



Instituto de Geociências e Ciências Exatas (IGCE)
Pós-Graduação em Ciência da Computação

Auxílio a Deficientes Visuais utilizando Redes Neurais Convolucionais e Competição e Cooperação de Partículas

Jefferson Antonio Ribeiro Passerini

Orientador: Prof. Dr. Fabricio Aparecido Breve

Disciplina: Estudos Especiais II

Agenda

- Introdução
- Objetivos e Motivação
- Fundamentação Teórica
- Trabalhos Relacionados
- Metodologia
- Resultados
- Cronograma de Execução
- Considerações Finais

Introdução

- 2,2 bilhões de pessoas com algum tipo de deficiência visual no mundo (ONU, 2019);
- 35 milhões de pessoas com deficiência visual no Brasil, onde aproximadamente 506 mil eram totalmente cegas (IBGE, Censo 2010);
- Evolução das tecnologias assistivas; Visão Computacional;
- Smartphone como recurso assistivo.

Objetivos

Objetivo Principal:

- Detecção de obstáculos a partir de imagens capturadas de dispositivos móveis sem a necessidade de servidores remotos para processamento.

Objetivos

Objetivo Estudos Especiais:

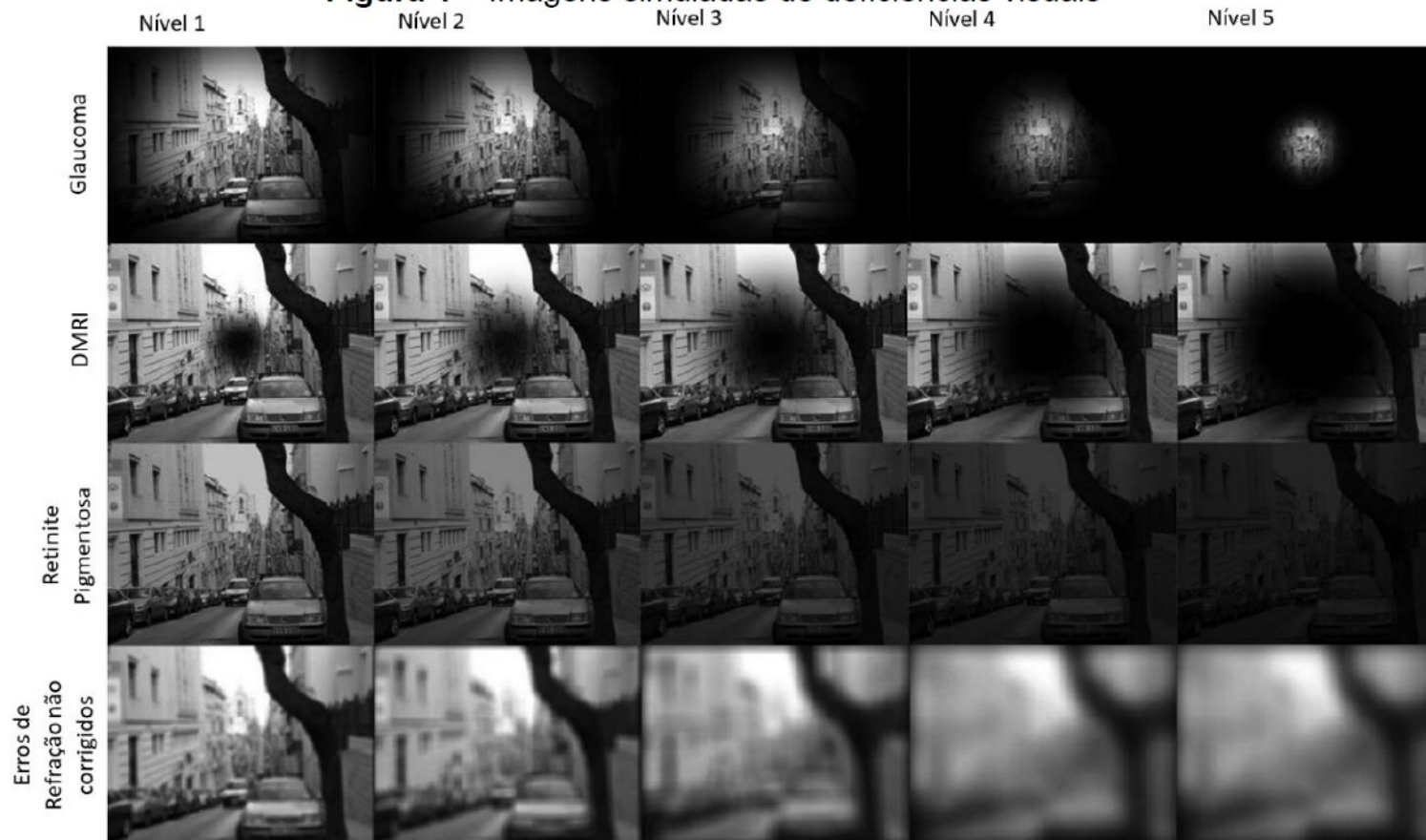
- Estudo bibliográfico dos conceitos a serem utilizados na pesquisa;
- Levantamento de trabalhos correlatos;
- Estabelecer um metodologia pra continuidade da pesquisa.

Fundamentação Teórica: Deficiência Visual

- Classificação da deficiência visual: 1-Def. visual leve; 2-Def. visual moderada; 3-Cegueira; 4-Cegueira severa; 5-Cegueira muito severa; 6-Cegueira total.
- Mudanças de comportamento como na forma de andar (GERUSCHAT *et al*, 2006; BICKET *et al*, 2020; TURANO *et al*, 2021);
- Diferenças no processo de aprendizagem (adaptação) do cego congênito e do cego tardio (PASQUALOTO; PROULX, 2012; PASQUALOTO; LAM; PROULX, 2013):
 - Cego tardio possui desenvolvimento do cérebro igual ou próximo a de uma pessoa com visão;
 - Cego congênito possuem habilidades superiores de memória auditiva.

Fundamentação Teórica: Deficiência Visual

Figura 1 – Imagens simuladas de deficiências visuais



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

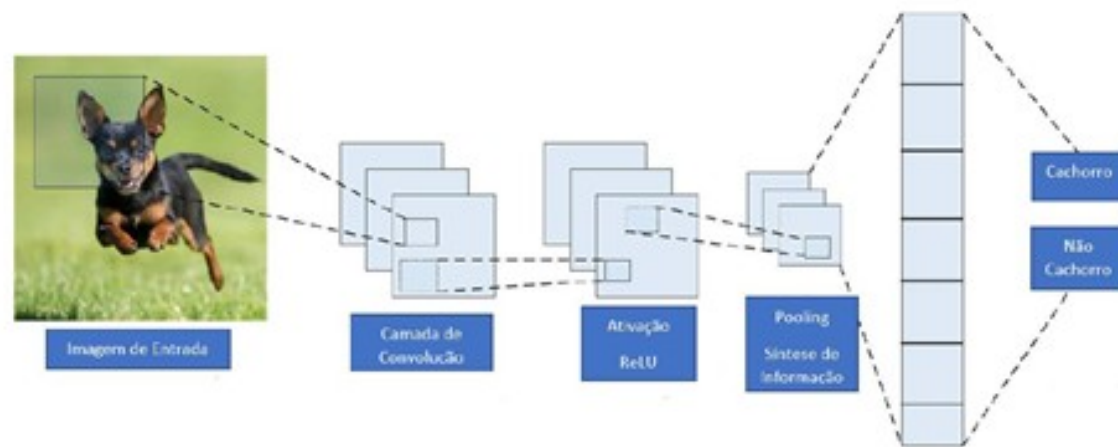
DMRI □ Degeneração macular relacionada a idade

Fundamentação Teórica: CNN e *Deep Learning*

- *Deep Learning* é utilizado em diferentes aplicações, sendo derivado das redes neurais convencionais, mas supera consideravelmente seu desempenho (LECUN; BENGIO; HINTON, 2015);
- É capaz de analisar funções complexas e não lineares, gerar representações de recursos distribuídos e hierárquicos e permite o uso eficaz de dados rotulados e não rotulados (PANG et al., 2018);
- Os modelos de aprendizado profundo tem uma representação de dos multicamadas simulando o que ocorre nas regiões sensoriais do cérebro humano. Sendo as redes neurais convolucionais (CNN) o modelo de rede que trouxe maior visibilidade para esse modelo de aprendizado (ALZUBAIDI et al., 2021).

