador atingiu sua maior acurácia em $k = 6$. Mas também fica evidente no mapa de calor que, diferentemente do que ocorreu com o PCA, aqui se tem maior estabilidade na acurácia em diversos pontos do gráfico. É relevante observar que, com $p = 21$ componentes principais do classificador também se obteve um bom desempenho.

Figura 14 – PCC+UMAP Componentes - Mapa de calor de acurácia.



Fonte: O autor (2021).

Na Tabela 10, estão demonstrados os melhores desempenhos para a combinação de PCC+UMAP. Observa-se que, apesar de o melhor desempenho tenha ocorrido com $p = 24$, em $p = 21$ tem-se a maior recorrência de bons resultados.

Tabela 10 – PCC+UMAP Componentes – melhores acurácias.

Posição	UMAP (p)	k	Acurácia (acc)	Tempo (s)
1	24	6	89,26% ±1,83%	10,73s ±3,09s
2	24	5	89,20% ±2,59%	12,86s ±4,23s
3	21	5	88,80% ±1,63%	14,11s ±4,63s
4	21	6	88,70% ±0,18%	11,57s ±3,50s
5	24	4	88,49% ±1,93%	14,08s ±4,16s
6	21	15	88,47% ±1,73%	5,03s ±1,47s
7	21	7	88,42% ±1,54%	8,94s ±2,19s
8	21	16	88,38% ±0,17%	4,66s ±1,58s
9	21	8	88,21% ±0,14%	8,89s ±3,06s
10	21	10	88,18% ±1,53%	6,83s ±1,79s
11	22	5	88,10% ±1,79%	12,35s ±4,74s
12	17	8	88,09% ±1,26%	9,05s ±3,06s
13	25	11	88,09% ±1,45%	6,10s ±2,00s
14	21	11	88,04% ±1,83%	6,27s ±1,73s
15	21	9	88,01% ±1,73%	7,05s ±1,85s

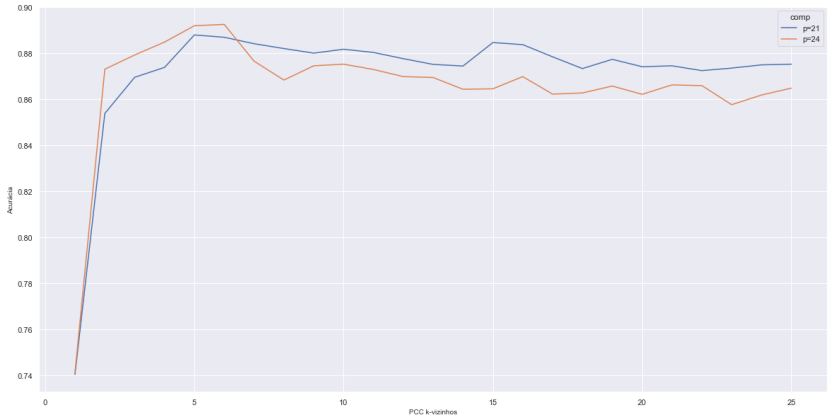
Fonte: O autor (2021).

Na Figura 15, observam-se as curvas de acurácia com $p = 21$ e $p = 27$ componentes principais variando no número de k vizinhos no classificador. O ponto de melhor desempenho para $p = 21$ está em $k = 5$ e, para $p = 24$, está em $k = 6$. No

gráfico foi traçada a variação de ambos os valores para p, pois foram os melhores resultados na combinação PCC+UMAP.

Percebe-se que, para $p = 21$, a curva é mais estável em k, mas o topo de acurácia ocorreu em $p = 24$. O k interfere no tempo de convergência do classificador onde quanto menor o k, maior o tempo para o classificador determinar o resultado.

Figura 15 – Variação da acurácia em k com componentes $p = 21$ e $p = 24$ (PCC+UMAP).

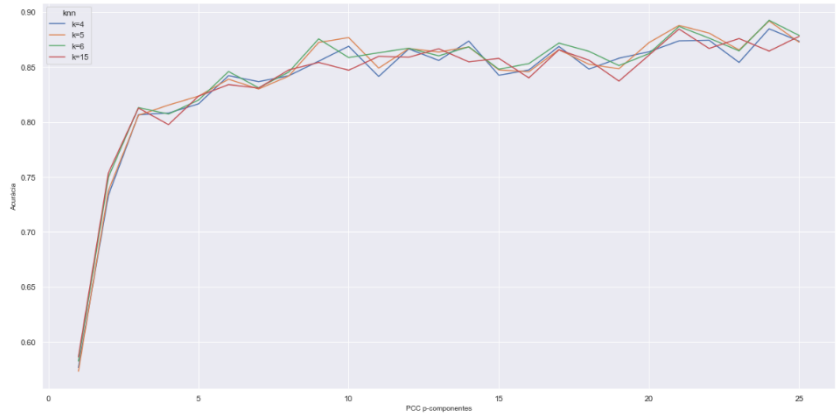


Fonte: O autor (2021).

Na Figura 16, observam-se as curvas de acurácia com $k = \{4, 5, 6, 15\}$ vizinhos próximos variando no número de p componentes principais no classificador. O ponto de melhor desempenho está em $k = 6$, mas se pode verificar que são muito próximos os valores de acurácia para os outros valores de k plotados no gráfico.

Na combinação do classificador PCC com a seleção de característica UMAP, não foi possível definir um melhor k. Deve-se levar em consideração a relação entre tempo de execução do classificador e a acurácia obtida, pois quanto maior o valor de k o tempo de convergência do PCC tende a ser menor.

Figura 16 – Variação da acurácia em p com vizinhos $k = \{4, 5, 6, 15\}$ (PCC+UMAP).



Fonte: O autor (2021).

5.3 COMPARATIVO DE RESULTADOS

Na comparação dos 5 resultados com maiores acurácias entre os dois métodos de classificação apresentados, o desempenho da combinação de PCC+UMAP ficou, em média, 10% acima do desempenho da utilização do PCC combinado ao algoritmo PCA (Tabela 14).

Se considerasse um ranqueamento único para os dois métodos, a classificação do método PCC+PCA com a maior acurácia (80,88%) obtida ficaria na posição 511 de 1.250 análises médias realizadas. A posição 510 ficaria com o método PCC+UMAP, utilizando $p = 4$ e $k = 3$, com uma acurácia de $80{,}88\%$ e um desvio padrão de $3{,}11\%$. Deve-se enfatizar que cada análise citada é a média de 50 execuções dos métodos citados.

