

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Paulo Honório Filho

**Uma abordagem de navegação de veículos aéreos não tripulados
com uso de mapa topológico visual monocular**

Fortaleza – CE

7 de março de 2019

Paulo Honório Filho

Uma abordagem de navegação de veículos aéreos não tripulados com uso de mapa topológico visual monocular

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Linha de Pesquisa: Inteligência Artificial

Orientador: Prof. Dr. A. Wendell de Oliveira Rodrigues

Coorientador: Prof. Dr. Pedro Pedrosa Rebouças Filho

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)

Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Fortaleza – CE

7 de março de 2019

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

- H769a Honório, Paulo Honório Filho.
 Uma abordagem de navegação de veículos aéreos não tripulados com uso de mapa topológico visual
 monocular / Paulo Honório Filho Honório. - 2019.
 82 f. : il. color.
- Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Ciência da Computação, Campus
Fortaleza, 2019.
 Orientação: Prof. Dr. A. Wendell de Oliveira Rodrigues.
 Coorientação: Prof. Dr. Pedro Pedrosa Rebouças Filho.
1. VANTS. 2. Visão Computacional. 3. Mapa Topológico. 4. Localização. 5. Navegação de VANTS. I.
 Titulo.

CDD 004

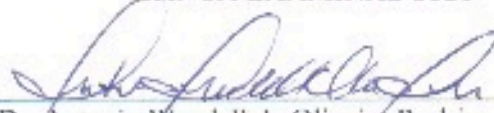
PAULO HONÓRIO FILHO

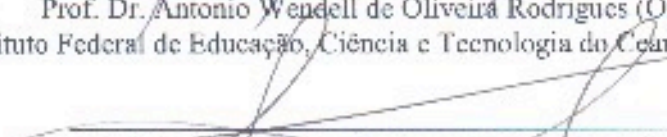
Uma abordagem de navegação de veículos aéreos não tripulados com uso de mapa topológico
visual monocular

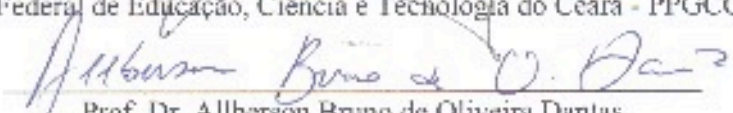
Dissertação submetida à Coordenação do Curso de Pós-graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação, área de concentração Ciência da Computação.

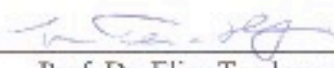
Aprovada em 22/02/2019.

BANCA EXAMINADORA


Prof. Dr. Antonio Wendell de Oliveira Rodrigues (Orientador)
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGCC/IFCE


Prof. Dr. Pedro Pedrosa-Rebouças Filho (Co-orientador)
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGCC/IFCE


Prof. Dr. Allberson Bruno de Oliveira Dantas
Universidade da Integração Internacional da Lusofonia Afro-Brasileira - UNILAB


Prof. Dr. Elias Teodoro da Silva Júnior
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE

Este trabalho é dedicado a minha família por todo apoio neste projeto.

Agradecimentos

Primeiramente a Deus, que sempre me deu uma luz que motiva a persistência nos desafios e acalenta nas horas difíceis.

A minha esposa Nair, por todo investimento de qualquer natureza que fizera em mim nesta etapa de nossas vidas. Por todas orações, companheirismo e amor.

A meu filho Artur, pelas aventuras contadas com seu caderninho de desenhos, me desleigando das preocupações e tensões do cotidiano.

A meus primeiros mentores: meus pais Paulo e Liduina. Por todo suporte a minha família enquanto eu estava isolado e focado em minhas pesquisas e estudos. E pelos momentos sem notícias nossas a custo de muitas saudades do netinho deles.

A meu irmão Dalton, por me convencer a fazer estes mestrado, me permitindo reencontrar uma paixão há muito esquecida nos anos de dedicação ao mercado de trabalho.

A meu orientador Prof. Dr. Wendell Rodrigues e co-orientador Prof. Dr. Pedro Pedrosa, pelos ensinamentos, amizade, confiança e dedicação depositados ao longo deste trabalho. Suas experiências e visão ampla foram cruciais para a conclusão desta maratona.

A todas as novas amizades e companheiros de pesquisa do PPGCC, em especial aos laboratórios LAPISCO, LAMP e LIT, ao Leandro Marinho, Madson Dantas, Lucas Silva, Suane Pires e Raul Medeiros pelas conversas e discussões que foram de suma importância na conclusão desse trabalho.

Ao Wellington Souza e Jefferson Almeida pela ajuda nos testes e montagem dos dados.

A todos os professores e companheiros do PPGCC que de alguma forma contribuíram para minha formação.

“Tenho a impressão de ter sido uma criança brincando à beira-mar, divertindo-me em descobrir uma pedrinha mais lisa ou uma concha mais bonita que as outras, enquanto o imenso oceano da verdade continua misterioso diante de meus olhos”
(Isaac Newton)

Resumo

Este trabalho apresenta uma análise comparativa do desempenho de diversos descritores de imagem e classificadores na tarefa de reconhecimento de locais no ambiente para localização e navegação de um veículo aéreo não tripulado (VANT) usando mapa topológico, por meio da visão monocular de uma câmera convencional embarcada na aeronave. Na metodologia adotada são avaliadas combinações de extratores de atributos clássicos e classificadores modernos existentes na literatura para reconhecimento de cenas, definindo a melhor configuração para esta tarefa. Essa dissertação também analisa a extração de características por meio de Redes Neurais Convolucionais (ou *Convolutional Neural Network* - CNN), aplicando na mesma base de imagens e combinando os mesmos classificadores na abordagem conhecida por transferência de aprendizagem. As melhores métricas obtidas com os extratores clássicos foram 99,77% de sensibilidade e 99,98% de acurácia através da combinação do extrator *Local Binary Patterns* (LBP) com os *Support Vector Machines* (SVM), cujo tempo de extração foi 17,15s e classificação foi 3,36ms. Contudo, os melhores tempos foram observados na extração usando *Gray Level Co-Occurrence* (GLCM) com 11,77s, no treinamento do *k-Nearest Neighbor* (kNN) com GLMC resultando em 45 μ s (e uma acurácia de 99,67%) e na classificação do LBP com *Multi-layer Perceptron* (MLP) repetindo a marca de 10 μ s (e uma acurácia de 99,96%). Quanto aos resultados obtidos pelas CNNs, verifica-se um grupo de empate obtendo a acurácia de 99,97%. No entanto, as melhores marcas de tempo foram obtidas no arranjo MobileNet e classificador Bayesiano com 88ms na extração, 208ms no treino e 6 μ s na classificação e acurácia de 97,83%. Conclui-se, portanto, que dentre as configurações avaliadas a combinação do classificador SVM com o extrator GLCM é a melhor escolha para este desafio de localização e navegação considerando o tempo de resposta na identificação de cenas por imagens monoculares capturadas pela câmera convencional embarcada na aeronave. Por fim, conclui-se que a CNN aumenta a eficácia e confiabilidade na navegação do VANT, caso o mesmo possa arcar com o custo computacional necessário, podendo ser utilizado para este fim em soluções de controladores de voo.

Palavras-chave: VANTS. Visão Computacional. Mapa Topológico. Localização. Navegação de VANTS.

Abstract

This work presents a comparative analysis of the performance among several image descriptors and classifiers using location recognition task applied in the environment for location and navigation of an unmanned aerial vehicle (UAV) using a topological map, through monocular vision of a conventional onboarded camera. In the adopted methodology, classic features extractors combined with modern scene recognition classifiers in the literature are evaluated, defining the best configuration for UAV location and navigation tasks. This dissertation also analyzes the feature extraction applying Convolutional Neural Networks (CNN), using the same database image and combining the same classifiers in the transfer learning approach. The best metrics obtained with the classical extractors were 99.77% of sensitivity and 99.98% of accuracy by combining the Local Binary Patterns (LBP) extractor with the Support Vector Machines (SVM), whose extracting time was 17,15s and 3.36ms in the classification step. However, the best performance times were observed in the extraction step using the Gray Level Co-Occurrence (GLCM) with a time of 10.22s. The GLCM trained by k -Nearest Neighbor (kNN) resulted in a response time of 45 μ s with an accuracy of 99.67%, while the LBP classification with Multi-layer Perceptron (MLP) achieved the time of 10 μ s with an accuracy of 99.96%. About the results obtained with the CNNs, it was verified that there is a tie group achieved the highest accuracy, with 99.97%. On the other hand, the best response times were obtained in the arrangement MobileNet plus Bayesian classifier, resulting in 88ms in extraction step, 208ms in training step and 6 μ s in classification step with an accuracy of 97.83%. Thus, the combination of SVM classifier with GLCM extractor is the best solution for proposed location and navigation tasks, considering the obtained time based in the identification of the monocular images taken by the conventional onboarded camera in the aircraft. Finally, assuming that the UAV can handle the computational cost, it is possible to conclude that CNN increases the effectiveness and reliability of the UAV navigation, being satisfactory used for the location and navigation in flight controller solutions.

Keywords: VANTS. Computer vision. Topological Map. Location. Navigation of VANTS.

Lista de ilustrações

Figura 1 – Exemplo de uma GLCM $g(i, j)$ em C , obtida a partir da imagem $f(x, y)$ em A composta por dez variações de intensidade mapeadas em B com níveis de cinza claro ao escuro em ordem crescente. Com a distância d de um pixel e direção α em 0° . A matriz $g(i, j)$ resulta em 36 pares, sendo $g(7, 8) = 1$ um exemplo destacado em C	35
Figura 2 – Tomando uma vizinhança 3×3 ($P = 8$ e $R = 1$) sobre um pixel arbitrário com a escala de cinza equivalente a 55, obtém-se o padrão $b_{ij} = 11111000_2$ pela comparação do pixel v_c com seus vizinhos. O padrão binário segue a ordem dos bits de conversão do menos significativo ao mais significativo apresentados na terceira matriz, gerando o valor 31_{10} computado no histograma LBP.	37
Figura 3 – Estrutura de uma rede MLP com N neurônios na camada de entrada, duas camadas escondidas com N neurônios cada e dois neurônios na camada de saída.	41
Figura 4 – (a) Conjunto de treinamento modelado como um gráfico completo; (b) Um cálculo da árvore de abrangência mínima sobre o conjunto de treinamento em que protótipos são destacados; (c) floresta de caminho ótimo sobre o conjunto de treinamento; (d) processo de classificação de uma amostra desconhecida; e (e) a amostra de teste é finalmente classificada.	43
Figura 5 – Exemplo de pré-processamento de um SVM, em que θ é a função discriminante que mapear os dados na entrada em um espaço de características.	44
Figura 6 – Matriz de confusão para um problema de classificação com 24 rótulos.	48
Figura 7 – Bebo PF722000	52
Figura 8 – Campus do IFCE e locais mapeados.	53
Figura 9 – A ConvNet A foi treinada com a base de dados Imagenet (DENG et al., 2009), sendo que os pesos convolucionais foram congelados e transferidos para a ConvNet B , reduzindo o treinamento da mesma à atualização dos pesos da camada densamente conectada.	55
Figura 10 – Ambientes e mapa topológico.	57
Figura 11 – Fluxo adotado para navegação do VANT nesta pesquisa.	58
Figura 12 – Estrutura do LeNet-5, um ConvNet, aqui utilizado para reconhecimento de dígitos. Cada quadro é um mapa de características, isto é, um conjunto de unidades cujos pesos são restritos para serem idênticos.	77
Figura 13 – Padrão de conectividade da CNN entre os neurônios das camadas adjacentes.	78

Figura 14 – Exemplo de uma ConvNet. 80

Lista de tabelas

Tabela 1	– Pesquisas de navegação com visão computacional - Parte 1.	29
Tabela 2	– Pesquisas de navegação com visão computacional - Parte 2.	30
Tabela 3	– Especificação do VANT Parrot Bebop PF722000.	51
Tabela 4	– Especificação da estação de controle externa.	51
Tabela 5	– Base de imagens por período e horário.	54
Tabela 6	– Quantidade de características extraídas em cada ConvNet.	56
Tabela 7	– Instruções para manobrar o VANT e rotas de navegação consideradas: Siga em Frente (SF), Guinada de 90° a esquerda (G90E), Guinada de 90° a direita (G90D) e Guinada de 180° (G180).	57
Tabela 8	– Especificidade (Sp), Sensibilidade (Se), Valor Preditivo Positivo (PPV) e Acurácia (Acc) obtida pelos arranjos de ferramentas de extração e classificação propostos, acompanhadas do desvio padrão à direita. . . .	59
Tabela 9	– Média dos tempos de treino, de teste e de extração obtidos pelos arranjos de ferramentas de extração e classificação propostos, acompanhado pelos respectivos desvios padrões à direita.. . . .	60
Tabela 10	– Acurácia (Acc) e $F1-Score$ (F1S) obtidos pelos arranjos de extratores de características e classificadores.	63
Tabela 11	– Acurácia (Acc), tempo de treino, tempo de teste e tempo de extração obtidos pelos classificadores e extratores de características respectiva- mente.	65

List of Algorithms

1	Algoritmo estocástico de Max Pool	79
---	---	----

Lista de abreviaturas e siglas

Ac	Acurácia
ConvNets	Convolutional networks (Redes convolucionais)
CNN	Convolutional networks (Redes convolucionais)
Es	Especificidade
GLCM	Gray-Level Co-occurrence Matrix (Matriz de Co-ocorrência)
GPS	Global Positioning System (Sistema de Posicionamento Global)
GPU	Graphic Processor Unit (Processadores dedicados para processamento gráfico e paralelo)
IFCE	Instituto Federal de Educação, Ciência e Tecnologia do Ceará
ILSVRC	ImageNet: Large Scale Visual Recognition Challenge (RUSSAKOVSKY et al., 2015)
LBP	Local Binary Pattern (Padrão Binário Local)
MLP	Multi Layer Perceptron (Perceptron de Múltiplas Camadas)
OACI	Organização de Aviação Civil Internacional
OPF	Optimum-Path Forest (Floresta de Caminhos Ótimos)
PDI	Processamento Digital de Imagens
PPV	Positive Predictive Value (Valor Preditivo Positivo)
ReLu	Rectified Linear Units (Unidade Linear Retificada)
RPA	Remotely Piloted Aircraft (Aeronave Remotamente Pilotada)
RNA	Redes Neurais Artificiais
SLAM	Simultaneous Localization and Mapping (Localização e Mapeamento. Simultâneos.)
SVM	Support Vector Machines (Máquinas de Vetores de Suporte)
Se	Sensibilidade
UAV	Unmanned Aerial Vehicle

VANT Veículo Aéreo Não Tripulado

VPP Valor Preditivo Positivo

Lista de símbolos

$\mathbb{R}$	Conjunto dos números reais
$f(x, y)$	Função bidimensional de intensidade de luz (imagem)
$I(x, y)$	Matriz bidimensional de intensidade de luz (imagem)
max	Máximo
min	Mínimo
M_{pq}	Momento geral de ordem p, q de uma imagem
m_{pq}	Momentos Espaciais
μ_{pq}	Momentos Centrais
η_{pq}	Momentos Centrais Normalizados
ϕ_{pq}	Momentos de Hu
G_D^θ	Matriz de co-ocorrência com distância D e direção
L	Número de níveis de cinza
X	Vetor de características
C	Classe de uma amostra
$P(X, Y)$	Probabilidade conjunta
$P(Y X)$	Probabilidade a posteriori
$sign(x)$	Função sinal
$K(x, x_i)$	Função de kernel
H	Matriz Hessiana
J	Matriz Jacobiana

Sumário

1	INTRODUÇÃO	27
1.1	Motivação do Trabalho	28
1.2	Trabalho relacionados	28
1.3	Problema a ser Resolvido	28
1.4	Objetivos Geral e Específicos	31
1.5	Produção científica	31
1.6	Estrutura da Dissertação	31
2	TÉCNICAS COMPUTACIONAIS UTILIZADAS	33
2.1	Reconhecimento de Ambiente	33
2.1.1	Extratores de Características	33
2.1.1.1	Fourier	33
2.1.1.2	Matriz de Coocorrência	34
2.1.1.3	Padrões Binários Locais	36
2.1.2	Classificadores de Padrões	38
2.1.2.1	Bayes	38
2.1.2.2	k-Nearest Neighbor	39
2.1.2.3	Multi Layer Perceptron	40
2.1.2.4	OPF	42
2.1.2.5	Support Vector Machine	43
2.2	Aprendizagem Profunda	45
2.2.1	Convolutional Neural Networks	45
2.2.1.1	VGGNet	45
2.2.1.2	ResNet	46
2.2.2	Transferência de Aprendizagem	47
2.3	Métricas de Avaliação	47
3	PROPOSTA E METODOLOGIA	51
3.1	Equipamento Utilizado	51
3.1.1	Quadrirrotores	52
3.2	Rotas Predefinidas	52
3.3	Base de imagens	54
3.4	Etapas de Extração de Características	54
3.5	Extração de Características e CNNs	55
3.6	Etapas de Classificação	55
3.7	Rotas de Navegação	56

3.8	Abordagem da Navegação Proposta	58
3.9	Avaliação de Desempenho	58
4	EXPERIMENTOS E RESULTADOS	59
4.1	Experimento com as ConvNets	61
5	CONCLUSÃO	67
5.1	Trabalhos Futuros	68
	REFERÊNCIAS	69
	APÊNDICES	75
	APÊNDICE A – CONVNETS	77

1 Introdução

Veículos aéreos não tripulados (VANTs), comercialmente conhecidos como *Drones* (do inglês Zangão), são o foco de várias pesquisas para as mais diferentes finalidades. Tal atração é uma resposta ao tamanho pequeno da aeronave, o baixo custo e o grande potencial de manipulação e manobras (STOJKOSKA et al., 2017). O termo adotado tecnicamente pela OACI (Organização de Aviação Civil Internacional) para identificar esse tipo de aeronave é o RPA (*Remotely Piloted Aircraft*), passando a substituir a nomenclatura VANT, considerada obsoleta na comunidade aeronáutica internacional (AERONÁUTICA, 2016), contudo como a regulação de voo e legislação não são o foco deste mestrado, este texto irá manter a definição oriunda do termo *Unmanned Aerial Vehicle* (UAV).