Fundamentação Teórica: CNN e *Deep Learning*

- As CNNs são redes neurais do tipo *feedforward* e extraem características dos dados através de convoluções (LI et al., 2020);
- São compostas por camadas de convolução; ativação (ReLU) e de síntese de informações (*Pooling*) para a extração das características e uma camada totalmente conectada (*fully connected*) para a classificação.



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

Fundamentação Teórica: CNN e *Deep Learning*

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
Inception-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).

Fundamentação Teórica: *Transfer learning*

- A transferência de aprendizado parte da premissa que os dados rotulados são escassos, assim afirma que é possível transferir representações aprendidas para outras tarefas;
- A transferência de aprendizado se justifica pela necessidade de grandes bases de dados para treinamento das CNNs (WEISS; KHOSHGOFTAAR; WANG, 2016). Em domínios especiais a quantidade de dados disponíveis para treinamento geralmente são insuficientes (TAN *et al.*, 2018);
- Vários trabalhos demonstram da utilização de treinamento com tipos de dados diferentes da aplicação destino: Cook, Feuz e krishnan (2013); Cao *et al.* (2013), Oquab *et al.* (2014), Shin *et al.* (2016), Alzubaidi *et al.* (2020);

Fundamentação Teórica: *Seleção de Características*

- Análise de Componentes Principais (PCA) (JOLLIFFE, 2002);
- Isomap (TENENBAUM; SILVA; LANGFORD, 2000);
- *Locally Linear Embedding* (LLE) (ROWEIS; SAUL, 2000);
- K_{LL} Isomaps (SAXENA; GUPTA; MUKERJEE, 2004);
- *t-Distributed Stochastic Neighbor Embedding* (T-SNE) (MAATEN; HINTON, 2008);
- *Uniform Approximation and Projection* (UMAP) (McINNES; HEALY; MELVILLE, 2020);

Fundamentação Teórica: *Classificador*

- Competição e Cooperação de Partículas (PCC) (Breve *et al.*, 2002);
 - Aprendizado Semi-supervisionado;
 - Utiliza rede complexa para representação dos dados, onde os vértices estão interligados utilizando o modelo Knn, admitindo a Distância Euclidiana como métrica;
 - Cada vértice representa uma instância dos dados;
 - Deve ser fornecidos alguns rótulos para o modelo;
 - Partículas navegam pela rede complexa competindo e cooperando para dominar os vértices não rotulados
 - Navegação pode seguir: regra gulosa ou aleatória.

Trabalhos Correlatos

Tabela 3 - Trabalhos Analisados por área de domínio

Domínio	Trabalhos Pesquisados
Detecção de Obstáculos	Neha e Shakib (2021); Ansari <i>et al.</i> (2020); Romadhoni <i>et al.</i> (2020); Jiang <i>et al.</i> (2019); Parikh <i>et al.</i> (2018); Yang <i>et al.</i> (2018); Buchs <i>et al.</i> (2017); Rizzo <i>et al.</i> (2017); Hoang <i>et al.</i> (2017); Lin <i>et al.</i> (2017); Tepelea <i>et al.</i> (2017); Dang <i>et al.</i> (2016); Everding <i>et al.</i> (2016); Krishnan <i>et al.</i> (2016); Saffory <i>et al.</i> (2016); Pisa <i>et al.</i> (2016); Poggi e Mattoccia (2016); Yasuno <i>et al.</i> (2016); Ye <i>et al.</i> (2016); Gupta <i>et al.</i> (2015); Kumar e Meher (2015); Takizawa <i>et al.</i> (2015); Tanveer <i>et al.</i> (2015); Kumar <i>et al.</i> (2014); Niitsu <i>et al.</i> (2014); Sadi <i>et al.</i> (2014); Vera <i>et al.</i> (2014); Tapu <i>et al.</i> (2017a); Tapu <i>et al.</i> (2017b); Tapu <i>et al.</i> (2013); Scherlen <i>et al.</i> (2007).
Detecção de Buracos	Islam e Said (2018).
Identificação de Locais	Alghamdi <i>et al.</i> (2013).
Identificação de faixas de pedestres	Berriel <i>et al.</i> (2017a); Berriel <i>et al.</i> (2017b); Poggi <i>et al.</i> (2015).
Navegação Interna	Fusco e Coughlan (2020); Cheraghi <i>et al.</i> (2017); Kassim <i>et al.</i> (2016); Stoll <i>et al.</i> (2015); Nakajima e Haruyama (2013).
Navegação Externa	Krishnan <i>et al.</i> (2016); Gupta <i>et al.</i> (2015); Fan <i>et al.</i> (2014).
Identificação de Pessoas	Kumar <i>et al.</i> (2017); Majeed e Baadel (2016); Neto <i>et al.</i> (2016).
Identificação de Portas e Maçanetas	Niu <i>et al.</i> (2017); Wang <i>et al.</i> (2013).
Auxílio a utilização de interfaces de dispositivos inteligentes	Akkapusit; Ko (2021).
Descrição de ambientes	Malek <i>et al.</i> (2017).

Fonte: O autor (2021).

Trabalhos Correlatos

Tabela 4 - Trabalhos analisados - unidades de processamento

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

Fonte: O autor (2021).

Trabalhos Correlatos

- Sensores:
 - RGB □ 22 trabalhos
 - RGB-D □ 7 trabalhos
 - Stereo □ 4 trabalhos
 - IR □ 4 trabalhos
 - Ultrassônico □ 12 trabalhos
 - Laser □ 4 trabalhos
 - Rfid □ 4 trabalhos
 - GPS □ 6 trabalhos
 - IMU □ 6 trabalhos
 - Radar □ 1 trabalho

Trabalhos Correlatos

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 <i>et al.</i> , 2019)
Berriel <i>et al.</i> (2017a)	Identificação de faixas de pedestres	CNN - VGG (SIMONYAN; ZISSERMAN, 2015)
Berriel <i>et al.</i> (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), <i>Local Binary Patterns</i> (LBP) (OJALA; PIETIKAINEN; HARWOOD, 1994), <i>Histogram of Oriented Gradients</i> (HoG) (DALAL; TRIGGS, 2005), <i>Support Vector Machines</i> (SVM) (CORTES; VAPNIK, 1995)
Islam e Sadi (2018)	Detecção de buracos	CNN – AlexNet (KRIZHEVESKY; SUTSKEVER; HINTON, 2012)
Jiang <i>et al.</i> (2019)	Detecção de obstáculos	CNN – Resnet (HE <i>et al.</i> , 2016)
Krishnan <i>et al.</i> (2016)	Detecção de obstáculos e navegação	<i>Speeded Up Robust Features</i> - SURF (BAY; TUYTELAARS; VAN GOOL, 2006)
Kumar e Meher (2015)	Detecção e identificação de objetos	<i>Convolutional-Recursive deep learning</i> (SOCHER <i>et al.</i> , 2012)
Kumar <i>et al.</i> (2017)	Reconhecimento facial e navegação	Rede Neural Artificial – RNA (YONG; CHEN; WAN, 2013)