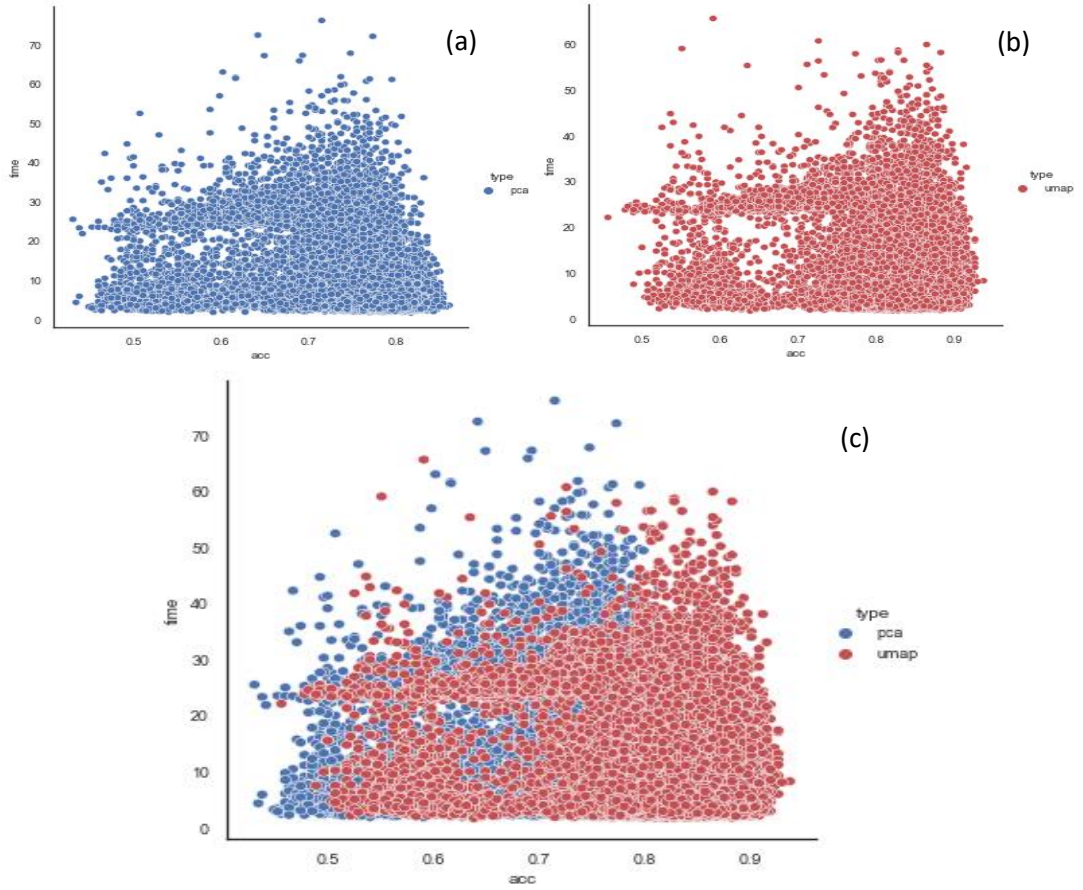
Tabela 11 - Acurácias: PCC+UMAP e PCC+PCA

Modelo	**Posição**	$\boldsymbol{p}$	$\boldsymbol{k}$	**Acurácia ($\boldsymbol{acc}$)**	**Tempo (s)**
PCC+UMAP	1	24	6	89,26% ±1,83%	10,73s ±3,09s
PCC+UMAP	2	24	5	89,20% ±2,59%	12,86s ±4,23s
PCC+UMAP	3	21	5	88,80% ±1,63%	14,11s ±4,63s
PCC+UMAP	4	21	6	88,70% ±0,18%	11,57s ±3,50s
PCC+UMAP	5	24	4	88,49% ±1,93%	14,08s ±4,16s
PCC+PCA	1	7	25	80,88% ±2,36%	3,66s ±1,09s
PCC+PCA	2	7	18	80,57% ±2,93%	5,02s ±1,51s
PCC+PCA	3	7	22	80,55% ±2,63%	4,57s ±1,29s
PCC+PCA	4	7	16	80,53% ±2,09%	5,19s ±1,42s
PCC+PCA	5	7	14	80,34% ±2,21%	6,27s ±1,82s

Fonte: O autor (2021).

Na Figura 17, podem-se observar as 62.500 execuções dos métodos analisados de forma granular, onde cada um dos métodos foi executado 31250 vezes com diferentes combinações de p e k. Na Figura 17(a), tem-se a dispersão de execuções do método PCC+PCA na relação de acurácia e tempo de execução. O mesmo ocorre na Figura 17(b) para o método PCC+UMAP. Na Figura 17 (c), tem-se a sobreposição das análises, onde fica evidente a maior acurácia obtida com o método PCC+UMAP.

Figura 17 – Análise de dispersão acurácia (acc) x tempo de execução ($time$) para os métodos PCC+PCA e PCC+UMAP – (a) PCC+PCA (b) PCC+UMAP (c) PCC+PCA e PCC+UMAP.



Fonte: O autor (2021).

Breve e Fischer (2020) divulgaram seu estudo com redes neurais convolucionais para extração de características, utilizando o classificador de competição e cooperação de partículas (PCC). Para isso, utilizaram as CNNs VGG16, VGG19 e a junção de ambas na extração de características e PCA para a seleção e redução da dimensionalidade das características extraídas pelas redes neurais. Os autores também testaram os modelos adicionando a camada de *polling* à rede, com o classificador exposto a 10% de amostras classificadas e com 20% de amostras classificadas do conjunto de dados.

Breve e Fischer (2020), em seu estudo, utilizaram a linguagem Python para as CNNs e a extração de características, e o restante do processo com a seleção de características e a classificação sendo utilizada a ferramenta Matlab.

Como neste trabalho, foram utilizados 20% de amostras classificada. Na Tabela 12 estão demonstrados os resultados comparados com Breve e Fischer (2020) utilizando 20% de amostras classificadas.

