**UNIVERSIDADE ESTADUAL DO CENTRO-OESTE, UNICENTRO-PR**

# UTILIZAÇÃO DE REDES NEURAIS CONVOLUCIONAIS E IMAGENS OBTIDAS POR *RPA* PARA O MAPEAMENTO DE PALMEIRAS NA AMAZÔNIA OCIDENTAL

DISSERTAÇÃO DE MESTRADO

**MAURO ALESSANDRO KARASINSKI**

**IRATI-PR**

**2021**

MAURO ALESSANDRO KARASINSKI

# UTILIZAÇÃO DE REDES NEURAIS CONVOLUCIONAIS E IMAGENS OBTIDAS POR *RPA* PARA O MAPEAMENTO DE PALMEIRAS NA AMAZÔNIA OCIDENTAL

Dissertação apresentada à Universidade Estadual do Centro-Oeste, como parte das exigências do Programa de Pós-Graduação em Ciências Florestais - Mestrado, área de concentração em Manejo Florestal, para obtenção do título de Mestre.

Prof. Dr. Henrique Soares Koehler
Orientador

Prof. Dr. Afonso Figueiredo Filho
Coorientador

Profa. Dra. Sabina Cerruto Ribeiro
Coorientadora

**IRATI-PR**
**2021**

Catalogação na Publicação
Rede de Bibliotecas da Unicentro

K18u

Karasinski, Mauro Alessandro
Utilização de redes neurais convolucionais e imagens obtidas por *RPA* para o mapeamento de palmeiras na Amazônia Ocidental / Mauro Alessandro Karasinski. – – Irati, 2021.
vi, 103 f. : il. ; 28 cm

Dissertação (mestrado) - Universidade Estadual do Centro-Oeste, Programa de Pós-Graduação em Ciências Florestais, área de concentração em Manejo Florestal, 2021.

Orientador: Henrique Soares Koehler
Coorientador: Afonso Figueiredo Filho
Coorientadora: Sabina Cerruto Ribeiro
Banca examinadora: Matheus Pinheiro Ferreira, Evandro Orfanó Figueiredo, Ana Paula Dalla Corte

Bibliografia

1. Redes neurais convolucionais. 2. YOLOv4. 3. Sensoriamento remoto. 4. Floresta Amazônica. I. Título. II. Programa de Pós-Graduação em Ciências Florestais.

CDD 634.9

# TERMO DE APROVAÇÃO

Defesa Nº 155

**Mauro Alessandro Karasinski**

## "UTILIZAÇÃO DE REDES NEURAIS CONVOLUCIONAIS E IMAGENS OBTIDAS POR RPA PARA O MAPEAMENTO DE PALMEIRAS NA AMAZÔNIA OCIDENTAL"

Dissertação aprovada em 15/04/2021, como requisito parcial para obtenção do grau de Mestre, no Programa de Pós-Graduação em Ciências Florestais, área de concentração em Manejo Sustentável de Recursos Florestais, da Universidade Estadual do Centro-Oeste, pela seguinte Banca Examinadora:

Dr. Matheus Pinheiro Ferreira
Instituto Militar de Engenharia
Primeiro Examinador

Dr. Evandro Orfanó Figueiredo
Embrapa - Acre
Segundo Examinador

Dra. Ana Paula Dalla Corte
Universidade Federal do Paraná,
Terceira Examinadora

Dr. Henrique Soares Koehler
Universidade Federal do Paraná/Universidade Estadual do Centro-Oeste
Orientador e Presidente da Banca Examinadora

Irati - PR
2021

## SUMÁRIO

## LISTA DE FIGURAS

## LISTA DE TABELAS

## AGRADECIMENTOS

Aos meus pais, pelo amor, dedicação, compreensão, aprendizado e todo carinho demonstrado.

Ao Professor Dr. Henrique Soares Koehler, pela orientação e todos os conhecimentos adquiridos durante esta etapa.

Ao Professor Afonso Figueiredo Filho, pela coorientação, ensinamentos, amizade e incentivo.

À Professora Dra. Sabina Cerruto Ribeiro pela coorientação, amizade e incentivo.

Ao Programa de Pós-Graduação em Ciências Florestais da Universidade Estadual do Centro-Oeste pela oportunidade de aperfeiçoamento.

Ao Programa de Pós-Graducação em Ciência Florestal da Universidade Federal do Acre, pela oportunidade de mobilidade acadêmica.

À Empresa Brasileira de Pesquisa Agropecuária – Acre (EMBRAPA-ACRE), pela parceria firmada para o desenvolvimento de minha pesquisa, em especial ao Daniel de Almeida Papa, pela amizade e incentivo.

Ao Grupo de Pesquisa de Mapeamento Florestal na Amazônia (GPMAP) da EMBRAPA-ACRE, pela parceria e cooperação, especialmente ao Professor Matheus Pinheiro Ferreira pelos conhecimentos transmitidos.

Aos membros da banca examinadora, por aceitar o convite. Meus sinceros agradecimentos.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela bolsa de estudos concedida.

Ao Ramon de Sousa Leite, pelo companheirismo, amizade, carinho, paciência e incentivo durante esta jornada. Sou eternamente grato.

E a todos que, de alguma forma, contribuiram para minha formação.

## RESUMO

As palmeiras (Arecaceae) são um dos recursos mais importantes do ponto de vista social e econômico para as comunidades locais na Amazônia, porque garantem rendimentos e oferecem recursos como alimentos e matéria-prima para a construção, artesanato e indústria. A complexidade das florestas amazônicas limita a obtenção de informações cruciais para a exploração e gestão comercial das palmeiras, tais como a densidade e distribuição espacial. Em vista disso, neste estudo avaliou-se o desempenho da Rede Neural Convolucional YOLOv4 para a detecção e classificação automática das palmeiras nas florestas tropicais nativas. O estudo foi realizado num remanescente de Floresta Ombrófila Aberta no sudoeste da Amazônia. Primeiramente foi gerada uma ortofoto *RGB* a partir de imagens obtidas com uma aeronave remotamente pilotada. Em seguida, a ortofoto foi subdividida em 960 parcelas de 37,5 x 37,5 metros. Foram rotuladas, manualmente, 1098 palmeiras identificadas por fotointerpretação pertencentes a quatro espécies de palmeiras: *Attalea butyracea* (Mutis ex L.f.) Wess. Boer, *Euterpe precatoria* Mart., *Iriartea deltoidea* Ruiz & Pav e *Oenocarpus bataua* Mart. Realizou-se um aumento de dados para elevar a capacidade de aprendizagem do modelo. Selecionou-se aleatoriamente 80% dos dados para treinamento e 20% dos dados para validação. Para fazer as previsões da localização e classificação das palmeiras, a Rede Neural Artificial para a detecção de objetos YOLOv4 foi utilizada. O método alcançou precisão média geral de 91,08% e a precisão média para *A. butyracea*, *E. precatoria*, *I. deltoidea* e *O. bataua* foi 92,07% ±2,85%; 96,2% ±1,48%; 93,83% ±3,09% e 92,48% ±2,82%, respectivamente. O YOLOv4 é uma ferramenta efetiva para o mapeamento das palmeiras em florestas nativas, podendo ser utilizada no âmbito do planejamento e manejo florestal.



Palavras-chave: Redes neurais artificiais, YOLOv4, deep Learning sensoriamento remoto, floresta Amazônica.

**ABSTRACT**

The palm trees (Arecaceae) are one of the most important resources from the social and economic point of view for the local communities in the Amazon, because they guarantee income and provide resources such as food, raw material for construction, handicrafts, and industry. The complexity of Amazonian forests limits obtaining crucial information for commercial exploitation and management of palm trees, such as density and spatial distribution. We evaluated the performance of the YOLOv4 Convolutional Neural Network in the automatic detection and classification of palm trees in native tropical forests. The study was conducted in a remnant of Open Ombrophylous Forest in southwestern Amazonia. First, an RGB orthophoto was generated from images obtained with an Unmanned Aerial Vehicle. The orthophoto was then subdivided into 960 plots of 37.5 x 37.5 meters. We manually labeled 1098 palm trees identified by photointerpretation belonging to four palm species: Attalea butyracea (Mutis ex L.f.) Wess. Boer, Euterpe precatoria Mart., Iriartea deltoidea Ruiz & Pav and Oenocarpus bataua Mart. Data augmentation was performed to increase the learning ability of the model. It randomly selected 80% of the data for training, 20% for validation. To make the predictions, the Artificial Neural Network for object detection YOLOv4 was used. The method achieved overall average accuracy of 91.08% and the average accuracy for *A. butyracea*, *E. precatoria*, *I. deltoidea* and *O. bataua* was 92.07% ±2.85%; 96.2%; ±1.48%; 93.83%; ±3.09% and 92.48% ±2.82%, respectively. YOLOv4 is an important tool for mapping palm trees in native forests, serving as a support for forest planning and management.

Keywords: Artificial Neural Networks, YOLOv4, Deep Learning, Remote Sensing, Amazon Forest.

# 1 INTRODUÇÃO

Nas últimas décadas, a região conhecida como o Arco do Desmatamento na Amazônia passou por uma intensa modificação no uso da terra devido à expansão agrícola, industrialização e urbanização, o que resultou na fragmentação florestal. Isto exige da sociedade uma análise mais profunda do aproveitamento socioeconômico dos recursos, o que inclui o conhecimento sobre a sua biodiversidade.

Os produtos florestais não madeireiros têm sido uma importante alternativa econômica que favorece a conservação e o uso de recursos florestais aliados à movimentação de economias locais. As palmeiras (Arecaceae), por exemplo, são um dos recursos mais importantes do ponto de vista social e econômico para as comunidades extrativistas na Amazônia, pois fornecem produtos florestais não madeireiros como frutas e palmitos (*Euterpe* spp., *Astrocaryum* spp.), combustível e óleos (*Attalea butyracea* (Mutis ex L.f.) Wess.Boer, *Oenocarpus bataua* Mart.), tecidos, fibras e materiais de construção (*Iriartea deltoidea* Ruiz & Pav, *Attalea* spp.) (BERNAL *et al.* e EISERHARDT *et al.*, 2011). Diante disto, é de grande interesse melhorar os métodos para avaliar a extensão e distribuição das populações das espécies de palmeiras, principalmente na porção oeste da Amazônia, onde se encontra a maior riqueza de espécies de palmeiras região (VORMISTO, 2002).

Um dos pontos iniciais para o manejo de qualquer produto não madeireiro é o mapeamento, inventário e localização das árvores de interesse. Isso proporciona a quantificação do potencial produtivo, identificando a viabilidade da comercialização de tal produto, possibilitando acesso a fontes de financiamentos agrícolas e o licenciamento ambiental do plano de manejo junto aos órgãos competentes. Após a localização dos indivíduos, a etapa de planejamento para o acesso e coleta destes exemplares é de grande importância para a estruturação do manejo. Para ambas as etapas é importante o uso de tecnologias, proporcionando maior eficiência e precisão nos dados coletados e, principalmente, de otimização do tempo e do esforço laboral para a coleta e extração. Assim, o sensoriamento remoto aliado aos avanços da visão computacional apresenta um conjunto de ferramentas que auxilia no tratamento desse tipo de informação espacial.

Informações sobre a densidade e distribuição espacial das palmeiras são cruciais para a exploração e o manejo comercial dos produtos não madeireiros por elas fornecidos. Essas informações, na maioria das vezes, são obtidas a partir de inventários de campo, por exemplo, com a contagem *in situ* dos indivíduos alvos (ROCHA, 2004). Embora esse

seja um bom método, torna-se inviável para áreas extensas, principalmente em florestas de alta complexidade como é o caso das Florestas Abertas com Palmeiras e Bambu no estado do Acre. Portanto, o desenvolvimento de ferramentas para avaliar a distribuição espacial e densidade dessas palmeiras ajuda a estimar o valor econômico potencial total das florestas, além de promover o manejo sustentável dessas espécies (VORMISTO, 2002; NEVALAINEN *et al.*, 2017).

Técnicas de visão computacional, aplicadas a imagens de Sensoriamento Remoto, especialmente de aprendizado profundo baseadas em Redes Neurais Convolucionais (*CNNs*), têm se mostrado adequadas para a identificação de palmeiras em áreas de florestas (DOS SANTOS *et al*., 2017, FERREIRA *et al*., 2020). As *CNN*s se destacam quando comparadas a outros métodos de classificação, uma vez que, no processo de aprendizagem, conseguem armazenar características espaciais de uma imagem além de depender de um nível mínimo de pré-processamento, sendo amplamente utilizadas no reconhecimento de padrões em imagens.

Os métodos de aprendizado profundo baseados em *CNN*s geralmente dividem a imagem inteira em vários fragmentos de imagem com tamanho de janela específico e, em seguida, classificam-nos em plano de fundo ou copa da árvore por meio de diferentes *CNN*s (ZHENG *et al.*, 2021).

Diferentes abordagens vêm sendo utilizadas para a detecção, classsificação e segmentação de árvores em floretas naturais. Alguns métodos, são capazes de identificar a presença de diferentes classes de objetos nas imagens, enquanto outros são baseados na segmentação de instâncias em que cada pixel dentro de uma instância de um objeto é atribuido a uma categoria com base na sua localização e tamanho.

Ferreira *et al.* (2020) desenvolveram um método baseado em mapas de pontuação derivados de um modelo de rede totalmente convolucional para detectar e classificar copas individuais de palmeiras no oeste da Amazônia. Embora a taxa de precisão do classificador obtida pelos autores tenha sido relativamente alta (98,6%, 96,6% e 78,6% para *E. precatoria*, *I. deltoidea* e *A. butyracea*, respectivamente), o algoritmo de detecção e classificação apresentou dificuldades ao segmentar indivíduos muito próximos uns aos outros. Freudenberg *et al.* (2019) aplicaram uma rede neural profunda do tipo *U-net* para detectar dendezeiros a partir de imagens de sensores de satélites em grandes plantações em Jambi na Indonésia e Coqueiros na região metropolitana de Bengaluru na Índia.

Recentemente, uma variedade de métodos de detecção de objetos de ponta a ponta foi aplicada ao campo de detecção de copas de árvores como *Faster R-CNN* (ZHENG *et*

*al.*, 2019), RetinaNet (SELVARAJ *et al.*, 2020), YOLOv2 e YOLOv3 (PUTTEMANS *et al.*, 2018, ITAKURA E HOSOI, 2020) e *Mask R-CNN* (BRAGA *et al.*, 2020). Em geral, o método baseado em detecção de objetos é mais rápido e mais robusto quando comparado a outros métodos de detecção de copas de árvores, principalmente tratando-se de florestas com maior diversidade de espécies (ZHENG *et al.*, 2021).

Dessa maneira, esta pesquisa utiliza um novo método para o mapeamento de palmeiras economicamente importantes, usando rede neural convolucional (YOLOv4) aplicada a imagens *RGB* obtidas por Aeronaves Remotamente Pilotadas (*RPA*, do inglês, *Remotely Piloted Aircraft*) em Florestas Tropicais nativas.

# 2 OBJETIVOS

## 2.1 Geral

Detectar e classificar palmeiras automaticamente em Floresta Tropical Amazônica usando imagens *RGB* obtidas por *RPA* e rede neural convolucional.

## 2.2 Específicos

- Avaliar o desempenho da rede YOLOv4 na detecção de palmeiras.
- Avaliar o efeito do aumento de dados no processo de aprendizagem.
- Detectar automaticamente as palmeiras na área de estudo.
- Estimar a densidade e a distribuição espacial das palmeiras.

# 3 REVISÃO DE LITERATURA

## 3.1 Fitofisionomia Florestal

O sistema fitogeográfico oficial adotado no Brasil foi publicado em 1992 por Veloso. A "formação" propriamente dita é determinada pelo ambiente (forma e relevo), sendo caracterizada pela hierarquia topográfica, além dos atributos climáticos onde a mesma está situada. As duas principais formações em domínio amazônico são: Floresta Ombrófila Densa (FOD) e Floresta Ombrófila Aberta (FOA).

A Floresta Ombrófila Densa ocupa grande parte da bacia hidrográfica do rio Amazonas. Ocorre em climas tropicais de alta temperatura (média 25 ºC) e de alta precipitação bem distribuída durante o ano. Apresenta dossel de 30-40 m, com árvores emergentes que podem chegar até 60 m, subdossel de 5-20 m e submata com espécies arbóreas e arbustivas de 2 a 5 m (RIZZINI, 1997). É subdividida em cinco formações: alto-montana, montana, submontana, terras baixas e aluvial.

A Floresta Ombrófila Aberta é considerada uma área de transição entre a Floresta Amazônica e as áreas extra-amazônicas. Esta floresta apresenta quatro fisionomias específicas (faciações florísticas) que alteram a fisionomia ecológica da Floresta Ombrófila Aberta: floresta com palmeiras, floresta de bambu, floresta de sororoca e floresta com cipó. Tem como características apresentar climas mais secos, que chegam de 2 a 4 meses de secas por ano, com temperaturas de 24 a 25°C (VELOSO, 1992).

Segundo IBGE (2005) as tipologias florestais encontradas no Estado do Acre são: Floresta Aberta com Bambu Dominante (9,40%), Floresta Aberta com bambu mais Floresta Aberta com Palmeiras (26,20%), Floresta Aberta com Palmeiras das Áreas Aluviais (5,48%), Floresta Aberta com Palmeiras (7,77%), Floresta Aberta com Palmeiras e Floresta Densa (12,12%), Floresta Densa mais floresta Aberta com Palmeiras (7,20%), Floresta Aberta com Palmeiras mais Floresta Aberta com Bambu (21,02%) Floresta Aberta com Bambu em Áreas Aluviais (2,04%), Floresta Densa (0,53%), Floresta com Bambu mais Floresta Densa (0,36%) e Floresta Densa Submontana (0,47%).

## 3.2 Produtos Florestais Não Madeireiros

Os PFNMs são considerados importantes fontes de renda para trabalhadores rurais ou extrativistas, e de matéria prima para indústrias. A extração comercial dos PFNMs vem sendo defendida como uma das formas mais sustentáveis de conservação das florestas, assegurando os modos de vida tradicionais de comunidades rurais em diversos

países, principalmente daqueles em desenvolvimento (REGO, 1999; LESCURE, 2000; ROCHA, 2004; NYEGREN *et al.*, 2006).

Segundo Pedrozo *et al.* (2011), os PFNMs são recursos provenientes de florestas naturais, sistemas agroflorestais e plantações incluindo também plantas medicinais e de uso alimentício, frutas, fungos, fauna e madeira para fabricação de artesanato, sendo a floresta amazônica, a maior fonte de fornecimento desses produtos.

Na região amazônica, os PFNMs obtidos de palmeiras frutíferas, dentre estes a polpa do açaizeiro, têm grande potencial agronômico, tecnológico, nutricional e econômico (YUYAMA *et al.*, 2011).

No Brasil, em 2018, a participação de PFNMs somou R$ 1,6 bilhões, registrando um crescimento de 1,8% em relação ao ano anterior. O grupo dos produtos alimentícios, maior entre os não madeireiros da extração vegetal, novamente apresentou valor de produção crescente (4,1%), totalizando R$ 1,3 bilhões. O açaí foi o produto que registrou maior participação no valor de produção dentro deste grupo (46,3%) (IBGE, 2019).

### 3.3 Palmeiras

As Palmeiras (Arecaceae) estão entre os grupos de plantas mais notáveis e diversos, com 181 gêneros e aproximadamente 2.500 espécies, que desde os primórdios da humanidade prestam uma ampla gama de serviços (TOMLINSON, 2006; CÁMARA – LERET *et al.*, 2017; LEVIS *et al.*, 2017). Muitas dessas espécies são consideradas de extrema importância ecológica, pois permitem que um grande número de animais usufrua de seus frutos e flores (ONSTEIN *et al.*, 2017). Na região Amazônica, seis a cada dez espécies de plantas mais comuns na floresta são palmeiras (TER STEEGE *et al.*, 2013), compreendendo cerca 35 gêneros compostos por mais de 160 espécies (ALVEZ-VALLES *et al.*, 2018).

As palmeiras exibem uma variedade de formas de crescimento, que vão de características semelhantes a pequenos arbustos até palmeiras análogas ao porte de grandes árvores. Segundo Kissling *et al.* (2019), aproximadamente 40% das palmeiras são capazes de cultivar hastes com diâmetro ≥10 cm a 1,30 metros acima do solo. No estado do Acre são registradas 78 espécies de palmeiras, das quais 28 alcançam o estrato superior da floresta tendo potencial para ser monitoradas a partir do uso de *RPA.*

As espécies *Attalea* butyracea, *Eutepe precatoria* Mart., *Iriartea deltoideae* Ruiz & Pav amp e *Oenocarpus bataua* Mart. serão detalhadas a seguir por apresentarem características morfológicas de copa capaz de se diferenciar entre as demais quando

visualizadas em nível de fotointerpretação como, por exemplo, tamanho da copa, disposição das folhas ao longo do caule, orientação das pinas na raque etc.

### 3.3.1 *Attalea butyracea* (Mutis ex L.f.) Wess.Boer

*Atallea burtyraceae* (Mutis ex L.f.) Wess. Boer, popularmente conhecido como jací, é uma das palmeiras neotropicais mais abundantes (HENDERSON *et al.*, 1995), com uma haste grossa de até 50 cm de diâmetro e 15-20 m de altura. Possui uma grande coroa de 15 a 40 folhas pinadas que alcançam entre 6 a 7 metros de comprimento e 80 cm de largura. Os folíolos são regularmente dispostos na raque de forma linear e no mesmo plano. Possuem grandes inflorescências fechadas em uma espessura bráctea pedunculares lenhosas. As infrutescências são grandes e pendentes, e carregam numerosos frutos densamente dispostos, elíptico, com 4,5 a 8,5 cm de comprimento e 2,5 a 4,5 cm de diâmetro, de cor amarelada, laranja e marrom, com um endocarpo lenhoso espesso e com 2 a 3 sementes estreitamente elípticas, de 3 a5 cm de comprimento e 0,5 a 1,2 mm de espessura (HENDERSON, 1995; LORENZI *et al.*, 2000; FERREIRA *et al.*, 2020). Suas folhas são amplamente utilizadas em comunidades amazônicas para a construção de telhados e artesanatos (BERNAL *et al.*, 2010).