Lin <i>et al.</i> (2017)	Detecção de obstáculos	<i>Processo desconectado: Histogram of Oriented Gradients</i> (HoG) (DALAL; TRIGGS, 2005) e Haar Cascades (VIOLA; JONES, 2001) e <i>processo conectado: F-RCNN</i> (REN <i>et al.</i> , 2015) e YOLO (REDMON <i>et al.</i> , 2016)
Majeed e Baadel (2016)	Reconhecimento facial	Haar Cascades (VIOLA; JONES, 2001) e Fisherfaces (BELHUMEUR; HESPANHA; KRIEGMAN, 1997)
Malek <i>et al.</i> (2017)	Descrição de ambientes	Extração de características: <i>Local Binary Pattern - LBP</i> (OJALA; PIETIKAINEN; HARWOOD, 1994), <i>Histogram of Oriented Gradients</i> - HoG (DALAL; TRIGGS, 2005), <i>Bag of Words</i> - BoW (WU; HOI; YU, 2010); Aprendizado de características: Auto-Encoder neural network- AE (BALDI, 2012); Classificação: Regressão Logística
Neha e Shakib (2021)	Detecção de obstáculos	Transformada de Hough e <i>Region of Interest</i> – ROI (FERNANDES; OLIVEIRA, 2008)
Neto <i>et al.</i> (2016)	Reconhecimento facial	<i>Histogram of Oriented Gradients</i> (HoG) (DALAL; TRIGGS, 2005), <i>Principal Components Analysis</i> - PCA (JOLLIFFE, 2002); KNN (BISHOP, 2006)
Niu <i>et al.</i> (2017)	Identificação de portas e maçanetas	CNN - YOLOv2 (REDMON; FARHADI, 2016);
Parikh <i>et al.</i> (2018)	Detecção de obstáculos	CNN - InceptionV3 (SZEGEDY <i>et al.</i> , 2016a)
Poggi e Mattoccia (2016)	Detecção de obstáculos	Ransac framework (CHOI; KIM; YU, 2009), Filtro de Kalman (KALMAN, 1960), CNN – LeNet (LECUN <i>et al.</i> , 1989)
Poggi <i>et al.</i> (2015)	Identificação de faixas de pedestres	Algoritmo SGM (HIRSHMULLER, 2008), Ransac framework (CHOI; KIM; YU, 2009), Filtro de Kalman (KALMAN, 1960), CNN – LeNet (LECUN <i>et al.</i> , 1989)

Trabalhos Correlatos

Rizzo <i>et al.</i> (2017)	Detecção de obstáculos	CNN
Tapu <i>et al.</i> (2013)	Detecção e reconhecimento de objetos	<i>Scale-Invariant Feature Transform</i> (SIFT) (LOWE, 1999); <i>Speeded Up Robust Features</i> - SURF (BAY; TUYTELAARS; VAN GOOL, 2006); <i>Bag of Visual Words</i> - BoVW (CSURKA <i>et al.</i> , 2004); <i>Histogram of Oriented Gradients</i> -HoG (DALAL; TRIGGS, 2005)
Tapu <i>et al.</i> (2017a)	Detecção e reconhecimento de objetos	<i>Bag of Visual Words</i> - BoVW (CSURKA <i>et al.</i> , 2004); representação de imagem – VLAD (JEGOU; DOUZE; SCHMID, 2011); <i>Histogram of Oriented Gradients</i> -HoG (DALAL; TRIGGS, 2005); <i>Support Vector Machines</i> (SVM) (CORTES; VAPNIK, 1995)
Tapu <i>et al.</i> (2017b)	Detecção e reconhecimento de objetos	CNN - YOLO (REDMON <i>et al.</i> , 2016)
Yang <i>et al.</i> (2018)	Detecção de obstáculos	CNN SegNet (BADRINARAYANAN; KENDAL; CIPOLLA, 2017)
Ye <i>et al.</i> (2016)	Detecção de obstáculos	Gaussian Mixture Model – GMM (BISHOP, 2006)

Fonte: O autor (2021).

Trabalhos Correlatos

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 <i>et al.</i> (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 <i>et al.</i> (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 <i>et al.</i> (2019)	Base Própria Imagens (200 imagens estereoscópicas)	Precisão de 76,6%
Krishnan <i>et al.</i> (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 <i>et al.</i> (2017)	Base Própria	Acurácia: 90% na identificação de pessoas
Lin <i>et al.</i> (2017)	Base Própria – 1710 imagens	Acurácia: 60%
Majeed e Baadel (2016)	Base Própria	Acurácia acima de 90%

Malek <i>et al.</i> (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 <i>et al.</i> (2016)	Base Própria	Acurácia 94,26%
Niu <i>et al.</i> (2017)	Base Própria	Introduziram a base de imagens de mãos e maçanetas e o modelo precisa de ajustes para a correta detecção
Parikh <i>et al.</i> (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 <i>et al.</i> (2015)	2500 imagens de celular de cenários urbanos	Acurácia de 88% a 94% no reconhecimento de faixas de pedestres dependendo da categoria
Rizzo <i>et al.</i> (2017)	Base Própria	Demonstração de processo de fusão de sensores
Tapu <i>et al.</i> (2013)	4500 imagens extraídas do Pascal Dataset	Acurácia por tipo do objeto: Cars – 95%; Bikes – 87%; People – 94%; Obstacles – 90%;
Tapu <i>et al.</i> (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 <i>et al.</i> (2017b)	ImageNet (treinamento) e VOT2016 (testes)	Acurácia: Vehicle: 94%; Bikes: 91%; Pedestrian: 95%; Static Obstruction: 90%
Yang <i>et al.</i> (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 <i>et al.</i> (2016)	Base Própria	Acurácia de detecção acima de 86,7% dependendo da classe do objeto.

Fonte: O autor (2021).

Trabalhos Correlatos

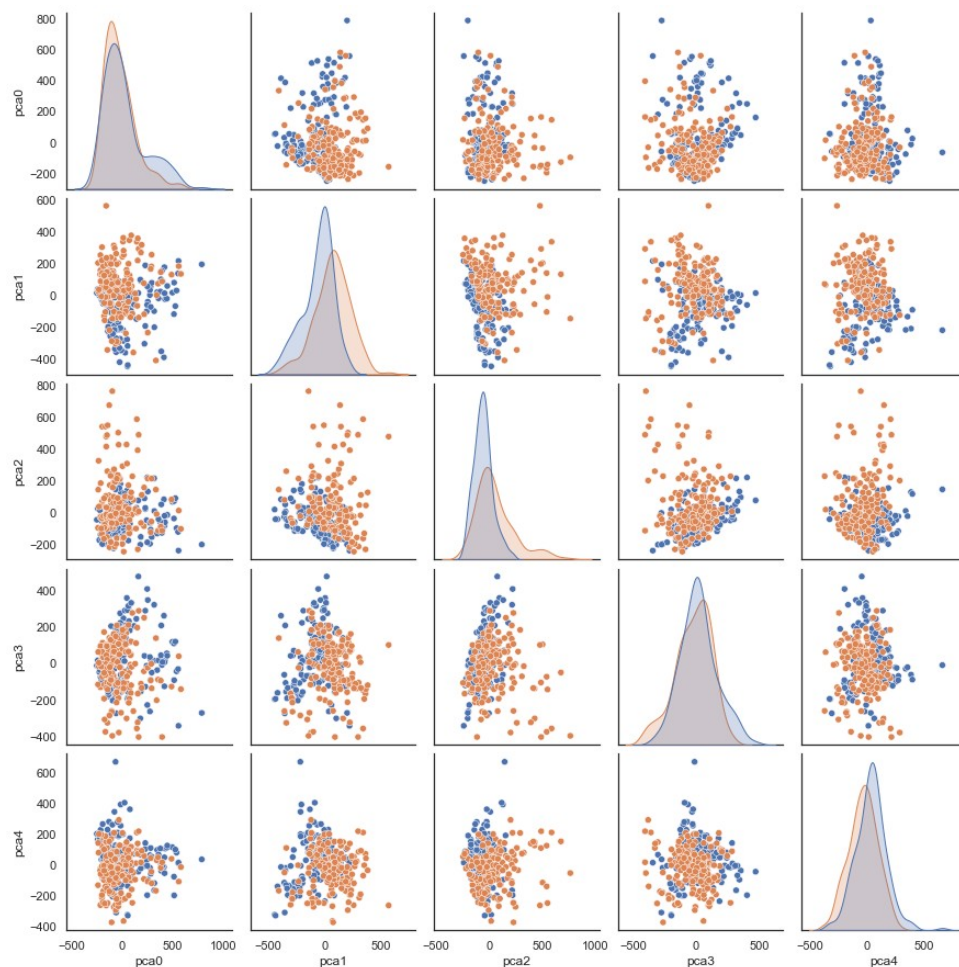
- Considerações
 - Diversas abordagens aplicadas ao auxílio ao deficiente visual;
 - Métodos de verificação baseados na percepção em testes de uso;
 - Acurácias variando de 60% a 97%;
 - Falta de um conjunto de imagens “ideal” para realização de testes;
 - Utilização de bases próprias.