Tabela 12 – Comparativo de acurácia utilizando classificador PCC (com 20% de amostras classificadas)

Trabalho	Extração	Modelo	*Polling*	p	k	Acurácia (acc)
Proposto	VGG16+VGG19	PCC+UMAP	Não	24	6	**89,26% ±1,83%**
Proposto	VGG16+VGG19	PCC+PCA	Não	7	25	80,88% ±2,36%
Breve e Fischer (2020)	VGG16	PCC+PCA	Não	10	7	79,53% ±2,40%
Breve e Fischer (2020)	VGG19	PCC+PCA	Não	10	8	79,35% ±2,65%
Breve e Fischer (2020)	VGG16+VGG19	PCC+PCA	Não	14	4	79,43% ±2,65%
Breve e Fischer (2020)	VGG16	PCC+PCA	AVG(Global)	7	6	72,51% ±3,04%
Breve e Fischer (2020)	VGG19	PCC+PCA	AVG(Global)	15	3	71,52% ±3,28%
Breve e Fischer (2020)	VGG16+VGG19	PCC+PCA	AVG(Global)	10	6	73,43% ±3,10%
Breve e Fischer (2020)	VGG16	PCC+PCA	MAX(Global)	7	7	74,30% ±2,80%
Breve e Fischer (2020)	VGG19	PCC+PCA	MAX(Global)	20	8	72,28% ±3,87%
Breve e Fischer (2020)	VGG16+VGG19	PCC+PCA	MAX(Global)	20	4	73,19% ±3,35%

Fonte: O autor (2021).

Pode-se observar, na Tabela 12, que o modelo proposto, utilizando VGG16+VGG19 como extratores de características, UMAP para a seleção de características e PCC como classificador, obteve uma acurácia de 89,26%, a maior acurácia dentre as abordagens apresentadas.

Breve e Fischer (2020) também realizaram testes comparativos com a classificação, utilizando CNNs na classificação do conjunto de dados proposto. Na Tabela 13, estão demonstrados os dados de classificação destes testes para as CNNs que obtiveram as maiores acurácias apresentadas no trabalho citado.

A CNN VGG16, com ajuste de treinamento e sem camada de *polling* obteve uma acurácia de 89,40% com um desvio padrão de 6,50%, e a CNN Xception com uma camada de *polling* utilizando média global e ajuste no treinamento obteve uma acurácia de 91,68% com um desvio padrão de 3,58%. Estas são as CNNs que, na classificação do conjunto de dados proposto, obtiveram uma acurácia maior que a abordagem PCC+UMAP.

Tabela 13 – Comparativo de acurácia utilizando classificador PCC e CNNs (*Transfer learning*)

Trabalho	Extração	Modelo	*Polling*	p	k	Acurácia (acc)
Proposto	VGG16+VGG19	PCC+UMAP	Não	24	6	**89,26% ±1,83%**
Proposto	VGG16+VGG19	PCC+PCA	Não	7	25	80,88% ±2,36%
Breve e Fischer (2020)	VGG16	Softmax	Não (*fine-tunable*)			**89,40% ±6,50%**
Breve e Fischer (2020)	VGG16	Softmax	AVG (*fine-tunable*)			89,23% ±7,52%
Breve e Fischer (2020)	Xception	Softmax	AVG (*fine-tunable*)			**91,68% ±3,58%**
Breve e Fischer (2020)	MobileNet	Softmax	AVG (*fine-tunable*)			88,89% ±3,36%

Fonte: O autor (2021).

A abordagem PCC+UMAP apresentou uma acurácia (89,26%) muito próxima às CNNs e com uma menor variação no desempenho médio (desvio padrão de 1,83%), o que, provavelmente, representaria maior estabilidade na classificação das imagens em uma utilização real.

6 CRONOGRAMA DE EXECUÇÃO

A seguir, apresenta-se a programação das atividades previstas para o desenvolvimento da pesquisa aqui proposta.

Tabela 14 - Cronograma de Execução

	2021		2022		2023		2024	
	1ºS	2ºS	1ºS	2ºS	1ºS	2ºS	1ºS	2ºS
Disciplina: Estrutura de dados								
Disciplina: Aprendizado de Máquina								
Disciplina: *Digital Speech Processing*								
Disciplina: Estudos Especiais II								
Qualificação								
Realização de testes com o modelo								
Publicação de Artigos								
Defesa								

Fonte: O autor (2021).

Durante o primeiro semestre de 2021, foram cursadas as disciplinas de Estrutura de Dados, Aprendizado de Máquina e *Digital Speech Processing*, enquanto, no segundo semestre de 2021, está sendo cursada a disciplina de Estudos Especiais II.

Desta forma, com o aproveitamento de créditos provenientes do programa de pós-graduação em nível de mestrado e a conclusão das referidas disciplinas, conclui-se a totalidade necessária de créditos em disciplinas.

Durante o primeiro semestre de 2022, serão realizados novos testes, utilizando mais redes convolucionais e combinações entre elas na extração de características. Também devem ser testados outros modelos de seleção de características e novos classificadores. Para cumprimento dos prazos relativos à qualificação, a entrega da versão para os membros da banca deve ser feita 60 dias antes da data do exame de qualificação; o prazo máximo para que o exame aconteça é 05/10/2022.

Após o exame de qualificação deverão ser estudados métodos de melhoria na rede complexa gerada para o classificador PCC e ampliação do conjunto de imagens utilizada nos testes. Também deverá ser trabalhada a publicação de artigos no período. A defesa deve ocorrer até o final do 1º semestre de 2025.

7 CONSIDERAÇÕES FINAIS

Neste trabalho, foi apresentada a abordagem de classificação PCC+UMAP e PCC+PCA com extração de características utilizando CNNs VGG16+VGG19, ambas aplicadas ao mesmo conjunto de imagens para a classificação entre caminho limpo ou caminho com obstáculos.

O método baseado em PCC+UMAP demonstrou ser uma abordagem promissora, pois, com uma taxa de acurácia média de 89,26%, foi melhor que qualquer abordagem utilizando o algoritmo PCA para seleção de características. Este método também obteve uma acurácia muito próxima quando comparado a redes neurais convolucionais amplamente utilizadas, como a VGG e a Xception.

Como este estudo ainda está em fase inicial, tem-se um caminho a trilhar buscando novas abordagens para seleção de características, novas combinações de dados extraídos de CNNs diferentes, possíveis ajustes na construção do grafo utilizado pelo classificador PCC e, ainda, na realização de testes comparativos com outros classificadores.

Também um objetivo desta pesquisa é trabalhar para a ampliação do conjunto de imagens proposto por Breve e Fischer (2020), colaborando para uma melhor qualidade nos testes a serem realizados na continuidade deste estudo.

8 REFERÊNCIAS

Agarwal, A.; Philips, J.M.; Venkatasubramanian, S. *Universal Multi-Dimensional Scaling. In **16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining***, 2010.