### 3.3.2 *Euterpe precatoria* Mart.

*Euterpe precatoria* Mart., (açaí-solteiro) ocorre em vários habitats, em terrenos alagados e também em terras não alagadas. Pode ser comum na várzea, mas também ocorre em rampas andinas íngremes a 2000 m de altitude (KAHN, 1993; HENDERSON, 1995). Possui raízes adventícias continuamente na base do estipe, formando um anel espesso na base (1,5 cm) de cor púrpura, podendo alcançar 80 cm do nível do solo (BOVI e CASTRO, 1993). As inflorescências *E. precatoria* Mart. têm numerosas flores masculinas (4,5 x 2,7 mm) e femininas (3,2 x 2,6 mm). As flores masculinas abrem e liberam o pólen antes que as flores femininas sejam receptivas e desta forma não ocorre autofecundação, sendo a polinização cruzada geralmente entomófila com predominância de besouros e abelhas como polinizadores potenciais (KÜCHMEISTER *et al.*, 1997; LORENZI *et al.*, 2010). Os frutos são globosos e de cor púrpura-escuro quando maduros, com mesocarpo suculento, existindo uma semente por fruto, com endosperma sólido e homogêneo (HENDERSON, 1995). A produção de frutos é anual nos períodos entre março/abril a setembro (ROCHA, 2004). Possui forma de copa linear, regularmente

espaçada, de padrão estrelado, com raio de copa inferior a 7 metros, sendo facilmente distinguível em imagens aéreas (FERREIRA *et al.*, 2020). Além da produção local para a subsistência de famílias locais, a produção do açaí vem cada vez mais conquistando o mercado nacional e de exportação (TAVARES, 2020), seu potencial nutricional e energético conquistou principalmente a indústria de alimentos e cosméticos.

### 3.3.3 *Iriartea deltoidea* Ruiz & Pav

*Iriartea deltoidea* Ruiz & Pav (paxiubão) possui caule único de até 35 metros de altura e DAP > 30 cm, com presença de raízes aéreas compactadas com mais de 50 cm de comprimento. Possui de 4 a 6 folhas com folíolos com dobras profundas e margens externas dentadas, dispostos em diferentes planos, conferindo-lhe a forma de grandes penas. As infrutescências são envoltas em uma grande bráctea verde de até 1 metro de comprimento, com frutos globosos de 2,5 a 3 cm de diâmetro de cor verde amarelado (PINARDI, 1993; ALVAREZ-LOAYZA, 2011). Seu caule possui alta densidade e boa resistência à flexão, compressão e tenacidade extrema, sendo utilizada principalmente pelas comunidades tradicionais, na construção de paredes e pisos de moradias rústicas.

### 3.3.4 *Oenocarpus bataua* Mart.

Uma espécie oleaginosa e comestível na região amazônica muito apreciada é a espécie *Oenocapus bataua* Mart., popularmente conhecida como patauá. É uma palmeira de estipe solitário, com 4 a 26 m de altura, amplamente distribuída na Amazônia brasileira, ocorre tanto em floresta úmida de várzea e de galeria quanto em florestas de terra firme, bem adaptado a solos pobres (GALEAN e BERNAL, 1987, BALICK 1992, MOSCOTE-RIOS *et al.*, 1998). *O. bataua* é uma planta monoica com inflorescência infra foliares que podem atingir até 2 metros de comprimento e pedúnculos florais até 40 cm (NÙÑEZ-AVELLANEDA e ROJASROBLES, 2008). Possui de 8 a 16 folhas arranjadas em forma de espiral, cada uma medindo de 3 a 10 m de comprimento. Os folíolos são largos e pêndulos, o que facilita na identificação pelo método de fotointerpretação.

A população natural de *O. bataua* produz anualmente cerca de 39 kg por palmeira, podendo gerar rendas substanciais e ecologicamente sustentáveis às comunidades amazônicas (MILLER, 2002). Da poupa do fruto é produzido o chamado "vinho do patauá", o qual é bastante nutritivo e do vinho é extraído o óleo, o qual pode substituir o azeite de oliva na culinária, por ter sabor e composição química semelhante.

## 3.4 A Fotogrametria e as Aeronaves Remotamente Pilotadas (*RPAs*)

A técnica de medir distâncias e dimensões por meio de fotografias (fotogrametria) é antiga, data de meados do século XIX, quando o topógrafo alemão Albrecht Meydenbauer observou, após sofrer um acidente, que as medições diretas poderiam ser realizadas de forma indireta por meio de imagens (ALBERTZ, 2001; MEYER, 1985). Com o avanço da fotogrametria e as técnicas de sensoriamento remoto, aviões tripulados com fotógrafos embarcados são utilizados para levantamentos em diversas áreas, como na engenharia, fiscalização, conservação, cultura e lazer.

Uma inovação tecnológica que ganha destaque nos últimos anos, são as Aeronaves Remotamente Pilotadas (*RPA*s) ou Veículos Aéreos Não Tripulados (VANTs), os quais são aeronaves capazes de serem pilotadas por controle remoto ou autonomamente (ALBERTZ, 2007). As *RPAs* apresentam vantagens quando comparadas às aeronaves tripuladas, pois possuem maior flexibilidade, baixo custo, não necessitam de área de decolagem e pouso, além de minimizar os riscos de acidentes com a tripulação (CASSEMIRO e PINTO, 2014).

A integração de métodos fotogramétricos com os avanços da visão computacional intensificou o interesse em imagens digitais obtidas com plataformas não tripuladas. Na área florestal, o uso de *RPA*s possui muitas aplicações, dentre elas podem-se citar o monitoramento e combate de incêndios florestais (PÉREZ-RODRÍGUEZ *et al.*, 2020), fiscalização de desmatamento ilegal (FONTES; POZZETTI, 2016), produção de mapas de uso de solo (JOSE; GUERRA, 2020), medição volumétrica de madeira em pátios, silvicultura de precisão (FIGUEIREDO *et al.*, 2016; SOBRINHO *et al.*, 2018), identificação e contagem de árvores, determinação de carbono e biomassa (ENE *et al.*, 2017; REX *et al.*, 2020, ABDULLAH *et al.*, 2021).

## 3.5 Inteligência Artificial

### 3.5.1 Aprendizagem de Máquina

De acordo com Bishop (2006), Aprendizagem de Máquina, ou do inglês *Machine Learning* é um sistema capaz de adquirir e armazenar conhecimento e, dessa forma, melhorar o desempenho em soluções de tarefas específicas, ou seja, é um dos tipos de algoritmos[1] usados na Inteligência Artificial, que desenvolve programas que aprendem a

[1]Série de instruções que devem ser seguidas passo a passo por uma máquina (MANASWI, 2018)

fazer previsões com base em dados, sem ser explicitamente programado. O *Machine Learning* é utilizado em diversas áreas do conhecimento, tais como: reconhecimento de padrões, processamento de imagens, filtro de spam, recomendações de música, detecção de fraudes etc.

Existem três tipos principais de aprendizagem de máquina (SNOW, 2018): aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço.

O aprendizado supervisionado é aplicado para problemas de classificação e regressão. Os dados utilizados nessa categoria de aprendizado necessitam ser rotulados, ou seja, é uma técnica que ensina a um algoritmo como resolver tarefas específicas usando dados que foram classificados anteriormente por humanos; no aprendizado não supervisionado as máquinas buscam aprender por si próprias, ou seja, as máquinas buscam de alguma forma dar um significado para os dados que recebem, sem nenhum rótulo para esses dados; e, no aprendizado por reforço, o algoritmo realiza um *feedback* sobre os resultados obtidos, atribuindo posições positivas à resultados considerados corretos e penalizando os considerados incorretos e, dessa forma, vai ajustando o comportamento do modelo para encontrar um melhor resultado.

### 3.5.2 Aprendizado Profundo

O Aprendizado Profundo, ou do inglês *Deep Learning*, é uma forma de aprendizado de máquina que possibilita aos computadores aprenderem com a experiência a compreender o mundo em termos de uma hierarquia de conceitos. Pelo fato de o computador armazenar o conhecimento com a experiência, não há a necessidade de um operador humano especificar todos os conhecimentos necessários ao computador. A hierarquia de conceitos permite que o computador aprenda conceitos em níveis mais complicados originários de outros níveis mais simples (GOODFELLOW; BENGIO; COURVILLE, 2016). O *Deep Learning* utiliza um modelo inspirado no neurônio humano, como uma rede neural artificial, cujos neurônios artificiais estão organizados em camadas interconectadas. Devido a grande quantidade de camadas entre a camada de entrada e a camada de saída o modelo se torna mais profundo, por isso recebe o nome de *Deep Learning* (ARROYO-FIGUERO *et al.*, 2000).

### 3.5.3 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) são formadas por sistemas computacionais paralelos de processamento simples, também denominados neurônios artificiais ou nodos,

interligados entre si de forma específica para desempenhar determinada tarefa (BINOTI, 2010; BINOTI *et al.*, 2013). Seu primeiro conceito foi introduzido em 1943, mas ganhou popularidade algumas décadas depois com a evolução dos computadores e a introdução de algoritmos de treinamento como o *backpropagation*, que permite a realização de um treinamento posterior para aperfeiçoar os resultados do modelo.

As RNAs são organizadas em camadas (Figura 1), que compõe sua arquitetura, podendo ser constituídas por apenas uma camada simples (*perceptron*) ou redes multicamadas, formadas por uma ou mais camadas intermediárias (“camadas ocultas”) ou pela combinação de várias redes de camadas simples (VENTURIERI e SANTOS, 1998).

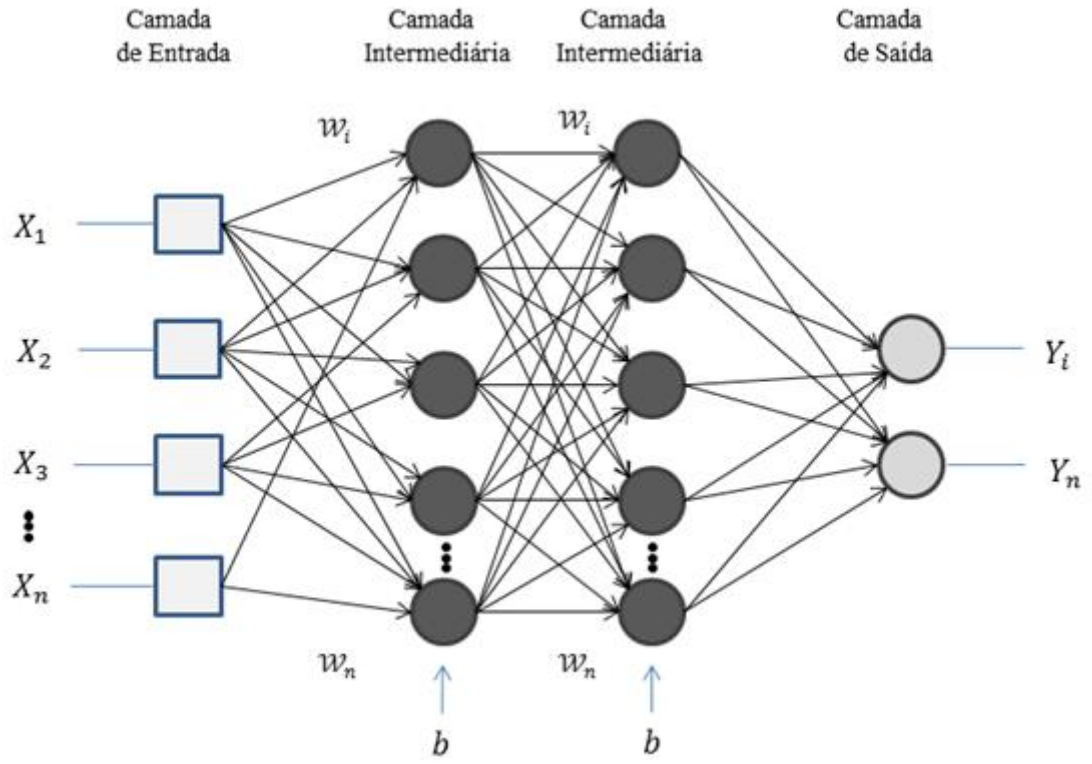


**Figura 1** Representação de uma Rede Neural Artificial de Multicamadas, quando um modelo possui um grande número de camadas intermediárias recebe o nome de *Deep Learning*. Adaptado de Neogi (2020).

O funcionamento geral de uma rede multicamadas está representado na Figura 1. Cada neurônio recebe todos os valores das entradas (*x*), que são multiplicados pelos pesos sinápticos (*w*) e somados entre si junto com uma constante chamada de polarização ou *bias* (*b*). Essa constante possui o papel de centralizar a curva da função de ativação em um valor conveniente. O somatório das várias entradas, ponderada pelos pesos de suas respectivas conexões, gera o potencial de ativação que é utilizado para propagar para os demais neurônios da próxima camada (LIPPMANN, 1987).

Quando uma rede neural artificial é inicializada, os pesos sinápticos recebem valores aleatórios que, quando multiplicados pelos valores recebidos, não atingem os valores desejados no momento do treinamento. Para corrigir os pesos sinápticos, uma das técnicas mais utilizadas é a retropropagação ou *backpropagation*, a qual corrige os valores dos pesos pela diferença entre os valores obtidos e o valor esperado pelo algoritmo. Em um treinamento bem sucedido, o erro diminui com o aumento do número de iterações e o procedimento converge para um conjunto estável de pesos (GALO, 2000).

As RNAs são aplicadas em diversas áreas do conhecimento: para a aproximação de funções, previsão de séries temporais, classificações e reconhecimento de padrões. Cada abordagem de classificação demanda diferentes arquiteturas de rede e parâmetros de treinamento, cuja definição influencia nos resultados de saídas, obtendo diferentes modelagens das classes de interesse (KOVÁCS, 2002).

### 3.5.4 Redes Neurais Convolucionais

As Redes Neurais Convolucionais (CNNs) são um tipo de aprendizado profundo que abrange um grande número de camadas convolucionais e de amostragem. Foram aplicadas com sucesso pela primeira vez por Lecun *et al.* (1998), para o reconhecimento de dígitos manuscritos. Em uma CNN a entrada da camada é geralmente uma matriz de imagem com dimensões arbitrárias e sua saída é um vetor de características correspondentes a diferentes classes. Os métodos de classificação baseados na CNN usam essas características em um algoritmo de classificação para encontrar o rótulo da classe, transformando os dados de entrada em resultados na medida em que o modelo aprende recursos de nível cada vez mais alto (LITJENS *et al.*, 2017).

Krizhevsky *et al.* (2017) usaram uma CNN para classificar 1,2 milhão imagens em 1000 classes no *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) de 2012. Eles venceram o desafio com resultados animadores e marcaram o avanço mais significativo para tarefas de classificação de imagens. A partir de então as CNNs vêm sendo amplamente utilizadas em diversas áreas do conhecimento.

As CNNs alcançam desempenho notável em tarefas de detecção de objetos, pois são complexas o suficiente para extrair características intrínsecas de alto nível para aprender a identificar e rotular objetos espacialmente (ZHU *et al.*, 2017).

Uma CNN pode ser dividida em extração de características e classificação. A extração de características passa por três etapas principais: a convolução, uma função de

ativação e *Pooling*. O processo de como cada uma destas etapas funciona está ilustrado na Figura 2 e serão descritas nos parágrafos seguintes.

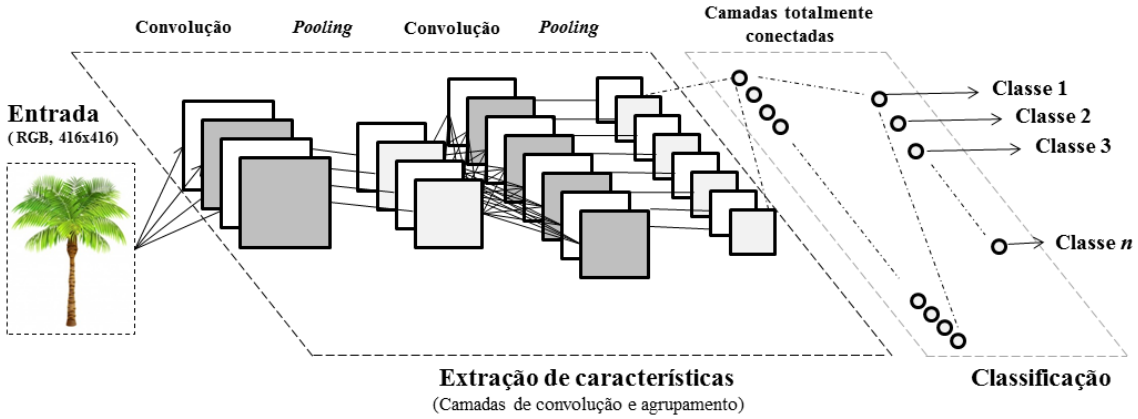


**Figura 2** Estrutura básica de uma Rede Neural Convolucional. *FC* = Camadas totalmente conetadas.

Matematicamente, convolução é uma operação linear que a partir de duas funções $f$ e $g$ gera uma terceira função, sendo normalmente vista como uma versão modificada de uma das funções originais. Num contexto de imagens pode-se entender esse processo como um filtro (*kernel*) que transforma a imagem de entrada, sendo utilizada para a detecção de bordas, suavização de imagens, extração de atributos, entre outras aplicações (PARKER, 2010).

Seja a função $f$ e $g$ para uma variável discreta $x$ a convolução é definida como:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\tau) \cdot g(x - \tau) d\tau \quad (1)$$

Onde $*$ representa o operador de convolução. Para as funções $f$ e $g$, quando está definido no conjunto $\mathbb{Z}$ de inteiros, a equação da convolução discreta é definida como:

$$f[x] * g[x] = \sum_{n=-\infty}^{\infty} f[n] \cdot g[x - n] \quad (2)$$

Quando se trata de utilizar a convolução em processamento de imagens para inteligência artificial, são necessários dois somatórios, pois temos duas dimensões, altura e largura.

$$f(x,y) * g(x,y) = \int_{\tau_1=-\infty}^{\infty} \int_{\tau_2=-\infty}^{\infty} f(\tau_1,\tau_2) \cdot g(x-\tau_1,\tau_2) d\tau_1 d\tau_2 \quad (3)$$

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x-n_1,x-n_2] \quad (4)$$

Um *kernel* é uma matriz de pesos utilizada para uma operação de multiplicação de matrizes. Essa operação é realizada diversas vezes em diferentes regiões da imagem (*patches)*. A cada aplicação, a região é alterada por um parâmetro conhecido como *stride* (Figura 3). Normalmente o *stride* possui o valor 1, o que significa que a transformação será aplicada em todos os *pixels* da imagem. O resultado dessa operação é denominado mapa de características (*feature maps*).