Metodologia

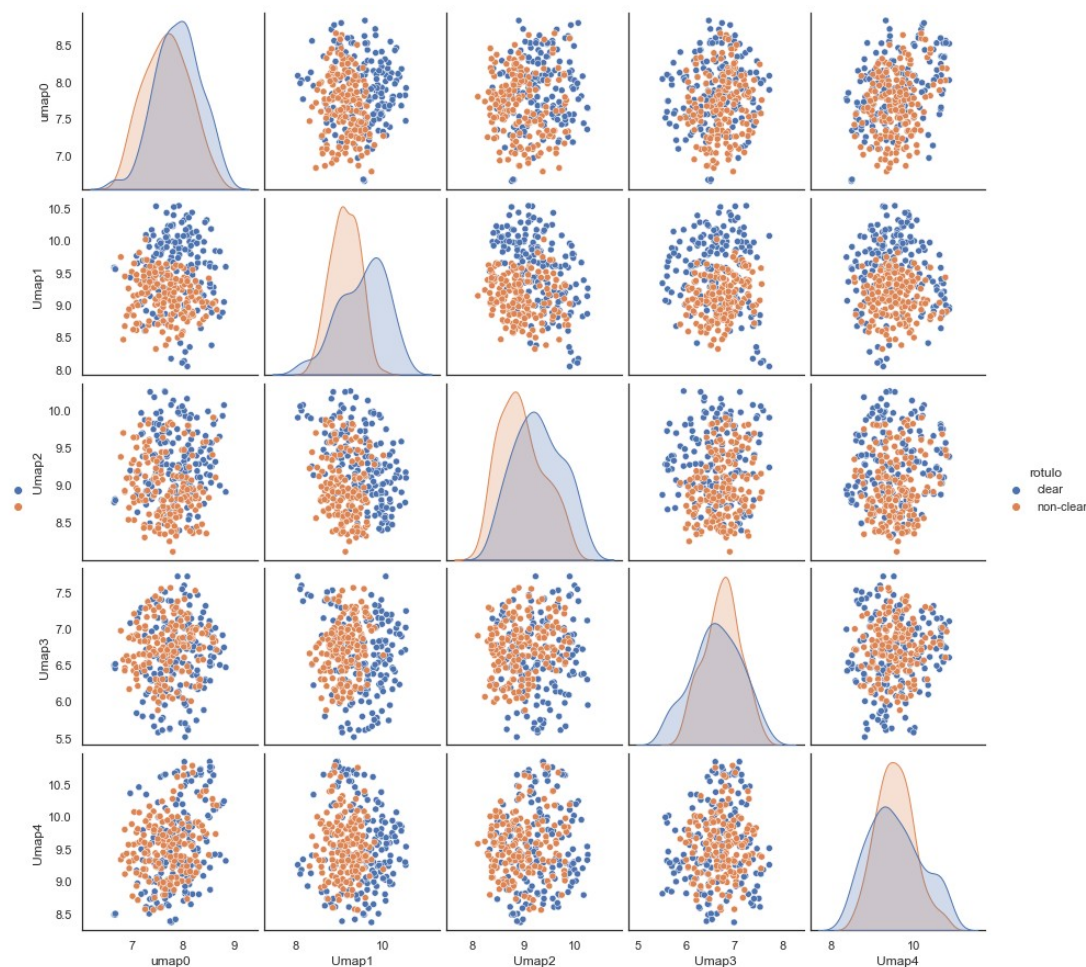
- Baseada no trabalho de Breve e Fischer (2020) (Python+Matlab);
- Modelo desenvolvido em linguagem Python;
- Adoção da premissa da combinação das CNNs VGG16+VGG19 que obtiveram melhor desempenho no trabalho publicado;
- Sem *pooling* (síntese da informação)
- Transferência de aprendizado: ImageNet
- Aferir o impacto da utilização de outro processo de seleção de características (PCA e UMAP);
- Utilizar o mesmo conjunto de imagens (342 imagens, onde 175 demonstram caminhos livres e 167 caminhos com obstáculos).

Metodologia

VGG16+VGG19
50.176 características (sem pooling)
Redução (5 componentes)



PCA



UMAP

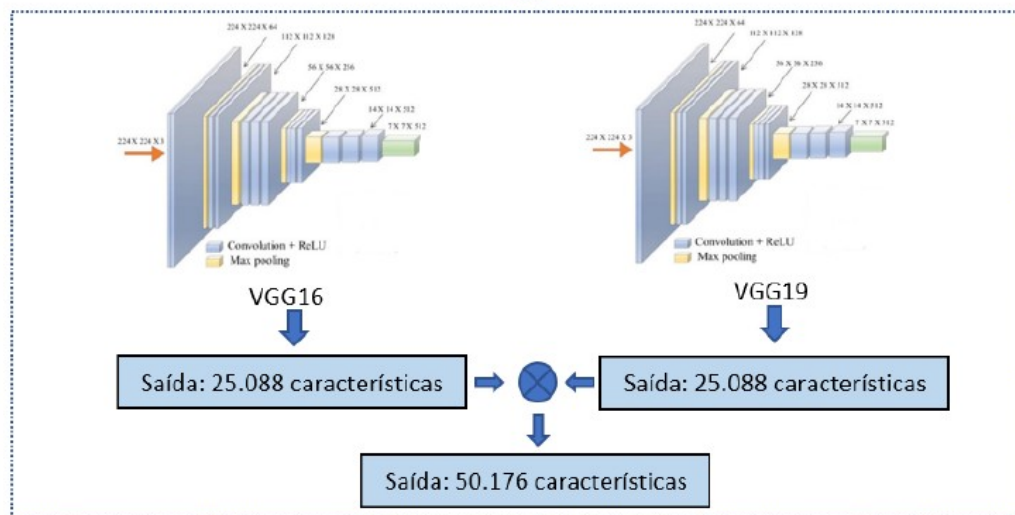
Metodologia

Figura 9 – Ilustração das etapas do método proposto.

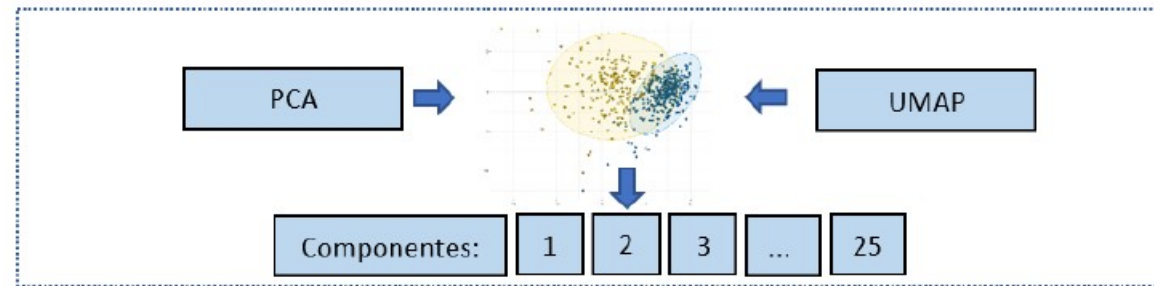
Conjunto de Dados



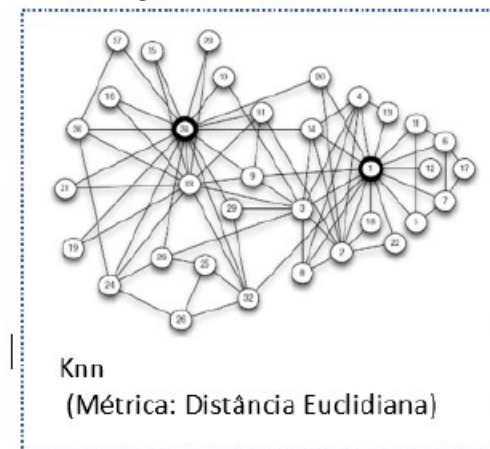
Extração de Características



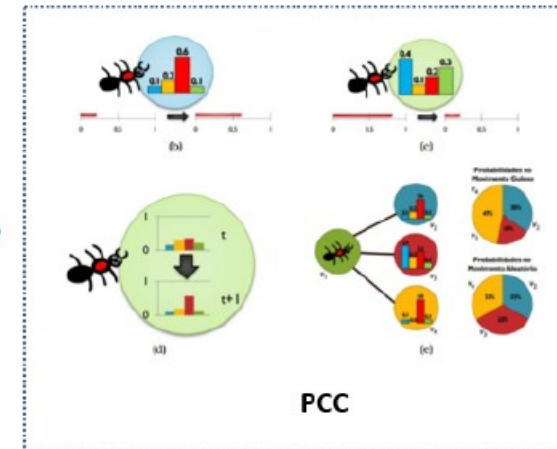
Seleção de Características



Construção do Grafo



Classificador



Fonte: O autor (2021).