PETKOVICS et al. (2017), utiliza uma aeronave empregada na agricultura de precisão, visando coletar dados para auxiliar uma fazenda de maneira mais eficiente e, assim, reduzir seus custos. Enquanto em RESHMA; RAMESH; SATHISHKUMAR (2016), um método de monitoramento do tráfego rodoviário usando VANT é proposto, abordando a dificuldade de gerenciar as redes rodoviárias, devido às suas grandes distâncias. Em KOBAYASHI et al. (2017), os autores apresentam soluções para a busca de vítimas em desastres naturais utilizando VANTs e tecnologia sem fio. Já NGUYEN; JENSSEN; ROVERSO (2018), propõem uma inspeção automática e autônoma de linha de energia baseada na visão obtida por VANT.

A navegação dos VANTs tem sido cada vez mais explorada, com o objetivo de elaborar mais sistemas modernos, com a capacidade de navegar de forma independente e precisa (CHEN, 2014). No geral, a tarefa de navegação de um VANT é executada usando *Inertial Navigation Systems* (INS), que fornece informações de posição, velocidade e altitude do VANT (NEMRA; AOUF, 2010). Contudo, este sensor possui pequenos erros de polarização, que aumentam continuamente com o tempo. Assim, informações adicionais sobre a posição do veículo são necessárias a partir de um sensor de navegação preciso, como Sistema de Posicionamento Global (GPS) (NEMRA; AOUF, 2010).

Vale salientar também que os sensores INS para navegação aérea tem um custo elevado, não sendo populares em pequenas aeronaves como um VANT. Embora o GPS ofereça navegação com um receptor barato, em algumas situações este sistema pode simplesmente não receber o sinal de satélite, seja devido a obstáculos, bloqueio ou cobertura do sinal (YOO; AHN, 2003; UNICOMB et al., 2017; SKULSTAD et al., 2015).

1.1 Motivação do Trabalho

A identificação da localização e a realização da navegação de VANTs usando diferentes recursos se tornou um interesse de investigação, motivado pela necessidade de obter alternativas independentes de satélites (NGUYEN; JENSSEN; ROVERSO, 2018; KOBAYASHI et al., 2017). Por exemplo, pode ser vista como uma alternativa ao INS/GPS, um dos meios mais comuns para a navegação de VANTs, o reconhecimento de imagens, sendo possível a obtenção de uma navegação mais precisa baseada em tratamentos visuais, independente de satélite (SKULSTAD et al., 2015).

Entre as principais vantagens dos sensores visuais, pode-se destacar a não dependência de sinais externos (KANIEWSKI, 2017), não é necessário o fornecimento de informações sobre meio ambiente, são altamente recomendados para percepção em ambientes dinâmicos (alta capacidade anti-interferência). Além disso, ainda dificultam a identificação do sistema de detecção, pois utilizam principalmente sensores do tipo passivo (SKULSTAD et al., 2015).

1.2 Trabalho relacionados

Alguns trabalhos são apresentados em uma breve revisão bibliográfica do conteúdo relacionado à navegação de veículos aéreos não tripulados. Neste sentido, esta revisão visa esclarecer a questão da localização de dispositivos móveis na robótica. Logo em seguida aborda as soluções de Localização e Mapeamento Simultâneos (SLAM) que muito avançaram na otimização de navegação de VANTs em ambientes desconhecidos. Por fim, serão abordadas formas de identificação de cenários para avaliação das abordagens analisadas nesta pesquisa.

Como pode ser encontrado nas Tabelas 1 e 2, os trabalhos solucionaram a navegação por imagem aplicando um único método, especializando-se em um único modelo de aeronave ou fazendo reconhecimento de marcos naturais. Estes artigos estão disponíveis nas plataformas de busca IEEE *Xplore*, Google Adêmico e Elsevier. De contrapartida, nesta dissertação foram avaliadas diversas abordagens em um VANT comercialmente disponível, possibilitando ser replicado em mais aeronaves e diversificando a unidade controladora ou a estação de comando.

1.3 Problema a ser Resolvido

Esse estudo busca uma solução de localização e navegação de um VANT por meio de um sensor ótico comumente encontrado nestas aeronaves, tendo como fonte de dados o ambiente em torno do campo de visão do VANT.

Tabela 1 – Pesquisas de navegação com visão computacional - Parte 1.

Título	Vantagem	Desvantagem	Referência
A Vision-Based Guidance System for UAV Navigation and Safe Landing using Natural Landmarks	Não requer qualquer marco artificial e é robusto para oclusões e variações de luz	Não possibilita interrupção na rota por detecção de algum impedimento ou evento	CESETTI et al. 2010
A Neuro-based Network for On-line Topological Map Building and Dynamic Path Planning	Esta abordagem é capaz de produzir um consistente e estável mapa topológico em ambientes naturais sem intervenção humana	Não houve treinos em condições reais com ambientes complexos de navegação	CHIN et al. 2017
Real-Time Robot Localization, Vision, and Speech Recognition on Nvidia Jetson TX1	O modelo proposto conta com um sistema embarcado com recursos de computação heterogênea, podendo processar imagens e SLAM, além de reconhecimento de voz.	Utiliza equipamento onde o peso da bateria e da unidade de processamento com GPU iriam exigir um VANT com categoria maior.	TANG; REN; LIU et al. 2017
A novel mobile robot localization approach based on topological maps using classification with reject option in omnidirectional images	Apresenta uma abordagem de localização para robôs móveis que se baseia em mapas topológicos visuais com processamento de imagens omnidirecionais, apresentando resultados promissores.	O hardware embarcado com processamento elevado, além de uma câmera omnidirecional com espelho esférico, são poucos comuns em um VANT de baixo custo.	MARINHO et al. et al. 2017
Developing of a Video-Based Model for UAV Autonomous Navigation	Mostra uma modelagem contínua em um VANT comercial com baixo custo computacional, adotando somente a câmera embarcada e uma estação de controle externa.	A análise de extração de recursos teve foco no algoritmo <i>Oriented FAST and Rotated BRIEF</i>	AGUILAR et al. et al. 2017

Fonte: Próprio autor.

Tabela 2 – Pesquisas de navegação com visão computacional - Parte 2.

Título	Vantagem	Desvantagem	Referência
Robust autonomous flight and mission management for MAVs in GPS-denied environments	Traz uma estimativa de localização por meio de um scanner a laser 2D, otimizada pelo gerenciamento de missões com visão computacional e embarcados em um hardware de poder computacional restrito.	Para embarcar uma tecnologia 2D laser scanner é necessário um VANT com porte e custo mais elevados.	BI et al. et al. 2017
Problems of a Trajectory Planning in Autonomous Navigation Systems Based on Technical Vision and Ai	O sistema foi desenhado de forma a se tornar um único chip de navegação por visão	A falta de dados da fabricação do chip leva a percepção do alto custo ao se ampliar os pontos de rotas ou novas inferências	ZHILENKOV et al. 2018
DroNet: Learning to Fly by Driving	Prevê simultaneamente a probabilidade de colisão e o ângulo de direção desejado, permitindo que um VANT reaja prontamente a eventos imprevistos e obstáculos	A aplicação foi treinada em ambiente urbano, que possui bases de imagens em abundância, auxiliando no treinamento	LOQUERCIO et al. 2018
Autonomous UAVs for Structural Health Monitoring Using Deep Learning and an Ultrasonic Beacon System with Geo-Tagging	A execução de um voo autônomo preciso em um VANT comercialmente disponível, substituindo o GPS por uma abordagem de localização com bluetooth.	Para efetuar a triangulação com os beacons são necessários quatro radios bluetooth instalados no ambiente de voo.	KANG; CHA et al. 2018

Fonte: Próprio autor.

1.4 Objetivos Geral e Específicos

O presente trabalho tem como objetivo principal o estudo sobre VANT, focando em contribuições que valorizem a precisão da navegação e assertividade na localização do mesmo. Baseando-se no estado de arte das técnicas de processamento de imagens, elaborando uma análise que identifique o menor tempo de resposta do controle do dispositivo e a maior inferência na identificação de cenários do ambiente em torno da aeronave.

Em seu sentido mais particular, os objetivos específicos são compostos por:

- Criar um banco de imagem dos ambientes por meio da câmera embarcada no VANT.
- Mapear as imagens com um mapa de rotas de navegação pelo campus do IFCE.
- Analisar combinações de extratores de características e classificadores consolidados com as imagens adquiridas.
- Aplicar mesma análise de combinações de CNNs¹ e classificadores, usando o conceito de transferência de aprendizado.
- Identificar a viabilidade do uso das técnicas na aplicação em tempo real.
- Propor uma abordagem para auxiliar a tomada de decisão durante a localização da aeronave em voo.
- Desenvolver um módulo de controle externo ao VANT para prover precisão e assertividade tanto na navegação quanto na localização com um mapa topológico.

1.5 Produção científica

Durante este projeto de mestrado, o seguinte trabalho científico foi aceito, a saber:

- **Paulo P. H. F. Honório Filho, Suan Pires Pinheiro da Silva, Jefferson Silva, Leandro B. Marinho, Tiago Carneiro, Antonio Wendell de O. Rodrigues e Pedro P Rebouças Filho. An Approach to Navigation in Outdoor and Indoor Environments with Unmanned Aerial Vehicle Using Visual Topological Map.** Em: *Workshop of Works in Progress (WIP)* do SIBGRAPI 2018, *31th Conference on Graphics, Patterns and Images*, em Foz do Iguaçu, Paraná, Brazil;

1.6 Estrutura da Dissertação

A dissertação está composto nos seguintes capítulos abaixo.

O **Capítulo 2** apresenta as técnicas clássicas empregadas em processamento digital de imagens para extração de características e classificação estado de arte. São apresentadas,

¹ Detalhes sobre Redes Convolucionais (CNNs) estão disponíveis no Apêndice A

também, as técnicas de Redes Neurais Convolucionais utilizadas para os arranjos dos algoritmos propostos de localização e navegação da VANT.

No **Capítulo 3** é apresentada a proposta do trabalho e a metodologia utilizada para a implementação dos algoritmos de localização e navegação.

O **Capítulo 4** apresenta os resultados e as discussões dos experimentos isolados, com as imagens sintéticas e reais utilizando os dois tipos de câmera, assim como do teste de navegação juntamente com a abordagem proposta para otimizar o processo de classificação.

No **Capítulo 5** estão as conclusões do estudo realizado nesta dissertação e as propostas para trabalhos futuros.

A definição das CNNs e o conceito de *Transfer Learning* são apresentados na Sessão 2.2.2.

2 Técnicas Computacionais Utilizadas

Nesse capítulo serão apresentadas as técnicas utilizadas para a realização do processamento das imagens coletadas pelo VANT para o reconhecimento de ambientes, bem como os algoritmos de aprendizagem profunda utilizados para a implementação de experimentos que validaram as análises propostas nesta pesquisa.

2.1 Reconhecimento de Ambiente

O reconhecimento de ambientes é uma das tarefas cruciais para a localização de um dispositivo móvel na robótica e a literatura de processamento digital de imagens (PDI) possui ferramentas que possibilitam soluções excelentes para esse desafio.

Neste capítulo serão abordadas técnicas clássicas de processamento digital de imagens para extração de características e classificação. Esses métodos possuem uma literatura rica e estão presentes em estudos com abordagens variadas como segmentação de imagens e reconhecimento de padrões, tornando-os promissores para investigação deste problema de localização.

2.1.1 Extratores de Características

Nesta sessão os algoritmos de extração adotados nesta dissertação serão explanados detalhadamente e a fim de se obter melhor entendimento do desempenho de cada um deles nos testes e experimentos de navegação de VANTs.

2.1.1.1 Fourier

A Transformada de Fourier é utilizada para decompor uma imagem em seus componentes de seno e co-seno. A saída da transformação representa a imagem no domínio de frequência (ou de Fourier), enquanto a imagem de entrada é o equivalente do domínio espacial. Na imagem do domínio de Fourier, cada ponto representa uma frequência particular contida no domínio espacial. A Transformada de Fourier é usada em uma ampla gama de aplicações, como análise de imagem, filtragem de imagem, reconstrução de imagem e compressão de imagem (GONZALEZ, 2009).

Em (LACHKAR, 2003), entende-se que na tarefa de extrair características de uma imagem digital, o foco da pesquisa restringi-se a Transformada Discreta de Fourier (DFT), sendo a mesma uma amostrada e, portanto, não possui todas as frequências que formam uma imagem, mas apenas um conjunto de amostras que é grande o suficiente para descrever completamente a imagem do domínio espacial, cujo número de frequências

corresponde ao número de pixels na imagem do domínio espacial, isto é, ambos domínios têm o mesmo tamanho. Logo, para uma imagem quadrada de tamanho $N \times N$, a DFT bidimensional é dada por:

$$F(k, l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) e^{\iota 2\pi \left(\frac{ki}{N} + \frac{lj}{N} \right)}, \quad (2.1)$$

onde $f(i, j)$ é a imagem no domínio espacial e o termo exponencial é a função base correspondente a cada ponto $F(k, l)$ no espaço de Fourier, tendo $\iota = \sqrt{-1}$. De forma análoga, a imagem de Fourier pode ser transformada novamente no domínio espacial, sendo a transformada inversa de Fourier dada por:

$$f(i, j) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k, l) e^{\iota 2\pi \left(\frac{ki}{N} + \frac{lj}{N} \right)}. \quad (2.2)$$

Observe o termo de normalização $\frac{1}{N^2}$ na transformação inversa que pode ser aplicada à transformação direta (2.1) em vez da transformação inversa (2.2), mas não deve ser usada para ambas.

Como a DFT é composta por um somatório duplo sobre cada ponto da imagem em análise, ela pode ser escrita como nas fórmulas:

$$F(k, l) = \frac{1}{N} \sum_{j=0}^{N-1} P(k, j) e^{-\iota 2\pi \frac{lj}{N}}, \quad (2.3)$$

onde

$$P(k, j) = \frac{1}{N} \sum_{i=0}^{N-1} f(i, j) e^{-\iota 2\pi \frac{ki}{N}}. \quad (2.4)$$

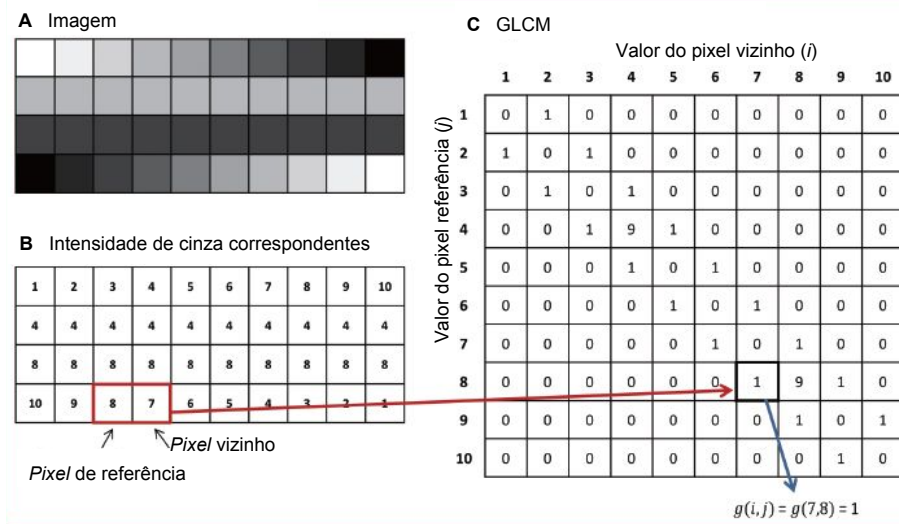
Desta forma, a imagem de entrada é transformada inicialmente em uma imagem intermediária usando N Transformadas de Fourier unidimensionais. Este resultado é então transformado na imagem do domínio da frequência, novamente usando N Transformadas de Fourier unidimensionais. Tal fragmentação reduz o custo computacional de uma DFT, contudo uma vez trocando as DFTs unidimensionais por Transformada Rápida de Fourier (FFT) obtem-se uma redução da complexidade computacional de $O(N^2)$ (referente a uma DFT unidimensionais) para $O(N \log_2 N)$. Esta é uma melhoria significativa, em particular para imagens grandes. Na literatura existem várias formas de FFT e em geral restringe o tamanho da imagem de entrada que pode ser transformada, muitas vezes para $O(N = 2^n)$, onde $n \in \mathbb{Z}$ (VOLNA EVA KOTYRBA, 2015).