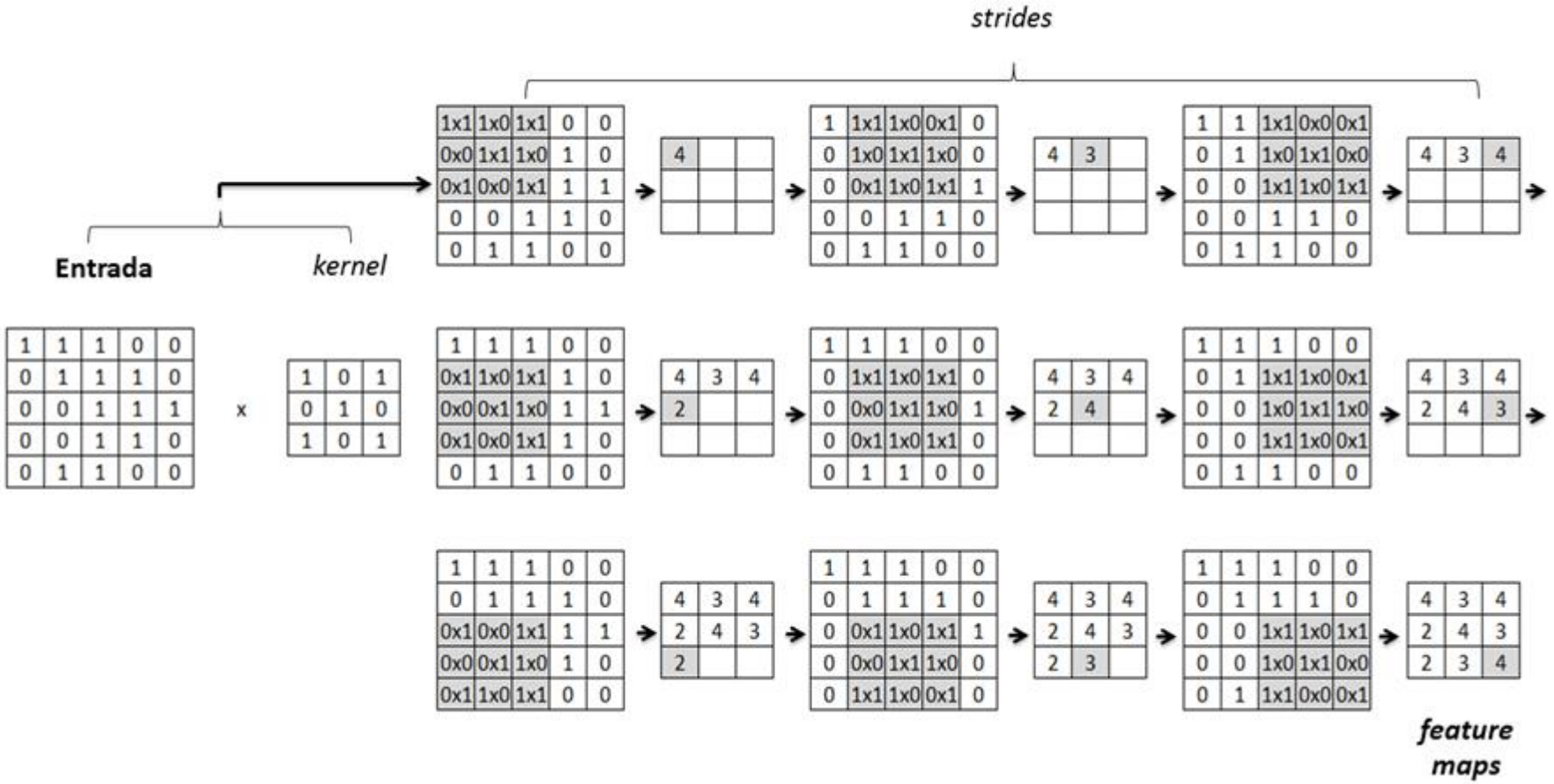


**Figura 3** Esquema do processo de convolução em uma imagem, onde uma matriz de entrada é multiplicada por um filtro (*kernel*) deslizante sobre a imagem pixel por pixel (*stride* = 1), retornando como saída o mapa de características (*feature maps).*

Por padrão, o processo de filtragem por uma matriz de *kernel* reduz a resolução da imagem original. Como normalmente usa-se pequenos *kernels*, para qualquer convolução dada, pode-se perder apenas alguns pixels, mas isso pode aumentar à medida que se aplica muitas camadas convolucionais sucessivas. Para contornar esse efeito, são adicionados *pixels* com valores iguais à zero (Figura 4) ao redor da imagem original antes da convolução, que quando multiplicado pelo *kernel* mantém a dimensionalidade na imagem resultante. Esse processo é denominado *padding* (HASHEMI, 2019).

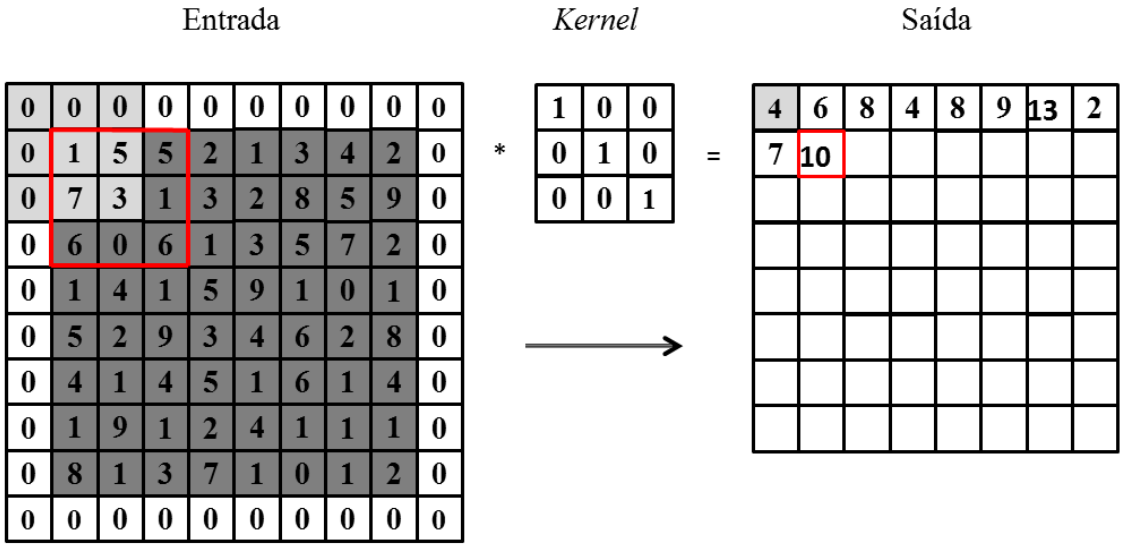


**Figura 4** *Padding* de tamanho 1 antes da convolução com um *kernel* de tamanho 3 × 3.

### 3.5.4.1 Função de ativação

Outro elemento extremamente importante nas redes neurais artificiais é a função de ativação, a qual possibilita que a rede decida se um neurônio deve ser ativado ou não. Isto significa que se uma informação recebida é relevante para o objetivo a ser alcançado, o modelo é ativado e caso contrário, essa informação é ignorada.

A função de ativação permite a não linearidade do modelo, o que torna a rede neural capaz de aprender e resolver problemas mais complexos. Para cada objetivo existe um diferente tipo de função de ativação mais adequado (KARLIK; OLGAC, 2011).

A escolha da função de ativação é fundamental para a compreensão do desempenho de uma rede neural. O processo de aplicação de uma função de ativação em uma camada de uma rede neural pode ser matematicamente representado por uma saída $z$ pertencente função de ativação $g(y)$, como segue:

$$z = g(y) = g\left(\sum_i w_i x_i + b\right) \quad (5)$$

Em que: $g$ é ativação em função dos pesos $w_i$ e entradas $x_i$ mais uma constante de polarização $b$.

No passado, as funções de ativação Sigmoide ou Função Logística (TURIAN; BERGSTRA; BENGIO, 2009) e Tangente Hiperbólica (TanH) (LECUN *et al.*, 1998) foram amplamente utilizadas, porém se tornaram ineficazes em redes neurais profundas, pois não permitem a retropropagação do modelo e, por ser uma função linear, não importa quantas camadas tenha uma rede neural, sua última camada sempre será uma função linear da primeira camada, o que transforma uma rede neural de multicamadas em uma rede neural de apenas uma camada.

Em aprendizagem profunda tornou-se mais relevante a utilização da função Unidade de Ativação Linear Retificada (*ReLU*, do inglês *Rectified Linear activation Unit*) e *Leakly Relu* (KRIZHEVSKY; SUTSKEVER; HINTON, 2017; MAAS*et al.*, 2013; NAIR; HINTON, 2010). A *ReLU* pode ser considerada como uma função linear por partes, devido sua característica de preservar propriedades da linearidade para valores maiores que zero, facilitando a otimização de modelos lineares com métodos baseados no gradiente e possuindo boa capacidade de generalização. Em contrapartida, é uma função não linear, pois valores negativos sempre são emitidos como zero. (GOODFELLOW *et al.*, 2016).

No entanto, com a evolução das redes neurais convolucionais, são necessárias funções de ativação que apresentem mais robustez e elevado grau de desempenho durante o treinamento. Para isso foram lançadas funções como a *Swish* (RAMACHANDRAN; ZOPH; LE, 2017) que apresenta melhora na precisão da classificação em redes neurais profundas, além da fácil implementação em qualquer rede neural devido a semelhança com a *ReLU*.

Outra função de ativação que se destaca é a função *Mish* proposta por Misra (2019), a qual é uma função não monotônica, autoregularizada e inspirada na propriedade de *Self-Gating* de Swish, onde a entrada não modulada é multiplicada pela saída de uma função não linear da entrada.

#### 3.5.4.2 Agrupamento

As camadas de agrupamento ou *Pooling* geralmente são usadas imediatamente após as camadas convolucionais e têm como principal função simplificar as informações na saída da camada convolucional, obtendo invariância espacial além de diminuir o

número necessário de parâmetros aprendíveis subsequente (YAMASHITA *et al.*, 2018). A camada de *Pooling* recebe cada saída de *feature maps* e as condensa. Tem-se o *Max Pooling* como exemplo, o qual reduz uma região na camada anterior aplicando uma função $u(x, y)$ ao *patch* de entrada e retorna, em uma nova camada, um único valor máximo entre a vizinhança para essa região (Figura 5). A função de *Max Pooling* é expressa por:

$$a_j = max_{N \times N}(a_i^{n \times n}) \quad (6)$$

Em que $a_j$ = valor de pixel máximo encontrado ($max_{N \times N}$) em função de uma região específica analisada ($a_i^{n \times n}$).

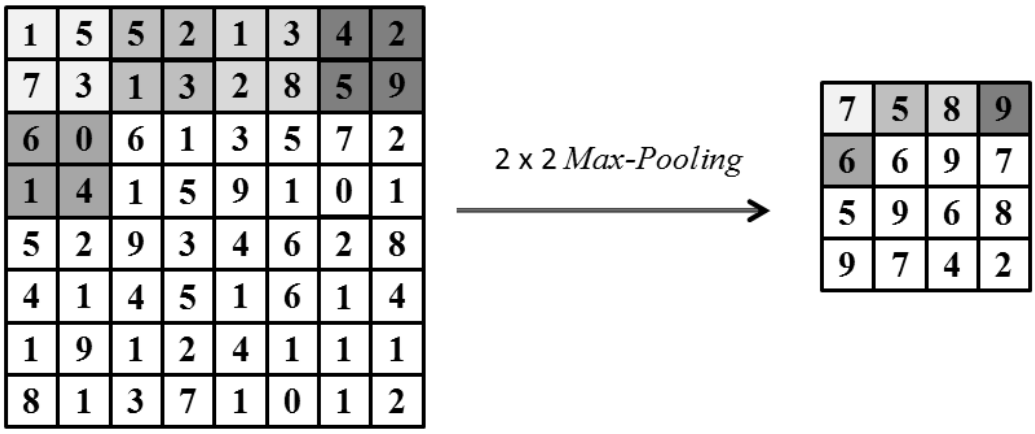


**Figura 5** *Max Poolling* $2 \times 2$ para uma entrada de $8 \times 8$ e *Stride* 2.

### 3.5.4.3 Camadas Totalmente Conectadas

As camadas totalmente conectadas (*FC*, do inglês *Fully Connected Layer*) têm como objetivo pegar os resultados do processo de convolução e *Pooling* e usá-los para classificar a imagem em um rótulo. A saída da convolução é transformada em um único vetor de valores, ou seja, são transformados em uma matriz unidimensional e ligados a uma camada totalmente conectada (Figura 2), também chamada de camada densa, onde cada vetor representa uma probabilidade de que um determinado recurso pertença a uma determinada classe. Normalmente, a *FC* possui o mesmo número de neurônios de saída quanto ao número de classes (YAMASHITA *et al.*, 2018).

### *3.5.4.4 Backbone*

*Backbone* é a "espinha dorsal" de uma *CNN*, sendo uma rede neural convolucional padrão que serve para a extração de recursos, onde as primeiras camadas detectam

recursos de níveis mais baixos como bordas, arestas e cantos e as camadas posteriores detectam recursos de níveis superiores como o rótulo de classe a que um determinado objeto pertence (CHEN *et al.*, 2018).

#### 3.5.4.5 Função de perda

A função de perda, também conhecida como função de custo, mede a compatibilidade entre as previsões de saída da rede neural por meio de propagação direta e determinados rótulos de verdade fundamental[2] (GOODFELLOW; BENGIO; COURVILLE, 2016). A função perda reduz todos os aspectos, sejam eles bons ou ruins, de um sistema altamente complexo a um único número, um valor escalar, o qual permite que as possíveis soluções sejam classificadas e comparadas (REED; MARKS, 1999).

A função de perda mais comumente usada para a classificação multiclasse é a entropia cruzada, enquanto para a regressão de valores contínuos normalmente é aplicado o erro quadrado médio (*MSE*, do inglês *Mean Squared Error*), sendo calculada a média das diferenças quadradas entre os valores preditos e os reais (YAMASHITA *et al.*, 2018).

#### 3.5.4.6 Taxa de aprendizagem

O gradiente descendente é comumente usado como um algoritmo de otimização que atualiza interativamente os parâmetros de aprendizagem da rede de modo a minimizar a perda. O gradiente da função de perda nos fornece a direção na qual a função tem a taxa de aumento mais acentuada e cada parâmetro aprendível é atualizado na direção negativa com um tamanho de passo arbitrário determinado com base em um hiperparâmetro denominado taxa de aprendizagem, ou seja, a taxa de aprendizagem representa a velocidade com que os pesos são atualizados em direção ao ponto ótimo (BUDUMA, 2015). O gradiente é, matematicamente, uma derivada parcial da perda ($\partial L$) com relação a cada parâmetro aprendível, e uma única atualização de um parâmetro é formulada da seguinte maneira:

$$w := w - \alpha * \left(\frac{\partial L}{\partial w}\right) \tag{7}$$

Onde $w$ representa cada parâmetro aprendível, $\alpha$ é taxa de aprendizagem e $L$ representa uma função de perda, sendo a taxa de aprendizagem um dos hiperparâmetros mais importantes a ser definido antes do treinamento da rede neural.

[2] É a localização real do objeto na imagem, aquela que foi sinalizada anterior ao treinamento.

Outro método utilizado para a atualização da taxa de aprendizagem e consequentemente a redução da perda é o programador de taxa de aprendizagem (Figura 6), também conhecido como "recozimento" da taxa de aprendizagem que foi proposto por Huang *et al.* (2017). Esse método usa uma função cosseno para atualização do hiperparâmetro. Por ser cíclica, a função cosseno tem como vantagem, em relação ao gradiente descendente, a possibilidade de sair dos mínimos locais mais facilmente. A taxa de aprendizagem diminuirá até o final do ciclo e então aumenta repentinamente, dando a possibilidade de extrair-se um mínimo local. Se a função a ser otimizada não for convexa, então a taxa começa a diminuir novamente e se escolhendo-se o número de ciclos é possível evitar os mínimos locais.

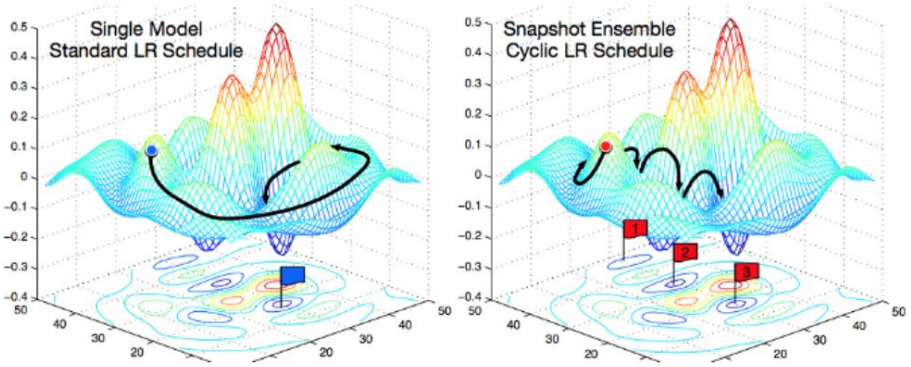


**Figura 6** Esquerda: ilustração da otimização Gradiente Descendente com uma programação de taxa de aprendizagem típica. O modelo converge a um mínimo no final do treinamento. À direita: ilustração da combinação de instantâneos. O modelo passa por vários ciclos de "recozimento" de taxa de aprendizagem, convergindo e escapando de múltiplos mínimos locais (HUANG *et al.* 2017).

#### 3.5.4.7 Sobreajuste (*Overfitting*)

Após o treinamento de uma rede neural profunda em dados rotulados conhecidos, geralmente ele é testado em dados não vistos para ver se tem capacidade de generalização. Quando o modelo tem boa capacidade de generalização, significa que ele tem um bom desempenho quando aplicado aos dados de teste (SALMAN; LIU, 2019).

O *Overfitting* refere-se a uma situação em que o modelo aprende regularidades estatísticas específicas ao conjunto de treinamento, ou seja, o modelo memoriza um ruído indesejável ao invés de aprender o sinal e, portanto, apresenta bom desempenho no conjunto de treinamento e pior desempenho em um conjunto de dados de validação. Isso é um grande desafio no processo de aprendizagem de máquina, uma vez que um modelo

não é generalizável para dados nunca vistos antes (PIOTROWSKI; NAPIORKOWSKI, 2013; ZHANG *et al.*, 2017).

Vários métodos foram propostos para minimizar o *overfitting*, como por exemplo, a obtenção de mais dados de treinamento. Quando o modelo apresenta um maior número de dados para o treinamento é provável que o modelo aumente sua capacidade de generalização. Outro exemplo é a normalização do lote (*batch normalization)*, que é um tipo de camada suplementar que normaliza de forma adaptativa os valores de entrada da camada seguinte, mitigando o risco de *overfitting*, bem como melhorando o fluxo de gradiente da rede, o que permite maiores taxas de aprendizagem e reduz a dependência da inicialização (IOFFE; SZEGEDY, 2015).

Na maioria das vezes, o conjunto de dados para o treinamento, apresenta-se de forma reduzida, pois numa situação real, nem sempre é possível adquirir grande número de amostras. Em contrapartida, quando o conjunto de dados é pequeno, o aumento de dados (*Data Augmentation*) torna-se uma ferramenta importante para evitar que o modelo apresente *Overfitting.* O aumento de dados ou *Data Augmentation* é um processo de modificação dos dados de treinamentos originais, por meio de transformações aleatórias, como inversão, translação, corte, rotação, aplicação de ruídos, contraste, nitidez e apagamento aleatório para que o modelo não veja exatamente as mesmas entradas durante as iterações de treinamento e alcance melhor desempenho (ZHONG *et al.*, 2020).

### 3.5.5 *You Only Look Once* (YOLO)

YOLO do inglês é a abreviatura para *You Only Look Once* ou "você só olha uma vez", também conhecida por YOLOv1, na sua primeira versão e foi proposta por Joseph Redmon e Ali Farhadi (2016) como uma nova abordagem para a detecção de objetos.

De acordo com Zhao *et al.* (2018), a detecção de objetos corresponde à tarefa de estimar com precisão a localização e as categorias de objetos em uma dada imagem e tem ganhado destaque devido a sua importância na análise de vídeos e compreensão de imagens.

Muitos sistemas de detecção de objetos precisam passar pela imagem mais de uma vez para poder detectar todos os objetos na imagem ou tem que passar por dois estágios para detectar esses objetos. As redes neurais detectoras de objetos pertencentes ao grupo das R-CNNs, por exemplo, criam propostas de região e examinam uma a uma para a identificação de um objeto. Em vez disso, YOLO coloca a detecção de objetos como um problema de regressão por caixas delimitadoras (*BoundingBox*) espacialmente separadas

e suas respectivas probabilidades de classes associadas. Uma única rede neural prevê a caixa delimitadora e as probabilidades de classe diretamente em imagens completas em um único estágio, ou seja, detecta todos os objetos analisando a imagem apenas uma vez. Seu design permite treinamento de ponta a ponta em velocidade de tempo real sem perder a precisão de detecção (REDMON *et al.*, 2016).

Na rede YOLO, a imagem é dividida em uma grade de tamanho $S \times S$ (Figura 7) e se o centro de um objeto cai em uma célula da grade, esta célula é responsável por detectar o objeto. Cada célula prevê caixas delimitadoras *B* (YOLO escolheu *B=2*) e para cada caixa o modelo emite uma pontuação de confiança *C* que reflete no modelo o quão confiante uma caixa delimitadora contém um objeto. Quando se usa essa pontuação, pode-se evitar que o modelo detecte planos de fundo se nenhum objeto existir na célula, ou seja, as pontuações de confiança devem ser zero. Caso contrário, deseja-se que a pontuação de confiança seja igual à intercessão sobre a união (*IoU*, do inglês, *Intersection over Union*) entre a caixa prevista e a verdade fundamental (REDMON *et al.*, 2016).

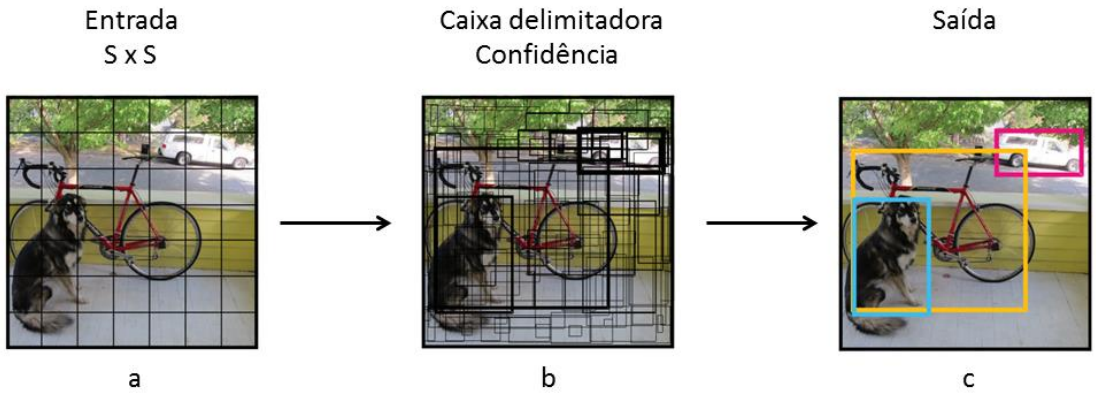


**Figura 7** Detecção de objetos usando YOLO, onde a é o grid de tamanho $S \times S$, b são as caixas delimitadoras possíveis de conter um objeto e c é a detecção final. Fonte: Adaptado de Redmon e Farhadi(2016).