Metodologia

- Configuração de testes:
 - VGG16, VGG16 → extração de características utilizando a saída completa da etapa convolucional do modelo (7x7x512) com 25.088 características para cada rede;
 - PCA, UMAP → seleção de características, nesta etapa foram geradas variações de conjuntos de 1 até 25 características (p);
 - Construção da rede complexa (grafo) com variação em k vizinhos próximos, utilizando a distância euclidiana como métrica;
 - Classificador PCC com configurações padrões e exposto a um conjunto com 20% de dados rotulados;
 - Acurácia utilizada como métrica de aferição de desempenho.

Metodologia

- Configuração de testes:

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

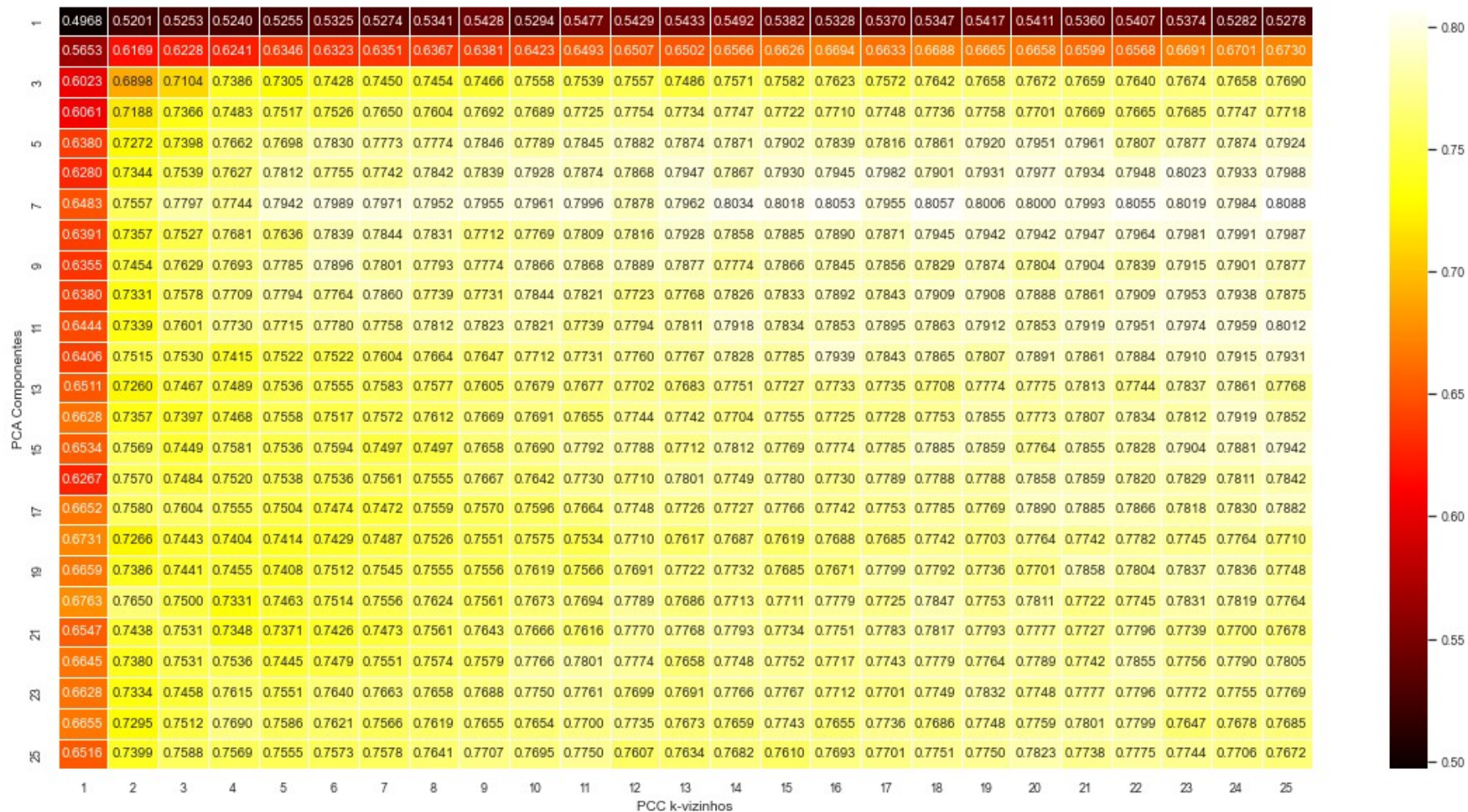
Fonte: O autor (2021).

Metodologia

- Configuração de testes:
 - Materiais
 - Notebook com processador Intel I7-9750H de 2,6GHz;
 - 32 GB de memória RAM;
 - SO Windows 10 Home;
 - Python versão 3.8.12, plataforma Anaconda e IDE Spyder 4.1.4.

Resultados

Mapa de calor de acurácia - PCC+PCA



Resultados

Mapa de calor de acurácia - PCC+PCA

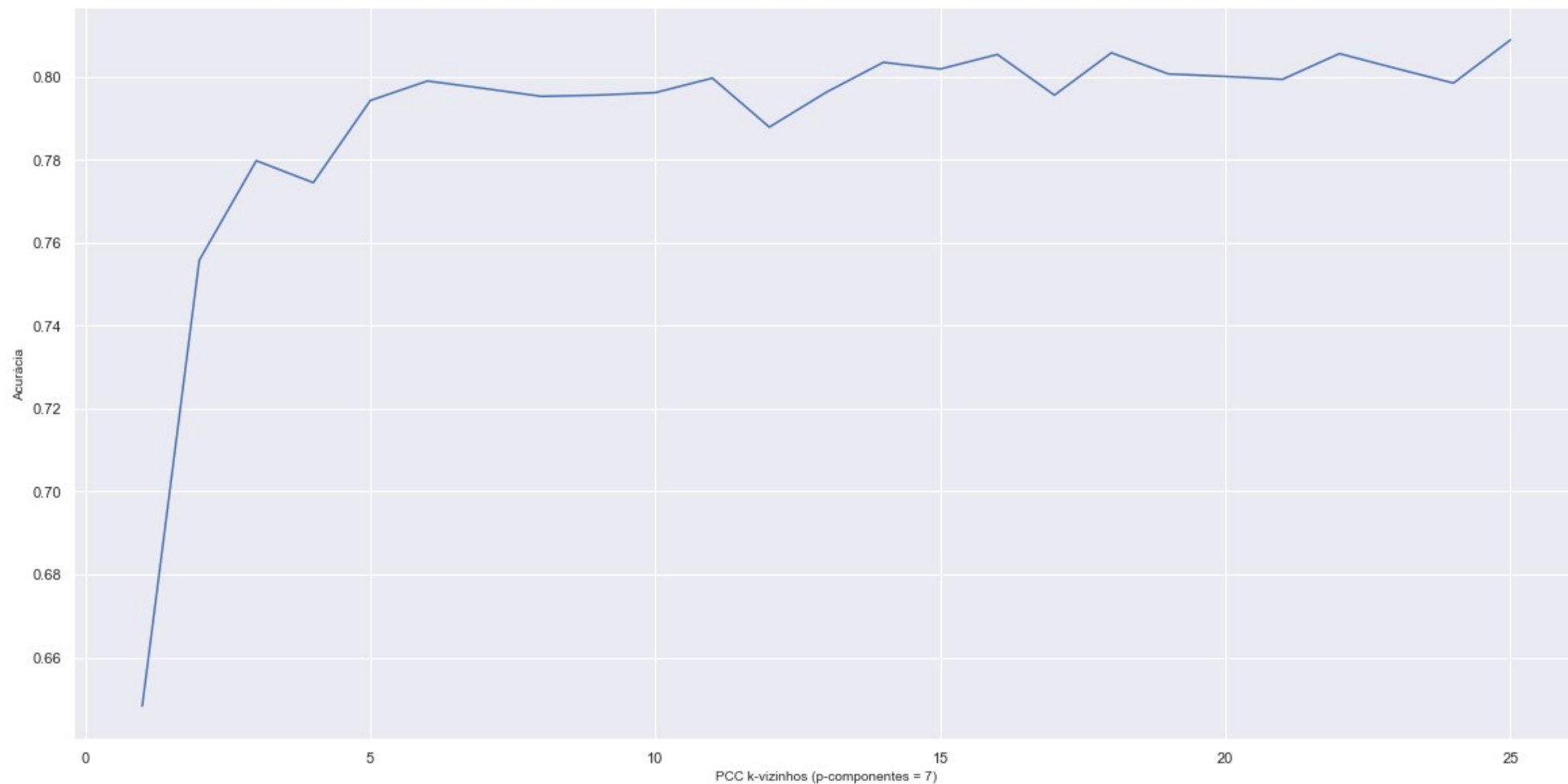
Tabela 9 - PCC+PCA Componentes – melhores acurácias

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

Fonte: O autor (2021).