2.1.1.2 Matriz de Coocorrência

A Matriz de Coocorrência apresentada por (HARALICK, 1973), conhecida também como GLCM (*Gray-Level Co-occurrence Matrix*), é um extrator de características de imagens para textura. Os parâmetros obtidos por ela são formas de caracterizar a posição relativa entre os pixels da região de interesse (Region of interest ou ROI) na imagem

selecionada (HUBER, 2000). A GLCM é um histograma bidimensional ($G \times G$) de níveis de cinza de uma imagem $f(x, y)$ e tamanho $M \times N$ da ROI que descreve a ocorrência de pares de pixels de valores $\{i, j\} \in [0, 1, \dots, G - 1]$, separados por uma distância d , em uma determinada direção α (AL-JANOBI, 2001) e dependente da quantidade de tons de cinza (Figura 1). De acordo com o estudo de CLAUSI, a quantidade de níveis de cinza é um

Figura 1 – Exemplo de uma GLCM $g(i, j)$ em **C**, obtida a partir da imagem $f(x, y)$ em **A** composta por dez variações de intensidade mapeadas em **B** com níveis de cinza claro ao escuro em ordem crescente. Com a distância d de um pixel e direção α em 0° . A matriz $g(i, j)$ resulta em 36 pares, sendo $g(7, 8) = 1$ um exemplo destacado em **C**.



Fonte: Adaptado de (MATTONEN, 2014).

importante fator na GLCM. Em princípio, quanto maior a quantidade maior é a exatidão da informação da textura extraída da imagem (CLAUSI, 2002).

A partir da GLCM obtida dentro da ROI escolhida na imagem, alguns parâmetros estatísticos podem ser obtidos:

a) **Contraste (*con*)**

O *con* é a medida da dispersão ou da variabilidade local da imagem. Imagens com muitas variações locais de níveis de cinza tendem a fornecer valores elevados de contraste. O *con* é dado por:

$$con = \sum_{i,j} (i - j)^2 \cdot m(i, j). \quad (2.5)$$

b) **Desvio padrão (*std*)**

O *std*, também chamado de grau de espalhamento dos níveis de cinza entre pares de pixels, é dado por:

$$std = \sqrt{1/G^2 \sum_{i,j} \langle m(i, j) - \bar{m} \rangle^2}. \quad (2.6)$$

onde G é o número de níveis de cinza presentes na imagem e m o valor médio da matriz $m(i, j)$.

c) **Segundo momento angular** (asm)

O asm mede a homogeneidade na transição entre níveis de cinza na textura da imagem. Uma maior homogeneidade ou um padrão regular de textura implica geralmente valores elevados de asm . O asm é obtido por:

$$asm = \sum_{i,j} m(i, j)^2. \quad (2.7)$$

d) **Entropia** (coo)

A coo mede quão aleatória é a textura da imagem. Valores elevados de coo indicam uma distribuição aleatória de níveis de cinza da GLCM. A coo pode ser expressa por:

$$coo = - \sum_{i,j} m^2(i, j) \cdot \log[m^2(i, j)]. \quad (2.8)$$

e) **Correlação** (cor)

A cor é a medida da dependência local entre os níveis de cinza em uma textura. Valores elevados de cor são obtidos para regiões com elevada similaridade entre os níveis de cinza. A cor pode ser expressa por:

$$cor = \frac{\sum_{i,j} i \cdot j \cdot m(i, j) - m_x m_y}{\sigma_x \cdot \sigma_y}. \quad (2.9)$$

Onde

$$\bar{m}_x = \sum_i i * (\sum_j m(i, j)|_{i=0,...,G-1}) \text{ e } \bar{m}_y = \sum_j j * (\sum_i m(i, j)|_{j=0,...,G-1}),$$

$$\sigma_x^2 = \sum_i (\sum_j m(i, j)|_{i=0,...,G-1}) * (i - \bar{m}_x)^2 \text{ e}$$

$$\sigma_y^2 = \sum_j (\sum_i m(i, j)|_{j=0,...,G-1}) * (j - \bar{m}_y)^2.$$

2.1.1.3 Padrões Binários Locais

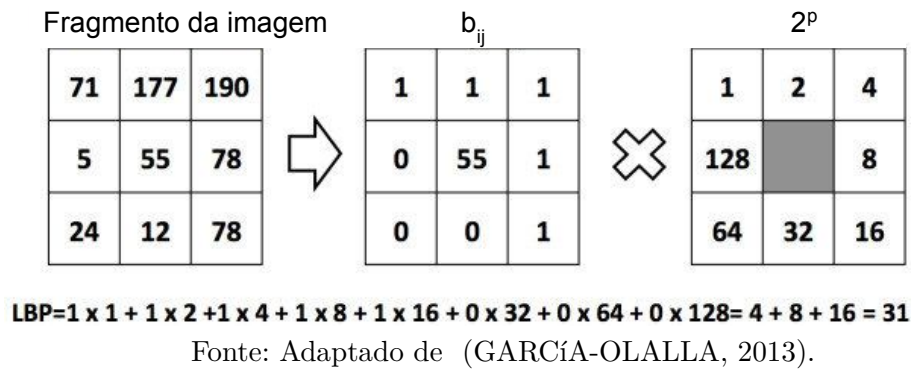
Os Padrões Binários Locais (*Local Binary Patterns* ou LBP) foram propostos em (OJALA, 1994) como um método simples e eficiente para descritores de texturas. Ele atribui um número binário para cada pixel verificado (g_c). Para atribuir esse padrão binário, é analisado se cada vizinho do pixel excede, ou não, um limiar (*threshold*) determinado pelo valor do pixel central.

Cada pixel vizinho que ultrapassar ou equivaler ao *threshold* é atribuído o valor binário 1, caso contrário, recebe o valor 0 e a junção desses dígitos representa padrão binário (b_{ij}) relacionado ao pixel central g_c . Neste ponto é importante o uso de um algoritmo para determinar qual é o dígito menos significativo (*Least Significant Bit* ou LSB), ou seja, a

partir de que vizinho o binário começará a ser formado, tendo em vista que a vizinhança é circular.

Torna-se notório que ao determinar o LSB está sendo atribuído um peso a cada um dos dígitos, como exemplificado na Figura 2. O somatório do produto de cada dígito pelo seu respectivo peso retornará o que é referenciado como código LBP.

Figura 2 – Tomando uma vizinhança 3×3 ($P = 8$ e $R = 1$) sobre um pixel arbitrário com a escala de cinza equivalente a 55, obtém-se o padrão $b_{ij} = 11111000_2$ pela comparação do pixel v_c com seus vizinhos. O padrão binário segue a ordem dos bits de conversão do menos significativo ao mais significativo apresentados na terceira matriz, gerando o valor 31_{10} computado no histograma LBP.



Desta forma, considerando-se uma imagem em escala de cinza I . Seja g_c o valor de intensidade de um pixel arbitrário $g_c = I(x, y)$, g_p para $p = 0, \dots, P - 1$ denota o valor de intensidade na vizinhança espacial circular, de raio R e composta por P vizinhos, entorno de g_c . O rótulo LBP de padrão de intensidade da vizinhança espacial do pixel central g_c é

$$LBP_{P,R}(x, y) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (2.10)$$

onde a vizinhança é composta por P pixels e $s(\cdot)$ é a função degrau:

$$s(x) = \begin{cases} 1, & \text{se } x \geq 0 \\ 0, & \text{se } x < 0 \end{cases} \quad (2.11)$$

e pode ser generalizado para qualquer vizinhança circular de P pixels e com raio R , onde o pixel g_p é então definido como:

$$g_p = I(x_p, y_p) \text{ para } p = 0, \dots, P - 1 \quad (2.12)$$

em que os valores de g_p , que não estão exatamente no centro de grades em I , são estimados por interpolação (HE, 2010).

$$x_p = x_c - R \sin\left(\frac{2\pi p}{P}\right) \quad (2.13)$$

$$y_p = y_c + R \cos\left(\frac{2\pi p}{P}\right) \quad (2.14)$$

sendo x_c e y_c as coordenadas do pixel central.

Novamente segundo (HE, 2010), uma das propriedades mais importante do operador LBP é a invariância de iluminação. Essa característica é percebida quando acontecem mudanças monotônicas nos níveis de cinza, que podem ser causadas por mudanças na iluminação do ambiente, na aquisição da imagem. Outra análise importante é a simplicidade computacional, que proporciona retornos rápidos, ideais para tarefas cujos tempos de resposta sejam de criticidade elevada.

Em um padrão binário LBP, há 2^P possibilidades de números binários diferentes que são formados diante da escolha do dígito LSB, alternando o mapeamento de pesos para cada pixel vizinho. É certo que alguns destes pixels terão mais importância que outros, sendo possível perceber que essa representação é sensível à rotação da imagem de entrada (SCHAEFER, 2017).

Diante desse problema, uma possível solução é executar um deslocamento bit a bit circular no sentido horário sobre cada número binário formado por P vezes e selecionado o menor padrão gerado. Desta forma, quantificando estatísticas de ocorrências individuais dos padrões invariantes a rotação, correspondendo a certos recursos na imagem de entrada. Estes padrões podem ser considerados como detectores de recursos (OJALA, 2002).

Nesta sessão foram apresentados três extratores de características distintos que provem bases de atributos ricas e variação e comportamentos auxiliando na generalização do treinamento dos classificadores que serão descritos logo adiante.

2.1.2 Classificadores de Padrões

Neste capítulo serão abordados métodos de classificação de padrões estado da arte, explorando desde suas naturezas probabilísticas às equações fundamentadas no calculo da distancia dos atributos de cada exemplo em análise e por fim, entendendo as estruturas de supervisão orientadas em camadas de aprendizagem.

2.1.2.1 Bayes

Em um desafio de rotular o dados desconhecido $y \in \{Y_1, Y_2, \dots, Y_k\}$ de um conjunto de características $x = (x_1, \dots, x_D) \in \mathbb{R}^D$, os classificadores Bayesianos aprendem o modelo da probabilidade conjunta $P(x, y)$ com os dados de treinamento e prediz o rótulo y mais provável através da regra de Bayes calculando $P(y|x)$. Ou seja, aplicando a regra de Bayes, a probabilidade a posteriori $P(y = Y_k|x)$ para o rótulo Y_k pode ser definida na equação 2.15 (BISHOP, 2007)

$$P(y = Y_k|x) = \frac{P(x|y = Y_k)P(y = Y_k)}{\sum_{j=1}^K P(x|y = Y_j)P(y = Y_j)}, \quad (2.15)$$

onde Y_k consiste no k – ésimo valor possível para y e o somatório no denominador é sobre todos os valores de y . A distribuição $P(y)$ fornece a probabilidade a posteriori de y e $P(x|y)$ modela a função de densidade de probabilidade condicional do rótulo, logo a probabilidade de x dado y . Essas estimativas são utilizadas na regra de Bayes para determinar um dado $P(y|x_z)$ para uma amostra sem rótulo x_z .

O classificador Bayesiano gera uma função que atribui um rótulo b_y à amostra x_z com base na regra de classificação apresentada na Equação 2.16

$$\hat{y} = \arg \max_{Y_k} P(y = Y_k | x), k \in \{1, \dots, K\}, \quad (2.16)$$

conhecida como máximo a posteriori (SERANG, 2012). Essa regra de decisão pode ser simplificada, ao desconsiderar o denominador em 2.15, tomando uma dada amostra x_z em que esse termo seja constante. Ou seja, a regra de decisão rotula uma dada amostra através da probabilidade a posteriori máxima. Uma das versões mais comumente apresentada é a Gaussiana ou Normal. A generalização multivariada de uma distribuição Gaussiana no espaço D – dimensional para um rótulo Y_k é apresentada em 2.17.

$$P(x|y = Y_k) = \frac{1}{(2\pi)^{\frac{D}{2}} |\sum_k|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (x - \mu_k)^T \sum_k^{-1} (x - \mu_k) \right\}, \quad (2.17)$$

em que μ_k é o vetor de médias e $\sum_k$ a matriz de covariância de tamanho $D \times D$. Logo, $|\sum_k|$ e $\sum_k^{-1}$ são o determinante e a inversa da matriz de covariância, respectivamente.

2.1.2.2 k-Nearest Neighbor

O classificador K-ésimo vizinho mais próximo (K-Nearest Neighbor - kNN) é um dos métodos de aprendizagem de máquinas mais fundamentais e simples, sendo uma excelente primeira escolha para uma investigação de classificação quando há pouco ou nenhum conhecimento prévio sobre a distribuição dos dados. O kNN foi desenvolvido a partir da necessidade de realizar uma análise discriminante quando estimativas paramétricas confiáveis de densidade de probabilidade são desconhecidas ou difíceis de determinar (STEINBACH, 2009).

Esta estratégia funciona da seguinte forma: dado um grupo de treinamento A e um objeto de teste B , o qual é um vetor de valores de atributos e tem uma classe desconhecida, o algoritmo computa a distância (ou similaridade) entre um objeto de entrada qualquer contida em B e todos os objetos de treinamento para determinar sua lista de vizinhos mais próximos e em seguida atribuir uma classe para tal entrada, tendo em conta a classe que é maioria entre os vizinhos selecionados.

A similaridade pode ser calculada por funções de distância como exemplo:

- Euclidiana

$$d = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (2.18)$$

- Mahalanobis

$$d = \sum_{i=1}^k |x_i - y_i| \quad (2.19)$$

- Minkowski

$$d = \left(\sum_{i=1}^k (|x_i - y_i|)^q \right)^{\frac{1}{q}} \quad (2.20)$$

O valor de k representa a quantidade de objetos que compõem a vizinhança e quando o valor dele é muito pequeno, o resultado pode ser sensível a pontos ruidosos existentes no conjunto de dados, e se o valor de K for muito grande, então a vizinhança pode incluir muitos pontos de outras classes, podendo levar a predição de uma classe que não é a real para determinada instância (STEINBACH, 2009).

2.1.2.3 Multi Layer Perceptron

As redes neurais de múltiplas camadas (Multi Layer Perceptron - MLP) são uma evolução do Perceptron de Camada Única publicado em (ROSENBLATT, 1958), possibilitando tratar problemas não linearmente separáveis, o que o torna uma vantagem sobre o classificador de camada única.

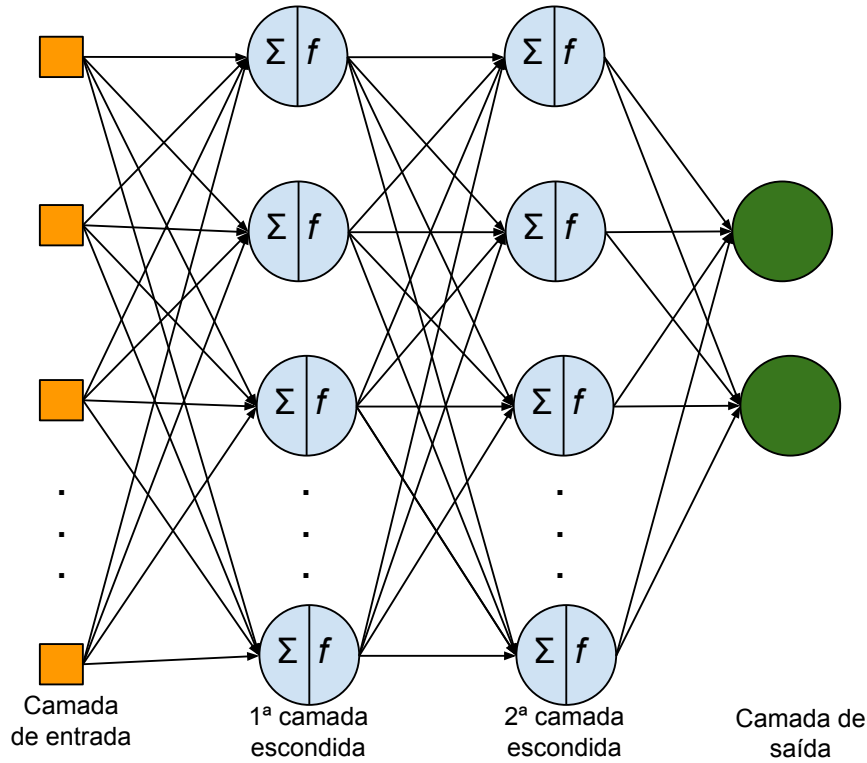
Os estudos de (CYBENKO, 1989) denotam a generalização na aproximação de funções contínuas em redes neurais de aprendizagem (RNA) alimentadas por uma camada escondida, utilizando função de ativação sigmoideal, e com uma camada de saída utilizando função linear dos neurônios. Na pesquisa de (BISHOP, 1995), a relação linear do número de parâmetros a serem ajustados na RNA em função do número de variáveis da função que se deseja aproximar tende a maximizar a capacidade de generalização.

Ambos trabalhos mostraram a viabilidade do uso de RNA como aproximadores universais de funções. Uma rede MLP é subdividida em camadas: camada de entrada, camada(s) intermediária(s) ou escondida(s) e camada de saída. A Figura 3 mostra o grafo arquitetural de uma rede MLP com duas camadas escondidas e uma camada de saída. Tipicamente, os neurônios em cada camada da rede têm como suas entradas apenas os sinais de saída da camada precedente, constituindo a característica feedforward desta rede neural. O conjunto de sinais de saída da camada de saída (final) constitui a resposta global da rede para o padrão de ativação fornecido pelos nós de fonte da camada de entrada (HAYKIN, 2001). Pode-se dizer também que esta rede é estritamente do tipo alimentada adiante (feedforward) ou acíclica. Um perceptron de múltiplas camadas tem três características distintas (HAYKIN, 2001):

$$y_k = \frac{1}{1 + \exp(-v_k)} \quad (2.21)$$

1. O modelo de cada neurônio (ver Apêndice A) da rede inclui uma função de ativação não-linear, sendo a não-linearidade diferenciável em qualquer ponto. Uma forma

Figura 3 – Estrutura de uma rede MLP com N neurônios na camada de entrada, duas camadas escondidas com N neurônios cada e dois neurônios na camada de saída.