YOLO usa a Supressão Não Máxima (*NMS*, do inglês, *Non Maximum suppression*) para manter a melhor caixa delimitadora. A primeira etapa na *NMS* é remover todas as caixas delimitadoras previstas que têm uma probabilidade menor que a pontuação de confiança *C*. Em seguida, as caixas delimitadoras com valor de *C* mais alto são selecionadas e remove-se as caixas delimitadoras que são muito semelhantes a esta. Esse processo é repetido até que todas as caixas delimitadoras não máximas tenham sido removidas para cada classe (REDMON *et al*, 2016).

#### 3.5.5.1 Arquitetura da rede

Assim como outras CNNs, YOLO é composto de três camadas de operações principais para detecção de objetos, que são: Convolução, *Max Pooling* e Classificação, que ocorre por meio de camadas totalmente conectadas.

YOLO usa uma rede neural inspirada no modelo GoogLeNet (SZEGEDY *et al.*, 2015) para a classificação de imagens, mas em vez dos módulos de iniciação usados pelo GoogLeNet, YOLO simplesmente usa camadas de redução $1 \times 1$ seguidas por camadas convolucionais $3 \times 3$. Possui 24 camadas convolucionais seguidas por 2 camadas totalmente conectadas. A saída da rede é um tensor de previsões $7 \times 7 \times 30$ (Figura 8).

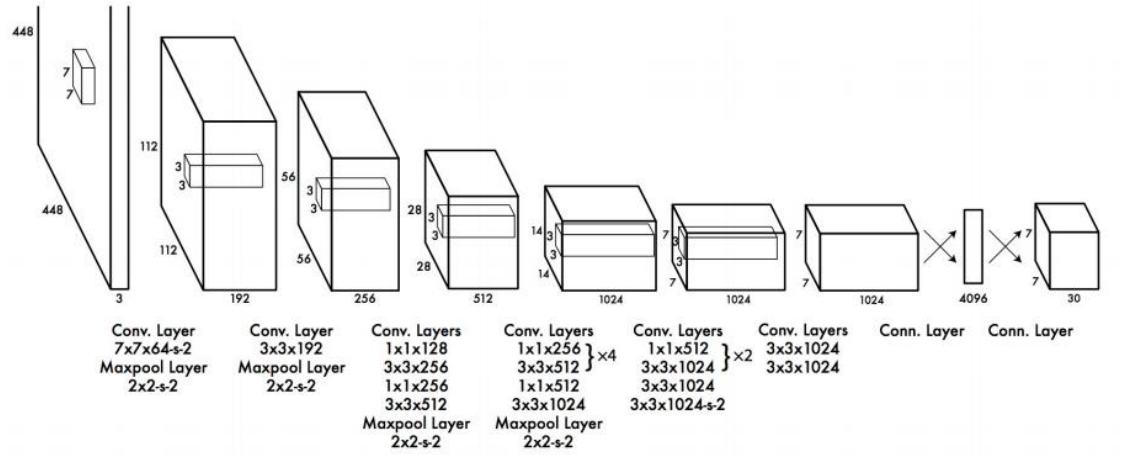


**Figura 8** Arquitetura da Rede YOLO. Fonte: Redmon e Farhadi (2016).

#### 3.5.5.2 Função de Perda

YOLO usa a Soma de Quadrado do Erro (*SSE*, do inglês *Sum-Squared Error*) para a função de perda, devido sua facilidade de otimizar. O algoritmo tenta otimizar a perda em 5 partes, onde as duas primeiras representam a perda de localização, a terceira e a quarta representam a perda de confiança e a quinta parte, representa a perda de classificação. Para o melhor entendimento é necessário considerar os seguintes pontos: I – A função de perda penaliza o erro de classificação apenas se houver um objeto naquela célula grade. II – Como temos B caixas delimitadoras para cada célula, deve-se escolher a caixa com maior *IoU* com a caixa da verdade fundamental para o cálculo da perda. Dessa forma a função penalizará a perda de localização se esta caixa for a responsável pela caixa da verdade fundamental. III – *SSE* pondera o erro de localização igualmente com o erro de classificação, o que pode não ser ideal.

#### 3.5.5.2.1 Perda de localização

A perda de localização é expressa pelas seguintes equações:

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \tag{8}$$

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \tag{9}$$

Este é uma *SSE* entre as coordenadas da caixa prevista $(x, y)$ e as coordenadas verdadeiras $(\hat{x}, \hat{y})$. É somado todas as grades da célula e para cada célula somam-se todas as caixas *B*.

Para cumprir os pontos I e II descritos, YOLO usa uma variável binária $1_{ij}^{obj}$ de modo que: $1_{ij}^{obj}$ é igual se um objeto aparece na célula *i* mais caixa delimitadora *j*, esta célula é a responsável por aquele objeto, caso contrário $1_{ij}^{obj} = 0$.

Visto que *SSE* pondera o erro de localização igualmente com o erro de classificação (ponto III), YOLO usa uma constante ($\lambda_{coord}$) para dar ao erro de localização um peso maior na função de perda.

Uma vez que a rede YOLO detecta objetos de tamanho diferentes e a *SSE* calcula o erro para caixas grandes e pequenas de forma igual. Pondera-se os valores de largura e altura $(w, h)$ da caixa para que a métrica reflita que pequenos erros em desvios de caixas grandes são menos importantes do que em caixas pequenas. Dessa forma a rede YOLO prevê a raiz quadrada da largura e altura da caixa delimitadora (Equação 9).

#### 3.5.5.2.2 O erro de confiança

O terceiro termo é o erro de confiança quando a grade possui um objeto e o quarto termo é o erro de confiança quando a grade não apresenta nenhum objeto. São representados pelas expressões:

$$\sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left(C_i - \hat{C}_i\right)^2 \quad (10)$$

Onde $C_i = 1$ e $0 \geq C_i \geq 1$.

$$\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} 1_{ij}^{noobj}\left(C_i - \hat{C}_i\right)^2 \quad (11)$$

Se não houver nenhum objeto na grade, não é preciso preocupar-se com a classificação e o erro de localização, apenas considera-se a confiança *C*, a qual deve ser igual à zero. E para isso utiliza-se a variável $1_{ij}^{noobj}$, que é igual a 1 se não há objeto dentro da célula *i* ou existe um objeto, mas a caixa *j* para esta célula não é responsável por aquele objeto, caso contrário $1_{ij}^{noobj} = 0$.

Uma vez que células da grade não contém nenhum objeto, isso empurra as pontuações de confiança dessas células para zero, ou seja, o valor da confiança da verdade fundamental. Isso pode levar o treinamento a convergir precocemente. Para amenizar esse efeito, diminui-se a perda de previsões de confiança para as caixas que não contém objetos usando parâmetros $\lambda_{noobj} = 0{,}5$.

#### 3.5.5.2.3 Perda de classificação

Para a perda de classificação deve-se somar os erros para todas as probabilidades de classes $p_i(c)$ para *S* x *S* células de grade, como segue:

$$\sum_{i=0}^{S^2} 1_i^{obj} \sum_{c\in classes}\left(p_i(c) - \hat{p}_i(c)\right)^2 \quad (12)$$

### 3.5.5.3 YOLOv2

Inicialmente, a arquitetura de rede YOLOv1 era formada por 24 camadas convolucionais mais duas totalmente conectadas. O YOLOv2 usa uma arquitetura *DCNN* (*Deep Convolutional Neural Network*) sendo totalmente convolucional, isto é, sem camadas totalmente conectadas. O *backbone* da rede é chamado de *Darknet-19*, composto por 19 camadas convolucionais e mais 11 camadas adicionais para as tarefas específicas de detecção, totalizando 30 camadas (REDMON e FARHADI, 2016).

Ao compararmos o YOLO com outros modelos de detecção de objetos, como o *FasterR CNN* e o *SSD* (REN *et al.*, 2015; LIU *et al.*, 2016), o YOLO, embora seja mais rápido, ainda apresenta maiores erros de localização. YOLOv2 é a segunda versão do YOLO, e tem como objetivo melhorar significativamente a precisão do modelo e ao mesmo tempo torná-lo mais rápido. Comparado com sua primeira versão (YOLOv2 apresenta algumas melhorias (REDMON *et al.*, 2016):

#### 3.5.5.3.1 Normalização em lote (*Batch Normalization*)

A normalização em lote é aplicada em todas as "camadas ocultas" para melhorar a convergência da rede, fornecendo mais estabilidade no treinamento e dispensa de outros métodos de regularização. Isso proporcionou um aumento de 2% na métrica de precisão.

#### 3.5.5.3.2 Classificador de alta resolução

A resolução inicial da rede tornou-se 448 x 448 pixels por dez épocas[3], enquanto na primeira versão era 224 x 224 ao treinar o classificador e 448 x 448 ao treinar detector. Esse aumento na resolução melhorou a métrica de precisão em 4%.

#### 3.5.5.3.3 Rede convolucional com caixas de âncora

YOLOv2 importa a técnica de caixas de âncora do *Faster R-CNN* (REN *et al.*, 2015), usando precedentes de caixa delimitadora para cada local na imagem, a rede só precisa prever os deslocamentos para as caixas de âncora, o que é uma tarefa mais fácil. Esse ajuste reduziu ligeiramente a métrica de precisão *mAP* (*mean Average Precision*) de 69,5% para 69,2%, porém melhorou o *Recall* de 81% para 88%, ou seja, aumentando a probabilidade de detectar todos os objetos da verdade fundamental.

A versão 2 do YOLO (REDMON *et al.*, 2016) prevê coordenadas de localização em relação a localização da célula de grade. Isso limita a verdade fundamental a ficar entre 0 e 1. A rede prevê cinco caixas delimitadoras para cada célula, sendo cinco coordenadas para cada caixa delimitadora, *tx*, *ty*, *tw*, *th* e *to*. Se a célula é deslocada do canto superior esquerdo da imagem por (*cx*, *cy)* e a caixa delimitadora anterior (caixa de âncora) tem largura e altura *pw*, *ph*, então as previsões correspondem a:

$$b_x = \sigma(t_x) + c_x \quad (13)$$

[3] Época é cada vez que uma amostra passa pela rede neural e retorna ajustando os pesos e limiar da rede.

$$b_y = \sigma(t_y) + c_y \tag{14}$$

$$b_y = \sigma(t_y) + c_y \tag{15}$$

$$b_x = p_w e^{t_w} \tag{16}$$

$$b_h = p_h e^{t_h} \tag{17}$$

$$Pr(object) * IOU\ (b, object) = \sigma(t_o) \tag{18}$$

Um exemplo dessa aplicação pode ser encontrado no estudo de Redmon e Farhadi (2016). Se consideradas duas caixas de âncora, a célula da grade (2,2) na Figura 9 irá gerar 2 caixas (a azul e a amarela). As caixas pontilhadas representam as duas caixas de âncora para essa célula.

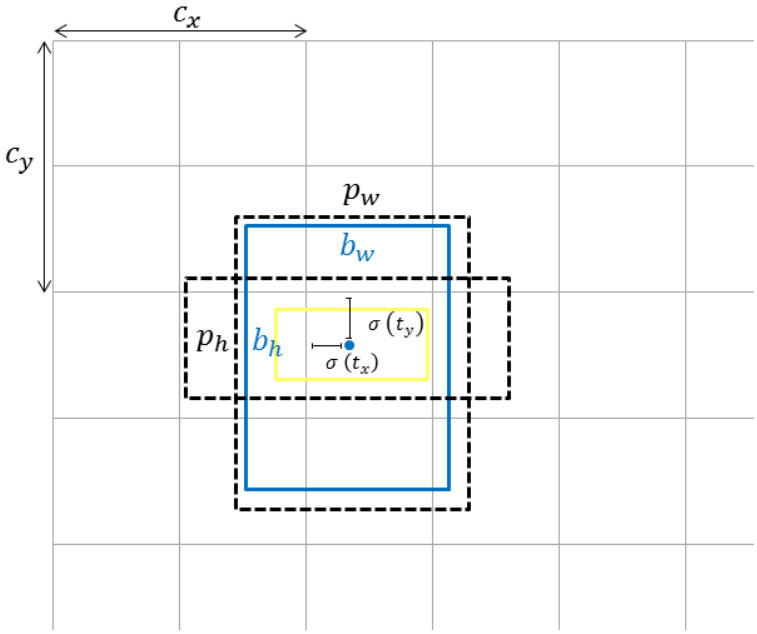


**Figura 9** Caixas delimitadoras com dimensões anteriores e predição de localização. Prevemos a largura *w* e altura *h* como compensações de centróides de cluster. Prevemos as coordenadas do centro da caixa relativa à localização da aplicação do filtro usando uma função sigmóide. Redmon e Farhadi (2016).

Ao se considerar apenas a caixa azul representada na Figura 9, em vez da caixa azul prevista ser conferida à célula da grade, como acontece na primeira versão do YOLO, em YOLOv2 atribui-se a caixa azul não apenas à célula de grade, mas também a uma das

caixas de âncora. Esta será sempre a caixa com valor de *IoU* mais alto em relação à caixa da verdade fundamental (REDMON e FARHADI, 2016).

#### 3.5.5.3.4 *Clusters* de dimensionalidade

Existem dois problemas ao utilizar as caixas de âncora com YOLO. Primeiro, as dimensões das caixas de ancora são escolhidas manualmente. A rede pode aprender para ajustar as caixas de forma mais adequada, mas ao escolher anteriores melhores para a rede começar, é possível tornar o aprendizado mais fácil e prever boas detecções (REDMON e FARHADI, 2016). Por tanto, ao invés de escolher anteriores manualmente, Redmon e Farhadi (2016) executaram o algoritmo de agrupamento por k-médias nas caixas delimitadoras do conjunto de treinamento, o que possibilita encontrar automaticamente bons antecedentes.

O segundo problema ao utilizar as caixas de âncora, está relacionado à instabilidade do modelo, principalmente nas camadas iniciais. Essa instabilidade vem da previsão dos locais para a caixa de âncora. Dessa forma qualquer caixa de âncora pode acabar em qualquer ponto da imagem, independente de qual local a rede previu a caixa. Com inicialização aleatória, o modelo leva muito tempo para estabilizar e prever compensações sensatas. Por conta disso, Redmon e Farhadi (2016) utilizaram uma função de ativação logística, a qual prevê coordenada de localização em relação à grade da célula, e proporciona uma restrição no número de previsões de caixas de âncora para cada célula de grade, tornando a rede mais estável.

#### 3.5.5.3.5 Recursos refinados

A primeira versão do YOLO não conseguia lidar bem com objetos de pequena escala. Para contornar isso, Redmon e Farhadi (2016) propuseram adicionar uma "camada de passagem" que usa recursos de camadas com um mapa de recursos mais refinado. Esta camada empilhou recursos de alta resolução com os de baixa resolução em diferentes canais, fornecendo maior robustez na detecção de objetos menores.

#### 3.5.5.3.6 Treinamento em múltiplas escalas

Outro feito importante proporcionado por Redmon e Farhadi (2016) foi o aumento da robustez do modelo ao implementar o suporte de imagem de tamanhos diferentes. Para conseguir isso, a entrada da rede é alterada a cada poucas iterações, selecionando aleatoriamente um novo tamanho de dimensão de imagem durante o treinamento. As

dimensões variam de múltiplos de 32, devido ao fator de *Downsampling* da rede, tendo como tamanho mínimo 320 x 320 e máximo 608 x 608 *pixels*. Este regime força a rede a aprender bem em uma variedade de tamanho de entradas.

#### 3.5.5.4 YOLOv3

Anos mais tarde após o lançamento do YOLOv2, Redmon e Farhadi (2018) lançaram o YOLOv3 com uma rede composta por 106 camadas, destas, 53 para o *backbone* (*Darknet-53*) e os outros 53 para tarefas de detecção de objetos, ainda sendo uma rede neural totalmente convolucional. Em comparação com o YOLOv2 os autores aplicaram algumas modificações:

##### 3.5.5.4.1 Classificação em multi-rótulos

A ativação *softmax* no previsor foi substituída pelo classificador logístico independente com funções binárias de perda cruzada. Essa alteração permitiu resolver domínios mais complexos como a classificação de multi-rótulos, ou seja, um objeto pode ser anexado a mais de uma classe, como por exemplo, o agrupamento de gêneros de plantas em uma determinada família.

##### 3.5.5.4.2 Previsão em três escalas

Três escalas diferentes são previstas nesta versão, para objetos de pequena, média e grande escala. Essa abordagem melhora principalmente a detecção de objetos pequenos. Para cada escala há uma grade diferente: 13 x 13 (Figura 10a) para objetos grandes, 26 x 26 (Figura 10b) para objetos médios e 52 x 52 (Figura 10c) para objetos pequenos.

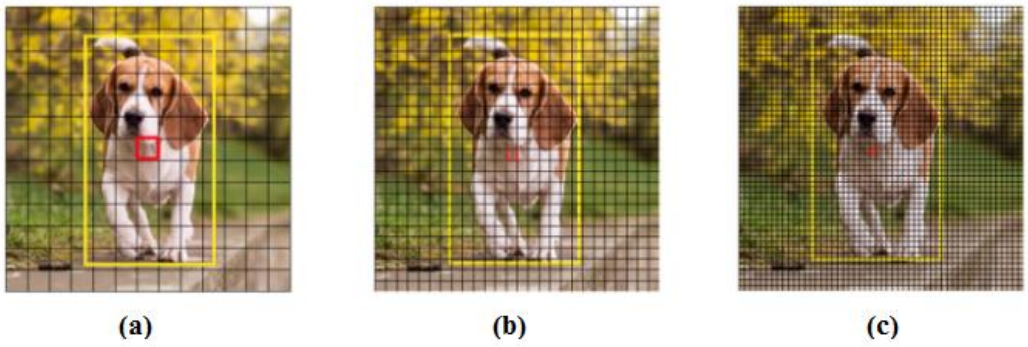


**Figura 10** Previsão de caixa delimitadora em três escalas diferentes. a) *grid* 13 x 13, b) *grid* 26 x 26 e c) *grid* 52 x 52 (REDMON e FARHADI, 2018).

#### 3.5.5.4.3 Aumento da previsão de âncoras

Em vez de 5 âncoras por célula, a quantidade é aumentada para 9, onde há 3 âncoras para cada grade. Embora YOLOv2 pudesse prever 845 caixas delimitadoras, com esta nova configuração com YOLOv3 é possível prever 10.647 caixas delimitadoras.

### 3.5.5.5 YOLOv4

Publicado por Alexey Bochkovsky *et al.* (2020) o artigo "YOLOv4: *Optimal Speed and Accuracy of Object Detection*" apresenta a quarta versão do YOLO, a qual possui uma melhoria na velocidade de inferência e acurácia, além de ser mais eficiente no processamento em Unidades de Processamento Gráfico (*GPU*, do inglês *Graphics Processing Unit*), pois foi otimizada para utilizar menos memória.

Os autores também verificaram a influência dos métodos conhecidos por "Bolsa de Brindes" (do inglês, *Bag-of-Freebies*). Isso significa que os autores realizaram mudanças na estratégia de treinamento, obtendo melhor precisão sem aumentar o custo de inferência (BOCHKOVSKY *et al.*, 2020). Um bom exemplo da aplicação deste método é o aumento de dados. O objetivo do aumento de dados é aumentar a variabilidade das imagens de entrada, de modo que o modelo de detecção de objeto projetado tenha maior robustez às imagens obtidas em diferentes ambientes.

#### 3.5.5.5.1 Estrutura da rede YOLOv4

Para compreender melhor a estrutura da rede neural YOLOv4, pode-se fazer uma analogia ao corpo humano, separando-a em três partes principais: a espinha dorsal, pescoço e cabeça.

Todos os detectores de objetos (Figura 11) utilizam uma imagem de entrada e compactam recursos por meio de um *backbone* de *CNN*. Na classificação de imagens, esses *backbones* compõe o fim da rede e a previsão pode ser feita a partir deles. Na detecção de objetos, várias caixas delimitadoras precisam ser criadas junto com a classificação, deste modo, as camadas de recursos do *backbone* convolucional precisam ser misturadas e mantidas à luz umas das outras. Essa combinação de camadas de recursos de *backbone* ocorre no "pescoço", (do inglês, *neck*) e a detecção ocorre na "cabeça", (do inglês, *head*) (BOCHKOVSKIY *et al.*, 2020).

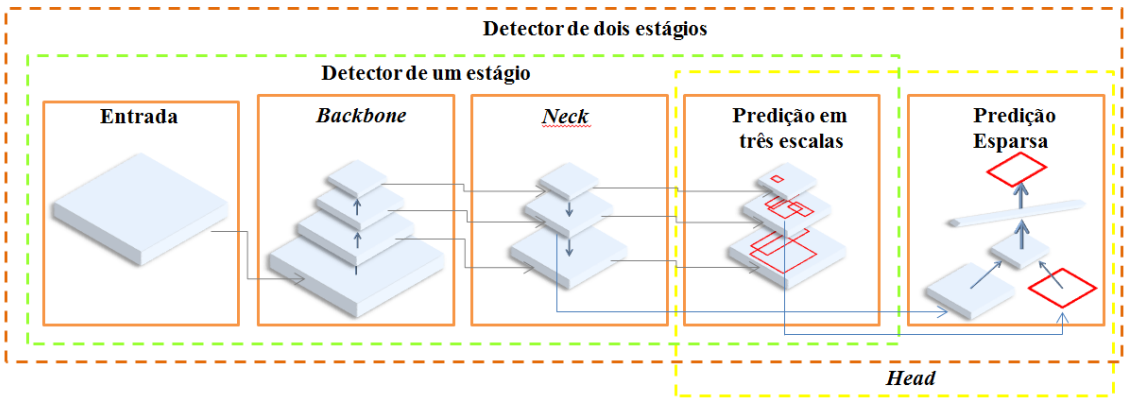


**Figura 11** Estrutura de um detector de objeto (BOCHKOVSKIY *et al.*, 2020).

#### 3.5.5.5.1.1 *Backbone*

A rede de *backbone* é geralmente pré-treinada na classificação *Imagenet* (DENG *et al.*, 2009) ou *MS COCO* (LIN *et al.*, 2014). O pré-treinamento significa que os pesos da rede já foram adaptados para identificar características relevantes em uma imagem, embora eles sejam ajustados na nova tarefa de detecção de objetos. Os *backbones* *CSPResNext50* (WANG *et al.*, 2020), *CSPDarknet53* (WANG *et al.*, 2020) e *EfficientNet-B3* (TAN *et al.*, 2019), foram testados por Bochkovsky *et al.* (2020) para o detector YOLOv4.