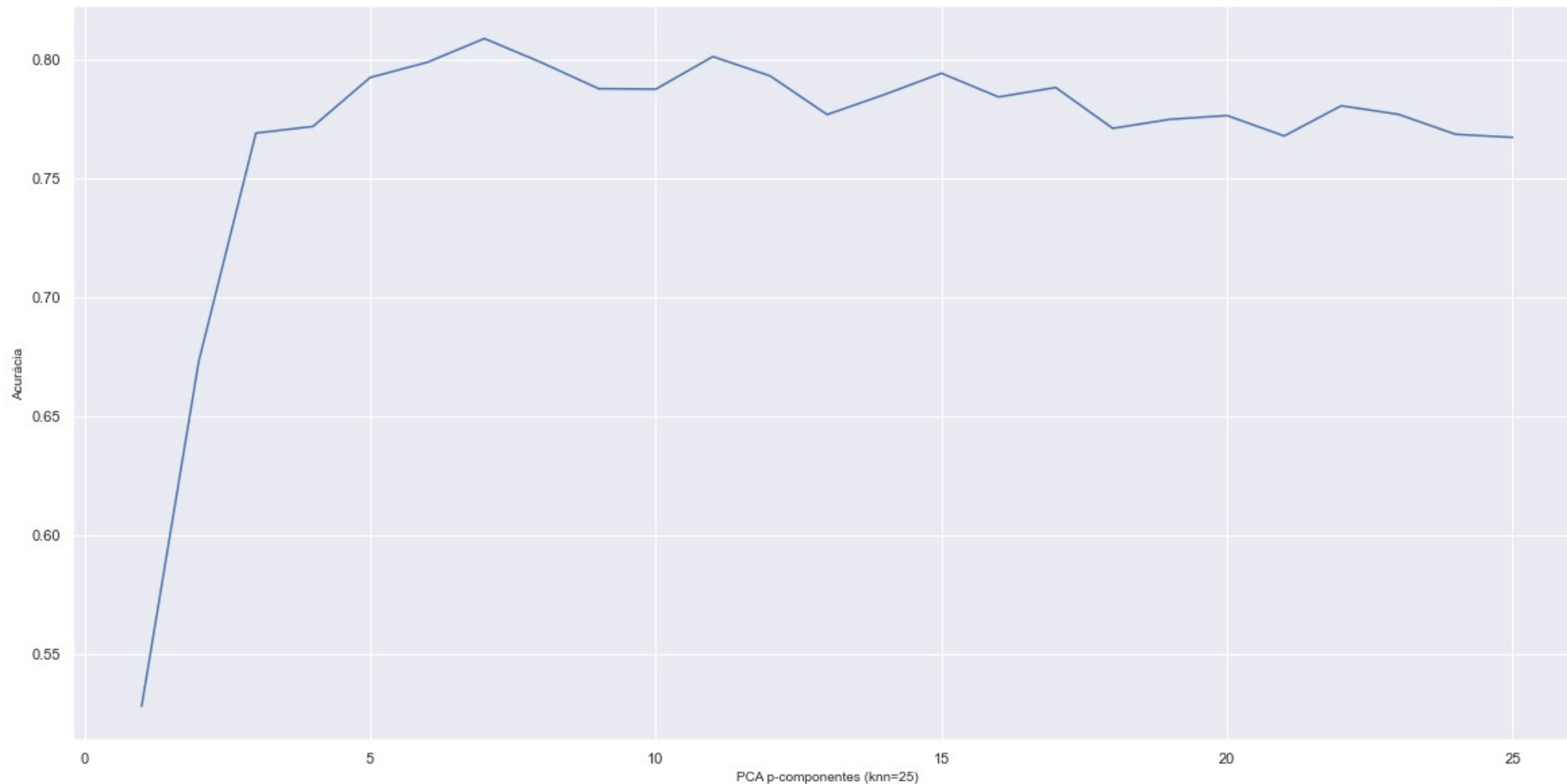
Resultados

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



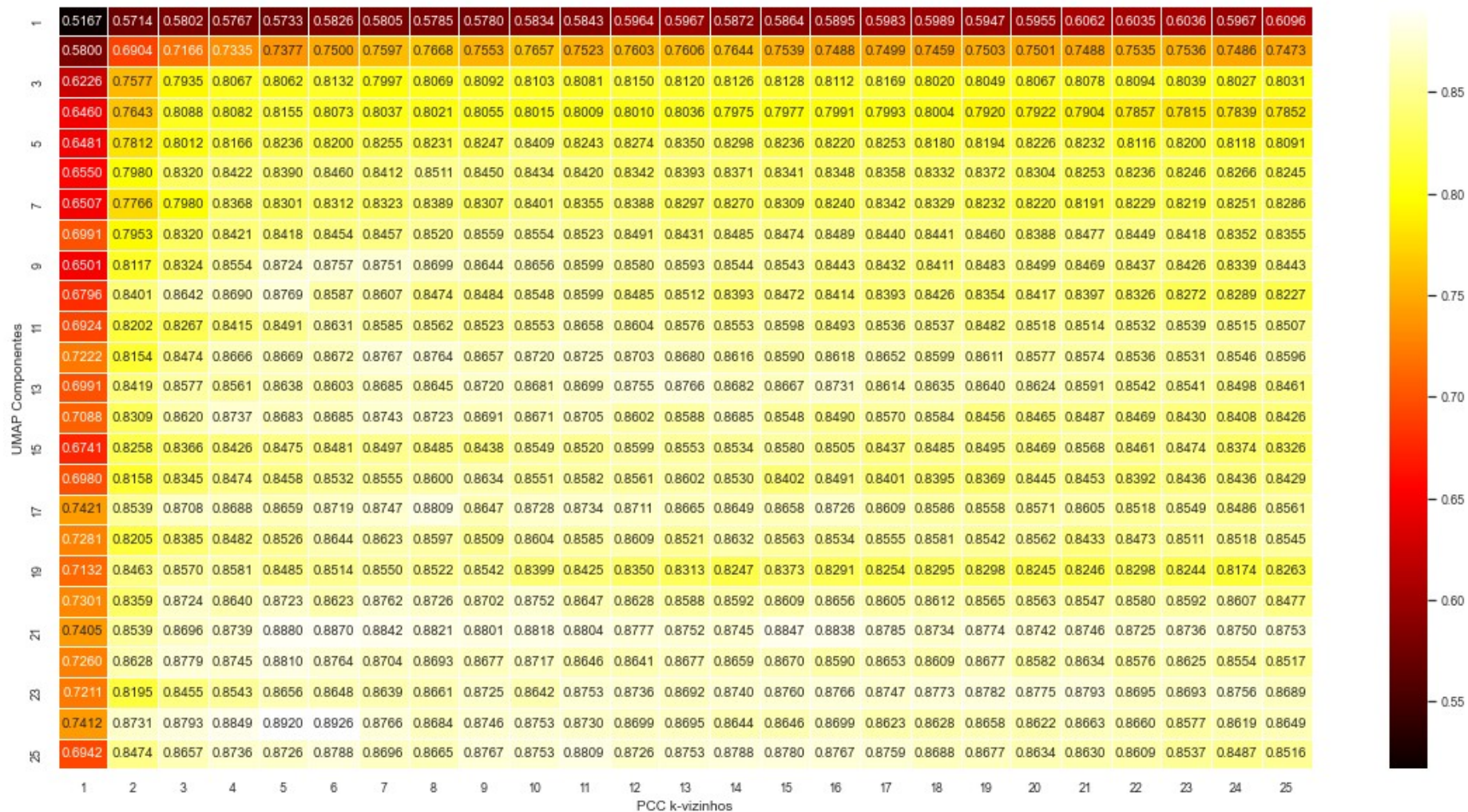
Resultados

Variação da acurácia em p com vizinho $k=25$ (PCC+PCA)