AKKAPUSIT, P.; KO, I.-Y. Task-oriented approach to guide visually impaired people during smart device usage. In: IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, 2021. p. 28-35.

Alghamdi, S.; van Schyndel, R; Khalil, I. Accurate positioning using long range active RFID technology to assist visually impaired people. **Journal of Network and Computer Applications**, v. 41, p. 135–147, January 2013.

Alzubaidi, L; Fadhel, M; Al-Shamma, O; Zhang, J; Santamaría, J; Duan, Y; Oleiwi, S. *Towards a Better Understanding of Transfer Learning for Medical Imaging: A Case Study. **Applied Sciences**,* vol. 10, nº 4523, 2020.

Alzubaidi, L.; Zhang, J.; Humaidi, A.J; *et al*. *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. **Journal of Big Data***, vol. 8, nº 53, 2021.

ANDERSON, D. I.; CAMPOS, J. J.; WITHERINGTON, D. C.; DAHL, A., RIVERA, M.; HE, M.; UCHIYAMA, I.; BARBU-ROTH, M. *The role of locomotion in psychological development*. ***Frontiers in Psychology**,* vol. 4, 2013.

Ansari, A.; Gautam, A.; Agarwal, A.; Kumar, A.; Goel, R. *Smart Cane 2.0 Walking Stick for Visually Impaired. **International Research Journal of Engineering and Technology (IRJET),** vol. 7, nº 5, 2020.*

Aspinall, P.A.; Borooah, S.; Al Alouch, C.; Roe, J.; Laude, A.; Gupta, R.; Gupta, M.; Montarzino, A.; Dhillon, B. *Gaze and pupil changes during navigation in age-related macular degeneration. **British Journal of Ophthalmology**, nº 98(vol. 10), p. 1393–1397, 2014.*

Badrinarayanan, V; Kendall, A; Cipolla, R. SegNet: *A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, **in IEEE Transactions on Pattern Analysis and Machine Intelligence**,* vol. 39, no. 12, p. 2481-2495, 2017.

Baldi, P. *Autoencoders, Unsupervised Learning, and Deep Architectures. **Proceedings of ICML Workshop on Unsupervised and Transfer Learning,*** vol. 27, p. 37-49, PMRL, 2012.

Bay, H; Tuytelaars, T; Van Gool, L. *SURF: Speeded Up Robust Features.* Lecture Notes in Computer Science, Computer Vision – ECCV, Vol. 3951, cap. 32, p. 404–417, 2006.

Becht, E; McInnes, L; Healy, J; Dutertre, C; Kwok, I W H; Ng, L. G; Ginhoux, F; Newell, E.W. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 2018.

Belhumeur, P; Hespanha, J; Kriegman, D. *Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection.* ***IEEE Trans. Pattern Anal. Mach. Intell***, vol. 19, p 711-720, 1997.

Berriel, R. F; Lopes, A. T; de Souza, A. F; Oliveira-Santos, T. *Deep Learning-Based Large-Scale Automatic Satellite Crosswalk Classification.* ***IEEE Geoscience and Remote Sensing Letters***, vol 14, n. 9, p. 1513-1517, 2017a.

Berriel, R. F; Rossi, F. S; de Souza, A. F; Oliveira-Santos, T. *Automatic Large-Scale Data Acquisition via Crowdsourcing for Crosswalk Classification: A Deep Learning Approach.* ***Computers & Graphics***, vol 68, p. 32-42, 2017b.

Bicket, A., Mihailovic, A., Jian-Yu, E., Nguyen, A., Mukherjee, M., Friedman, D., Ramulu, P. *Gait in Elderly Glaucoma: Impact of Lighting Conditions, Changes in Lighting, and Fear of Falling.* ***Translational Vision Science & Technology****.* Vol. 9, nº 13. TVST, 2020.

Bhowmick, A; Hazarika, S. *An insight into assistive technology for the visually impaired and blind people: state-of-the-art and future trends.* ***Journal on Multimodal User Interfaces.*** *Vol. 11, p. 1-24, 2017.*

Bishop, C. **Pattern Recognition and Machine Learning.** New York, NY, USA: Springer, 2006

Breve, F.; Zhao, L.; Quiles, M.; Pedrycz, W.; Liu, J. *Particle competition and cooperation in networks for semi-supervised learning.* ***IEEE Transactions on Knowledge and Data Engineering****,* vol. 24, nº 9, p. 1686-1698, set. 2012.

Breve, F.; Zhao, L. *Particle competition and cooperation in networks for semi-supervised learning with concept drift. In* ***The 2012 International Joint Conference on Neural Networks (IJCNN)****,* jun. 2012, p.1-6.

Breve, F.; Zhao, L. *Fuzzy Community structure detection by particle competition and cooperation.* ***Soft Computing***, vol 17, nº4, p. 659-673, abr. 2013.

Breve, F. *Active semi-supervised learning using particle competition and cooperation in networks. In The* ***2013 International Joint Conference on Neural Networks (IJCNN),*** ago. 2013, p.1-6.

Breve, F.A.; Zhao, L.; Quiles, M.G. *Particle competition and cooperation for semi-supervised learning with label noise.* ***Neurocomputing****,* vol. 160, p. 63-72, 2015a.

Breve, F.; Quiles, M.G.; Zhao, L*. Interactive image segmentation using particle competition and cooperation. In* ***2015 International Joint Conference on Neural Networks (IJCNN),*** jul. 2015b, p. 1-8.

Breve, F.; Fischer, C.N. *Visually Impaired Aid using Convolutional Neural Networks, Transfer Learning, and Particle Competition and Cooperation. In* ***International Joint Conference on Neural Networks (IJCNN),*** mar. 2020.

Buchs, G; Simon, N; Maidenbaum, S; Amedi, A. *Waist-up protection for blind individuals using the EyeCane as a primary and secondary mobility aid*. ***Restorative Neurology and Neuroscience***, vol. 35(nº 2), p. 225–235, 2017.

Cai, Z.; Fan, Q.; Feris, R.S.; Vasconcelos, N. *A unified multi-scale deep convolutional neural network for fast object detection. In* ***European Conference on Computer Vision****.* Springer, 2016, p. 354-370.

Cao, X; Zhong, W; Yan, P; Liu, W. *Transfer learning for pedestrian detection.* ***Neurocomputing,*** vol.100, p. 51–57, 2013.

Cheng, B; Xiao, B; Wang, J; Shi, H; Zhang, L. *HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation*. ***IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, p. 5385-5394, 2020.