Fonte: Produção do próprio autor.

normalmente utilizada de não-linearidade que satisfaz esta exigência é uma não-linearidade sigmoide¹ definida pela função logística 2.21. onde v_k é a soma ponderada de todas as entradas sinápticas acrescidas do *bias* do neurônio k , e y_k é a saída do neurônio. A existência de não-linearidades é importante pois, do contrário, a relação entrada-saída da rede poderia ser reduzida aquela de um perceptron de camada única (CYBENKO, 1989). A escolha da função logística tem motivação biológica, pois procura levar em conta a fase refratária de neurônios reais.

2. A rede contém uma ou mais camadas de neurônios escondidos, que não são parte da entrada ou da saída da rede. São estes neurônios escondidos que capacitam a rede aprender tarefas complexas extraindo progressivamente as características mais significativas dos padrões (vetores) de entrada.
3. A rede exibe um alto grau de conectividade, determinado pelas sinapses (ou elos de conexões) da rede. Uma modificação na conectividade da rede é obtida por uma mudança no número de sinapses ou de seus pesos.

¹ As funções sigmoides são chamadas assim porque seus gráficos apresentam a forma de “s”.

2.1.2.4 OPF

O Classificador Floresta de Caminhos Ótimos (Optimum Path Forest - OPF) é uma evolução da Transformada de Florestas da Imagem (IDT) apresentada em (FALCÃO, 2004). Fazendo uso de conceitos de teoria dos grafos o OPF em sua versão original apresenta um proposta de classificação supervisionada.

Um classificador supervisionado é construído a partir de uma fase inicial de treinamento, na qual é fornecido um conjunto de amostras representativas de cada uma das classes que se pretende caracterizar. Posteriormente, o classificador deve ser capaz de determinar a qual classe pertence uma nova amostra apresentada.

Este classificador permite rotular dados não linearmente separáveis, com amostras espacialmente dispersas (ALPAYDIN, 2009). O algoritmo OPF foi utilizado também em (ROCHA, 2009) e (CAPPABIANCO, 2012) como base de um algoritmo classificador não supervisionado.

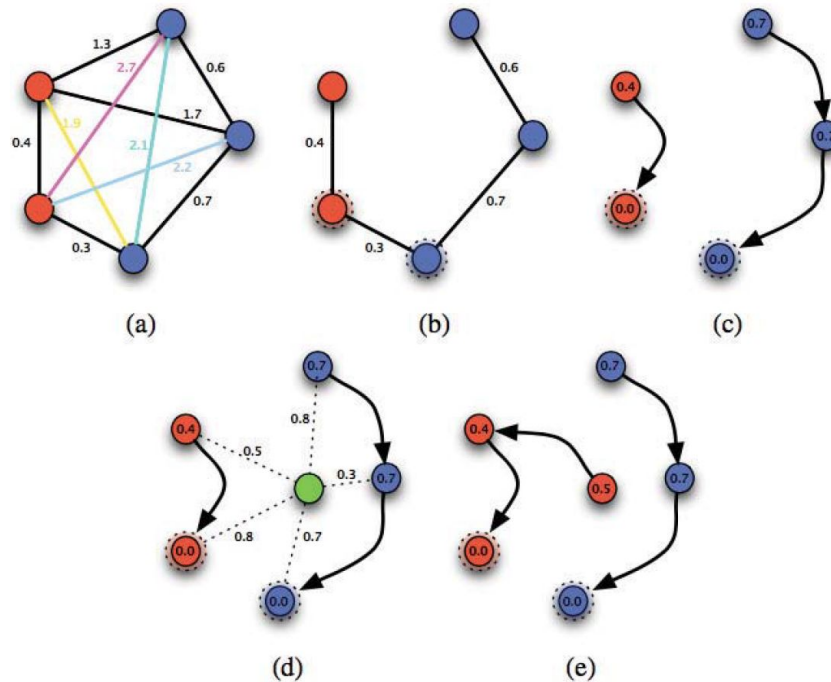
No caso do classificador OPF, o conjunto de amostras de treinamento é modelado através de um grafo rotulado no qual cada amostra é representada por um vértice como em Figura 4(a). Estes vértices são organizados em uma floresta, isto é, um conjunto de árvores. Cada árvore da floresta contém unicamente vértices de um mesmo rótulo, Figura 4(c), sendo que para uma mesma classe podem existir varias árvores. Desta forma, cada classe é caracterizada por conjunto de árvores dentro da floresta construída.

Para construir o conjunto de árvores representativas de cada rótulo, este classificador aplica um processo de maximização do mapa de conectividade entre amostras do mesmo rótulo. Neste processo, cada vértice recebe um valor que caracteriza o custo da sua conectividade com seu grupo, apresentado na Figura 4(b). Este valor está associado ao caminho de menor custo desse vértice com algum vértice protótipo da sua árvore. Os vértices protótipos são aqueles dentro da árvore que são os mais próximos aos vértices de outro rótulo.

Um detalhe importante do OPF é que um rótulo é caracterizado por estruturas sem restrições espaciais. O conjunto de árvores de um rótulo pode estar espacialmente separado, ou seja, não formando um único agrupamento espacial. Este conjunto de árvores de um rótulo também pode ter formatos diferentes, sendo assim, os diferentes rótulos não precisam, por exemplo, ser linearmente separáveis.

Para classificar uma nova amostra, procura-se em toda a floresta o vértice para o qual o custo de conectividade com a nova amostra seja menor, Figura 4(d). A classe desse vértice é atribuída à nova amostra, Figura 4(e).

Figura 4 – (a) Conjunto de treinamento modelado como um gráfico completo; (b) Um cálculo da árvore de abrangência mínima sobre o conjunto de treinamento em que protótipos são destacados; (c) floresta de caminho ótimo sobre o conjunto de treinamento; (d) processo de classificação de uma amostra desconhecida; e (e) a amostra de teste é finalmente classificada.



Fonte: Adaptado de (SILVA, 2015).

2.1.2.5 Support Vector Machine

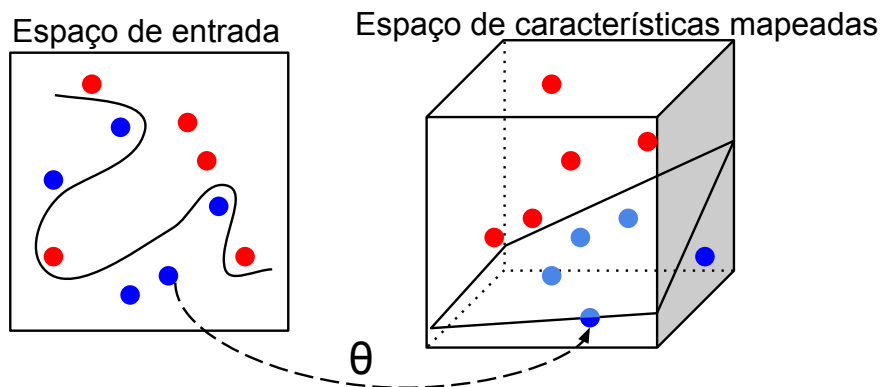
As Máquinas de Vetores de Suporte (Support Vector Machine - SVM) resolvem o problema de classificação de padrões assumindo ser possível separar os rótulos em um espaço de mais alta dimensão e está fundamentada na teoria *Vapnik-Chervonenkis*. (VAPNIK, 1963)

Suponha um conjunto de dados não linearmente separáveis. Tais amostras podem ser classificadas em um conjunto binário de rótulos usando curvas ou um hiperplano como superfícies de decisão, sendo que encontrar tais limiares é o maior desafio desta abordagem.

O princípio de um classificador SVM é pré-processar os dados de tal maneira que o problema de encontrar uma função discriminante não linear seja transformado em um problema de encontrar um hiperplano, ou seja, mapear os dados que estão em uma dimensão qualquer para outra maior, procurando tornar os mesmos linearmente separáveis. Isto é feito definindo um mapeamento o qual transforma o vetor de entrada em um conjunto maior (Figura 5) e espera-se que, escolhendo um mapeamento adequado, o novo conjunto de treinamento seja linearmente separável.

Embora isso permita uma boa assertividade, o custo computacional no treinamento é sensível ao tamanho do subconjunto de treino e ao número de vetores de suporte

Figura 5 – Exemplo de pré-processamento de um SVM, em que θ é a função discriminante que mapear os dados na entrada em um espaço de características.



Fonte: Produção do próprio autor.

na classificação (MAZZONI, 2006). (NETO, 2013), apresenta um método que tem como principal objetivo diminuir o conjunto de treinamento, selecionando os dados de treinamento a partir da similaridade no processamento de dois algoritmos de quantização vetorial, no contexto de classificações binárias.

Ainda que em (MAZZONI, 2006) tenha reduzido o número de vetores de suporte no caso de várias classes, a abordagem apresentou demora na convergência e complexidade elevada, visto que o trabalho propõe-se a resolver esse problema minimizando o número de vetores de suporte em várias SVMs binárias e depois compartilhando os sub-conjuntos de vetores de suporte de cada uma dessas máquinas.

Embora SVM tenha sido projetado, inicialmente, para ser um classificador binário, múltiplas SVMs são necessárias para resolver problemas com várias classes (CRAMMER, 2001). Na literatura encontram-se três abordagens principais:

1. Um contra todos (*One Versus All* - OVA)

Que utiliza SVMs para separar cada rótulo dos outros. A decisão é dada para o rótulo com maior valor de confiança.

2. Um contra um (*One Versus One* - OVO)

Esta abordagem requer SVMs que levam em consideração todas as combinações binárias entre as classes. A decisão é dada pela maioria de votos.

3. Gráfico acíclico dirigido (*Directed Acyclic Graph* - DAG)

Semelhante ao OVO, o DAG requer um SVM para cada par de rótulos diferentes. No entanto, o DAG reduz em um o número de classificadores invocados na fase de previsão fazendo uso de uma árvore de decisão binária. Em que cada nível, o

processo de predição segue o ramo esquerdo ou direito do nó atual, dependendo do resultado do classificador nesse nó.

Em (HSU, 2002) compararam muitas técnicas multi-classe SVM sobre a função kernel Gaussiano (*Radial-Basis Function* - RBF). Através de testes comparativos, os autores mostraram que OVO e DAG são mais preciso e apresentou menos tempo de treinamento e teste que o OVA.

2.2 Aprendizagem Profunda

2.2.1 Convolutional Neural Networks

Redes neurais convolucionais (Convolutional Neural Networks - CNN ou ConvNets) são redes neurais artificiais em que se aplica a operação de convolução em pelo menos uma de suas camadas. Esta estratégia é indicada para um conjunto particular de problemas em que cada amostra de uma base de dados segue uma topologia estrutural. Nessa topologia, considera-se a existência de uma relação entre valores de atributos próximos em uma representação da amostra. Mais detalhes estão presentes no **Apêndice A**.

A seguir serão apresentados dois algoritmos dentre os sete que se destacaram nos desafios de reconhecimento visual de grande escala da ImageNet (*ImageNet Large Scale Visual Recognition Challenge* - ILSVRC) (RUSSAKOVSKY et al., 2015). Nesta competição é avaliado o desempenho dos algoritmos com desafios de localização e detecção de objetos a partir de imagens, vídeos e análise de cenas em escala, sendo a ImageNet uma vasta base de dados bastante explorada na pesquisa de visão computacional (DENG et al., 2009).

2.2.1.1 VGGNet

A VGGNet é uma ConvNet de alto desempenho com destaque no ILSVRC em 2014, sendo bastante aplicada devido à estrutura simples e bom desempenho. Ela utiliza um tamanho de 224×224 como imagem de entrada, a qual sofre um pré-processamento, em que o valor médio do canal RGB de todo o conjunto de treinamento é subtraído por cada pixel da imagem de entrada para centralizar a entrada no zero (SIMONYAN; ZISSERMAN, 2014).

ConvNets em geral usam filtros grandes, no entanto a VGGNet usa o menor filtro (3×3 , $s = 1$ e $p = 1$) em todas as camadas, podendo usar várias funções não lineares ReLU. Como uma camada de filtros grandes são divididos em várias camadas de filtros menores, a não linearidade ReLU tem mais possibilidades. Desta forma a decisão se torna mais discriminativa.

O número de pesos para ajustar, no treinamento, é bastante reduzido. Em analogia, suponha que se tem três camadas convolucionais 3×3 e uma camada convolucional 7×7 . Neste contexto, ambas as camadas possuem canais C . calculando-se o número de parâmetros em duas camadas. O número de parâmetros na camada convolucional da 7×7 é de $7^2 \times C^2 = 49C^2$. As três camadas convolucionais 3×3 são $3 \times (3^2 \times C^2) = 27C^2$. Desta forma se o número de parâmetros a serem ajustados é menor, o uso de filtros menores se torna mais vantajoso em termos de regularização.

Em (SIMONYAN; ZISSERMAN, 2014) é apresentado também a aplicação de uma camada convolucional 1×1 para a rede de canais C . Provendo uma noção inicial de que um filtro 1×1 não altera o dado de entrada na camada de saída, contudo este filtro realiza uma transformação linear de canais por produto de ponto entre canais. Neste momento, os canais de entrada e saída são definidos para serem os mesmos. Ao criar uma camada convolucional de 1×1 em uma rede, haverá mais espaço para que a ReLU se torne uma rede mais discriminativa.

Recursos adicionais do VGGNetwork incluem:

- Uma camada 2×2 Max-pool ($s = 2$) foi usada como a camada de *pooling*.
- Camada densamente conectada (*fully connected layer* ou *fc*) é $4096 - 4096 - 1000$ e a última camada usa *softmax*.
- Todas as camadas usam a não linearidade de ReLU.
- Resultados experimentais mostram que a melhoria desta rede conciste no uso da memória e tempo de computação.

O VGGNet requer cerca de 93MB de memória no estágio *Forward* para cada imagem e quase 200MB de memória no estágio de backprop. O número total de parâmetros é de aproximadamente 138MB. Em suma a maior parte da memória vem da camada conv da frente, e a maioria dos parâmetros aumenta a partir da camada *fc*.

2.2.1.2 ResNet

A Rede Residual foi desenvolvida por (HE et al., 2015) e vencedora do ILSVRC de 2015. Esta rede possui como ideia central a composição de uma ConvNet *feed-forward* (ou propagação direta) padrão, incrementada de conexões que ignoram (*bypass*) algumas camadas de convolução. Cada *bypass* dá origem a um bloco residual no qual as camadas de convolução preveem um resíduo que é adicionado ao tensor (estrutura de dados transportador de atributos pelas camadas) de entrada do bloco. As redes convolucionais profundas tendem a sofrer com a dificuldade de otimização. Além de uma certa profundidade, adicionar camadas extras resulta em maior erro de treinamento e maior erro de validação, mesmo quando a normalização de lote é usada. Na ResNet, os autores julgam que é improvável que esse *underfitting* seja causado por desaparecimento do gradiente, pois essa dificuldade ocorre mesmo com redes normalizadas por lotes. A arquitetura de rede residual resolve isso

adicionando conexões de *bypass* que são somadas com a saída das camadas de convolução.

2.2.2 Transferência de Aprendizagem

No trabalho de (MENEGOLA et al., 2017), para prover redução no tempo do treinamento completo de redes convolucionais (ou *fine-tuning*) e suprir a carência de grandes bases de dados para treinamento de um novo modelo, é apresentado o conceito de ré-aproveitamento do conhecimento previamente treinado ou transferência de aprendizagem entre modelos de inferência.

Tal abordagem consiste no uso dos vetores de características gerados por uma ConvNet previamente treinada com seus pesos sinápticos configurados para reconhecer e extrair características de um determinado conjunto de dados para o qual foi previamente treinado. Esses vetores gerados serão os descritores de cada entrada presente na base de dados.

Após gerados esses vetores de características do novo conjunto do qual se deseja fazer a classificação, esses vetores podem ser utilizados como entrada para um outro classificador para que seja treinado com esses vetores ou podem ser dados como entrada para a camada completamente conectada de classificação da própria arquitetura que foi utilizada para extração das características (YOSINSKI et al., 2014).

Em analogia, o processo de transferência de conhecimento é como um observador que está vendo um objeto e começa a descrever os elementos presentes neste objeto para outra pessoa, onde esta pessoa não está vendo o objeto descrito. A partir das características relatadas pelo observador o ouvinte irá traçar um perfil dos elementos que ele conhece e das características recebidas da primeira pessoa e associar a um objeto que ele conhece e que possua características semelhantes.

Como observado em (MENEGOLA et al., 2017) e (HOO-CHANG et al., 2016) a transferência de conhecimento provê bons resultados e é uma alternativa muito vantajosa, pois dispensa o tempo que seria gasto treinando uma rede neural robusta do zero para um domínio específico.