O *CSPResNet50* e o *CSPDarknet53* são baseados em *DenseNet* (HUANG *et al.*, 2017). O *DenseNet* foi projetado para conectar camadas em redes neurais convolucionais com as seguintes motivações: para aliviar o problema de desaparecimento do gradiente no caso de redes muito profundas, para reforçar a propagação de recursos, encorajar a rede a reutilizar recursos e reduzir o número de parâmetros de rede (HUANG *et al.*, 2017).

Em *CSPResNet50* e *CSPDarknet53*, o *DenseNet* foi editado para separar o mapa de feições da camada base, copiando-o e enviando uma cópia através do bloco denso e enviando a outra para o próximo estágio. Dessa forma, foi possível remover os gargalos computacionais no *DenseNet* e melhorar o aprendizado (WANG *et al.*, 2020). Já o EfficientNet foi projetado pelo *Google Brain Team*[4] para resolver principalmente o problema de dimensionalidade das *CNN*s (TAN *et al.*, 2019).

Os autores do YOLOv4, no entanto, testaram também, outras redes na configuração de detecção de objetos, porém concluíram que o *CPSDarknet53* é o melhor para implementar a rede de *backbone*.

[4]*Google Research, Brain Team, Mountain View, CA.*<https://research.google/teams/brain/>

#### 3.5.5.5.1.2 *Neck*

A próxima etapa na detecção de objetos é misturar e combinar os recursos formados no *backbone* de rede neural convolucional para se preparar para a etapa de detecção. Os componentes do *neck* geralmente fluem para cima e para baixo entre as camadas e conectam apenas algumas camadas no final da rede convolucional (BOCHKOVSKIY *et al.*, 2020).

Conforme a imagem passa pela rede, a complexidade dos recursos em camadas consecutivas aumenta, desde as representações dos recursos de baixo nível, como bordas e texturas, até a codificação de partes inteiras do objeto (recursos de alto nível), como boca e nariz, por exemplo. No entanto, como visto na seção 3.5.4.2, a resolução espacial dos mapas de características diminui devido a várias convoluções e *pooling* que ocorrem no processo.

Para contornar esse efeito o *neck* entra em ação. Os Bochkovskiy *et al.* (2020) sugerem a rede *PANet* (LIU *et al.*, 2018) para a agregação dos recursos. A rede *PANet* emprega um caminho de cima para baixo para combinar recursos semanticamente ricos de camadas de alto nível com informações de localização precisas que residem nos mapas de características de alta resolução das camadas inferiores e, ainda, emprega um caminho de baixo para cima, fazendo uso de um caminho mais curto com conexões laterais limpas do nível inferior para o superior, o que permite maior facilidade no fluxo de informações (LIU *et al.*, 2018).

Além disso, o YOLOv4 adiciona um bloco *SPPNet* (HE *et al.*, 2015) após o *CSPDarknet53* para aumentar o campo receptivo e separar os recursos mais importantes do *backbone* (BOCHKOVSKIY *et al.*, 2020). O *SPPNet* é uma estratégia de "*pooling* de pirâmide espacial", a qual faz com que a *CNN* elimine a exigência de um tamanho fixo de imagem de entrada, aumentando a precisão do reconhecimento, além de dar robustez à possíveis deformações de objetos (HE *et al.*, 2015).

#### *3.5.5.5.1.3 Head*

O YOLOv4, assim como na sua terceira versão, utiliza a detecção baseada em caixa de âncoras e três níveis de granularidade (escalas de detecção) como *head*. Dessa maneira, é possível a detecção de objetos de tamanho pequenos com maior acurácia e rapidez (BOCHKOVSKIY *et al.*, 2020).

### 3.5.6 Transferência de conhecimento (*Transfer Learning*)

A maioria dos problemas que envolvem visão computacional usa conjunto de dados extremamente grandes, como por exemplo, a base de dados *MS COCO*, *ImageNet*, *PASCAL* e *SUN* (DENG *et al.*, 2009; EVERINGHAM *et al.*, 2010; XIAO *et al.*, 2010; XIAO *et al.*, 2010; DOLLÁR *et al.*, 2011). Na maioria das vezes, a base de dados de entrada não é grande o suficiente para o treinamento de uma CNN, levando muitas vezes ao problema de *overfitting*. Uma maneira de contornar esse problema é a utilização do método de transferência de conhecimento (do inglês, *transfer learning*), o qual tem como princípio a utilização de um conhecimento já obtido em domínios específicos e, em seguida, aplicar esse conhecimento para resolver problemas de diferentes domínios (PAN e YANG, 2010).

Um exemplo da utilização de *transfer learning* é o aproveitamento dos pesos pré-treinados, sendo estes, geralmente, os pesos das camadas iniciais de um modelo, ou seja, as camadas responsáveis por extração de características como borda, cantos e forma, aplicáveis em qualquer imagem, tendo assim um ganho significativo no treinamento em relação a um modelo que é treinado desde o início (PAN e YANG, 2010).

## 3.6 Trabalhos correlatos

Os inventários florestais, na maioria das vezes, são realizados com base na mensuração e contagem das árvores *in situ* no campo (ROCHA, 2004). Isto exige muito tempo e recursos para a execução o que implica na dificuldade de ser aplicado em áreas extensas (LIANG *et al.*, 2016, FERREIRA *et al.*, 2020), especialmente, em florestas com alta diversidade e complexidade como é o caso das florestas amazônicas. O Sensoriamento Remoto expandiu a escala e reduziu o custo de inventários florestais usando varreduras a laser, satélite e imagens aéreas (WULDER *et al.*, 2012; BARRET *et al.*, 2016; WEINSTEIN *et al.*, 2019). Adicionalmente, com o avanço das tecnologias associadas a visão computacional, novas abordagens têm se mostrado adequadas para a identificação espécies em áreas de florestas (DOS SANTOS *et al.*, 2017; FERREIRA*et al.*, 2020).

Tagle Casapia *et al.* (2020) desenvolveram um método baseado em segmentação da imagem por crescimento de regiões para a identificação e quantificação da abundância de palmeiras economicamente importantes no noroeste do Peru usando imagens de RGB a partir de VANTs.

Weinstein *et al.* (2019) usaram redes neurais de aprendizado profundo semisupervisionado e imagens áereas *RGB* para detecção individual de copas de árvores de Carvalho e Pinus em uma Floresta Aberta na California. Os autores alcançaram precisão média de 61% para os dados anotados visualmente e a taxa de detecção foi de 82% para as árvores coletadas no campo. Também, Barré *et al.* (2017) desenvolveram um sistema de aprendizado profundo para classificar espécies com base em imagens de folhas usando CNN. O sistema conseguiu classificar com precisão de 83,7% as espécies e plantas a partir de imagens obtidas por *smartphones* no nordeste dos Estados Unidos.

Freudenberg *et al.* (2019) alcançaram precisões superiores a 90% utilizando uma rede neural profunda do tipo U-Net, para detectar dendezeiros em grandes plantações em Jambi na Indonésia e coqueiros na região metropolitana de Bengaluru na Índia. Mubin *et al.* (2019) utilizaram duas redes neurais convolucionais para detectar dendezeiros jovens e adultos em plantações na Malásia, encontrando precisões gerais de 95,11% e 92,96% para jovens e adultos, respectivamente.

# 4 MATERIAL E MÉTODOS

## 4.1 Localização e caracterização física da área de estudo

O presente estudo foi conduzido no campo experimental do Acre da Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA ACRE), localizada no município de Rio Branco, Região do Baixo Acre, Estado do Acre, no trecho Rio Branco – Porto Velho, RO, à margem direita da BR 364 - Km 14, com coordenadas geográficas de 10º01’22”S e 67º40’3”W (Figura 12). A EMBRAPA ACRE possui aproximadamente 1.200 hectares, sendo 960 hectares coberto por floresta natural.

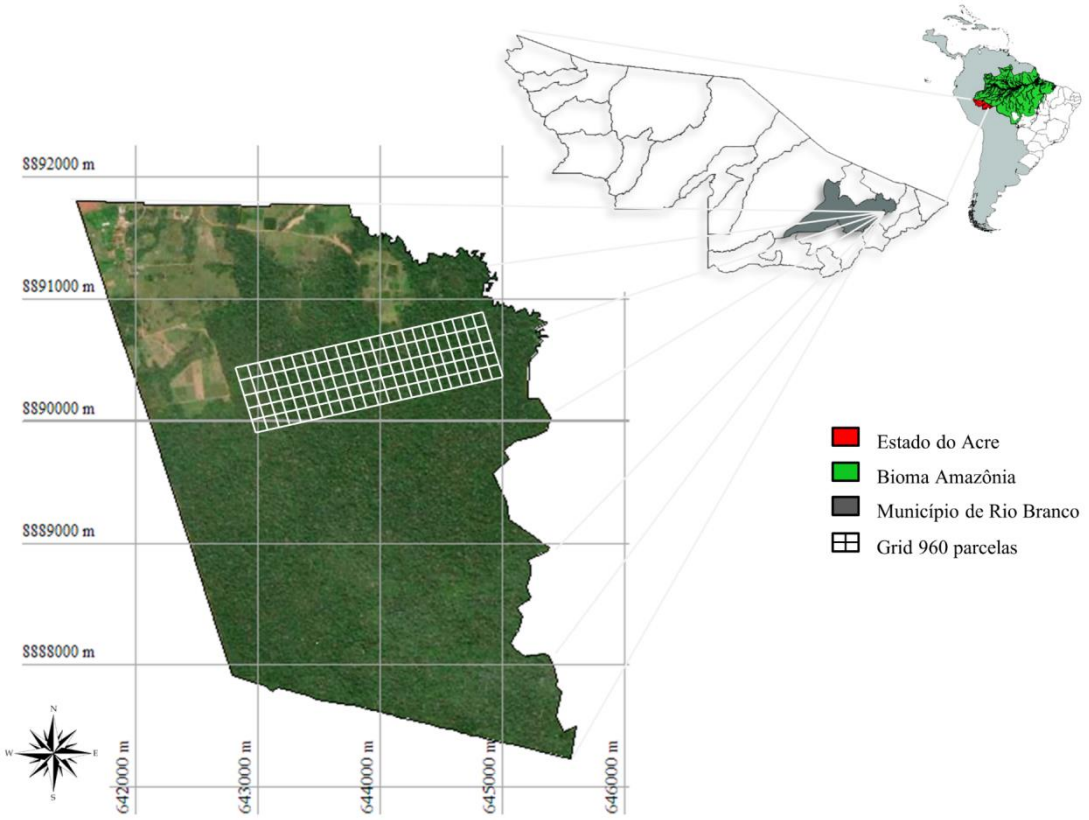


**Figura 12** Localização da área de estudo - Campo Experimental da EMBRAPA ACRE.

O clima da região, segundo classificação de Köppen é do tipo “Am”, clima tropical de monções, com período seco anual de 3 meses (IBGE, 1997; CUNHA e DUARTE, 2005; ALVARES, 2013), com temperatura e precipitação média anual de 24,3ºC e 1.950 mm, respectivamente. A hidrografia é caracterizada por uma rede de drenagens derivada do igarapé Liberdade, que corta a área no sentido Sul/Norte, situado na bacia hidrográfica do rio Acre. A altitude varia de 147 a 210 metros, com relevo plano

a ondulado e solo com alta concentração de argila e de baixa permeabilidade (RODRIGUES *et al.*, 2001).

## 4.2 Caracterização da vegetação

No campo experimental da EMBRAPA ACRE foi observado a ocorrência de Floresta Ombrófila Densa (FOD) e Floresta Ombrófila Aberta (FOA), passando por um gradiente que vai de um dossel fechado, com baixa densidade de regeneração e incidência de luz e com presença de espécies de grande porte como, por exemplo, *Bertholletia excelsa* (Bonpl.), *Dipteryx odorata* (Aubl.) Willd, *Apuleia leiocarpa* (Vogel.) JFMacbr.; indo para uma vegetação mais aberta, caracterizada por uma vegetação de porte médio, presença de palmeiras, cipós, bambu e indivíduos arbóreos dominantes distribuídos de forma mais casual, permitindo a incidência maior de luz, com maior abertura do dossel (VELOSO *et al.*, 1991; RODRIGUES *et al.*, 2001). Existem, no campo experimental da EMBRAPA ACRE, aproximadamente 235 espécies arbóreas pertencentes a 65 famílias botânicas diferentes, com volume médio de 130 $m^3$ $ha^{-1}$, abundância média de 32 indivíduos por hectare e área basal média de 10 $m^2$ $ha^{-1}$, isto referente aos indivíduos com diâmetro a 1,30 metros do solo acima de 40 cm (PAPA, 2018).

## 4.3 Obtenção das imagens *RGB*

Os dados utilizados para o desenvolvimento deste estudo foram disponibilizados pelo projeto de pesquisa GEOFLORA da EMBRAPA ACRE, financiado com recursos da União previstos no orçamento do Ministério da Agricultura, Pecuária e Abastecimento.

As imagens aéreas foram adquiridas usando uma *RPA* da marca *DJI Phantom 4 Professional*, no mês de março de 2017. A câmera *RGB* (bandas *Red, Green* e *Blue*) embarcada na *RPA*, captura imagens de 20 megapixels e possui uma lente de foco automático de 24 mm. Para assegurar que as imagens obtidas fossem orientadas na ortogonal, obedecendo ao nadir, a câmera foi conectada a um sistema de estabilização eletrônica de *cardan* de três eixos. A altura de voo foi de 120 metros acima do dossel da floresta a uma velocidade de cruzeiro de 13,0 m $s^{-1}$, derivando uma distância de amostra do solo (*GSD*, do inglês *Ground Sample Distance*) de 4,3 cm. Ao total foram capturadas 1423 imagens com 86% de sobreposição lateral e longitudinal em oito voos consecutivos (FERREIRA *et al.*, 2020). Todos os voos foram autorizados pelo sistema de Solicitação de Acesso de Aeronaves Remotamente Pilotadas (SARPAS).

Anteriormente aos voos foram estabelecidos nas bordas do fragmento florestal três pontos de controle no solo (*GCPs*, do inglês, *Ground Control Points*). Para cada ponto de controle, foi instalado um receptor *GNSS* de dupla frequência que, durante 241 minutos, coletou dados de *GPS* e *GLONASS*, obtendo após processamento uma precisão média vertical e horizontal de 10 e 3 cm, respectivamente. Finalmente, o algoritmo de características invariáveis em escala (*SIFT*), disponível no *software Pix4D*, foi utilizado para gerar um ortomosaico da área de estudo.

### 4.4 Demarcação das copas individuais de palmeiras

A delimitação das copas individuais das palmeiras foi realizada a partir da fotointerpretação das imagens *RGB* de alta resolução proveniente da *RPA*. Primeiramente, foi realizado um trabalho de campo a fim de avaliar as características fenotípicas de cada espécie de palmeira, o que permitiu o estabelecimento de uma chave de identificação botânica, auxiliando no trabalho de fotointerpretação.

O ortomosaico referido na seção 4.3 foi analisado na composição de bandas de cores reais e em uma escala de 1:50, o que possibilitou a localização e geração dos polígonos com a forma das copas individuais. Na sequência, cada polígono foi analisado por uma equipe de cinco foto-intérpretes os quais detinham conhecimento especializado no reconhecimento de palmeiras, que após consenso, identificaram as espécies de palmeiras individuais.

A partir desta análise visual foram identificadas quatro espécies de palmeiras, sendo 84 indivíduos de *Attalea butyracea* Mutis exLf Wess.Boer (jací), 403 de *Euterpe precatoria* Mart. (açaí), 263 exemplares de *Iriartea deltoidea* Ruiz & Pav. (paxiubão), 43 de *Oenocarpus bataua* Mart. (patauá). Além destes, 221 indivíduos foram classificados como "Palmeiras Não Identificadas".

Posteriormente, utilizando o software *QGIS* (*QGIS Develvelopment Team*, 2019) foi gerado um *grid* de 960 parcelas de 37,5 m x 37,5m (0,1406 $ha^{-1}$) sobre a ortofoto e com o auxílio da ferramenta "*crop raster*" dividiu-se a imagem sobre o *grid*, de forma a obter um número maior de entradas para o modelo durante o treinamento, além de facilitar a identificação de objetos menores pelo algoritmo. Das 960 imagens geradas, foram selecionadas 430 imagens para o processo de aprendizagem, sendo que nestas, foi verificado maior presença de palmeiras pela interpretação visual. Para as inferências e estimativa da densidade populacional foram utilizadas todas as 960 imagens.

## 4.5 Rotulagem de dados

Para a rotulagem dos dados, foram estabelecidas cinco classes de palmeiras (Tabela 1). Inicialmente foi criado um retângulo envolvente (*Bounding Box*) ao redor de cada palmeira a partir da ferramenta "*Polygon from layer extent"* localizada no menu "*Extent"* da caixa de ferramentas de processamento do *QGIS*. Na sequência, as palmeiras rotuladas foram atribuídas a cada imagem correspondente e às suas respectivas classes. Então, foram gerados arquivos de texto (.txt) com o mesmo nome que a imagem correspondente, sendo que cada linha representava um "*Bounding Box*" de cada palmeira. Posteriormente, as imagens e os arquivos de textos foram carregados na plataforma *Make Sense*, a fim de gerar os arquivos rotulados no formato de entrada para a rede neural YOLO. A *Make Sense* é uma ferramenta de código fonte aberta e gratuita sob a licença GPLv3 e que não requer nenhuma instalação avançada, mas apenas um navegador *web* para executá-la.

**Tabela 1** Número de palmeiras e *Bounding Boxes* anotados nos respectivos rótulos de classe.

| **Classe** | **Rótulo** | ***N*** | ***BBox*** | ***BBox (%)*** | **Ø$_{mín.\ copa}$ (m)** | **Ø$_{médio\ copa}$ (m)** | **Ø$_{máx.copa}$ (m)** |
|---|---|---|---|---|---|---|---|
| 0 | *A. butyracea* | 84 | 96 | 8,74% | 3,27 | 11,20 | 23,62 |
| 1 | *E. precatoria* | 403 | 424 | 38,62% | 1,84 | 3,51 | 5,67 |
| 2 | *I. deltoidea* | 263 | 291 | 26,50% | 2,41 | 4,91 | 7,75 |
| 3 | N.I. | 221 | 244 | 22,22% | 1,56 | 7,08 | 21,91 |
| 4 | *O. bataua* | 43 | 43 | 3,92% | 7,16 | 11,53 | 20,52 |

*N*= Número de indivíduos; *BBox*= Número de *Bounding Boxes*; N.I. = palmeiras não identificadas; **Ø $_{mín.\ copa}$ =** diâmetro mínimo de copa; **Ø $_{médio\ copa}$** = diâmetro médio de copa; **Ø $_{máx.\ copa}$**=diâmetro máximo de copa.

Pelo fato do ortomosaico original ter sido segmentado em um *grid* sistemático, e sendo cada célula do *grid* uma nova imagem, alguns indivíduos foram divididos ao meio, ficando metade em cada imagem. Portanto, como a anotação foi realizada para cada imagem, o número de *Bounding Boxes* obviamente aumentou em relação à quantidade de indivíduos identificados no ortomosaico original. Os valores dos diâmetros mínimos, médios e máximos (Tabela 1) foram calculados com base nos *Bounding Boxes* das palmeiras do ortomosaico original, ou seja, apenas com relação as palmeiras inteiras. Outro fator importante a ser considerado é o grau de sobreposição das copas, uma vez que árvores dominantes no dossel da floresta podem cobrir parcialmente a copa de algumas palmeiras e influenciar na determinação do diâmetro de copa.

### 4.5.1 Descrição da caixa delimitadora (*Bounding Box*)

Para descrever a caixa delimitadora (Figura 13 e Figura 14), foi necessária a extração de cinco variáveis: as coordenadas $b_x$ e $b_y$ do centro da caixa delimitadora; a largura *w*; a altura *h* da caixa delimitadora, além do nome da classe pertencente.

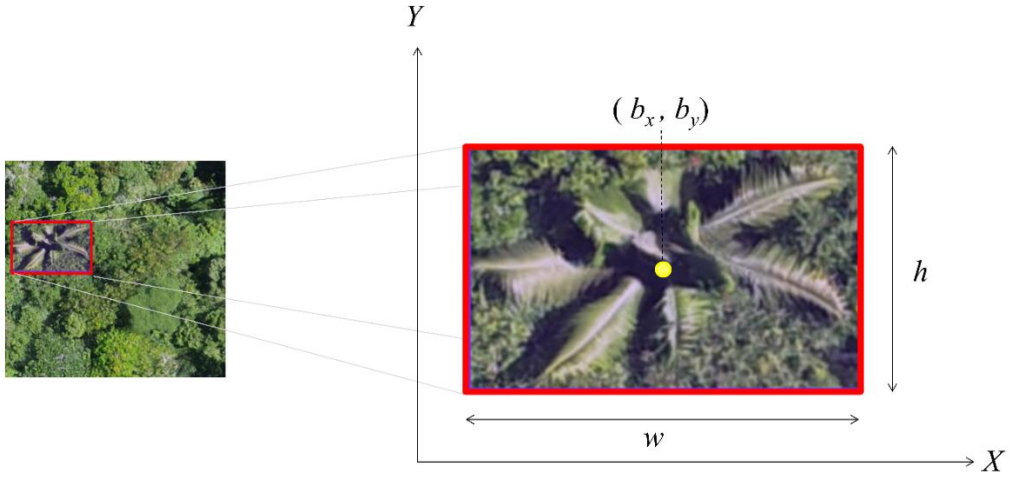


**Figura 13** Variáveis componentes de uma caixa delimitadora (*Bounding Box*). ($b_x$, $b_y$) são as coordenadas *X* e *Y* correspondente ao centro da caixa delimitadora, *w* representa a largura e *h* a altura.