Cheraghi, S. A; Namboodiri, V; Walker, L. *GuideBeacon: Beacon-based indoor wayfinding for the blind, visually impaired, and disoriented.* ***IEEE International Conference on Pervasive Computing and Communications (PerCom)***, vol. 2017, p. 121–130, Kona, Big Island, HI, USA, 2017.

Choi, S; Kim, T; Yu, W. *Performance evaluation of ransac family.* ***in British Machine Vision Conference, BMVC***, p. 1-12, 2009.

Chollet, F. *Xception: Deep Learning with Depthwise Separable Convolutions*. ***IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, p. 1800-1807, 2017.

Cook, D; Feuz, K; Krishnan, N. *Transfer Learning for Activity Recognition: A Survey.* ***Knowledge and information systems,*** vol. 36, p. 537-556, 2013.

Cortes, C; Vapnik, V. *Support-vector networks.* ***Machine Learning****,* vol. 20, p. 273–297, 1995.

Csurka, G; Dance, C; Fan, L; Willamowski, J; Bray, C. *Visual categorization with bags of keypoints.* ***Work Stat Learn Comput Vision, ECCV,*** Vol. 1, 2004.

Dalal, N; Triggs, B. *Histograms of oriented gradients for human detection.* ***IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),*** vol. 1, pags 886-893, IEEE, 2005.

Dang, Q; Chee, Y; Pham, D; Suh, Y. *A Virtual Blind Cane Using a Line Laser-Based Vision System and an Inertial Measurement Unit.* ***Sensors****,* nº 16 (vol 1), p. 95–, 2016.

Everding, L; Walger, L; Ghaderi, V.S.; Conradt, J. *A mobility device for the blind with improved vertical resolution using dynamic vision sensors.* ***IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom),*** *p. 1-5, IEEE, 2016.*

Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. *The pascal visual object classes (voc) challenge.* ***International Journal of Computer Vision,*** vol. 88, nº 2, p. 303-338, jun. 2010.

Fan, M; Bao, J; Tang, H. *A Guide Cane System for Assisting the Blind in Travelling in Outdoor Environments*. ***Applied Mechanics and Materials***, vol. 631-632, p. 568-571, 2014.

Fernandes, L. A. F; Oliveira, M. M. *Real-time line detection through an improved Hough transform voting scheme.* ***Pattern Recognit****.*, vol. 41, no. 1, p. 299–314, 2008.

Fusco, G; Coughlan, J.M. *Indoor Localization for Visually Impaired Travelers Using Computer Vision on a Smartphone.* ***Proceedings of the 17th International Web for All Conference,*** *Association for Computing Machinery, New York, USA, 2020.*

Geruschat, D.R., Hassan, S., Turano, K.A., Quigley, H., Congdon, N. *Gaze Behavior of the Visually Impaired During Street Crossing.* ***Optometry and vision science: official publication of the American Academy of Optometry,*** *vol. 83, p. 550-558, 2006.*

Gopalakrishnam, K.; Khaitan, S.K.; Choudhary, A.; Agrawal, A. *Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection.* ***Construction and Building Materials***, vol. 157, p. 322-330, 2017.

Gupta, S.;Sharma, I.; Tiwari, A. Chitranshi, G. Advanced Guide Cane for the Visually Impaired People. **International Conference on Next Generation Computing Technologies**, 2015, p. 452-455.

Hakim, H; Fadhil, *A. Survey: Convolution Neural networks in Object Detection.* ***Journal of Physics: Conference Series,*** vol. 1804, pag. 012095, 2021.

He K; Zhang X; Ren S; Sun J. *Deep residual learning for image recognition. In:* ***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, p. 770–8, 2016.

Hinton, G; Roweis , S.T. *Stochastic neighbor embedding.* ***Neural Information Processing Systems – NIPS***, vol. 15. 2002.

Hirschmüller, H. *Stereo processing by semiglobal matching and mutual information.* ***IEEE Trans. Pattern Anal. Mach. Intell***. Vol. 30, nº 2, pags 328–341, 2008.

Hoang, V.N.; Nguyen, T.H.; Le, T.L.; Tran, T.H.; Vuong, T.P.; Vuillerme, N. *Obstacle detection and warning system for visually impaired people on electrode matrix and mobile Kinect.* ***Vietnam Journal of Computer Science***, vol. 4, nº 2, p. 71-83, 2017.

Hossin, M; Sulaiman, M.N. *A Review on Evaluation Metrics for Data Classification Evaluations.* ***International Journal of Data Mining & Knowledge Management Process***, vol. 5, nº 2, p. 01-11, 2015.

Hu, J; Shen, L; Sun, G. *Squeeze-and-Excitation Networks.* ***IEEE/CVF Conference on Computer Vision and Pattern Recognition***, p. 7132-7141, 2018.

Huang, G; Liu, Z; Van Der Maaten, L; Weinberger, K.Q. *Densely connected convolutional networks. In:* ***Proceedings of the IEEE conference on computer vision and pattern recognition***, p. 4700-08, 2017.

Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. ***Mobilinets: Efficient convolutional neural networks for mobile vision applications***, 2017.

Hu, M.; Chen, Y.; Zhai, G.; Gao, Z.; Fan, L. *An Overview of Assistive Devices for Blind and Visually Impaired People*. ***International Journal of Robotics and Automation,*** vol. 34, 2019.

IBGE, BRASIL. **Censo Demográfico 2010**. Instituto Brasileiro de Geografia e Estatística. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf, Acesso: 05/11/2020. Brasilia-DF, 2010.

Islam, M.M.; Sadi, M.S. *Path hole detection to assist the visually impaired people in navigation. In* ***2018 4th International Conference on Electrical Engineering and Information Communication Technology (iCEEiCT)***, Set. 2018, p. 268-273.

Jegou, H; Douze, M; Schmid C. *Product quantization for nearest neighbor search.* ***In PAMI***, vol. 33, nº 1, p. 117–128, 2011.

Jiang, B.; Yang, J.; Lv, Z.; Song, H. *Wearable vision assistance system based on binocular sensors for visually impaired users.* ***IEEE Internet of Things Journal***, vol. 6, nº 2, p. 1375-1383, 2019.

Jolliffe, I. Springer-Verlag. ***Principal Components Analysis***. *Springer Series in Statistics*. Springer, 2002.