2.3 Métricas de Avaliação

Com objetivo de validar o desempenho de algoritmos de classificação, cinco métricas de avaliação são empregadas na literatura e definidas a seguir: acurácia, sensibilidade, especificidade, valor preditivo positivo e F1-score. Essas medidas são obtidas a partir da matriz de confusão (Figura 6). Para melhor entendimento da matriz de confusão, aplica-se ao contexto do problema o desafio de localização de um VANT, fazendo com que cada rótulo represente o cenário que se pretende identificar na navegação da aeronave.

- Verdadeiro Positivo (VP): corresponde ao número de vezes que o robô é localizado corretamente pelo modelo de classificação.
- Falso Negativo (FN): corresponde ao número de vezes que a localização em análise é prevista erroneamente como outra localização pelo modelo de classificação.
- Falso Positivo (FP): corresponde ao número de vezes que outra localização é prevista erroneamente como a localização em análise pelo modelo de classificação
- Verdadeiro Negativo (VN): corresponde ao número de vezes que outra localização é prevista corretamente pelo modelo de classificação

A matriz de confusão que resume os rótulos previstos corretamente ou incorretamente é apresentada na Figura 6. Evidenciando que para cada coordenada desta matriz obtém-se uma contagem de ocorrências entre de 0 ao numero de inferências realizadas relacionado às marcações VP, FN, FP e VN. Por fim, observa-se que o número de rótulos, determina a ordem da matriz.

Figura 6 – Matriz de confusão para um problema de classificação com 24 rótulos.

		Rótulo Previsto					
		1	2	3	4	...	24
Rótulo Real	1	VP	FN	FN	FN	FN	FN
	2	FP	VP	FN	FN	FN	FN
	3	FP	FP	VP	FN	FN	FN
	4	FP	FP	FP	VP	FN	FN
	:	FP	FP	FP	FP	VP	FN
	24	FP	FP	FP	FP	FP	VP

Fonte: Produção do próprio autor.

A acurácia (Ac) é a proporção de predições corretas. Essa medida é influenciada pelo desbalanceamentos do conjunto de dados e pode facilmente induzir a uma conclusão errada sobre o desempenho do sistema.

$$Ac = \frac{totaldeacertos}{totaldeamostras} \quad (2.22)$$

A sensibilidade (Se) é a proporção de verdadeiros positivos: a capacidade do sistema em prever corretamente a localização para casos verdadeiro.

$$Se = \frac{VP}{VP + FN}, \quad (2.23)$$

em que VP e FN representam o número de amostras de um determinado rótulo corretamente e incorretamente classificados, respectivamente.

A especificidade (Es) é a proporção de verdadeiros negativos: a capacidade do sistema em prever corretamente as outras localizações para os casos que realmente estão nelas.

$$Es = \frac{VN}{VN + FP}. \quad (2.24)$$

em que VN representa o número de amostras que não pertencem a um determinado rótulo classificado como não pertencentes ao rótulo considerado, enquanto FP representa o número de amostras classificadas incorretamente como pertencendo a um determinado rótulo.

O valor preditivo positivo (VPP) mede a proporção de verdadeiros positivos em relação a todas as predições positivas.

$$VPP = \frac{VP}{VP + FP}. \quad (2.25)$$

O VPP e a sensibilidade podem ser resumidas em outra métrica conhecida como a F-Score, que é a média harmônica ponderada desses valores. Na fórmula F-Score, o coeficiente de ponderação do VPP é a sensibilidade.

$$F - Score = 2 \cdot \frac{VPP \cdot Se}{VPP + Se}. \quad (2.26)$$

Nessa dissertação, o balance-weighted F-Score, conhecido como F1-Score, é utilizado. Essa fórmula varia o valor de resultado no intervalo entre 0 e 1, sendo que atinge o seu melhor valor em 1 e pior resultado em 0.

$$F1 - Score = 2 * \frac{VPP * Se}{VPP + Se}. \quad (2.27)$$

3 Proposta e Metodologia

Nesse capítulo são apresentados os equipamentos adotados assim como os cenários e locais utilizados na navegação do VANT. Para tanto faz-se necessária apresentar as rotas aplicadas ao estudo assim como a base de imagens gerada para o mesmo.

3.1 Equipamento Utilizado

Neste trabalho foi selecionado um VANT comercialmente disponível para efetuar aquisição de imagens no ambiente de testes. Trata-se de um quadricóptero modelo Bebop do fabricante Parrot, que pode ser observado na Figura 7, e com especificações disponíveis na Tabela 3.

Tabela 3 – Especificação do VANT Parrot Bebop PF722000.

Conectividade	
WiFi	802.11a/b/g/n/ac
SDK	(DLOUHÝ, 2015)
Antenas	Duas antenas de banda dual para 2.4 e 5 GHz
Câmera	
Sensor	CMOS 14Mpx
Resolução de Streaming	1700 x 1070 pixels
Vídeo encode	H264
Memória interna	Flash 8 GB

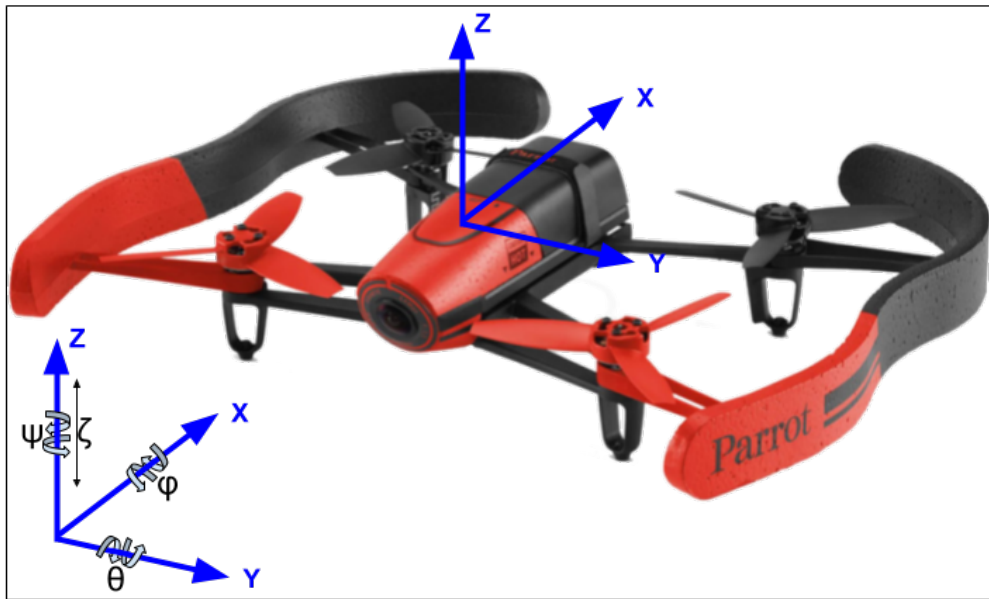
Destas especificações do VANT, duas foram cruciais para seleção deste equipamento, sendo a comunicação via WiFi e a biblioteca de interface com os comando de voo da aeronave. Tais características permitiram a interação do veículo com uma estação de controle externa.

Como estação de controle externa, foi utilizado o mesmo notebook em que foram realizados os treinamentos dos extratores, classificadores e ConvNets. No qual também ocorreu a aquisição do banco de imagens via *streaming* de cada cenário do mapa topológico. O provisionamento está apresentado na Tabela 4.

Tabela 4 – Especificação da estação de controle externa.

MacBook Pro	
Processador	2,3GHz Intel Core i5
Memória RAM	16GB
GPU	NVIDIA Geforce 9400M 256MB
Sistema Operacional	MacOS Sierra 10.12.6

Figura 7 – Bebop PF722000



Fonte: Adaptação do próprio autor.

3.1.1 Quadrirotores

Aeronaves do tipo quadrirotores possuem quatro hélices (ou propulsores) dispostos sobre motores, dois que efetuam rotação no sentido horário e os outros no sentido anti-horário. O voo é controlado pela variação na velocidade de giro em cada motor, possibilitando manobras como arfagem (θ), guinada (ψ), movimento vertical (ζ) e rolagem (ϕ), se assemelhando aos movimentos de helicópteros (MOHANTY; MOHANTY, 2018). A Figura 7 mostra o sistema de coordenadas destas manobras.

3.2 Rotas Predefinidas

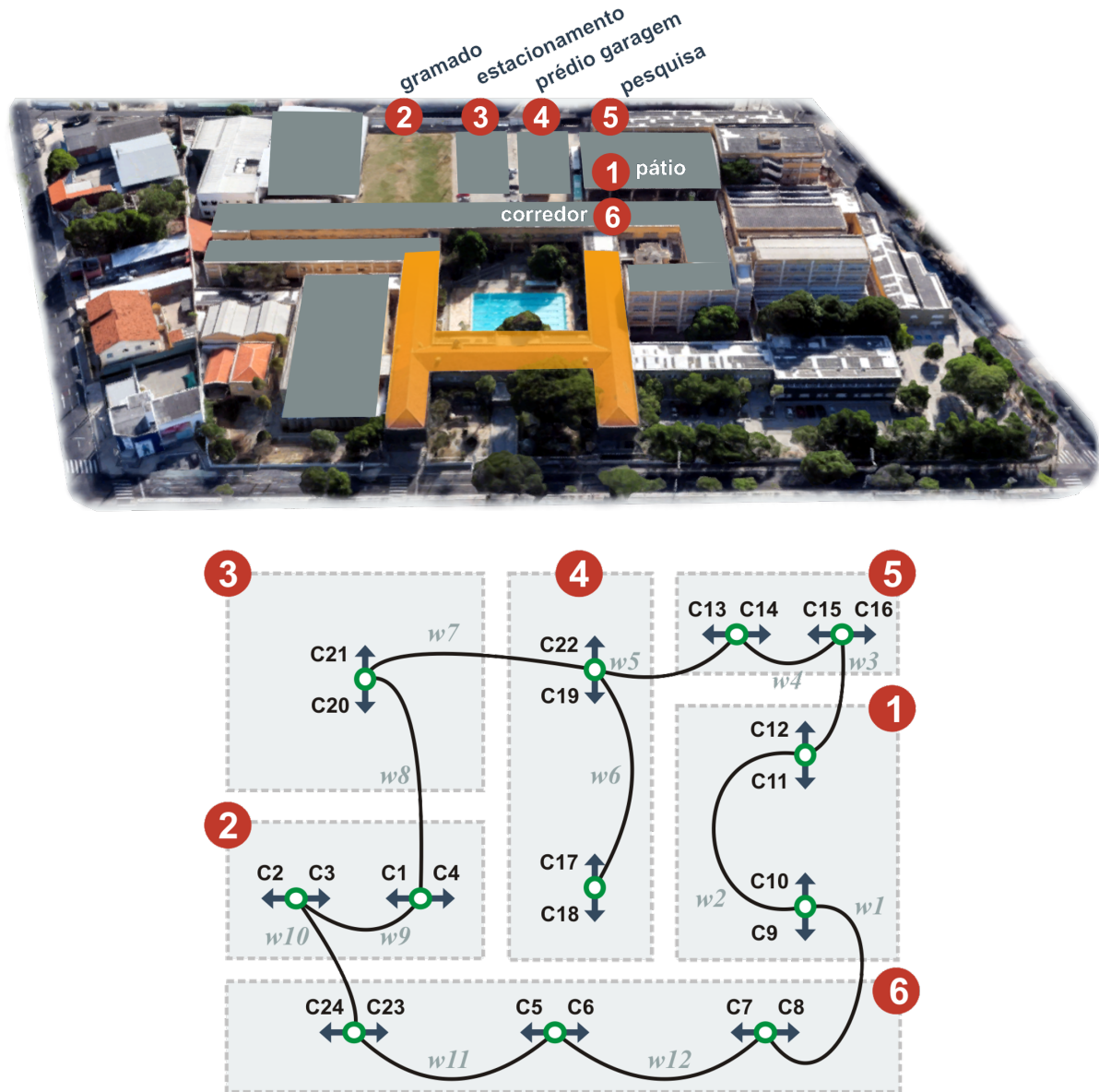
A abordagem consiste em obter um *streaming* de imagens de um VANT comercialmente disponível em operação, identificar o posicionamento do mesmo dentre os lugares pré-rotulados e treinados em um mapa topológico por uma estação de controle remota, e por fim, executar manobras que permitam à aeronave se deslocar de uma localização de partida para uma outra de destino executando a rota de voo.

Estas rotas foram definidas em um ambiente sem a existência de uma base de imagens em abundância que no caso se trata do campus do IFCE ¹ em Fortaleza (Figura 8) explorando ambientes abertos como quadras e pátios; estreitos como corredores e garagens; e dinâmicos como áreas comunais de alunos. Tais imagens foram coletadas em horários e

¹ O IFCE foi escolhido pela conveniência de ser o endereço do laboratório responsável pelo equipamento utilizado nesta pesquisa, além de possuir bastante diversidade nas áreas internas, provendo segurança e a variedade de cenários necessária para execução dos experimentos.

em períodos diferentes, para ter alteração da luminosidade solar e do transito de pessoas do local.

Figura 8 – Campus do IFCE e locais mapeados.



Fonte: Produção do próprio autor.

Nesta pesquisa o VANT navega em ambiente misto com espaços internos e externos, em que o mapa topológico é composto de grupos de nós representando cada uma das seis localizações. Estes nós (marcações em verde na Figura 8) foram realizados voos e captura de imagens em *streaming* do cenário com o posicionamento em questão, sendo ao todo vinte e quatro cenários ($C1, C2, \dots, C24$) que identificam a orientação para a localização em cada nó.

Como exemplo, a aeronave pode estar localizada no primeiro nó do patio coberto

no Bloco de Pesquisa (Local 5) com a câmera posicionada para o segundo nó (cenário C14) e com uma guinada de 180° o VANT estará em posicionamento C13 podendo se dirigir à Garagem do campus (local 4).

O mapa topológico elaborado nesta dissertação é capaz de identificar a orientação de cada cenário e possibilitar a navegação para um novo cenário em qualquer uma das seis localizações registradas por meio das arestas que interligam os nós ($w_1, w_2, \dots, w_{12}$). A Figura 8 também apresenta o mapa topológico utilizado nos arranjos de extratores e classificadores.

3.3 Base de imagens

Ao todo foram capturadas quatro mil e duzentas imagens por meio de voos com o Beebop nos ambientes evidenciados na Sessão 3.2, distribuídas de forma equivalente em vinte e quatro rótulos (240 imagens por cenário) de localização e posicionamento. Todas nas resolução 1700×1070 pixels. A Tabela 5 mostra as variações de períodos e horários que melhor diversificaram a aquisição das imagens devido a variação de luminosidade solar e interferência dinâmica de pessoas nos cenários.

Tabela 5 – Base de imagens por período e horário.

Luminosidade	Periodo	Quantidade de imagens
Alta	Manhã	1000 para ambientes com pessoas
		1100 para ambientes sem pessoas
Baixa	Fim de tarde	1000 para ambientes com pessoas
		1100 para ambientes sem pessoas

Fonte: Próprio autor.

3.4 Etapa de Extração de Características

Antes de extrair as características todas as imagens passaram por um pré-processamento de corte de tamanho de 1700×1070 para 889×500 pixels, o qual melhor se adequou aos algoritmos de extração adotados.

Dentre os algoritmos de extração de atributos existentes na literatura, foram selecionadas duas categorias de métodos consolidados baseadas em textura (GLCM e LBP) e na frequência, Fourier. Os extratores Fourier, GLCM e LBP foram aplicados em todas as imagens obtidas na aquisição da base de dados.

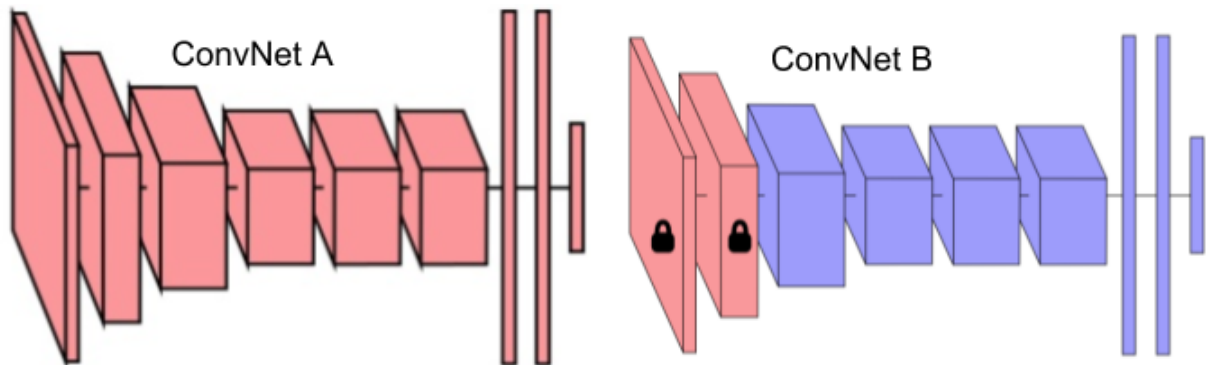
Para a GLCM, adotou-se a distância $D = 1$ e direção $\theta = 0$. No LBP, utilizou-se a versão com padrões uniformes com objetivo de diminuir a dimensão do problema. Em

seguida, as amostras obtidas nesta fase foram utilizadas no processo de treino e testes dos classificadores.