Resultados

Mapa de calor de acurácia - PCC+UMAP



Resultados

Mapa de calor de acurácia - PCC+PCA

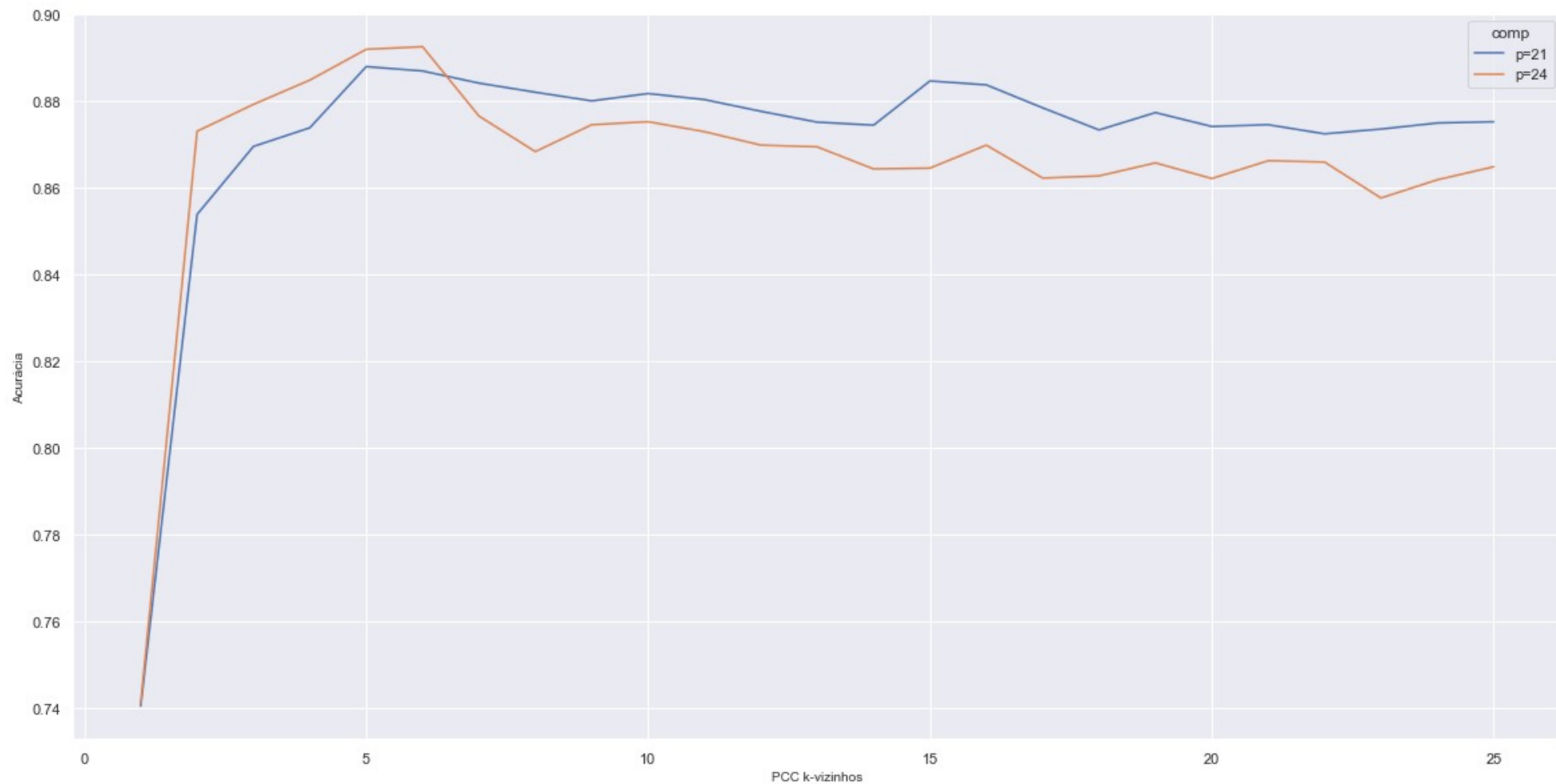
Tabela 10 – PCC+UMAP Componentes – melhores acurácias.

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

Fonte: O autor (2021).

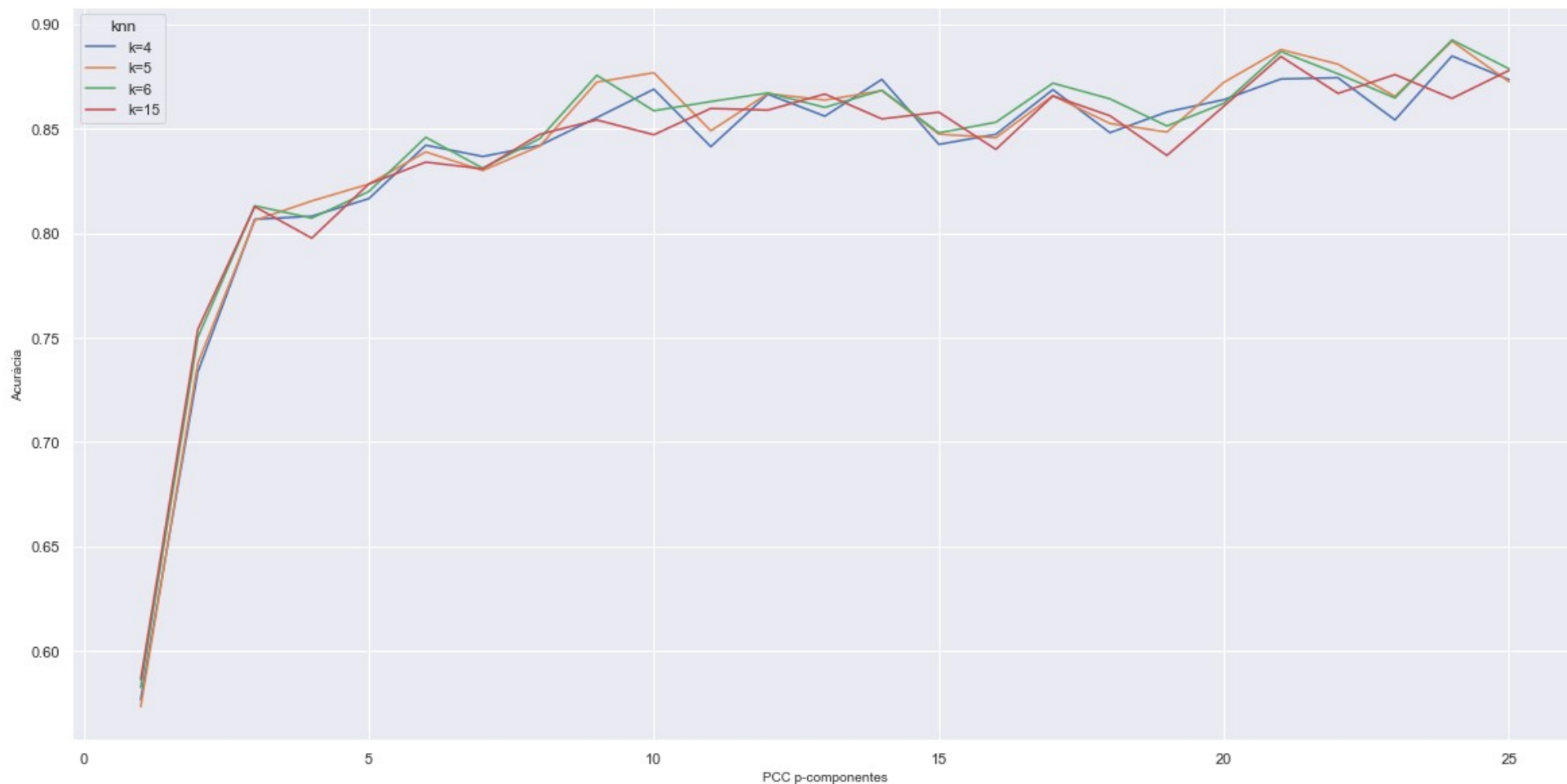
Resultados

Variação da acurácia em k com componentes $p=7$ (PCC+UMAP)