Kassim, A.M; Yasuno, T; Suzuki, H; Jaafar, H. I; Mohd Aras, M. *Indoor Navigation System based on Passive RFID Transponder with Digital Compass for Visually Impaired People.* ***International Journal of Advanced Computer Science and Applications,*** *vol. 7, 2016.*

Kalman, R. E. *A new approach to linear filtering and prediction problems.* ***Transactions of the ASME–Journal of Basic Engineering****,* vol. 82, no. Series D, p. 35–45, 1960.

Krishnan, A; Deepakraj, G; Nishanth, N; Anandkumar, K. M. *Autonomous walking stick for the blind using echolocation and image processing*. ***2nd International Conference on Contemporary Computing and Informatics (IC3I),*** p. 13–16, IEEE 2016.

Krizhevsky, A.; Sutskever, I.; Hinton, G.E. *Imagenet classification with deep convolutional neural networks. In* ***Advances in Neural Information Processing Systems 25****.* Curran Associates Inc., 2012, p. 1097-1105.

Kobak, D.; Linderman, G. C. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature Biotechnology*, vol. 39(nº 2), p. 156–157, 2021.

Kumar, K.; Champaty, B.; Uvanesh, K.; Chachan, R. Development of an ultrasonic cane as a navigation aid for the blind people. International Conference on Control, Instrumentation, Communication and Computational Technologies, 2014, p. 475-479.

Kumar, P.M; Gandhi, U; Varatharajan, R; Manogaran, G; Jidhesh, R; Vadivel, T. *Intelligent face recognition and navigation system using neural learning for smart security in Internet of Things.* **Cluster Comput**, vol. 22, p. 7733–7744, Springer, 2017.

Kumar, R.; Meher, S. *A novel method for visually impaired using object recognition. In* ***2015 International Conference on Communications and Signal Processing (ICCSP)***, p. 772-776, 2015.

Lakde, C.K.; Prasad, P.S. *Review paper on navigation system for visually impaired people*. ***International Journal of Advanced Research in Computer and Communication Engineering***, vol. 4, nº 1, 2015.

LeCun, Y; Boser, B; Denker, J.S; Henderson, D; Howard, R.E; Hubbard, W; Jackel, L.D. *Backpropagation Applied to Handwritten Zip Code Recognition.* ***In Neural Computation***, vol. 1, no. 4, p. 541-551, 1989.

LeCun, Y; Jackel, L.D; Bottou, L; Cortes, C; Denker, J.S; Drucker, H; Guyon, I; Muller, U.A; Sackinger, E; Simard, P. *et al*. *Learning algorithms for classification: a comparison on handwritten digit recognition.* ***Neural Networks Stat Mech Perspect***., p. 261-276, 1995.

LeCun, Y.; Bengio, Y.; Hinton, G. *Deep learning.* ***Nature***, vol. 521, nº 7553, p. 436, 2015.

Li, Z; Yang, W; Peng, S; Liu, F. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects, arXiv preprint https://arxiv.org/abs/2004.02806, 2020.

Lin, M.; Chen, Q.; Yan, S. Network-in-Network*. arXiv:1312.4400, 2014.*

Lin, B.S.; Lee, C.C.; Chiang, P.Y. *Simple smartphone-based guiding system for visually impaired people.* ***Sensors****,* vol. 17, nº 6, p. 1371, 2017.

Lowe, D. G. *Object recognition from local scale-invariant features*. ***Proceedings of the Seventh IEEE International Conference on Computer Vision,*** vol. 2, p. 1150-1157, 1999.

Majeed, A.; Baadel, S. *Facial Recognition Cane for the Visually Impaired.* ***Global Security, Safety and Sustainability – The Security Challenges of the Connected World***, p. 394-405. Springer International Publishing, 2016.

Malek, S; Melgani, F; Mekhalfi, M; Bazi, Y. *Real-Time Indoor Scene Description for the Visually Impaired Using Autoencoder Fusion Strategies with Visible Cameras.* ***Sensors***, vol. 17, nº 11, p. 2641–, 2017.

Manduchi, R; Coughlan, J. *(Computer) vision without sight.* ***Communications of the ACM***, vol. 55, nº 1, p. 96–104, 2012.

McInnes, L.; Healy, J.; Melville, J. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.* Disponível em: arxiv.org/abs/1802.03426v3, 2020.

Monteiro, J.; Aires, J.P.; Granada, R.; Barros, R.C.; Meneguzzi, F.; *Virtual guide dog: An application to support visually impaired people through deep convolutional neural networks. In* ***2017 International Joint Conference on Neural Networks (IJCNN)***, maio 2017, p. 2267-2274.

Nakajima, M; Haruyama, S. *New indoor navigation system for visually impaired people using visible light communication.* ***EURASIP Journal on Wireless Communications and Networking***, vol. 2013, nº 1, 2013.

Neha, F. F; Shakib, K. H. *Development of a Smartphone-based Real Time Outdoor Navigational System for Visually Impaired People.* ***2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)***, p. 305-310, 2021.

Neto, L. B; Grijalva, F; Maike, V. R. M. L; Martini, L. C; Florencio, D; Baranauskas, M. C. C.; Rocha, A.; Goldenstein, S. *A Kinect-Based Wearable Face Recognition System to Aid Visually Impaired Users.* ***IEEE Transactions on Human-Machine Systems***, vol. 47, nº 1, p. 1–13, 2016.

Niitsu, Y; Taniguchi, T; Kawashima, K. Detection and notification of dangerous obstacles and places for visually impaired persons using a smart cane. ***7th International Conference on Mobile Computing and Ubiquitous Networking***, p. 68-69, ICMU, 2014.

Niskanen, M.; Silvén, *O. Comparison of Dimensionality Reduction Methods for Wood Surface Inspection. In* ***6th International Conference on Quality Control by Artificial Vison****,* vol 5132, p. 178-188, 2003.

Niu, L; Qian, C; Rizzo, J; Hudson, T; Li, Z; Enright, S; Sperling, E; Conti, K; Wong, E; Fang, Y. *A Wearable Assistive Technology for the Visually Impaired with Door Knob Detection and Real-Time Feedback for Hand-to-Handle Manipulation.* ***IEEE International Conference on Computer Vision Workshop (ICCVW) – Venice***, p. 1500–1508, IEEE, 2017.

Ojala, T; Pietikainen, M; Harwood, D. *Performance evaluation of texture measures with classification based on Kullback discrimination of distributions.* ***Proceedings of 12th International Conference on Pattern Recognition****,* vol.1, p. 582-585, 1994.

ONU. ***World Report on vision.*** Disponível em: https://www.who.int/publications/i/item/world-report-on-vision. Acessado em 05/11/2020. Organização das Nações Unidas, Genebra, Suíça, 2019.

Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. *Learning and transferring mid-level image representative using convolutional neural networks. In* ***The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)****,* June, 2014.

Ottaiano, J.A.A.; Ávila, M.P.; Umbelino, C.C.; Taleb, A.C. **As Condições de Saúde Ocular no Brasil 2019**. Edição 1. Conselho Brasileiro de Oftalmologia (CBO), 2019.

Pang, Y; Sun, M.; Jiang, X.; Li, X. *Convolution in Convolution for Network in Network. In* ***IEEE Transactions on Neural Networks and Learning Systems***, vol. 29, no. 5, p. 1587-1597, 2018.

Parikh, N; Shah, I; Vahora, S. Android *Smartphone* Based Visual Object Recognition for Visually Impaired Using Deep learning. 2018 International Conference on Communication and Signal Processing (ICCSP), pag. 420-425, 2018.

Pasqualotto, A.; Proulx, M. *The role of visual experience for the neural basis of spatial cognition.* ***Neuroscience and biobehavioral reviews.*** Vol. 36, p. 1179-1187, 2012.

Pasqualotto, A.; Lam, J.; Proulx, M. *Congenital blindness improves semantic and episodic memory.* ***Behavioural brain research,*** *v*ol. 244, 2013.

Pisa, S; Pittella, E; Piuzzi, E. *Serial Patch Array Antenna for an FMCW Radar Housed in a White Cane*. ***International Journal of Antennas and Propagation***, p. 1–10, 2016.

Poggi, M.; Nanni, L.; Mattoccia, S. *Crosswalk recognition through point-cloud processing and deep-learning suited to a wearable mobility aid for the visually impaired. In* ***International Conference on Image Analysis and Processing****,* p. 282-289. Springer, 2015.

Poggi, M.; Mattoccia, S. *A wearable mobility aid for the visually impaired based on embedded 3d vision and deep learning. In* ***2016 IEEE Symposium on Computers and Communication (ISCC)***, p. 208-213. IEEE, 2016.

Redmon, J; Divvala, S; Girshick, R; Farhadi, A. *You Only Look Once: Unified, Real-Time Object Detection.* ***In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)****,* p. 779-788, 2016.

Redmon, J; Farhadi, A. Yolo9000: Better, faster, stronger. arXiv preprint arXiv:1612.08242, 2016

Ren, S; He, K; Girshick, R; Sun, J. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.* ***IEEE Transactions on Pattern Analysis and Machine Intelligence****,* vol. 39, 2015.

Rizzo, J.R.; Pan, Y.; Hudson, T.; Wong, E.K.; Fang, Y. *Sensor fusion ecologically valid obstacle identification: Building a comprehensive assistive technology platform for the visually impaired. In* ***2017 7th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*****,** p. 1-5. IEEE, 2017.

Romadhon, A.S.; Husein, A.K. *Smart Stick for the Blind Using Arduino*. ***Journal of Physics: Conference Series***, vol. 1569, 2020.

Roweis, S.T.; Saul, L.K. *Nonlinear Dimensionality Reduction by Locally Linear Embedding*. ***Science***, vol. 290, p. 2323-2326, set. 2000.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M; Berg, A.C.; Fei-Fei, L. *ImageNet Large Scale Visual Recognition Challenge.* ***International Journal of Computer Vision (IJCV),*** vol. 115. nº 3, p. 211-252, 2015.

Sadi, M. S; Mahmud, S; Kamal, Md. M; Bayazid, A I. *Automated walk-in assistant for the blinds.* ***International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)***, p. 1-4, IEEE, Dhaka, Bangladesh, 2014.

Saffoury, R.; Blank, P.; Sessner, J.; Groh, B.H.; Martindale, C.F.; Dorschky, E.; Franke, J.; Eskofier, B.M. *Blind path obstacle detector using smartphone camera and line laser emitter, in* ***2016 1st International Conference on Technology and Innovation in Sports, Health and Wellbeing*** (TISHW), p. 1-7. IEEE, 2016.

Saleh, K.; Zeineldin, R.A.; Hossny, M.; Nhavandi, S.; El-Fishawy, N.A. *Navigational path detection for the visually impaired using fully convolutional networks. In* ***2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)***, out. 2017, p. 1399-1404.

Sandler, M; Howard, A; Zhu, M; Zhmoginov, A; Chen; L. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. arXiv: 1801.04381, 2019.

Saxena, A.; Gupta, A.; Mukerjee, A. *Non-linear Dimensionality Reduction by Locally Linear Isomaps. In International Conference on Neural Information Processing (ICONIP),* ***Lecture Notes in Computer Science***, vol. 3316, p. 1038-1043, Springer, 2004.

Schapire, R.E. *The boosting approach to machine learning: An overview.* ***Nonlinear estimation and classification.*** p. 149–171, Springer, New York, 2003.

Scherlen, A.C; Dumas, J.C; Guedj, B; Vignot, A. "RecognizeCane" : *The new concept of a cane which recognizes the most common objects and safety clues.* ***29h Annual International Conference of the IEEE Engineering in Medicine and Biology Society***, p. 6356–6359, IEEE, Lyon – France, 2007.

Schmidhuber, J. *Deep learning in neural networks: An overview.* ***Neural Networks****,* vol. 61, p. 85-117, 2015.

Shin, H.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. *Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning.* ***IEEE Transactions on Medical Imaging*****,** vol. 35, nº 5, p. 1285-1298, maio 2016.

Simonyan, K.; Zisserman, A. *Very deep convolutional networks for large-scale image recognition.* ***Computational and Biological Learning Society***, 2015, p. 1-14.

Socher, R; Huval, B; Bhat, B; Manning, C; Ng, A.Y. *Convolutional-Recursive Deep Learning for 3D Object Classification.* ***Proceedings of the 25th International Conference on Neural Information Processing Systems,*** vol. 1, p. 656-664, 2012.

Stoll, C; Palluel-Germain, R; Fristot, V; Pellerin, D; Alleysson, D; Graff, C. *Navigating from a Depth Image Converted into Sound.* ***Applied Bionics and Biomechanics,*** vol. 2015, nº 4, p. 1–9, 2015.

Srivastava, R; Greff, K; Schmidhuber, J. Highway Networks. Disponível em: arXiv:1505.00387v2, 2015.

Sun, K; Cheng, T, Xiao, B; Wang, J. *Deep High-Resolution Representation Learning for Visual Recognition.* ***IEEE Trans Pattern Anal Mach Intell,*** vol.43, nº 10, p. 3349-3364, 2020.