3.5 Extração de Características e CNNs

As extrações de características foram conduzidas, aplicando o conceito de transferência de aprendizagem já explorada na Sessão 2.2.2 (e resumida na Figura 9), no qual as imagens de cenários foram processadas pelas camadas convolucionais de cada ConvNet selecionada. A camada densamente conectada não foi aplicada neste processo e na última camada convolucional o *feature map* resultante do *Max Pool* (Apêndice A) foi transformado em um vetor unidimensional que será aplicado na entrada dos classificadores adotados nesta pesquisa.

Figura 9 – A **ConvNet A** foi treinada com a base de dados Imagenet (DENG et al., 2009), sendo que os pesos convolucionais foram congelados e transferidos para a **ConvNet B**, reduzindo o treinamento da mesma à atualização dos pesos da camada densamente conectada.



Fonte: Adaptado de (YOSINSKI et al., 2014).

Foram consideradas as redes neurais convolucionais que se destacaram nos desafios da ILSVRC (RUSSAKOVSKY et al., 2015) como: VGG(16 e 19), MobileNet, ResNet50, InceptionV3, Xception, DenseNet(121 e 201) e NASNetLarge. Todas estas ConvNets foram pré-treinadas com o banco de dados ImageNet (DENG et al., 2009).

A Tabela 6 mostra o número de características gerado por cada CNN.

3.6 Etapa de Classificação

De forma similar aos extratores de características, foram selecionados diferentes tipos de classificadores com objetivo de encontrar a melhor configuração para a tarefa de navegação e localização de VANTs com visão computacional. E dentre as categorias de técnicas existentes, utilizou-se o classificador Bayesiano, o qual é baseado em probabilidade e estatística, o classificador embasado pela Teoria de Aprendizado, o SVM, a rede neural

Tabela 6 – Quantidade de características extraídas em cada ConvNet.

ConvNet	Número de características
VGG16	512
VGG19	512
MobileNet	1024
ResNet50	2048
InceptionV3	2048
Xception	2048
DenseNet121	1024
DenseNet201	1920
NASNetLarge	4032

artificial MLP, o classificador baseado em distância, kNN e por fim baseado na Teoria dos Grafos, OPF. Todas as codificações dos classificadores e extratores clássicos foram adquiridos no *framework* scikit-learn v0.19.1 (BUITINCK et al., 2013).

A validação cruzada aplicada na base de amostras consistiu na divisão aleatória em dois grupos, 80% para o subconjunto de treinamento e 20% para o subconjunto de teste, seguindo o método *holdout*, aferindo a generalização de um modelo. O conjunto de treino foi normalizado (média zero e variância unitária) e os conjuntos de teste também foram normalizados utilizando as mesmas medidas de normalização que o conjunto de treinamento.

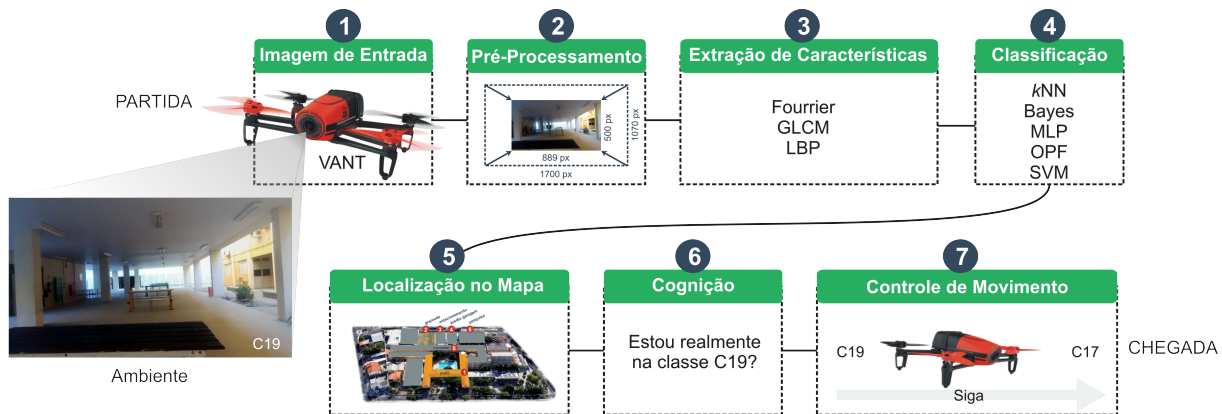
A divisão das classes foi balanceada em ambos conjuntos de treino e teste. Utilizou-se o kernel linear para o classificador SVM. A rede MLP foi treinada com um intervalo de neurônios na camada oculta de 1 a 50. Adotou-se a função de densidade de probabilidade Gaussiana (Normal) no classificador Bayesiano. kNN e OPF foram configurados com a distância Euclidiana e o número de vizinhos k foi variado no intervalo de 1 à 5.

3.7 Rotas de Navegação

Para realização dos teste de navegação, foram elaboradas dez rotas distintas, tomando como referencia o mapa topológico apresentado anteriormente. Os comandos usados para mover VANT, assim como, os pontos iniciais e finais de localização de cada rota são apresentados na Tabela 7.

Com objetivo de auxiliar o entendimento da navegação, a primeira rota é exemplificada na Figura 10. Nesta rota, o VANT inicia na posição c11 da localização 1 e recebe dois comandos para seguir em frente. Depois disso, ele vira (guinada) à direita e recebe novamente dois comandos para seguir em frente. Por fim uma guinada à direita, um

Figura 11 – Fluxo adotado para navegação do VANT nesta pesquisa.



Fonte: Produção do próprio autor.

3.8 Abordagem da Navegação Proposta

A Figura 11 mostra a sequência de passos adotados na abordagem proposta. Em que se inicia com a recepção da imagem via *streaming* com a aeronave (passo 1). Em seguida, durante os passos 2 ao 6 as imagens são processadas na estação de controle externa conectada via WiFi ao VANT, sendo os dados pré-processados, extraídos características e classificados. Por fim, após a manipulação dos dados pela estação de controle, o VANT recebe pela WiFi a sequência de comandos para a execução das manobras a fim chegar na posição seguinte da rota analisada.

3.9 Avaliação de Desempenho

Na avaliação dos resultados obtidos nos testes e análise de desempenho dos algoritmos extração e classificação, da abordagem proposta, foram aplicadas cinco métricas: especificidade, sensibilidade, valor preditivo positivo, acurácia e *F1-Score*. Estes valores são obtidos a partir da matriz de confusão e foram melhor aprofundados anteriormente na Sessão 2.3 desta dissertação.

4 Experimentos e Resultados

Neste capítulo são apresentados os experimentos executados norteados pela motivação e objetivos apresentados no capítulo introdutório, bem como a análise dos resultados de cada arranjo e, por fim, aferindo o custo computacional de cada metodologia a fim de melhor validar a abordagem da proposta.

Os testes são basicamente exploratórios e foram conduzidos com a intenção de avaliar a eficácia e eficiência de cada arranjo aplicado à tarefa de navegação de VANTs, sendo calculados através de dez voltas em cada uma das dez rotas descritas na metodologia, totalizando cem realizações na avaliação da classificação.

Propositamente, estes testes foram divididos em dois experimentos, afim de se observar os desempenhos dos algoritmos clássicos separadamente da abordagem com ConvNets, pois estas ultimas fazem uso de paralelismo com GPUs, tornando o ambiente de experimento diferente da abordagem clássica e reduzindo bastante o tempo de performasse.

Os resultados foram analisados usando as métricas de avaliação citadas na sessão anterior e podem ser observados na Tabela 8, evidenciando valores médios e desvios padrões das mesmas.

Tabela 8 – Especificidade (Sp), Sensibilidade (Se), Valor Preditivo Positivo (PPV) e Acurácia (Acc) obtida pelos arranjos de ferramentas de extração e classificação propostos, acompanhadas do desvio padrão à direita.

Extrator	Classificador	Sp (%)	Se (%)	PPV (%)	Acc (%)
Fourier	Bayes(Normal)	99,591±0,024	91,802±0,471	92,589±0,373	99,223±0,045
	kNN	99,718±0,021	94,155±0,415	94,122±0,427	99,438±0,039
	MLP	97,122±0,761	42,392±15,202	42,712±16,209	94,512±1,447
	OPF(Euclidiano)	99,806±0,028	95,993±0,554	95,976±0,566	99,617±0,053
	SVM(Linear)	99,611±0,024	92,226±0,475	92,167±0,491	99,259±0,045
	Bayes(Normal)	99,951±0,009	98,971±0,192	98,973±0,191	99,905±0,018
	kNN	99,839±0,012	96,500±0,242	96,502±0,236	99,672±0,023
	MLP	97,613±0,226	52,137±4,528	51,361±3,525	95,436±0,431
	OPF(Euclidiano)	99,915±0,009	98,253±0,198	98,254±0,199	99,831±0,019
	SVM(Linear)	99,918±0,007	98,249±0,135	98,241±0,137	99,835±0,013
LBP	Bayes(Normal)	99,984±0,006	99,619±0,119	99,615±0,118	99,958±0,011
	kNN	99,972±0,006	99,458±0,115	99,453±0,115	99,951±0,011
	MLP	99,983±0,008	99,532±0,161	99,538±0,162	99,964±0,015
	OPF(Euclidiano)	99,987±0,006	99,699±0,123	99,694±0,122	99,969±0,012
	SVM(Linear)	99,991±0,005	99,774±0,069	99,776±0,069	99,985±0,007

Fonte: Próprio autor.

De acordo com a Tabela 8, o LBP se destaca nesta abordagem por obter o melhor resultado entre todas as métricas apresentadas. De fato, o arranjo do extrator LBP com o

classificador SVM apresentou sempre o maior valor retornado dos testes, obtendo 99,991% de capacidade de determinar se o cenário em análise não é à localização e posição desejada, 99,774% para identificar o cenário corretamente, 99,776% na precisão em identificar o cenário corretamente em relação as proposições de predições corretas e 99,985% na realização de proposições de predições corretas.

Ainda, na Tabela 8, é possível ver que o arranjo Fourier combinado com o MLP apresentam desempenho inferiores, tendo as seguintes métricas: 97,122% de especificidade, 42,392% de sensibilidade, 42,712% de PPV e 94,512% de acurácia. Não sendo, aparentemente, uma alternativa confiável para essa tarefa e navegação a frente dos demais arranjos analisados e tomando como critério o acurácia.

O problema analisado nesta pesquisa apresenta características de resposta em tempo real, tornando importante observar o custo computacional dos arranjos analisados. Logo, os tempos de resposta para as etapas em cada arranjo foram obtidas e analisadas. Esses valores são indispensáveis na aplicação desta abordagem em ambientes embarcados. Sendo assim, a Tabela 9 apresenta os valores de tempo de treino, tempo de teste e tempo de extração para os arranjos de ferramentas de extração e os classificadores.

Tabela 9 – Média dos tempos de treino, de teste e de extração obtidos pelos arranjos de ferramentas de extração e classificação propostos, acompanhado pelos respectivos desvios padrões à direita..

Classificador	Tempo de treino (s)	Tempo de teste (ms)	Tempo de extração (s)
Fourier			
Bayes(Normal)	0,01551±0,00184	55,96±55,94	18,49±2,84
kNN	0,00017±0,00012	432,57±432,37	
MLP	22,26473±6,45220	0,01±0,01	
OPF(Euclidiano)	0,24186±0,01320	143,40±143,33	
SVM(Linear)	14,24819±5,53269	20,73±20,72	
GLCM			
Bayes(Normal)	0,00601±0,00448	34,30±34,28	11,77±8,67
kNN	0,00007±0,00002	345,74±345,58	
MLP	13,04043±3,79468	0,01±0,01	
OPF(Euclidiano)	0,13773±0,03154	80,30±80,26	
SVM(Linear)	5,13358±1,21721	9,53±9,53	
LBP			
Bayes(Normal)	0,06373±0,00299	80,76±80,72	17,15±13,21
kNN	0,00039±0,00026	1321,10±1320,47	
MLP	2,74550±1,03566	0,01±0,01	
OPF(Euclidiano)	0,64364±0,32249	361,82±361,64	
SVM(Linear)	5,49787±1,22487	80,91±80,87	

Fonte: Próprio autor.

De acordo com a Tabela 9, kNN apresenta o menor tempo de treino independente do extrator combinado com o mesmo. Sendo uma comportamento esperado dado a natureza do treinamento do mesmo, consistindo no armazenamento das amostras de treino. Os valores apresentados são: $70\mu s$ para o GLCM, $170\mu s$ para o Fourier e $390\mu s$ para LBP. O classificador que apresentou o tempo mais prolongado de treinamento foi o MLP nos arranjos GLCM e Fourier, apresentando tempos de treinamento, respectivamente, 13,04043s e 22,26473s. Vale, ainda, ressaltar que a combinação SVM com LBP termina o estágio de treinamento em 5,49787s.

Para os resultados dos tempos de teste, pode-se destacar o MLP como sendo o mais rápido, obtendo um tempo de $10\mu s$ independentemente do extrator utilizado no arranjo. O maior valor de tempo observado foi na combinação utilizando o classificador kNN, realizando o procedimento de teste em 345,74ms, 432,57ms e 1,3211s, em combinação respectivamente, com GLCM, Fourier e LBP.

Pode-se evidenciar, ainda, na Tabela 9 que o algoritmo de GLCM foi quem cumpriu a tarefa de extração de característica mais rapidamente, com o tempo de 11,77s. Por outro lado, O extrator Fourier apresentou o tempo mais prolongado de extração, sendo 18,49s. Note que os resultados mencionados anteriormente foram destacados em ambas as tabelas, seguindo a legenda de verde e vermelho para melhores e piores escores respectivamente.

Como o foco deste mestrado está na navegação do VANTs, é importante salientar que o treinamento dos classificadores ocorre somente na concepção de um modelo de inferência de cenários. Desta forma, os estágios envolvidos na navegação são somente a extração de características e o teste de classificação. Logo sendo destacado em azul na Tabela 9 o arranjo GLCM com SVM por apresentar melhor tempo de extração e o segundo melhor tempo de teste, com 11,77s e 9,53ms, sendo acurácia do mesmo 0,15% menor do que a melhor combinação destacada na Tabela 8.

4.1 Experimento com as ConvNets

Os experimento com as redes neurais convolucionais foram conduzidos com a formação de arranjos das ConvNets e os classificadores. Em que as camadas convolucionais substituíram os extratores clássicos e as camadas densamente conectadas foram trocadas pelos cinco classificadores clássicos adotados nesta pesquisa. As implementações das ConvNets foram obtidas no *framework* Keras 2.1.4 (CHOLLET et al., 2015), assim como a codificação dos classificadores foram novamente adquiridos no *framework* scikit-learn v0.19.1 (BUTINCK et al., 2013).

A validação cruzada também fora adotada neste caso, em que os cenários foram particionados aleatoriamente em dez grupos, quatro quintos para treinamento e o restante para teste, em cada uma das dez iterações. O conjunto de treinamento foi então normalizado

(média zero e variação unitária) e os conjuntos de teste também normalizado utilizando as mesmas medidas de normalização do conjunto de treinamento. As proporções das classes foram mantidas equilibradas em ambos os conjuntos de treino e teste. (RODRIGUEZ; PEREZ; LOZANO, 2010)

A Tabela 10 apresenta os valores médios e os desvios padrões da acurácia e do *F1-Score*. Analisando esta tabela, pode-se concluir que a maioria das combinações alcançaram valores maiores que 99%, em ambas as métricas. Em que as melhores Acc e F1S foram obtidas nas arquiteturas VGG19, MobileNet, InceptionV3, Xception, DenseNet121 e DenseNet201, com um valor de 99,97% para ambas as métricas em combinação com diferentes classificadores.

Entre os classificadores o MLP se destaca por estar associado com todas essas ConvNets vencedoras, neste experimento. Em contraste, a menor acurácia e o menor *F1-Score* obtidas pela rede ResNet50 em combinação com o classificador Bayesiano, com valores de 93,18% e 93,06%, respectivamente, logo 6,79% menos confiável em relação aos melhores arranjos.

É importante notar que a ResNet50 foi uma das segundas maiores na quantidade de características apresentados na Tabela 6, elevando a ocorrência de sobre-ajustes (ou *overfitting*) dos classificadores, podendo justificar a perca de generalização observada neste caso. (SHI, 2018; PARTAOURIDES, 2018) Os melhores resultados da Tabela 10 estão destacados em verde, assim como o pior está em vermelho.

Como foi abordado anteriormente, a análise do custos computacionais é essencial para esta pesquisa. Portanto, a Tabela 11 mostra os tempos de treinamento, de teste e de extração de características. De acordo com esta tabela, o classificador Bayesiano alcançou os menores tempos no treino, em conjunto com a maioria das ConvNets. Os valores de 0,094s, 0,117s, 0,192s, 0,208s, 0,328s e 0,366s foram obtidos quando o Bayes estava associado com VGG16, VGG19, DenseNet121, MobileNet, InceptionV3 e ResNet50, respectivamente.

O classificador que obteve o maior tempo de treinamento foi o OPF, obtendo o valor de 2011,031s quando combinado com a CNN Xception. Ainda avaliando a Tabela 11, quanto ao tempo de teste, o classificador Bayesiano foi novamente o mais rápido em combinação com todas as arquiteturas de CNN, realizando essa etapa em 4 μ s quando associado a VGG16 e VGG19 e em 6 μ s em conjunto com MobileNet, DenseNet121 e NASNetMobile, as melhores marcas do mesmo. O classificador mais lento na fase de classificação foi kNN, quando associado à NASNetLarge, retornando um tempo de teste de 27,990ms.