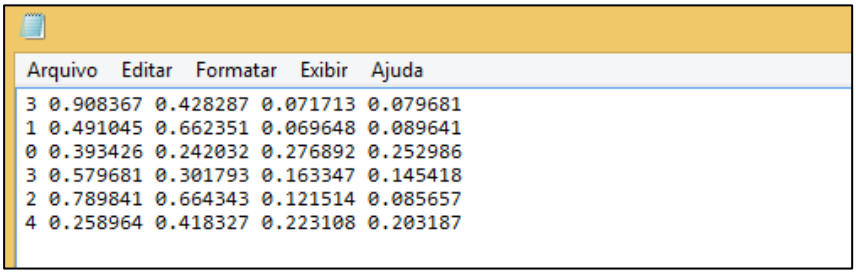


**Figura 14** Arquivo ".txt" com as informações provenientes da anotação, onde a primeira coluna faz referência ao nome da classe, a segunda e a terceira são as coordenadas relativas *X* e *Y* do centro da caixa delimitadora e a quarta e quinta coluna são, respectivamente, a largura *w* e altura *h*, da caixa delimitadora.

### 4.6 Customização dos dados

Foram estabelecidos dois cenários (Tabela 2) diferentes para o treinamento. No Cenário I, foi treinado o YOLOv4 com sua configuração original, a qual foi proposta por Bochkovsky *et al.* (2020), porém sem aumento de dados. Para o Cenário II, foi efetuado um pequeno ajuste na configuração e, principalmente, foi realizado um aumento da base de dados original, para dar maior suporte ao treinamento e validação.

**Tabela 2** Configuração do YOLOv4 para os dois cenários testados.

| **Configuração** | **Cenário I** | **Cenário II** |
|---|---|---|
| *Batch* | 64 | 64 |
| *subdivisions* | 12 | 12 |
| *Width* | 416 | 416 |
| *Height* | 416 | 416 |
| *channels* | 3 | 3 |
| *momentum* | 0,949 | 0,949 |
| *Decay* | 0,0005 | 0,0005 |
| *learning_rate* | 0,001 | 0,001 |
| *burn_in* | 1000 | 1000 |
| *max_batches* | 10000 | 10000 |
| *Policy* | *steps* | *steps* |
| *Steps* | 8000,0; 9000,0 | 8000.0; 9000.0 |
| *Scales* | 0,1; 0,1 | 0,1; 0,1 |
| *mosaic* | 1 | 1 |
| ***label_smooth_eps*** | **--** | **0,1** |
| ***train_size*** | **344** | **2000** |
| ***validation_size*** | **86** | **86** |

Em que: *Batch* = tamanho do lote; *subdivisions* = tamanho do mini-lote; *width* = largura da imagem em *pixels*; *height* = altura da imagem em *pixels*; *channels* = bandas *RGB*; *momentum* = acúmulo de movimento, o quanto o histórico afeta a posterior mudança dos pesos; *decay* = elimina o desequilíbrio no conjunto de dados; *learning_rate* = taxa de aprendizagem; *burn_in* = aumento gradual da taxa de aprendizagem; *max_batches* = número máximo de iterações; *policy* = política para alterar a taxa de aprendizagem; *steps* = nesse número de iterações a taxa de aprendizagem é multiplicada pelo fator de escala; *scales* = fator de escala; *mosaic* = combinação de imagens; *label_smooth_eps* = Suavização de rótulo; *train_size* = tamanho do conjunto de dados de treino (aumentado para cenário II); *validation_size* = tamanho do conjunto de dados de validação.

#### 4.6.1 Ajuste de dimensão e distribuição dos dados

Para o treinamento de ambos os cenários, utilizamos a Rede Neural Convolucional de detecção de objetos YOLOv4 (BOCHKOVSKY *et al.*, 2020). A rede YOLOv4 aceita variações de tamanhos de entrada, mas tem como restrição o tamanho que deve ser múltiplo de 32, porém quanto maior o tamanho da entrada, maior custo de computação é necessário. Com base nisso, foi adotado como entrada dos dados, por padrão, imagens

com resolução de 416 x 416 *pixels*. Por este motivo, cada imagem de entrada sofreu um ajuste em sua dimensionalidade, sendo redimensionada de aproximadamente 800 x 800 *pixels* para o tamanho final de 416 x 416 *pixels*. Observou-se que mesmo com este ajuste a imagem não perdeu suas características importantes para a aprendizagem do modelo.

Os conjuntos de dados, no Cenário I e II, foram subdivididos e organizados aleatoriamente em: conjunto de treinamento (80%), conjunto de validação ou teste (20%). No conjunto de treinamento estão as imagens utilizadas para o aprendizado do modelo, já o conjunto de validação são as imagens utilizadas para validar o experimento, ou seja, é a amostra utilizada para fornecer uma avaliação imparcial do modelo no conjunto de dados de treinamento durante o ajuste dos pesos, sendo que estas ainda não foram, até então, visualizadas pelo modelo.

### 4.6.2 *Data Augmentation*

A precisão da previsão dos modelos de aprendizado profundo supervisionado depende amplamente da quantidade e da diversidade de dados disponíveis durante o treinamento. Muitas vezes, ao trabalhar com tarefas específicas, como identificar palmeiras em uma floresta nativa com grande diversidade, como é o caso da Floresta Ombrófila Aberta, é difícil se obter grandes quantidades de dados necessários para treinar o modelo.

Dessa forma, para o Cenário II, anteriormente e durante o treinamento, foi aplicada a técnica de aumento de dados ou *data augmentation,* com o objetivo de alterar as imagens de treinamento para gerar um conjunto de dados sintético maior do que o conjunto de dados original e, consequentemente, melhorar o desempenho do modelo.

Outro fator importante que deve ser levado em consideração é a distribuição das anotações das caixas delimitadoras dentro do conjunto de dados. As espécies *O. bataua* e *A. butyracea,* por apresentarem menor densidade quando comparadas com *E. precatoria*, por exemplo, ficaram sub-representadas em relação às demais classes, o que é indesejado para o treinamento de uma rede neural. Para um treinamento mais robusto o ideal é que se tenha o mesmo número de objetos para todas as categorias. Portanto, realizou-se o aumento dos dados dando ênfase às espécies que estavam sub-representadas de forma a garantir uma proporção mais igualitária entre as classes.

Assim, foram realizadas as seguintes operações para o aumento dos dados: a) Distorção geométrica: inclui alterações no dimensionamento, recorte, inversão ou translação, cisalhamento e rotação; b) Distorção fotométrica: inclui alteração no brilho,

matiz, contraste, saturação, desfoque e ruído; c) Oclusão: inclui técnicas de recorte da imagem, sobreposição e mosaico (Tabela 3).

**Tabela 3** Lista de operações realizadas para o aumento de dados.

| **Operações** | **Tratamento Aplicado** |
|---|---|
| Rotação | 15º, 25º, 90º e -95º |
| Translação | horizontal, vertical e horizontal + vertical |
| Recorte | 10% e 20% |
| Cisalhamento | horizontal, vertical |
| Matiz | + 80% |
| Saturação | -10% e + 10% |
| Desfoque | até 1 *pixel* |
| Ruído | Até 1% |

Inicialmente, aplicou-se uma rotação na imagem, tanto no sentido horário quanto no sentido anti-horário. Além disso, foram realizadas uma inversão horizontal e inversão vertical na imagem e depois uma inversão vertical e horizontal, simultaneamente. Dessa forma, foi garantido ao modelo maior sensibilidade a qualquer orientação da câmera. Em seguida, aplicou-se a ferramenta "*crop*", com recorte de 10% e 20% para dar maior variabilidade ao posicionamento e tamanho, dando resistência a variações de tamanho do objeto e quanto à posição da câmera. Então, foram ajustados os valores de matiz para -10% e +10% e saturação para 80% em relação aos valores de referência da imagem original. Depois, aplicou-se um desfoque gaussiano, para dar resistência à diferentes focagens da câmera, além de estabelecer um percentual de ruídos que simula, por exemplo, a presença de um pássaro, ou qualquer outra forma de obstrução que possa acontecer na imagem. E, por último, foi utilizada a técnica de oclusão em mosaico que ao combinar quatro imagens de treinamento em uma, permite que o modelo aprenda a identificar objetos em uma escala menor do que a normal.

### 4.6.3 Suavização de rótulos de classe

A suavização de rótulo de classe não é uma técnica de tratamento de imagem, mas sim uma mudança intuitiva no rótulo de classe. Geralmente a classificação correta para uma caixa delimitadora é representada por um vetor de classes [0, 0, 0, 1, 0, 0, ...] e a função de perda é calculada com base nesta representação.

No entanto, quando um modelo se torna excessivamente seguro com uma precisão próxima de 1, ele geralmente está errado, super-ajustado e provavelmente desconsidera

as complexidades de outras previsões de alguma forma. Seguindo essa intuição, é mais razoável codificar a representação do rótulo da classe para avaliar essa incerteza em algum grau.

Para cada objeto, as redes de detecção costumam calcular uma distribuição de probabilidade em todas as classes (*pi*) com a função *softmax* (ZHANG *et al.*, 2019):

$$p_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{19}$$

onde $z_i$'s são os *logits* não normalizados diretamente da última camada linear para previsão de classificação, ou seja, uma função exponencial padrão é aplicada a cada elemento do vetor de entrada $z_i$ e normaliza esses valores dividindo pela soma de todos esses exponenciais ($\sum_j e^{z_j}$). Para detecção de objetos durante o treinamento, modificou-se a perda de classificação comparando a distribuição de saída $p$ contra a distribuição da verdade fundamental $q$ com entropia cruzada, como sugerido por Zhang *et al.* (2019):

$$L = \sum_i q_i \, log p_i \tag{20}$$

onde $q$ é uma distribuição conhecida por *one-hot*, em que a classe correta tem probabilidade igual a 1, enquanto todas as outras têm probabilidade igual a 0. Quando o modelo está muito confiante em suas previsões o mesmo fica sujeito a ajustes excessivos. Portanto, a suavização de rótulos foi sugerida por Szegedy *et al.* (2016) como forma de regularização.

Dessa maneira, converteram-se os rótulos em uma distribuição de probabilidade suavizada pela Equação 21. Essa técnica reduz a confiança do modelo, medido pela diferença entre o maior o menor *logit*.

$$q_i = \begin{cases} 1-\varepsilon & se\ i=y \\ \varepsilon(K-1) & se\ i\neq y \end{cases} \tag{21}$$

em que, $K$é o número de classes, $\varepsilon$ é uma constate e $q$ é a distribuição da verdade fundamental.

Com base nos parâmetros descritos, adotou-se o valor de $\varepsilon$ = 0,1. Isso significa que o modelo tem como parâmetro para uma certeza a probabilidade $q = 0{,}9$, quando próxima de 1 e $q = 0{,}1$, quando próximo de 0.

## 4.7 Ambiente de experimentação

Para computação, foi utilizado o *Google Collaboratory* também conhecido como *Google Colab*. O *Google Colab* é um serviço em nuvem para divulgação, educação e pesquisa em aprendizado de máquina. Tecnicamente, é um serviço hospedado de *notebook Python Jupyter*[5] executado em uma *GPU* robusta. O serviço é vinculado a uma conta *Google Drive* e é gratuito por um período de 12 horas diárias, ou opcionalmente, U$$ 10,00 por mês para atualizar para uma versão Pro, garantindo acesso ilimitado (GOOGLE, 2020).

O YOLOv4 é construído originalmente sobre a estrutura de código aberto *Darknet*. O *Framework*[6] *Darknet* foi escrito por Joseph Redmon (2016) em linguagem de programação *C* em *CUDA* (*Compute Unified Device Architecture*). Embora o *Darknet* não seja tão intuitivo de usar, é imensamente flexível. Dessa forma, utilizou-se o *Darknet* para implementar o YOLOv4 no *Google Colab*.

Os processamentos foram executados no *Google Colab* em um servidor *web Google Chorme.* O *Google Colab* opera no *Ubuntu 17.10 64 bits* e é composto por um processador *Intel Xeon* com dois núcleos de 2,3 *GHz* e 13 *GB* de *RAM* (CARNEIRO *et al.*, 2018). Foi utilizada uma *GPU NVIDIA Tesla V100-SXM2* de 16 *GB* de *RAM*.

## 4.8 Configurações do treinamento

Nesta seção, foram elencados os principais ajustes nas configurações do YOLOv4 para a detecção das palmeiras. Com o objetivo de otimizar o treinamento, o YOLOv4 foi inicializado com os pesos pré-treinados originalmente do banco de dados *MS COCO* (LIN *et al.*, 2014). Utilizaram-se os pesos pré-treinados para as 137 primeiras camadas convolucionais do modelo. Isso garantiu um ganho de processamento, uma vez que essas camadas iniciais são responsáveis pela detecção de características de níveis inferiores, como borda e textura, por exemplo. A estrutura completa sobre cada camada, tamanho de entrada e saída, número de filtros e passos realizados pelo YOLOv4 estão apresentados Tabela 7 do Anexo II.

Para a atualização da taxa de aprendizagem, foi utilizado o programador de taxa de aprendizagem. Foi estabelecido o valor da taxa de aprendizagem inicial igual a 0,001,

[5]*Notebook Jupyter* é uma ferramenta de código aberto baseada em navegador que integra linguagens interpretadas, bibliotecas e ferramentas para visualização (PÈREZ, 2007).

[6]Em desenvolvimento de software, é uma abstração que une códigos comuns entre vários projetos provendo uma funcionalidade genérica.

com *Burn in* equivalente a 1000 épocas. Isso significa que nas primeiras mil épocas a taxa de aprendizagem aumenta gradativamente de 0 até 0,001 e a partir de então a função cosseno inicia a atualização.

O número máximo de épocas ideais para o treinamento (*max_batches*) do YOLOv4 é de, no mínimo, 2000 vezes o número de classes (Equação 22) ou até que a perda se apresente constante à medida que o número de épocas aumenta. No presente estudo foram definidas cinco classes de palmeiras, portanto, determinaram-se *max_batches* = 10000.

$$\max\ \_batches = 2000 \times num\_classes \tag{22}$$

Além do número máximo de épocas também foi definido o tamanho do lote (*batch_size*). O tamanho do lote diz respeito ao número de imagens que passam pela rede neural a cada iteração e está estreitamente relacionado ao desempenho computacional necessário para o processamento dos dados, além dos processos de amostragem envolvidos. Um lote muito grande necessita mais tempo de processamento. Neste estudo foi definido um *batch_size* = 64 com 12 subdivisões ou minilotes (*mini_batches*). Ao agrupar as imagens em minilotes acelera-se o tempo do processo de treinamento e se aumenta a capacidade de generalização do modelo.

A normalização de quaisquer dados consiste em encontrar a média e a variância dos dados e normalizá-los de forma que tenham média 0 e variância unitária. Por conta disso, aplicou-se a normalização de lote (*BN*, do inglês *Batch Normalization*; Eq. 23), mas a *BN* não é executada quando o tamanho do lote se torna muito pequeno. A estimativa do desvio padrão e da média é influenciada pelo tamanho da amostra. Quanto menor o tamanho da amostra, maior a probabilidade de não representar a integridade da distribuição.

$$\hat{x}_{t,i}(\theta_t) = \frac{x_{t,i}(\theta_t) - \mu_t(\theta_t)}{\sqrt{\sigma_t(\theta_t)^2 + \epsilon}} \tag{23}$$

em que: $\epsilon$ é uma pequena constante adicionada para dar estabilidade numérica e $\mu_t(\theta_t)\ e\ \sigma_t(\theta_t)$ é a média e a variância respectivamente calculada para todos os exemplos do minilote atual.

Para resolver esse problema foi executada a normalização cruzada de minilotes (Equação 24), que usa estimativas de lotes recentes para melhorar a qualidade da estimativa de cada lote.

$$\hat{x}_{t,i}{}^{l}(\theta_t) = \frac{x_{t,i}{}^{l}(\theta_t) - \mu_{t,k}{}^{-l}(\theta_t)}{\sqrt{\sigma_{t,k}{}^{-l}(\theta_t)^2 + \epsilon}} \tag{24}$$

em que: a média e a variância são calculados a partir das *N* médias e variâncias anteriores e aproximadas usando fórmulas dos polinômios de Taylor para expressá-las como uma função de parâmetros $\theta_t$ em vez de $\theta_t - N$.

Um desafio em computar estatísticas em várias iterações é que as ativações de rede de diferentes iterações não são comparáveis entre si devido a mudanças nos pesos da rede pela retropropagação. Dessa maneira os polinômios de Taylor são utilizados para aproximar qualquer função indefinidamente diferençável (YAO, 2020).

As redes neurais costumam ter melhor desempenho se forem capazes de generalizar melhor, para isto, geralmente, são utilizadas técnicas de regularização como o *Dropout*, desativando-se certos neurônios durante o treinamento. Isso garante, na maioria das vezes, um aumento na precisão durante a validação. Mas o *Dropout* não funciona bem para camadas totalmente convolucionais, onde os recursos são espacialmente correlacionados, uma vez que este método descarta recursos de forma aleatória. Dessa forma, aplicou-se o método de regularização *Drop Block*, onde os recursos em um bloco, ou seja, uma região contígua em um mapa de características, de uma área correlacionada são descartados simultaneamente, fazendo com que as redes procurem um outro lugar por evidências para ajustar os dados (GHIASI, 2018).

## 4.9 Métricas de avaliação

### 4.9.1 Interseção Sobre a União

Para avaliar os resultados do treinamento foram utilizadas algumas métricas de avaliação. Uma das métricas mais fundamentais relacionadas à detecção de objetos é a Interseção Sobre a União (*IoU*) (Figura 15), também conhecida como índice de *Jaccard*.

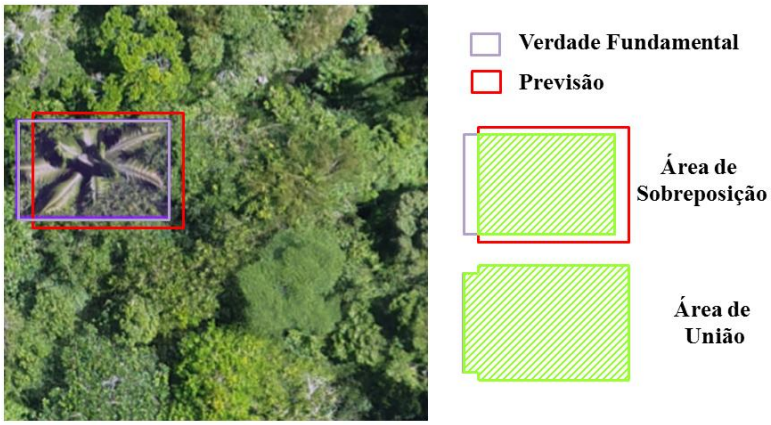


**Figura 15** Exemplo de detecção para uma um palmeira em uma imagem. A caixa delimitadora prevista é desenhada em vermelho e a caixa delimitadora da verdade fundamental em lilás.

Essa métrica mede a similaridade espacial entre uma caixa delimitadora prevista e a caixa delimitadora da verdade fundamental (RAHMAN e WANG, 2016). Definindo a proporção entre a interseção de ambas as caixas delimitadoras sobre a união (Equações. 25 e 26).

$$IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|} \tag{25}$$

simplificando,

$$IoU = \frac{Área\ de\ sobreposição}{Área\ de\ união} \tag{26}$$

Observa-se na Figura 15 e nas equações 25 e 26 que *IoU* é uma razão, que possui em seu numerador a área de sobreposição ($B \cap B^{gt}$) entre a caixa delimitadora prevista $B$ e a caixa delimitadora de verdade $B^{gt}$ e no denominador é a área ocupada tanto pela caixa delimitadora prevista quanto a caixa delimitadora da verdade fundamental ($B \cup B^{gt}$). A *IoU* possui valores resultantes entre 0 e 1; quanto mais próximo de 1, significa uma previsão perfeita (100%), logo se não existir interseção entre as caixas o valor será 0.

Para medir as distâncias entre as caixas delimitadoras Rezato fighi *et al.* (2019) sugerem a determinação da perda de *IoU*, (Equação 27):

$$IoU = 1 - \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|} \quad (27)$$

Embora o *IoU* seja um uma métrica eficiente para caixas delimitadoras sobrepostas, tem como limitação o problema do gradiente de desaparecimento devido a ocorrência de caixas não sobrepostas. Com base nisso, Zheng *et al.* (2018) sugerem a inserção da*IoU* Generalizada (*GIoU*; Equação 28).

$$GIoU = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \quad (28)$$

*GIoU* possui um termo de penalidade junto com a função de perda de *IoU* (*1-IoU*), onde *C* é a menor caixa cobrindo a caixa delimitadora prevista $B$ e caixa delimitadora da verdade fundamental $B^{gt}$. Devido à introdução do termo de penalidade, a perda *GIoU* expande o tamanho da caixa prevista até que se sobreponha a caixa de destino, maximizando a área de sobreposição da caixa delimitadora, sem que ocorra perda de sua coordenada original. Consequentemente, a perda *GIoU* tem convergência lenta, especialmente para caixas delimitadoras retangulares.