Szegedy, C; Wei, L; Yangging, J; Sermanet, P; Reed, S; Anguelov, D; Erhan, D; Vanhoucke, V; Rabinovich, A. *Going deeper with convolutions.* ***IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, p. 1-9, 2015.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. *Rethinking the inception architecture for computer vision. In The* ***IEEE Conference on Computer Vision and Pattern Recognition (CPVR),*** Jun. 2016, p. 2818-2826, 2016a.

Szegedy, C; Ioffe, S; Vanhoucke, V; Alemi, A. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.* ***AAAI Conference on Artificial Intelligence,*** *2016b.*

Takizawa, H; Yamaguchi, S; Aoyagi, M; Ezaki, N; Mizuno, S. *Kinect cane: an assistive system for the visually impaired based on the concept of object recognition aid.* ***Personal and Ubiquitous Computing***, vol. 19(5-6), p. 955–965, 2015.

Tan C.; Sun F.; Kong T.; Zhang W.; Yang C.; Liu C. *A Survey on Deep Transfer Learning*. In: ***Artificial Neural Networks and Machine Learning – ICANN 2018, Lecture Notes in Computer Science***, vol 11141. Springer, Cham., 2018.

Tenembaum, J.B.; Silva, V.; Langford, J.C. *A Global Geometric Framework for Nonlinear Dimensionality Reducton.* ***Science****,* vol. 290, pag. 2319-2323. Dez. 2000.

Tanveer, E.M.S; Hashem, M.M.A; Hossain, K. *Android Assistant EyeMate for Blind and Blind Tracker. In Proc.* ***18º International Conference on Computer and Information Technology (ICCIT),*** *p. 266,271, Dhaka, 2015.*

Tapu, R.; Mocanu, B.; Bursuc, A.; Zaharia, T. *A smartphone-based obstacle detection and classification system for assisting visually impaired people. In* ***The IEEE International Conference on Computer Vision (ICCV) Workshops****,* jun. 2013.

Tapu, R; Mocanu, B; Zaharia, T*. A computer vision-based perception system for visually impaired.* ***Multimed Tools Appl.*** Vol. 76, p. 11771–11807. Springer, 2017a

Tapu, R.; Mocanu, B.; Zaharia, T. *Deep-see: Joint object detection, tracking and recognition with application to visually impaired navigational assistance*. ***Sensors***, vol. 17, nº 11, p. 2473, 2017b.

Tepelea, L; Gavrilut, L; Gacsadi, *A. Smartphone application to assist visually impaired people.* ***14th International Conference on Engineering of Modern Electric Systems (EMES),*** vol 2017, p. 228–231, Oradea, Romania, 2017.

Terven, J.R; Salas, J; Raducanu, B. *New Opportunities for Computer Vision-Based Assistive Technology Systems for the Visually Impaired.* ***Computer***, vol. 47, nº 4, p. 52-58, 2014.

Turano, K.A., Geruschat, D.R., Baker, F.H., Stahl, J.W., Shapiro, M.D. *Direction of gaze while walking a simple route: Persons with normal vision and persons with retinitis pigmentosa.* ***American Journal of Optometry and Physiological Optics.*** *Vol. 78, nº 9, p. 667-675. Lippincott Willians and Wilikins, 2001.*

Van der Maaten, L.; Hinton, G. *Visualizing Data using t-SNE.* ***Journal of Machine Learning Research***, vol. 9, p. 2579-2605, 2008.

Vera, P.; Zenteno, D.; Salas, J. A *smartphone*-based virtual white cane**. Pattern Analysis and Applications**, vol. 17, p. 623–632, 2014.

Viola, P; Jones, M. *Rapid Object Detection using a Boosted Cascade of Simple Features.* ***IEEE Conf Comput Vis Pattern Recognit,*** vol. 1, p. I-511, 2001.

Wang, S; Yang, X; Tian, Y. *Detecting signage and doors for blind navigation and wayfinding.* **Network modeling and analysis in health informatics and bioinformatics**, vol. 2, nº 2, p. 81–93, 2013.

Wang, F; Jiang, M; Qian, C; Yang, S; Li, C; Zhang, H; Wang, X; Tang, X. *Residual Attention Network for Image Classification.* ***IEEE Conference on Computer Vision and Pattern Recognition (CVPR),*** p. 6450-6458, 2017.

Weiss, K.; Khoshgoftaar, T.M.; Wang, D. *A survey of transfer learning*. ***Journal of Big Data***, vol. 3, nº 9, 2016.

WHO. ***International Statistical Classification of Diseases and Related Health Problems 10th revision Current version Version for 2003 Chapter VII H54 Blindness and low vision***. Disponível em: http://www.who.int/classifications/icd/en/. World Health Organization, 2003.

Wu, L; Hoi, S. C. H; Yu, N. *Semantics-Preserving Bag-of-Words Models and Applications.* ***In IEEE Transactions on Image Processing***, vol. 19, no. 7, p. 1908-1920, 2010.

Yang, K; Bergasa, L. M; Romera, E; Cheng, R; Chen, T; Wang, K. *Unifying terrain awareness through real-time semantic segmentation.* ***Intelligent Vehicles Symposium (IV)***, p. 1033-1038, IEEE, 2018.

Yasuno, T; Kassim, A.M; Suzuki, H; Shahrieel Mohd Aras, M; Izzuan J. H; Azni Jafar, F; Subramonian, S. *Conceptual design and implementation of electronic spectacle-based obstacle detection for visually impaired persons.* ***Journal of Advanced Mechanical Design, Systems, and Manufacturing***, vol. 10, nº 7, p. 1-12, 2016.

Ye, C.; Hong, S.; Qian, X.; Wu, W. *Co-Robotic Cane: A New Robotic Navigation Aid for the Visually Impaired.* ***IEEE Systems, Man, and Cybernetics Magazine****,* vol.2, p. 33–42, 2016.

Yong, S; Chen, Y; Wan, C. *Seismic image recognition tool via artificial neural network.* ***CINTI 2013 - 14th IEEE International Symposium on Computational Intelligence and Informatics, Proceedings,*** p.399-404, 2013.

Zagoruyko, S; Komodakis, N. Wide residual networks. **arXiv preprint**, arXiv: 1605.07146, 2016.

Zeiler, M.D; Fergus R. *Visualizing and understanding convolutional networks.* ***In: European conference on computer vision*****,** p. 818-833, Springer, 2014.