Analisando a última coluna da Tabela 11, pode-se observar que a CNN MobileNet foi a mais rápida no cumprimento de sua tarefa de extração, obtendo uma marca de 88,33s.

Tabela 10 – Acurácia (Acc) e $F1-Score$ (F1S) obtidos pelos arranjos de extratores de características e classificadores.

ConvNet	Classificador	Acc (%)	F1S (%)
VGG16	Bayes(Normal)	99,10±0,27	99,09±0,27
	kNN	99,96±0,06	99,96±0,06
	MLP	99,96±0,06	99,96±0,06
	OPF(Euclidean)	99,94±0,09	99,94±0,09
	SVM(Linear)	99,96±0,06	99,96±0,06
	Bayes(Normal)	99,15±0,23	99,15±0,23
VGG19	kNN	99,96±0,09	99,96±0,09
	MLP	99,97±0,06	99,97±0,06
	OPF(Euclidean)	99,97±0,06	99,97±0,06
	SVM(Linear)	99,97±0,06	99,97±0,06
	Bayes(Normal)	97,83±0,54	97,81±0,55
MobileNet	kNN	99,96±0,06	99,96±0,06
	MLP	99,97±0,06	99,97±0,06
	OPF(Euclidean)	99,96±0,06	99,96±0,06
	SVM(Linear)	99,97±0,06	99,97±0,06
	Bayes(Normal)	93,18±0,64	93,06±0,66
ResNet50	kNN	99,92±0,09	99,92±0,09
	MLP	99,83±0,16	99,83±0,16
	OPF(Euclidean)	99,89±0,10	99,89±0,10
	SVM(Linear)	99,94±0,07	99,94±0,07
	Bayes(Normal)	97,65±0,40	97,62±0,41
InceptionV3	kNN	99,94±0,07	99,94±0,07
	MLP	99,97±0,06	99,97±0,06
	OPF(Euclidean)	99,89±0,12	99,89±0,12
	SVM(Linear)	99,96±0,06	99,96±0,06
	Bayes(Normal)	97,33±0,64	97,34±0,65
Xception	kNN	99,94±0,09	99,94±0,09
	MLP	99,97±0,09	99,97±0,09
	OPF(Euclidean)	99,85±0,15	99,85±0,14
	SVM(Linear)	99,97±0,06	99,97±0,06
	Bayes(Normal)	97,07±0,46	97,05±0,46
DenseNet121	kNN	99,97±0,08	99,97±0,08
	MLP	99,97±0,06	99,97±0,06
	OPF(Euclidean)	99,94±0,09	99,94±0,09
	SVM(Linear)	99,97±0,06	99,97±0,06
	Bayes(Normal)	98,25±0,37	98,24±0,38
DenseNet201	kNN	99,97±0,06	99,97±0,06
	MLP	99,97±0,07	99,97±0,07
	OPF(Euclidean)	99,96±0,06	99,96±0,06
	SVM(Linear)	99,97±0,06	99,97±0,06
	Bayes(Normal)	95,58±0,60	95,56±0,60
NASNetLarge	kNN	99,89±0,12	99,89±0,12
	MLP	99,86±0,12	99,86±0,12
	OPF(Euclidean)	99,82±0,21	99,82±0,21
	SVM(Linear)	99,90±0,11	99,90±0,11

Fonte: Próprio autor.

No entanto, o maior tempo de extração foi retornado pela DenseNet201, com um valor de 830,58s. Os melhores resultados desta tabela foram destacados em verde e tempos mais elevados foram destacados em vermelho.

Tabela 11 – Acurácia (Acc), tempo de treino, tempo de teste e tempo de extração obtidos pelos classificadores e extratores de características respectivamente.

Classificador	Acc (%)	Tempo de treino(s)	Tempo de teste(ms)	Tempo de extração(ms)
CNN - VGG16				
Bayes(Normal)	99,10±0,27	0,094±0,016	0,004±0,000	
kNN	99,96±0,06	0,391±0,059	2,343±0,040	
MLP	99,96±0,06	144,715±21,454	0,096±0,008	142,72±0,10
OPF(Euclidean)	99,94±0,09	719,578±3,243	0,918±0,006	
SVM(Linear)	99,96±0,06	16,431±0,629	1,329±0,018	
CNN - VGG19				
Bayes(Normal)	99,15±0,23	0,117±0,023	0,004±0,000	
kNN	99,96±0,09	0,425±0,078	2,059±0,033	
MLP	99,97±0,06	142,092±16,449	0,092±0,005	198,90±0,22
OPF(Euclidean)	99,97±0,06	1024,026±1,473	0,891±0,005	
SVM(Linear)	99,97±0,06	15,200±0,949	1,233±0,027	
CNN - MobileNet				
Bayes(Normal)	97,83±0,54	0,208±0,034	0,006±0,001	
kNN	99,96±0,06	1,209±0,067	12,312±0,274	
MLP	99,97±0,06	132,106±16,050	0,142±0,019	88,33±0,34
OPF(Euclidean)	99,96±0,06	771,108±0,761	1,046±0,004	
SVM(Linear)	99,97±0,06	53,990±1,800	3,993±0,019	
CNN - ResNet50				
Bayes(Normal)	93,18±0,64	0,366±0,103	0,014±0,000	
kNN	99,92±0,09	2,023±0,361	1,806±0,048	
MLP	99,83±0,16	866,698±134,928	0,240±0,037	186,18±0,04
OPF(Euclidean)	99,89±0,10	38,629±0,166	1,498±0,022	
SVM(Linear)	99,94±0,07	47,543±2,445	3,068±0,077	
CNN - InceptionV3				
Bayes(Normal)	97,65±0,40	0,328±0,085	0,012±0,002	
kNN	99,94±0,07	2,484±0,521	20,239±0,111	
MLP	99,97±0,06	294,088±40,820	0,153±0,024	269,43±0,08
OPF(Euclidean)	99,89±0,12	1566,315±1,326	1,258±0,008	
SVM(Linear)	99,96±0,06	148,204±24,955	5,944±0,080	
CNN - Xception				
Bayes(Normal)	97,33±0,64	0,467±0,068	0,012±0,001	
kNN	99,94±0,09	2,411±0,269	15,701±0,238	
MLP	99,94±0,09	101,329±12,414	0,091±0,015	252,63±0,53
OPF(Euclidean)	99,85±0,15	2011,031±3,627	1,237±0,025	
SVM(Linear)	99,97±0,06	98,338±1,919	4,733±0,048	
CNN - DenseNet121				
Bayes(Normal)	97,07±0,46	0,192±0,018	0,006±0,000	
kNN	99,97±0,08	0,990±0,198	7,689±0,182	
MLP	99,97±0,06	309,402±36,089	0,137±0,013	552,67±0,06
OPF(Euclidean)	99,94±0,09	174,098±0,937	1,128±0,022	
SVM(Linear)	99,97±0,06	36,676±2,935	2,843±0,061	
CNN - DenseNet201				
Bayes(Normal)	98,25±0,37	0,457±0,093	0,011±0,001	
kNN	99,97±0,06	1,849±0,049	11,323±0,222	
MLP	99,94±0,07	340,647±54,037	0,181±0,033	830,58±0,21
OPF(Euclidean)	99,96±0,06	1222,878±1,303	1,274±0,013	
SVM(Linear)	99,97±0,06	106,656±2,312	5,162±0,046	
CNN - NASNetLarge				
Bayes(Normal)	95,58±0,60	0,607±0,130	0,019±0,003	
kNN	99,89±0,12	2,644±0,342	27,990±0,260	
MLP	99,86±0,12	296,059±15,791	0,222±0,009	821,11±0,25
OPF(Euclidean)	99,82±0,21	101,787±0,265	1,421±0,012	
SVM(Linear)	99,90±0,11	219,342±4,464	10,048±0,100	

Fonte: Próprio autor.

5 Conclusão

Esta dissertação apresentou uma nova abordagem para a realização da navegação e localização de VANTs comerciais de pequeno porte. A solução apresentada aplica um mapa topológico com algoritmos de extração de características somados a técnicas de aprendizagem de máquinas em imagens obtidas a partir da câmera monocular embarcada na aeronave.

O ambiente selecionado para o experimento é composto tanto por áreas fechadas e estreitas quanto por ambientes abertos, permitindo a validação dos arranjos para ambos os tipos de posicionamento, sendo consideradas também as variações de luminosidade do ambiente e a interferência dinâmica de pessoas nos cenários.

Uma breve revisão bibliográfica foi apresentada, demonstrando o intenso esforço de investigação dedicada à navegação de VANTs com precisão ou autônomos, por parte da comunidade científica nos últimos anos. Características particulares da aplicação também foram evidenciadas, como exemplo sua limitada autonomia, sendo necessária a implementação de estruturas de algoritmos eficientes com baixo custo computacional.

Dessa forma a possibilidade de se realizar uma comunicação com uma estação de controle externo (*cloud*, *laptop* ou GPU) com a finalidade de tratar os dados enviados pelo VANT e posteriormente gerar tomadas de decisão quanto sua navegação e localização, se tornaram factíveis.

De acordo com os resultados apresentados das combinações de extratores clássicos e classificadores, foi possível verificar que a melhor métrica é alcançada quando usada a combinação de LPB com SVM, obtendo 99,99% de especificidade, 99,77% de sensibilidade e PPV e 99,98% de acurácia. Já os melhores tempos obtidos nas etapas de extração, treino e teste, foram verificados com os seguintes algoritmos, respectivamente: GLCM com 11,77s, kNN com 70 μ s e MLP com 10 μ s.

Percebe-se, portanto, que dentre os arranjos avaliados neste experimento a combinação do classificador SVM com o extrator GLCM é a melhor escolha para este desafio de localização e navegação considerando o tempo de resposta na identificação de cenários por câmera convencional embarcada no VANT em estudo, mesmo tendo atingido a acurácia de 0,15% a menos de que a melhor combinação.

Este trabalho também avaliou o desempenho da abordagem proposta com a utilização de ConvNets na etapa de extração de características, obtendo a acurácia e *F1-Score* de 99,97% com as arquiteturas VGG19, MobileNet, InceptionV3, Xception, DenseNet121 e DenseNet201 em combinação com os mesmos classificadores do primeiro experimento.

Destacando-se como melhor escolha arranjo da MobileNet com o classificador MLP, pois além de atingir a melhor métrica de precisão, o tempo de extração de características desta ConvNet foi o mais rápido com o valor de 88,33ms. De fato, as ConvNets aumentaram a eficácia e a confiabilidade na navegação do VANT com as marcas de tempo e acurácia atingidas, porém está solução arcar com o custo computacional elevado das camadas convolucionais.

A maior contribuição do trabalho encontra-se na avaliação de diferentes arranjos de extratores e classificadores na solução do problema de navegação de VANTs, podendo esses arranjos serem implementados de acordo com as especificações da aeronave que vá ser utilizada.

5.1 Trabalhos Futuros

Para trabalhos futuros, pode-se avaliar os melhores desempenhos dos arranjos apresentados em uma plataforma embarcado com uma GPU, verificando a eficácia dos arranjos em ambientes com baixa autonomia energética¹. Outra possível contribuição futura está na aplicação dos arranjos apresentados em outras classes de VANT (modelos profissionais ou linhas industriais), avaliando o desempenho dos algoritmos frente as particularidades de cada tipo de aeronave. Também se torna relevante a re-codificação dos extratores clássicos em paradigmas de paralelização com o uso de GPUs, desta forma podendo unificar as duas abordagens em um único experimento. Por fim, ainda pode-se propor, a implementação de um mapa topológico colaborativo particularmente nos arranjos que utilizam ConvNets, podendo um conhecimento adquirido por um VANT ser compartilhado com outras aeronaves na mesma formação de voo em enxame.

¹ Existem, no mercado, processadores dedicados a visão computacional, tal como Intel/Movidius Myriad 2 que viabiliza o processamento embarcado e ConvNets (MARANTOS et al., 2018).

Referências

AERONÁUTICA, B. C. da. Sistemas de aeronaves remotamente pilotadas e o acesso ao espaço aéreo brasileiro. Comando da Aeronáutica, 2016. Citado na página 27.

AGUILAR, W. G. et al. Developing of a video-based model for uav autonomous navigation. In: SPRINGER. *Latin American Workshop on Computational Neuroscience*. [S.l.], 2017. p. 94–105. Citado na página 29.

AL-JANOBI, A. Performance evaluation of cross-diagonal texture matrix method of texture analysis. *Elsevier Pattern Recognition*, v. 34, n. 1, p. 171–180, 2001. Citado na página 35.

ALPAYDIN, E. *Introduction to machine learning*. [S.l.]: MIT press, 2009. Citado na página 42.

BI, Y. et al. Robust autonomous flight and mission management for mavs in gps-denied environments. In: IEEE. *Control Conference (ASCC), 2017 11th Asian*. [S.l.], 2017. p. 67–72. Citado na página 30.

BISHOP, C. M. *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, Inc., 1995. ISBN 0198538642. Citado na página 40.

BISHOP, M. S. C. M. Pattern recognition and machine learning. 2007. Citado na página 38.

BUITINCK, L. et al. API design for machine learning software: experiences from the scikit-learn project. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. [S.l.: s.n.], 2013. p. 108–122. Citado 2 vezes nas páginas 56 e 61.

CAPPABIANCO, A. X. F. C. L. Y. J. K. U. F. A. Brain tissue mr-image segmentation via optimum-path forest clustering. *Computer Vision and Image Understanding*, Elsevier, v. 116, n. 10, p. 1047–1059, 2012. Citado na página 42.

CESETTI. A vision-based guidance system for uav navigation and safe landing using natural landmarks. *Journal of intelligent and robotic systems*, Springer, v. 57, n. 1-4, p. 233, 2010. Citado na página 29.

CHEN, B. Navigation and control of unmanned aerial vehicles in gps-denied environments. In: *13th International Conference on Control Automation Robotics & Vision, ICARCV 2014, Singapore, December 10-12, 2014*. [S.l.: s.n.], 2014. Citado na página 27.

CHIN. A neuro-based network for on-line topological map building and dynamic path planning. p. 2805–2810, 2017. Citado na página 29.

CHOLLET, F. et al. *Keras*. 2015. <<https://keras.io>>. Citado na página 61.

CLAUSI, D. A. An analysis of co-occurrence texture statistics as a function of grey level quantization. *Canadian Journal of remote sensing*, Taylor & Francis, v. 28, n. 1, p. 45–62, 2002. Citado na página 35.

- CRAMMER, Y. S. K. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, v. 2, n. Dec, p. 265–292, 2001. Citado na página 44.
- CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, Springer, v. 2, n. 4, p. 303–314, 1989. Citado 2 vezes nas páginas 40 e 41.
- DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2009. p. 248–255. ISSN 1063-6919. Citado 3 vezes nas páginas 15, 45 e 55.
- DLOUHÝ, M. *Katarina*. [S.l.]: GitHub, 2015. <<https://github.com/robotika/katarina>>. Citado na página 51.
- FALCÃO, J. S. R. de A. L. A. X. The image foresting transform: Theory, algorithms, and applications. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 26, n. 1, p. 19–29, 2004. Citado na página 42.
- GARCÍA-OLALLA, E. A. L. F.-R. M. G.-O. D. G.-O. O. Adaptive local binary pattern with oriented standard deviation (albps) for texture classification. v. 2013, 12 2013. Citado na página 37.
- GONZALEZ, R. C. W. R. C. *Processamento digital de imagens*. [S.l.]: Pearson Educação, 2009. Citado na página 33.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>. Citado na página 78.
- HARALICK, K. S. I. D. R. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, v. 3, n. 6, p. 610–621, 1973. Citado na página 34.
- HAYKIN, S. *Redes Neurais - 2ed.* [S.l.]: BOOKMAN COMPANHIA ED, 2001. ISBN 9788573077186. Citado na página 40.
- HE, K. et al. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <<http://arxiv.org/abs/1512.03385>>. Citado na página 46.
- HE, N. S. C. G. Y. Pyramid-based multi-structure local binary pattern for texture classification. In: SPRINGER. *Asian Conference on Computer Vision*. [S.l.], 2010. p. 133–144. Citado 2 vezes nas páginas 37 e 38.
- HOO-CHANG, S. et al. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, NIH Public Access, v. 35, n. 5, p. 1285, 2016. Citado na página 47.
- HSU, C.-W. C.-J. L. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, IEEE, v. 13, n. 2, p. 415–425, 2002. Citado na página 45.
- HUBEL, D. H.; WIESEL, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, Wiley Online Library, v. 195, n. 1, p. 215–243, 1968. Citado na página 77.