Ao minimizar diretamente a distância entre duas caixas delimitadoras, o modelo converge muito mais rápido do que a perda *GIoU,* especialmente no caso de não sobreposição. Portanto utiliza-se a distância de interseção sobre a união (Equação 29) para minimizar a distância entre as caixas:

$$DIoU = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} \quad (29)$$

em que: $DIoU$ é a Distância-Interseção sobre a União e $\rho(b, b^{gt})$ é a distância euclidiana entre as coordenadas centrais da caixa delimitadora prevista e da verdade fundamental, normalizadas por um comprimento diagonal *c* da menor caixa delimitadora cobrindo duas caixas.

Embora a perda *DIoU* englobe dois fatores geométricos, *i.e.*, área de sobreposição e distância, ainda deixa de contabilizar um terceiro fator importante: o aspecto. Por conta disso, com base na perda *DIoU,* Zheng *et al.* (2018) sugerem a perda *CIoU* (Interseção Sobre a União Completa). A perda *CIoU* é proposta pela imposição da consistência da razão de aspecto e é calculada pela equação (30).

$$CIoU = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{30}$$

em que: $v$ mede a consistência da razão de aspecto e é calculado pela Eq. 31.

$$v = \frac{4}{\pi^2}\left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right)^2 \tag{31}$$

sendo:

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{32}$$

O parâmetro de compensação $\alpha$ quando multiplicado por $v$, representa um termo de troca, que determina a prioridade da regressão. Os parâmetros $w$, $h$ e $w^{gt}, h^{gt}$ são, respectivamente, os comprimentos e as alturas das caixas delimitadoras previstas e da verdade fundamental.

Diante do exposto, utilizou-se a perda *CIoU* da mesma forma como sugerido por Bochkovskiy*et al.* (2020) como métrica para o treinamento de YOLOv4, uma vez que a função supera todos os problemas de perda de gradiente, velocidade e precisão.

Alguns termos são amplamente conhecidos no treinamento e no teste de modelos de RNAs que se referem à aprendizagem e às previsões. O Verdadeiro Positivo (*TP*, do inglês, *True Positive*) corresponde aos resultados ótimos que o modelo deve buscar durante o treinamento, ou seja, o número de palmeiras que foram corretamente classificadas. O Falso Positivo (*FP*, do inglês, *False Positive*) corresponde às previsões erradas feitas pelo modelo, ou seja, quando o modelo determinou a existência de uma palmeira em uma localização errada, ou quando o modelo encontrou a localização da palmeira, porém a classificou incorretamente. O Falso Negativo (*FN*, do Inglês *False Negative*), corresponde a todos *TP*s que o modelo não conseguiu prever.

Os *TP*s são normalmente determinados com a ajuda de um limite mínimo de *IoU*. O limite define a precisão espacial mínima desejada para definir uma previsão como correta. Escolheu-se o valor de 0,5 (50%) de limite *IoU* para nosso modelo, assim como sugerido por Bochkovskiy *et al.* (2020). Isso significa que se o modelo apresentar *IoU* maior ou igual a 0,5 a previsão é TP e, caso contrário, a previsão ser FP.

### 4.9.2 *Recall*, Precisão e Limiar de Confiança

Outras duas métricas amplamente utilizadas para avaliar os resultados de experimentos de aprendizado de máquina são o *Recall* e a Precisão (POWERS, 2020). O *Recall* (Equação 33) foi utilizado para determinar a probabilidade de que as caixas delimitadoras para uma determinada espécie fossem detectadas corretamente.

$$Recall_i = \frac{TP}{TP + FN} \tag{33}$$

A precisão (Equação 34) representa a probabilidade de uma caixa delimitadora que foi detectada como uma determinada espécie represente aquela espécie. A Precisão é calculada como a razão entre o número de caixas corretamente detectadas de uma determinada espécie e o número de caixas que foram previstas pelo detector como daquela espécie.

$$Precisão_i = \frac{TP}{TP + FP} \tag{34}$$

Ao otimizar o modelo para determinar o *Recall* e a Precisão, é improvável que um detector de objetos produza valores elevados tanto de *Recall* quanto de Precisão em uma classe de objeto em todos os momentos, principalmente por causa de uma compensação entre as duas métricas. Essa compensação depende do limiar de confiança.

Cada caixa delimitadora prevista é associada a um limite de confiança, que vai de 0 a 1 (baixa e alta confiança, respectivamente), o qual é usado para avaliar a probabilidade de a classe de palmeira aparecer na caixa delimitadora. Ao escolher um limite de alta confiança, o modelo se torna robusto para exemplos positivos (ou seja, caixas contendo um objeto), portanto, haverá menos previsões positivas. Como resultado, os falsos negativos aumentam e os falsos positivos diminuem, o que reduz o *Recall* e melhora a Precisão. Da mesma forma, diminuir ainda mais o limite faz com que a Precisão diminua e o *Recall* aumente.

Para a detecção definiu-se um limite de confiança igual a 0,25 e as detecções acima desse valor foram consideradas *TP*s, enquanto as que estão abaixo foram consideradas *FP*s.

### 4.9.3 *F1-score*

A pontuação *F1-score* com o objetivo de avaliar o quão o modelo é capaz de generalizar, ou seja, o quanto o modelo é capaz de apresentar eficácia quando testado em uma base de dados nunca vista anteriormente (Equação 35).

$$F1\_score = 2 * \frac{precisão * Recall}{Precisão + Recall} \quad (35)$$

## 4.10 Validação cruzada

Para validar nosso experimento, foi realizada a validação cruzada *k-fold* (do inglês, *k-fold cross validation*) (ANTHONY e HOLDEN, 1998). O conjunto de dados original (Cenário I) foi dividido aleatoriamente em cinco subconjuntos de partes iguais (Figura 16). O treinamento foi realizado cinco vezes (k=5). No primeiro treinamento, um dos subconjuntos foi selecionado para validação (20%) e os quatro subconjuntos restantes (80%) para treinamento. No segundo treinamento, um segundo subconjunto, diferente do primeiro, foi selecionado para validação e os subconjuntos restantes para treinamento. Esse procedimento foi repetido por cinco vezes até que todos os dados passassem exatamente uma vez pelo conjunto de validação. Garantiu-se assim, a redução do erro de viés que pode ocorrer durante a detecção. Para o Cenário II, utilizou-se a mesma subdivisão realizada para o Cenário I, porém, o aumento de dados foi aplicado sempre para os quatro subconjuntos de treinamento, pertencentes a cada *k-fold*.

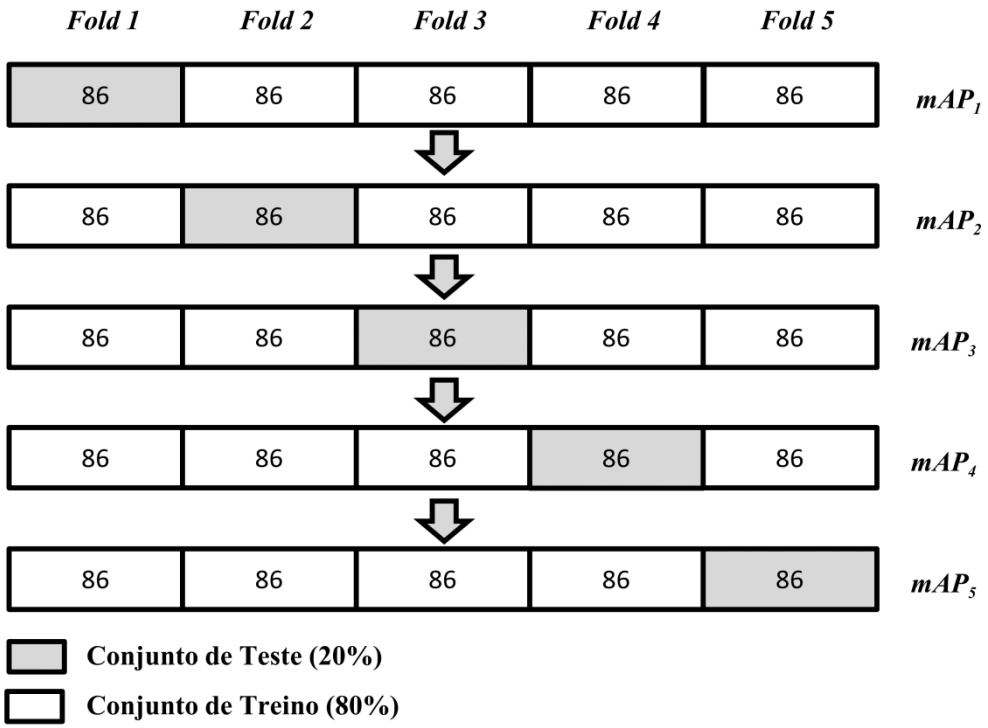


**Figura 16** Esquema de validação Cruzada *k-fold* com k=5, para um conjunto de 430 imagens subdividido em 5 partes iguais (Cenário I). *mAP*= Precisão Média para todas as classes.

### 4.11 Densidade e distribuição espacial

Com base nos resultados do treinamento do modelo YOLOv4 para a detecção de palmeiras, o melhor conjunto de pesos ajustados foi selecionado para realizar as inferências sobre a área de estudo e consequentemente foi estimada a densidade populacional para as cinco classes de palmeiras estudadas.

Da mesma forma que a entrada do modelo para o treinamento exige um tamanho fixo de 416 x 416 pixels, para realizar as inferências foi necessário ajustar as imagens de entrada com este tamanho. Para tal, foram utilizadas as 960 imagens provenientes do ortomosaico original (Seção 4.3) como base para as estimativas da densidade.

A densidade absoluta das palmeiras (*DA*) foi obtida pela Equação (37):

$$DA = \frac{N_i}{A} \tag{37}$$

Em que: *DA* é o número total de palmeiras de uma mesma espécie (*Ni*) por unidade de área (*A*) em hectares. Enquanto que, a Densidade Relativa foi calculada pela equação (38):

$$DR = \frac{n_i}{N} \quad (38)$$

Em que: *DR* indica a participação de cada espécie em relação ao número total de palmeiras (*N*), sendo $n_i$ o número total de indivíduos da i-ésima espécie de palmeira.

Além disso, calculou-se a frequência absoluta e relativa a fim de exprimir a distribuição espacial de cada espécie. A frequência Absoluta (*FA*), dada em percentagem, representa a distribuição espacial das espécies de palmeiras na área estudada e foi calculada pela equação (39):

$$FA = \frac{pi}{P} \times 100 \quad (39)$$

Em que: $pi$ é o número de unidades amostrais com ocorrência da espécie *i* e *P* é o número total de parcelas da amostra. Já a frequência relativa (*FR*) (Equação 40) é a relação entre a frequência absoluta de determinada espécie ($FA_i$) de palmeira com a soma das frequências absolutas de todas as espécies de palmeira.

$$FR = \frac{FA_i}{\sum FA} \times 100 \quad (40)$$

A frequência fornece informações a respeito da dispersão das espécies. Espécies com elevado número de indivíduos podem apresentar baixos valores de frequência se os indivíduos apresentarem padrão espacial agrupado. Ao passo que outras espécies podem apresentar 100% de frequência se os indivíduos correspondentes estiverem presentes em todas as parcelas.

Com base nisso, foi realizada a aplicação da função *K* proposta por Ripley (1976) para analisar a distribuição espacial das palmeiras. Sendo esta, uma função de densidade que considera a variância de todas as distâncias (m) entre todos os eventos.

A análise é feita graficamente, para facilitar a visualização dos desvios em relação à hipótese nula (hipótese de aleatoriedade), através de um gráfico cuja abscissa representa *m* e, na ordenada a função transformada *L(m)* da função *K* (RIPLEY, 1979). Para nosso estudo, foi definida uma distância *m* de 5 metros em mil simulações de *Montecarlo*[7], isto

[7] Série de cálculos de probabilidade que estimam a chance de um evento futuro acontecer

significa que a função de *K* de Ripley (Equação 41) avaliou a relação existente entre pares de eventos a cada 5 metros.

$$\widehat{L(m)} = \sqrt{\frac{\widehat{K}}{\pi}} - m \tag{41}$$

Se o padrão for completamente aleatório, a função *L(m)* se apresenta como uma linha horizontal sobre o eixo das abcissas. O padrão espacial agregado resulta em um número de indivíduos maior que o esperado pela hipótese de aleatoriedade, ou seja, distanciando em valores do eixo das abcissas em e assumindo valores positivos. No caso de padrão espacial uniforme, indica regularidade na localização das palmeiras e o número de indivíduos será menor do que o esperado pela hipótese de aleatoriedade e a função *L(m)* assumirá valores negativos.

Quando a imagem passa pelo modelo, o valor das coordenadas georreferenciadas não é considerado, ou seja, as coordenadas da imagem tornam-se referentes a cada *pixel* e não a representação da localização geográfica do *pixel*. Assim, cada objeto é localizado em relação a sua posição dentro da imagem e não geograficamente. Portanto, após as inferências foi necessário realizar o georreferenciamento das imagens.

Da mesma forma, as coordenadas do centro das caixas delimitadoras previstas foram extraídas e georreferenciadas. Consequentemente, o centro da caixa delimitadora foi definido como a posição exata de cada palmeira na floresta e suas coordenadas foram utilizadas para a analisar a distribuição espacial pela função *K* de Ripley.

Outra questão que deve ser levada em consideração é o fato de que ao segmentarmos o ortomosaico original, algumas palmeiras ficaram divididas pela metade e durante a previsão o modelo considera toda ou parte de palmeiras localizadas em cada imagem, gerando duplicatas das que foram divididas pela metade. Portanto, antes computar a quantidade de palmeiras identificadas pelo modelo durante a inferência, foi necessário remover as duplicatas. O processo de georreferenciamento e remoção das duplicatas foram realizados no software QGIS e como critério para remoção, foi selecionada a menor caixa delimitadora. Já os gráficos para análise de distribuição espacial pela função *K* de Ripley, foram gerados no *Software R*.

# 5 RESULTADOS E DISCUSSÕES

## 5.1 Desempenho do modelo

Para avaliar o modelo foram testados dois cenários distintos. No primeiro cenário (Cenário I), foram utilizadas 430 imagens para a aplicação do modelo YOLOv4 e o tempo de duração média para realizar o treinamento pelo método de validação cruzada foi de 5 horas a cada 10 mil iterações. No segundo cenário (Cenário II), a aplicação do YOLOv4 foi realizada com um total de 2086 imagens (430 imagens originais + 1656 imagens sintéticas) e levou em média 8 horas para finalizar o treinamento a cada 10 mil iterações. A precisão média geral (*mAP)* para as cinco classes foi de 72,61% e 91,08%, respectivamente, para os cenários I e II (Tabela 4).

**Tabela 4** Precisão Média Geral para os Cenários I e II.

| Cenários | Métricas | *1-fold* | *2-fold* | *3-fold* | *4-fold* | *5-fold* | Geral |
|---|---|---|---|---|---|---|---|
| I | *mAP* (%) | 75,26 | 70,52 | 76,06 | 69,51 | 71,72 | 72,61 |
| | *Precisão* | 0,73 | 0,82 | 0,81 | 0,7 | 0,72 | 0,75 |
| | *Recall* | 0,77 | 0,75 | 0,83 | 0,71 | 0,75 | 0,77 |
| | *F1-score* | 0,75 | 0,78 | 0,82 | 0,70 | 0,73 | 0,76 |
| | *IoU* (%) | 52,48 | 60,01 | 59,15 | 50,2 | 52,69 | 54,9 |
| II | *mAP* (%) | 90,95 | 88,16 | 94,24 | 90,44 | 91,61 | **91,08** |
| | *Precisão* | 0,82 | 0,88 | 0,91 | 0,82 | 0,91 | 0,87 |
| | *Recall* | 0,9 | 0,89 | 0,93 | 0,88 | 0,94 | 0,91 |
| | *F1-score* | 0,86 | 0,88 | 0,92 | 0,85 | 0,92 | 0,89 |
| | *IoU* (%) | 60,74 | 67,61 | 68,43 | 59,79 | 73,61 | 66,04 |

*mAP* = Precisão média para todos os treinamentos considerando todas as classes; *F1-score* = média harmônica da Precisão e *Recall; IoU* = Interseção sobre a União.

A precisão média (*AP*) por espécie nos cenários I e II variou entre 46,56% e 96,20% (Tabela 5).

**Tabela 5** Precisão média para as cinco classes de palmeiras em um remanescente de Floresta Ombrófila Aberta na Amazônia Ocidental.

| Métricas | *k-fold* | *A. butyracea* | | *E. precatoria* | | *I. deltoidea* | | *N.I.* | | *O. bataua* | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | I | II | I | II | I | II | I | II | I | II |
| *AP (%)* | *1-fold* | 78,67 | 91,16 | 90,81 | 96,24 | 76,99 | 92,14 | 51,79 | 85,2 | 77,99 | 90,00 |
| | *2-fold* | 81,89 | 89,29 | 91,57 | 96,05 | 76,76 | 90,62 | 35,13 | 69,85 | 67,27 | 95,00 |
| | *3-fold* | 59,60 | 96,78 | 91,49 | 96,50 | 88,55 | 96,83 | 43,39 | 87,15 | 97,27 | 93,94 |
| | *4-fold* | 79,61 | 92,37 | 92,21 | 94,04 | 61,71 | 92,12 | 56,87 | 79,07 | 57,14 | 94,57 |
| | *5-fold* | 56,73 | 90,77 | 87,00 | 98,19 | 79,47 | 97,46 | 45,61 | 82,73 | 89,77 | 88,89 |
| *mAP* (%) | | 71,3 | **92,07** | 90,62 | **96,2** | 76,7 | **93,83** | 46,56 | **80,8** | 77,89 | **92,48** |
| *s* (%) | | 12,09 | 2,85 | 2,08 | 1,48 | 9,66 | 3,09 | 8,30 | 6,82 | 16,28 | 2,82 |

I = Cenário I; II = Cenário II; *AP* = Precisão média; *s* = Desvio padrão; *N.I.* = Palmeiras Não Identificadas.

Em geral, o desempenho das técnicas de aprendizado profundo tende a melhorar na medida em que o tamanho do conjunto de dados aumenta (ANTONIOU *et al.*, 2017; PEREZ e WANG,2017; PEREZ MALLA *et al.*, 2019; HAO *et al.*, 2020). Na Tabela 4 e na Figura 17 é possível observar o efeito do aumento do conjunto de dados sobre a métrica de precisão média geral. Obteve-se um ganho de 18,47% (a métrica aumentou de 72,61% para 91,08%, respectivamente para os Cenários I e II) na precisão média geral ao treinar um conjunto de dados aumentado em relação ao conjunto de dados original. A mesma tendência com ganhos de 8,71% foi observada por Al-Masdi *et al.* (2018) ao estudarem a detecção e classificação simultânea de massas mamárias em mamografias digitais por meio de um sistema de diagnóstico auxiliado por computador baseado em YOLO.

Além disso, a validação cruzada *k-fold* (ANTHONY e HOLDEN, 1998) permitiu mitigar possíveis erros de viés durante o treinamento, pois os métodos de *Deep Learning* são sensíveis a variabilidade dos dados e, dependendo do arranjo dos dados, o modelo pode apresentar maiores ou menores erros de generalização. Consequentemente, isto ocasionou um aumento no custo computacional e no tempo para o treinamento do modelo YOLOv4 (5 e 8 horas para cada *k-fold* nos Cenários I e II, respectivamente). Todavia, a predição, ou seja, a detecção e classificação para cada imagem (parcela) leva apenas 10 milésimos de segundo.

O desempenho da curva de aprendizagem do modelo YOLOv4 para a detecção de palmeiras foi superior no Cenário II (Figura 17). A configuração de treinamento utilizada inicia o cálculo do *mAP* na milésima iteração e, a partir de então, o *mAP* foi calculado a

cada 100 épocas e os pesos foram salvos a cada 1.000 épocas e sempre que atingia uma nova máxima no valor da precisão geral.

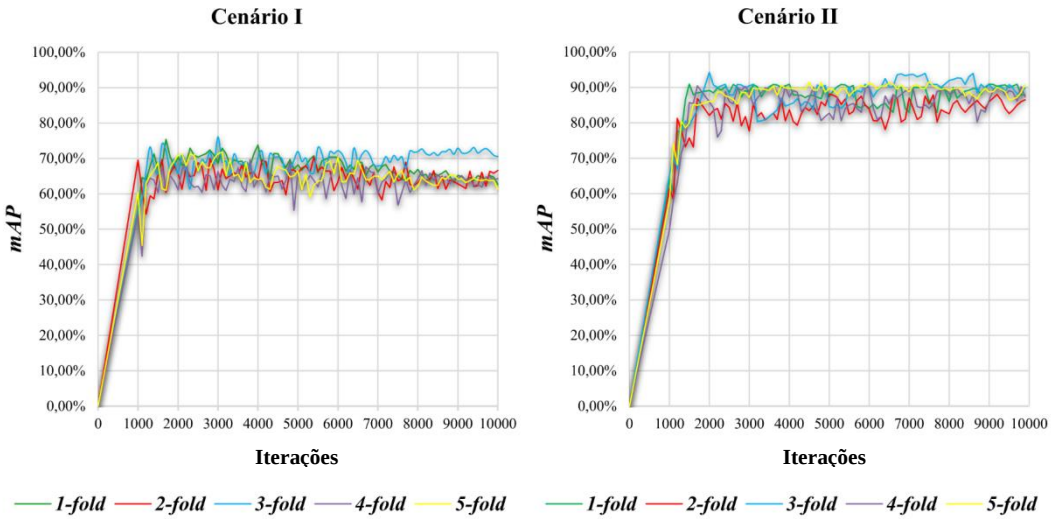


**Figura 17** Desempenho da curva de aprendizagem do YOLOv4 na detecção de palmeiras para os Cenários I e II e para 10 mil iterações.