Resultados

Variação da acurácia em p com vizinho $k=25$ (PCC+UMAP)



Resultados

Variação da acurácia em p com vizinho $k=25$ (PCC+UMAP)

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

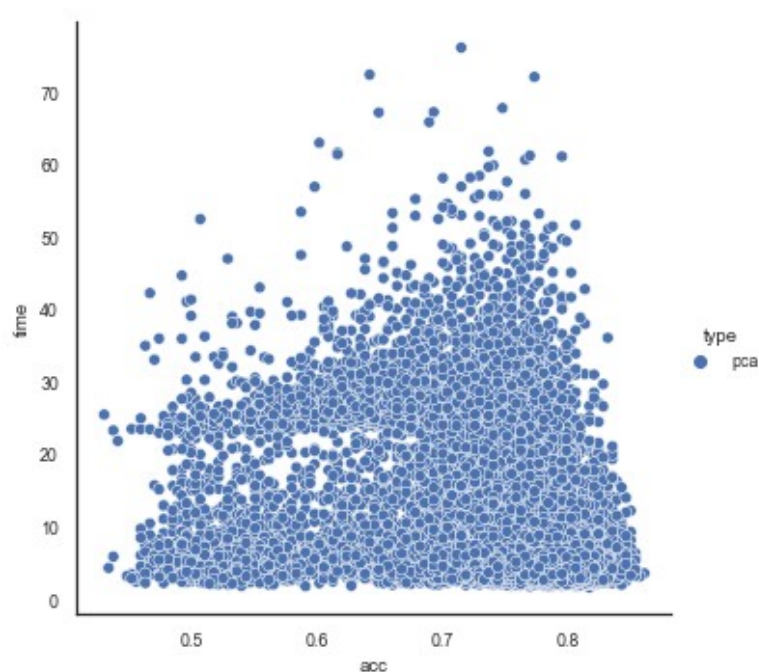
Modelo	Posição	p	k	Acurácia (acc)	Tempo (s)
PCC+UMAP	1	24	6	89,26% \pm 1,83%	10,73s \pm 3,09s
PCC+UMAP	2	24	5	89,20% \pm 2,59%	12,86s \pm 4,23s
PCC+UMAP	3	21	5	88,80% \pm 1,63%	14,11s \pm 4,63s
PCC+UMAP	4	21	6	88,70% \pm 0,18%	11,57s \pm 3,50s
PCC+UMAP	5	24	4	88,49% \pm 1,93%	14,08s \pm 4,16s
PCC+PCA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

Fonte: O autor (2021).

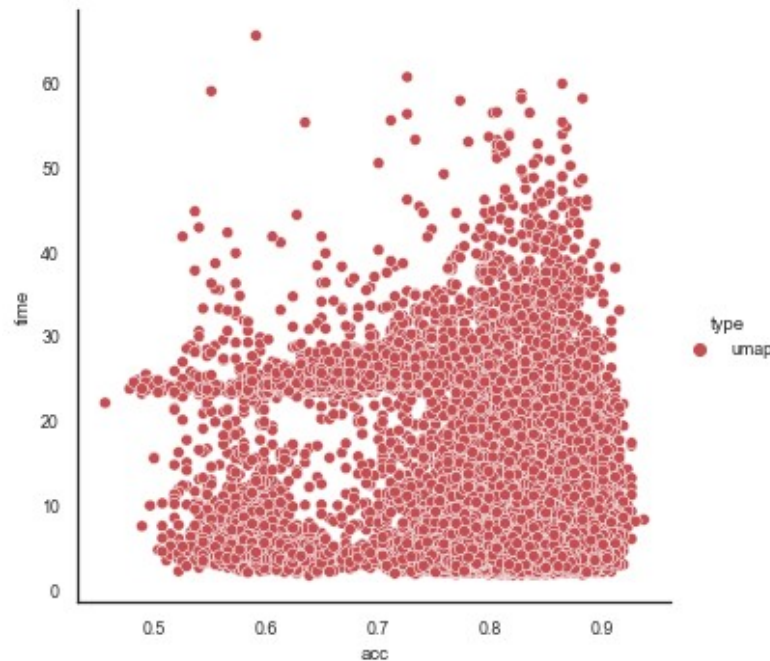
Ranking 62.500 execuções o melhor desempenho de PCC+PCA fica na posição 510.

Resultados

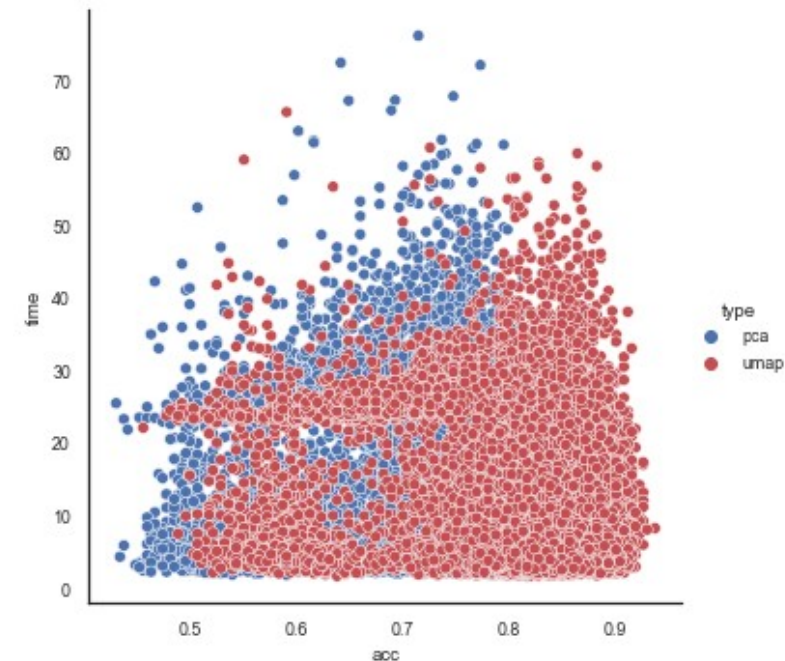
Figura 17 – Análise de dispersão acurácia (*acc*) x tempo de execução (*time*) para os métodos PCC+PCA e PCC+UMAP – (a) PCC+PCA (b) PCC+UMAP (c) PCC+PCA e PCC+UMAP.



(a) PCC+PCA



(b) PCC+UMAP



(c) PCC+PCA e PCC+UMAP

Resultados

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

Trabalho	Extração	Modelo	<i>Polling</i>	<i>p</i>	<i>k</i>	Acurácia (<i>acc</i>)
Proposto	VGG16+VGG19	PCC+UMAP	Não	24	6	89,26% ±1,83%
Proposto	VGG16+VGG19	PCC+PCA	Não	7	2	80,88% ±2,36%
					5	
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).

Resultados

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

Trabalho	Extração	Modelo	<i>Polling</i>	<i>p</i>	<i>k</i>	Acurácia (<i>acc</i>)
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 (<i>fine-tunable</i>)			89,40% ±6,50%
Breve e Fischer (2020)	VGG16	Softmax	AVG (<i>fine-tunable</i>)			89,23% ±7,52%
Breve e Fischer (2020)	Xception	Softmax	AVG (<i>fine-tunable</i>)			91,68% ±3,58%
Breve e Fischer (2020)	MobileNet	Softmax	AVG (<i>fine-tunable</i>)			88,89% ±3,36%

Fonte: O autor (2021).

Considerações Finais

- Resultados iniciais promissores para a continuidade dos estudos;
- Dificuldade de encontrar conjuntos de imagens padronizados para testes;
- Abordagens futuras.

Obrigado