- HUBER, J. D. I. Z. J. T. M. M. S. D. S. Relevance of sonographic b-mode criteria and computer-aided ultrasonic tissue characterization in differential/diagnosis of solid breast masses. *Elsevier Ultrasound in Medicine Biology*, v. 26, n. 8, p. 1243–1252, 2000. Citado na página 35.
- KANG, D.; CHA, Y.-J. Autonomous uavs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging. *Computer-Aided Civil and Infrastructure Engineering*, Wiley Online Library, v. 33, n. 10, p. 885–902, 2018. Citado na página 30.
- KANIEWSKI, W. G. P. Visual-based navigation system for unmanned aerial vehicles. In: *2017 Signal Processing Symposium (SPSymposium)*. [S.l.: s.n.], 2017. p. 1–6. Citado na página 28.
- KOBAYASHI, T. et al. Wireless technologies to assist search and localization of victims of wide-scale natural disasters by unmanned aerial vehicles. In: IEEE. *Wireless Personal Multimedia Communications (WPMC), 2017 20th International Symposium on*. [S.l.], 2017. p. 404–410. Citado 2 vezes nas páginas 27 e 28.
- LACHKAR, T. G. R. B. L. D. E. M. A. Textile woven-fabric recognition by using fourier image-analysis techniques: Part i: A fully automatic approach for crossed-points detection. *The Journal of The Textile Institute*, Taylor Francis, v. 94, n. 3-4, p. 194–201, 2003. Disponível em: <<https://doi.org/10.1080/00405000308630608>>. Citado na página 33.
- LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998. Citado na página 77.
- LOQUERCIO. Dronet: Learning to fly by driving. *IEEE Robotics and Automation Letters*, IEEE, v. 3, n. 2, p. 1088–1095, 2018. Citado na página 30.
- MARANTOS, C. et al. Efficient support vector machines implementation on intel/movidius myriad 2. In: IEEE. *Modern Circuits and Systems Technologies (MOCASST), 2018 7th International Conference on*. [S.l.], 2018. p. 1–4. Citado na página 68.
- MARINHO, L. B. et al. A novel mobile robot localization approach based on topological maps using classification with reject option in omnidirectional images. *Expert Systems with Applications*, Elsevier, v. 72, p. 1–17, 2017. Citado na página 29.
- MATTONEN, K. H. A. D. W. S. S. D. A. P. S. A. New techniques for assessing response after hypofractionated radiotherapy for lung cancer. *Journal of Thoracic Disease*, v. 6, n. 4, 2014. ISSN 2077-6624. Disponível em: <<http://jtd.amegroups.com/article/view/2121>>. Citado na página 35.
- MAZZONI, B. T. D. Multiclass reduced-set support vector machines. In: ACM. *Proceedings of the 23rd international conference on Machine learning*. [S.l.], 2006. p. 921–928. Citado na página 44.
- MENEGOLA, A. et al. Knowledge transfer for melanoma screening with deep learning. In: IEEE. *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on*. [S.l.], 2017. p. 297–300. Citado na página 47.
- MOHANTY, M. D.; MOHANTY, M. N. Design of a quadcopter uav for flood area data analysis. In: IEEE. *2018 2nd International Conference on Data Science and Business Analytics (ICDSBA)*. [S.l.], 2018. p. 3–7. Citado na página 52.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. [S.l.: s.n.], 2010. p. 807–814. Citado na página 78.

NEMRA, A.; AOUF, N. Robust ins/gps sensor fusion for uav localization using sdre nonlinear filtering. *IEEE Sensors Journal*, IEEE, v. 10, n. 4, p. 789–798, 2010. Citado na página 27.

NETO, G. A. B. A. R. Opposite maps: Vector quantization algorithms for building reduced-set svm and lssvm classifiers. *Neural processing letters*, Springer, v. 37, n. 1, p. 3–19, 2013. Citado na página 44.

NGUYEN, V. N.; JENSSEN, R.; ROVERSO, D. Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 99, p. 107–120, 2018. Citado 2 vezes nas páginas 27 e 28.

OJALA, M. P. D. H. T. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. v. 1, p. 582–585, Oct 1994. Citado na página 36.

OJALA, M. P. T. M. T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 24, n. 7, p. 971–987, 2002. Citado na página 38.

PARTAOURIDES, C. *Bayesian inference techniques for Deep Learning*. Tese (Doutorado) — Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών και ..., 2018. Citado na página 62.

PETKOVICS, I. et al. Selection of unmanned aerial vehicle for precision agriculture with multi-criteria decision making algorithm. In: IEEE. *Intelligent Systems and Informatics (SISY), 2017 IEEE 15th International Symposium on*. [S.l.], 2017. p. 000151–000156. Citado na página 27.

RESHMA, R.; RAMESH, T.; SATHISHKUMAR, P. Security situational aware intelligent road traffic monitoring using uavs. In: IEEE. *VLSI Systems, Architectures, Technology and Applications (VLSI-SATA), 2016 International Conference on*. [S.l.], 2016. p. 1–6. Citado na página 27.

ROCHA, F. A. C. A. X. F. L. M. Data clustering as an optimum-path forest problem with applications in image analysis. *International Journal of Imaging Systems and Technology*, Wiley Online Library, v. 19, n. 2, p. 50–68, 2009. Citado na página 42.

RODRIGUEZ, J. D.; PEREZ, A.; LOZANO, J. A. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 32, n. 3, p. 569–575, 2010. Citado na página 62.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. Citado na página 40.

RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, v. 115, n. 3, p. 211–252, 2015. Citado 3 vezes nas páginas 21, 45 e 55.

- SCHAEFER, N. D. G. Lbp vs. lbp variance for texture classification. In: SPRINGER. *International Conference on Data Mining and Big Data*. [S.l.], 2017. p. 156–164. Citado na página 38.
- SERANG, M. M. A. A. F. G. O. Efficient exact maximum a posteriori computation for bayesian snp genotyping in polyploids. *PLoS One*, Public Library of Science, v. 7, n. 2, p. e30906, 2012. Citado na página 39.
- SHI, H. *On Nonlinear Cointegration Methods for Structural Health Monitoring*. Tese (Doutorado) — University of Sheffield, 2018. Citado na página 62.
- SILVA, K. A. C. P. B. R. D. F. J. P. P. L. A. On the feasibility of optimum-path forest in the context of internet-of-things-based applications. *Recent Patents on Signal Processing*, Bentham Science Publishers, v. 5, n. 1, p. 52–60, 2015. Citado na página 43.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. Disponível em: <<http://arxiv.org/abs/1409.1556>>. Citado 2 vezes nas páginas 45 e 46.
- SKULSTAD, R. et al. Net recovery of uav with single-frequency rtk gps. In: IEEE. *Aerospace Conference, 2015 IEEE*. [S.l.], 2015. p. 1–10. Citado 2 vezes nas páginas 27 e 28.
- STEINBACH, P.-N. T. M. knn: k nearest neighbors. *The top ten algorithms in data mining*, CRC Press Boca Raton, FL, p. 151–162, 2009. Citado 2 vezes nas páginas 39 e 40.
- STOJKOSKA, B. R. et al. Indoor localization of unmanned aerial vehicles based on rssi. In: IEEE. *Smart Technologies, IEEE EUROCON 2017-17th International Conference on*. [S.l.], 2017. p. 120–125. Citado na página 27.
- TANG, J.; REN, Y.; LIU, S. Real-time robot localization, vision, and speech recognition on nvidia jetson tx1. *arXiv preprint arXiv:1705.10945*, 2017. Citado na página 29.
- UNICOMB, J. et al. Distance function based 6dof localization for unmanned aerial vehicles in gps denied environments. In: IEEE. *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. [S.l.], 2017. p. 5292–5297. Citado na página 27.
- VAPNIK, A. L. V. Generalized portrait method for pattern recognition. *Automation and Remote Control*, v. 24, n. 6, p. 774–780, 1963. Citado na página 43.
- VOLNA EVA KOTYRBA, M. J. R. Neural network approach to image steganography techniques. In: *Mendel 2015*. [S.l.]: Springer, 2015. p. 317–327. Citado na página 34.
- YOO, C.-S. Y. C.-S.; AHN, I.-K. A. I.-K. Low cost gps/ins sensor fusion system for uav navigation. In: IEEE. *Digital Avionics Systems Conference, 2003. DASC'03. The 22nd*. [S.l.], 2003. v. 2, p. 8–A. Citado na página 27.
- YOSINSKI, J. et al. How transferable are features in deep neural networks? In: *Advances in neural information processing systems*. [S.l.: s.n.], 2014. p. 3320–3328. Citado 2 vezes nas páginas 47 e 55.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: SPRINGER. *European conference on computer vision*. [S.l.], 2014. p. 818–833. Citado 2 vezes nas páginas 78 e 79.

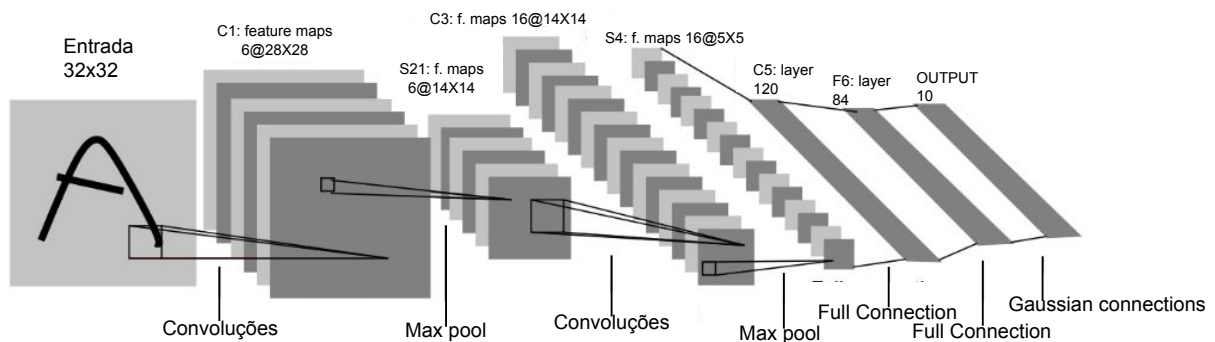
ZHILENKOV. Problems of a trajectory planning in autonomous navigation systems based on technical vision and ai. In: IEEE. *Young Researchers in Electrical and Electronic Engineering (EIconRus), 2018 IEEE Conference of Russian*. [S.l.], 2018. p. 1032–1035. Citado na página 30.

Apêndices

APÊNDICE A – ConvNets

Rede neural convolucional (Convolutional Neural Network - CNN ou ConvNet) foi introduzida por LECUN et al. 1998 e seus colaboradores apresentam um reconhecedor para dígitos manuscritos chamado LeNet. Ele usou o *backpropagation* em uma rede *feed forward* com muitas camadas ocultas, muitos mapas de unidades replicadas em cada camada, agrupando as saídas de unidades próximas, formando uma rede ampla que pode lidar com vários caracteres ao mesmo tempo, mesmo se eles se sobrepõem e uma inteligente maneira de treinar um sistema completo, não apenas um reconhecedor. Arquitetura apresentada na Figura 12.

Figura 12 – Estrutura do LeNet-5, um ConvNet, aqui utilizado para reconhecimento de dígitos. Cada quadro é um mapa de características, isto é, um conjunto de unidades cujos pesos são restritos para serem idênticos.

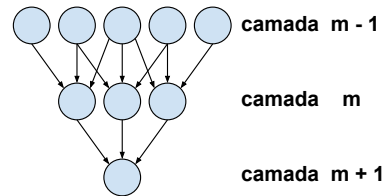


Fonte: Adaptado de (LECUN et al., 1998).

A topologia em camadas variantes do classificador Perceptron de Múltiplas Camadas (*Multi Layer Perceptron* - MLP), inspirada na biologia. Desde os primeiros trabalhos no córtex visual de animais (HUBEL; WIESEL, 1968), entende-se que existe um arranjo complexo de células dentro do córtex visual. Essas células são sensíveis a pequenas sub-regiões do espaço de entrada, chamadas de campo receptivo, e são pavimentadas de forma a cobrir todo o campo visual.

Os filtros na ConvNet, obtidos pela convolução de entradas e pesos, são locais no espaço de entrada e são, desta forma, mais adequados para explorar correlação espacialmente local presente em imagens naturais. Esta rede explora a correlação espacialmente local (coerência) reforçando um padrão de conectividade local entre os neurônios das camadas adjacentes. Os neurônios da entrada na m – ésima camada são conectadas a um subconjunto local de unidades na $(m - 1)$ – ésima camada, que possuem similaridade nos campos receptivos. Isso é ilustrado na Figura 13.

Figura 13 – Padrão de conectividade da CNN entre os neurônios das camadas adjacentes.



Fonte: Produção do próprio autor.

Tal arquitetura limita os **filtros** treinados a ter um padrão espacialmente local. Como ilustrado na Figura 13, o empilhamento de muitas camadas leva a **filtros** que se tornam cada vez mais **globais**. Logo, abrangendo uma região maior do espaço de entrada. Desta forma o neurônio na camada oculta $m + 1$ pode codificar uma característica não linear de largura maior.

Em uma ConvNet, cada filtro esparsos é adicionalmente replicado em todo o campo visual. Esses neurônios replicados formam mapas de características, que compartilham a mesma parametrização, ou seja, o mesmo vetor de peso e o mesmo viés. (GOODFELLOW; BENGIO; COURVILLE, 2016)

O gradiente de um peso compartilhado é basicamente a soma dos gradientes dos parâmetros que estão sendo compartilhados. Replicar as unidades dessa maneira permite que os recursos sejam detectados, independentemente de sua posição no campo visual. Além disso, o compartilhamento de peso oferece uma maneira muito eficiente de fazer isso, pois reduz bastante o número de parâmetros livres para aprender. Ao controlar a capacidade do modelo, a ConvNet tende a obter uma melhor generalização dos problemas de visão. (ZEILER; FERGUS, 2014)

Após uma camada convolução, é importante aplicar uma camada não linear (ou camada de ativação). O propósito dessa camada é introduzir a não-linearidade em um sistema que basicamente está computando operações lineares até o presente momento. Nos classificadores clássicos as funções não-lineares como tangente hiperbólica e sigmoide são bastantes exploradas, mas as pesquisas com ConvNets identificaram que camadas de unidade linear retificada (*ReLU*) é mais eficaz para esta tarefa.

Em termos simples, a *ReLU* apenas altera todas as ativações negativas para 0 no *feature map*. Essa camada aumenta as propriedades não-lineares do modelo e da ConvNet em geral sem afetar os campos receptivos da camada convolucional. (NAIR; HINTON, 2010)

Existem camadas agrupadas nas ConvNets que são projetadas para resumir as saídas de neurônios que pertencem a grupos vizinhos em um mesmo *feature map*. Também referida como uma camada de redução de resolução.

Dentre as soluções para tal seleção de características, o pool máximo é a implementação mais popular. Como apresentado no Algoritmo 1, em que k é um *kernel*, geralmente 2×2 , que integra aleatoriamente sobre as amostras do *feature map* resultante das convoluções.

Em cada coordenada (x, y) de k é mantida um registro acumulativo contador $[x][y]$, no qual se realiza a operação de maximização, que registra o maior valor de índice para todas as amostras aleatórias encontradas no *kernel*.

Vale ressaltar, o uso de três funções de apoio: $\text{senal}(k)$, $\text{indice}_x(k)$ e $\text{indice}_y(k)$, que retornam o sinal, o índice x e o índice y da amostra aleatória k respectivamente.

A ideia central deste algoritmo é produzir o número correto de amostras aleatórias correspondentes à localização (x, y) com o maior valor de probabilidade, desta forma o pool máximo promove uma taxa de convergência mais rápida, selecionando características invariantes superiores no *feature map* que melhoram o desempenho da generalização. (ZEILER; FERGUS, 2014)

Algorithm 1 Algoritmo estocástico de Max Pool

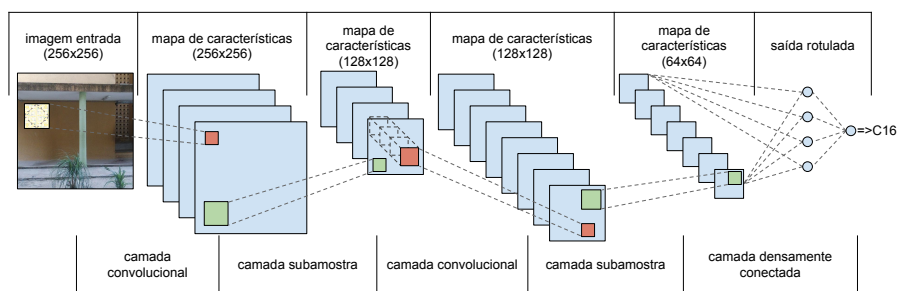
```

1: procedure MAX POOL( $\kappa$ )
2:    $\Delta_{max} \leftarrow -\infty$ 
3:   for  $i = 1, N$  do                                ▷ Inicializa todos contadores
4:     for  $j = 1, N$  do
5:       contador $[i][j] \leftarrow 0$ 
6:     end for
7:   end for
8:   while  $k$  do                                       ▷ Recebe amostra aleatória
9:      $i \leftarrow \text{indice}_x(k)$ 
10:     $j \leftarrow \text{indice}_y(k)$ 
11:    if  $\text{senal}(k) == +$  then
12:      contador $[i][j] ++$ 
13:    else
14:      contador $[i][j] --$ 
15:    end if
16:    if contador $[i][j] > \Delta_{max}$  then
17:       $\Delta_{max} \leftarrow \text{contador}[i][j]$ 
18:      libera  $k$ 
19:    end if
20:  end while
21: end procedure

```

Deforma simplificada as CNNs são modelos hierárquicos que alternam camadas convolucionais e de *pooling* para processar grandes espaços de entrada nos quais existe uma relação espacial ou temporal entre as entradas. No topo de um CNN, um MLP (por exemplo) pode ser adicionado para combinar os recursos e realizar previsão ou classificação. A Figura 14 mostra um exemplo do fluxo da classificação de um cenário de localização.

Figura 14 – Exemplo de uma ConvNet.



Fonte: Produção do próprio autor.