Para efeito de comparação e discussão sobre as métricas avaliadas, a partir deste momento, serão elencados apenas os principais resultados relacionados ao Cenário II, o qual teve um melhor desempenho no treinamento.

O detector de objetos YOLOv4 conseguiu prever satisfatoriamente as quatro classes de palmeiras previamente identificadas e ainda a classe de palmeiras não identificadas (Tabela 5). A espécie *E. precatoria (*açaí) foi detectada com 96,20% de acerto. *A. butyracea* (jací), *I. deltoidea* (paxiubão) e *O. bataua* (patauá) também apresentaram resultados satisfatórios para a configuração experimental considerada, com mais de 92% de acerto. A classe Palmeiras Não Identificadas apresentou a menor precisão de detecção (80,80%). Isto se deve ao fato desta classe possuir alta variabilidade (*s* = 6,82%), intrínseca da própria classe, *i.e.*, a classe é composta por diferentes espécies de palmeira, não apresentando um padrão de forma e tamanho dos indivíduos, o que dificulta a aprendizagem do modelo.

Após o treinamento do YOLOv4, para melhor compreensão dos resultados e do comportamento do modelo, foi construída uma matriz de confusão para cada *k-fold,* bem como para a média dos *5-folds* (Figura 18). Os valores que compõe a matriz foram obtidos a partir do conjunto de dados de validação aplicados ao método de classificação e comparando sua predição com a classe correta de cada *Bounding Box* da verdade

fundamental. É importante ressaltar que foram consideradas como verdade fundamental as palmeiras rotuladas no trabalho de fotointerpretação.

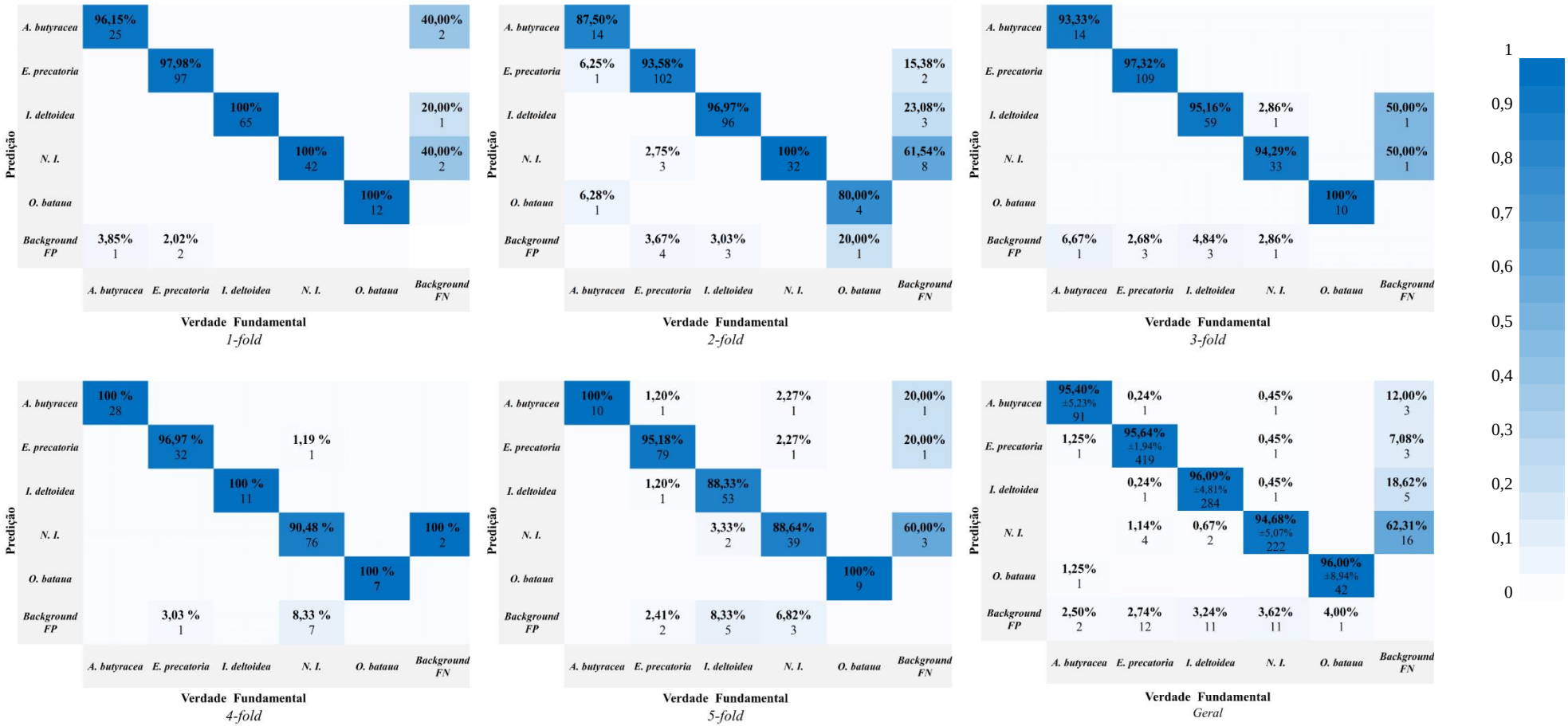


**Figura 18** Matriz de Confusão para as classes de palmeiras estudadas por *k-fold* e para o geral, em que *N. I.*= Classe de Palmeiras Não Identificadas; *Backgroud FP* = Classes de palmeiras que o modelo confundiu com o plano de fundo; *Background FN*= Número de palmeiras não detectadas pelo modelo.

As matrizes apresentam a precisão da classificação do modelo estudado (Figura 18). As precisões médias para cada espécie estão dispostas nas diagonais da matriz e o percentual de erro corresponde às células fora das diagonais. O vetor linha da matriz de confusão representa o valor verdadeiro, enquanto o vetor coluna representa o valor previsto em um limite de confiança de 0,25.

A matriz de confusão permite visualizar o quanto uma classe é capaz de confundir com as demais sendo que no presente estudo a maior confusão foi observada em relação ao plano de fundo, *background FP* (Figura 18), o que já era esperado para modelos de detecção de objetos (TING, 2017).

Ao considerar a média dos cinco treinamentos (geral) é possível observar que todas as espécies foram classificadas com precisão média superior a 95%, exceto a classe de Palmeiras Não Identificadas, que foram classificadas com 94,68% de taxa de acerto.

A classificação das palmeiras com elevada taxa de precisão, só foi possível devido as características particulares de cada espécie, como tamanho (Tabela 1) e forma (Figura 19). *O. bataua*, por exemplo, além de ser a classe que apresentou maior média de diâmetro de copa entre as demais palmeiras ($\mathbf{Ø_{médio\ copa}}$**=** 11,2 metros), sua copa possui de 8 a 16 folhas arranjadas em forma de espiral, facilitando a identificação pelo YOLOv4. Já *E. precatoria* apresenta um padrão de copa com diâmetro menor ($\mathbf{Ø_{médio\ copa}}$**=** 3,51 metros), seus folíolos são pêndulos ao longo da raque, o que confere um formato estrelado característico à copa.

Durante a análise de fotointerpretação para a delimitação das copas individuais não foi possível determinar quais espécies compõem a classe de palmeiras não identificadas, principalmente devido a não uniformidade de tamanho e forma, ou de alguma característica que as diferenciassem das demais ou, até mesmo, pelo fato de a copa estar parcialmente coberta pelo dossel da floresta. Da mesma forma, essa dificuldade também é observada pelo modelo YOLOv4, o que culminou em um maior erro de predição. Dessa forma, é importante investigar em novos estudos a utilização de imagens obtidas em diferentes épocas do ano, na tentativa de avaliar o efeito da fenofases sobre a detecção. Isto porque, que algumas espécies perdem suas folhas em um determinado período do ano, o que pode tornar as copas das palmeiras, que antes estavam parcialmente cobertas, mais expostas, facilitando na detecção.

Geralmente, os modelos de detecção de objetos apresentam um certo grau de dificuldade para detectar objetos de tamanhos pequenos (CHEN *et al.*, 2015, CHEN *et al.*, 2017; BOSQUET *et al.*, 2020; ZHANG *et al.*, 2020). YOLOv4 forma recursos

agregando pixels em camadas convolucionais e, no final da rede a previsão é feita com base na diferença entre a previsão e a verdade fundamental. Isto significa que se a caixa delimitadora da verdade fundamental não for grande, o sinal será pequeno durante o treinamento. Além disso, objetos pequenos são mais propensos a ter erros de rotulagem de dados. No entanto, a alta resolução espacial (*pixel* = 4 cm) da imagem obtida pela câmera *RGB* da *RPA* garantiu a identificação de forma precisa durante o processo de rotulagem das palmeiras e o mesmo para a detecção pelo YOLOv4.

Embora a alta resolução espacial da imagem tenha permitido a identificação de forma precisa, ao se plotar os resultados das detecções no *QGIS* (Figura 19), foi observado que algumas palmeiras classificadas como falso positivos poderiam ser, de fato, verdadeiros positivos que não foram observados durante a fotointerpretação, provavelmente devido ao tamanho do *zoom* adotado. Dessa forma, é evidente a importância de informações obtidas por meio de medições no campo para a comparação. Por isso, a próxima etapa subsequente seria, a partir das coordenadas obtidas pelo YOLOv4, realizar um levantamento de campo para validar os dados de sensoriamento remoto.

A principal limitação encontrada nos estudos para detecção de espécies florestais a partir de imagens de sensoriamento remoto está concentrada, principalmente, na identificação de indivíduos localizados muito próximos entre si. A maioria das investigações até agora foi desenvolvida para plantios florestais ou áreas abertas com baixa sobreposição de copas (SHAFRI *et al.*, 2011; SRESTASATHIER e RAKWATIN,2014; LI *et al.*, 2019; WU *et al*., 2020). Em contrapartida, pode-se observar na Figura 19 que o YOLOv4 detectou satisfatoriamente as palmeiras localizadas próximas umas das outras ou aquelas com as copas parcialmente cobertas pelo dossel da floresta.

A utilização de imagens *RGB*, mesmo que de alta resolução, limita o modelo YOLOv4 a detectar apenas os indivíduos que atingiram o dossel da floresta. Entretanto, isto não é um problema quando o objetivo do inventário é quantificar o potencial produtivo das palmeiras, uma vez que as palmeiras produtivas geralmente são indivíduos adultos que alcançaram estrato superior (HENDERSON, 1995; LORENZI *et al.*, 2000; ROCHA, 2004). Uma alternativa para ampliar a detecção a nível de subdossel é a realização de novas pesquisas com métodos que utilizam dados de imagens associados a *LiDAR* (*Light Detection and Ranging*). Além de obter o perfil da floresta para os estratos

inferiores, a nuvem de pontos *LiDAR* pode fornecer informações específicas da palmeira uma perspectiva tridimensional, o que poderia melhorar a aprendizagem.

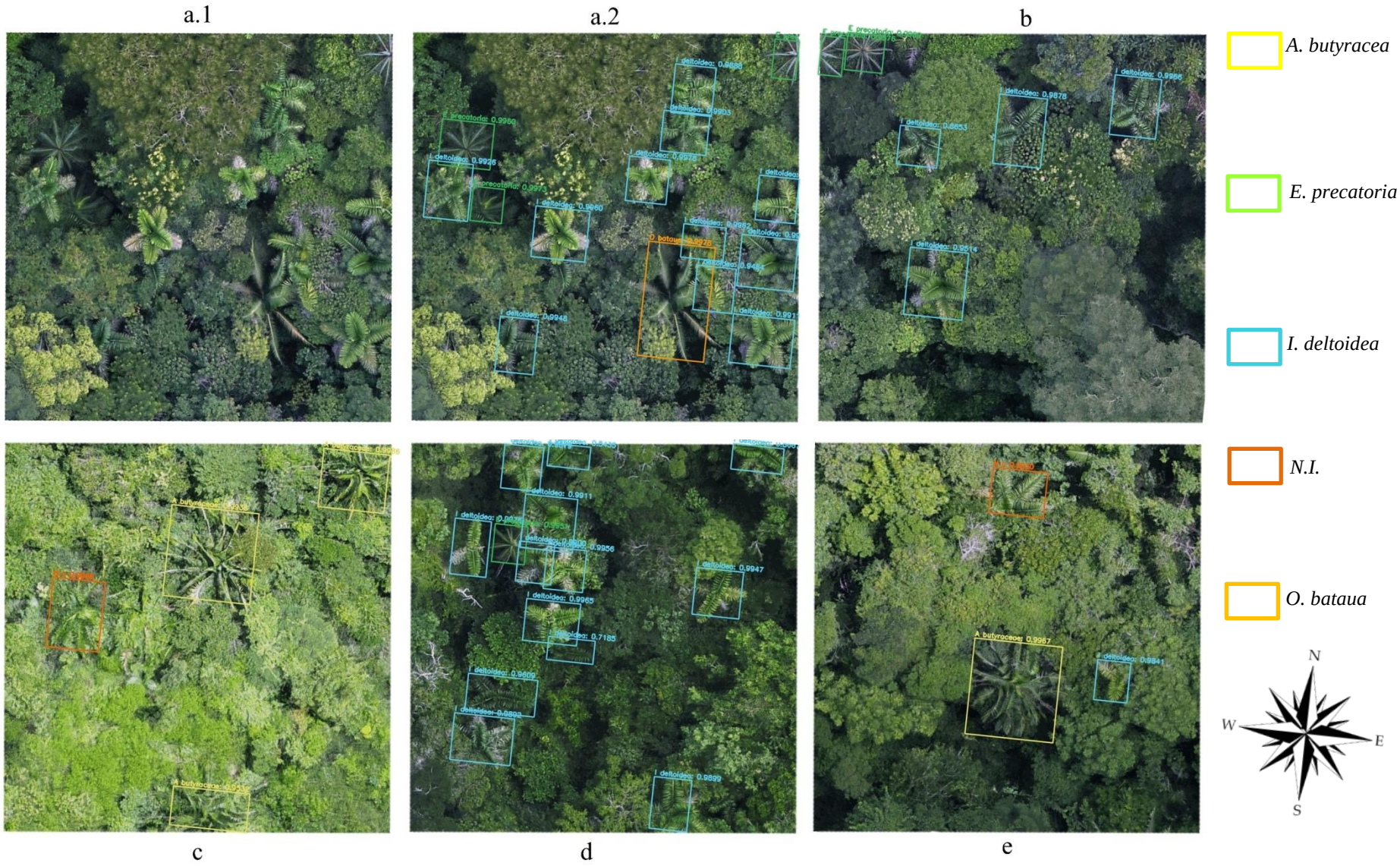


**Figura 19** Detecção de palmeiras pelo YOLOv4, em que, a.1 = parcela antes da predição; a.2 = predição para parcela a.1; *b*, *c*, *d* e *e* = outras predições.

Recentemente, estudos para a detecção de palmeiras individuais em nível de espécies vêm sendo desenvolvidos (DOS SANTOS *et al*., 2017; TAGLE CASAPIA *et al.*, 2020; FERREIRA *et al*., 2020). No entanto, até onde se sabe, este é o primeiro estudo que utilizou detector de objetos, de ponta a ponta (de único estágio), para a detecção de palmeiras em nível de espécie em floresta natural na região Amazônica.

Santos *et al.* (2017) desenvolveram um algoritmo "*ComptPalm*" adaptado a ambientes abertos para detectar cinco espécies de palmeiras com grande copa circular usando uma imagem de satélite pancromática *GeoEye* de alta resolução (*pixel* = 0,5 cm) no sudeste da Amazônia. No estudo, 75,45% das palmeiras foram detectadas com sucesso em áreas de pastagem.

Tagle Casapia *et al.* (2020) desenvolveram um método para a identificação e quantificação da abundância de palmeiras economicamente importantes no noroeste do Peru usando imagens de *RGB* a partir de VANTs. O método proposto por esses autores é baseado na segmentação da imagem por crescimento de regiões e é adequado para áreas com média a baixa densidade de palmeiras.

Ferreira *et al.* (2020) desenvolveram um método baseado em mapas de pontuação derivados de um modelo de rede totalmente convolucional para detectar e classificar copas individuais de palmeiras no oeste da Amazônia. A taxa de precisão do classificador obtida pelos autores foi de 98,6%, 96,6% e 78,6% para *E. precatoria, I. deltoidea* e *A. butyracea*, respectivamente.

O YOLOv4, uma vez que apresentou boa capacidade de generalização (*F1-score =0,89)*, pode ser utilizado para detecções de palmeiras em novas áreas, além de ser extremamente flexível pelo fato de ser construído sobre uma estrutura de código aberto, podendo ser executado em um navegador *web* sem a necessidade de grande capacidade computacional. Além disso, YOLOv4 pode ser facilmente treinado com o objetivo de incluir novas espécies.

Ainda que o modelo YOLOv4 tenha levado em média 8 horas para concluir o treinamento, o tempo necessário para as predições foi de 3 milésimos de segundo por parcela (0,140625 hectares), ou seja, o modelo levou 28,8 segundos para realizar as predições para a área total de estudo (135 hectares). O tempo total necessário para realizar a predição, desde o preparo das imagens de entrada, a predição pelo modelo e remoção das duplicatas, foi de aproximadamente 15 minutos.

## 5.2 Densidade de palmeiras e distribuição espacial

Com base nos resultados apresentados no item 5.1, o melhor conjunto de pesos foi selecionado e a densidade populacional foi estimada para área total de estudo. A densidade total

de palmeiras (incluindo todas as classes) foi de 12,2 indivíduos por hectare. Dentre as espécies identificadas (Tabela 6), *E. precatoria* foi a espécies que apresentou maior densidade na área de estudo, seguida por *I. deltoidea* e *O. bataua*. Em paralelo, comparando-se com as demais espécies, a classe de Palmeiras Não identificada apresentou elevado valor de densidade, exibindo 2,73 palmeiras por hectare.

**Tabela 6** Densidade para palmeiras detectadas a partir de imagens *RGB* obtidas por *RPA* em um fragmento de Floresta Ombrófila Aberta na Amazônia Ocidental.

| **Espécie** | ***N*** | ***DA*(ind.ha$^{-1}$)** | ***DR*(%)** | ***FA*(%)** | ***FR*(%)** | ***Distribuição*** |
|---|---|---|---|---|---|---|
| *A. butyracea* | 141 | 1,04 | 8,56 | 0,12 | 11,39 | Agregado e Aleatório |
| *E. precatoria* | 636 | 4,71 | 38,62 | 0,38 | 37,44 | Agregado |
| *I deltoidea* | 425 | 3,15 | 25,80 | 0,17 | 16,48 | Agregado |
| N.I. | 368 | 2,73 | 22,34 | 0,28 | 27,37 | Agregado e Aleatório |
| *O. bataua* | 77 | 0,57 | 4,68 | 0,08 | 7,32 | Agregado e Aleatório |

*N* =Número total indivíduos em 135 hectares; *DA*= Densidade Absoluta; *DR*= Densidade Relativa; *FA*= Frequência Absoluta; *FR*= Frequência Relativa; *N.I.*= Palmeiras Não Identificadas.

Diferentes estimativas de densidade de *E. precatoria* foram encontradas por estudos no estado do Acre (ROCHA, 2004; ACRE, 2006; TER STEEGE *et al.* 2013, LOPES *et al.*, 2019). Rocha (2004) ao avaliar o potencial ecológico para o manejo de frutos de *E. precatoria* em áreas extrativistas de terra firme no estado do Acre, encontrou densidade média de 23 indivíduos por hectare. Lopes *et al.* (2019) mapearam a favorabilidade de ocorrência e densidade de *E. precatoria* para o estado do Acre e encontraram as menores e maiores densidades para a espécie na região do Baixo Acre (0,2 a 280 palmeiras por hectare) e de 10 a 45 palmeiras por hectare para florestas de terra firme. No presente estudo, é importante ressaltar que foram considerados apenas os indivíduos de palmeiras que atingiram o dossel da floresta e que puderam ser visualizados a partir das imagens do VANT.

Os baixos valores de densidade para *A. butyracea* e *O. bataua*, podem estar diretamente relacionados ao bom estado de conservação do remanescente florestal estudado, bem como com as condições adaptativas da espécie. *A. butyracea*, por exemplo, tem tendência de prosperar em locais com maior antropização, uma vez que é uma palmeira exigente em luz e a taxa de sobrevivência da regeneração é maior devido à baixa incidência de predadores próximo a essas áreas perturbadas (URIBE *et al.*, 2001; WRIGHT e DUBER, 2001; MERTZLUFFT *et al.*, 2020). Já *O. bataua*, apesar de ocorrer em terra firme, é adaptado às áreas úmidas de baixio, justificando a baixa densidade encontrada na área do presente estudo.

Quanto à frequência, observa-se que a espécie *E. precatoria* ocorre em aproximadamente 37% das parcelas, evidenciado uma maior distribuição dessa espécie na área de estudo (Figura 20).

Para determinar o padrão espacial das espécies de palmeiras estudadas, foi realizada uma análise gráfica (Figura 21) pela função *K* de Ripley (RIPLEY, 1979).

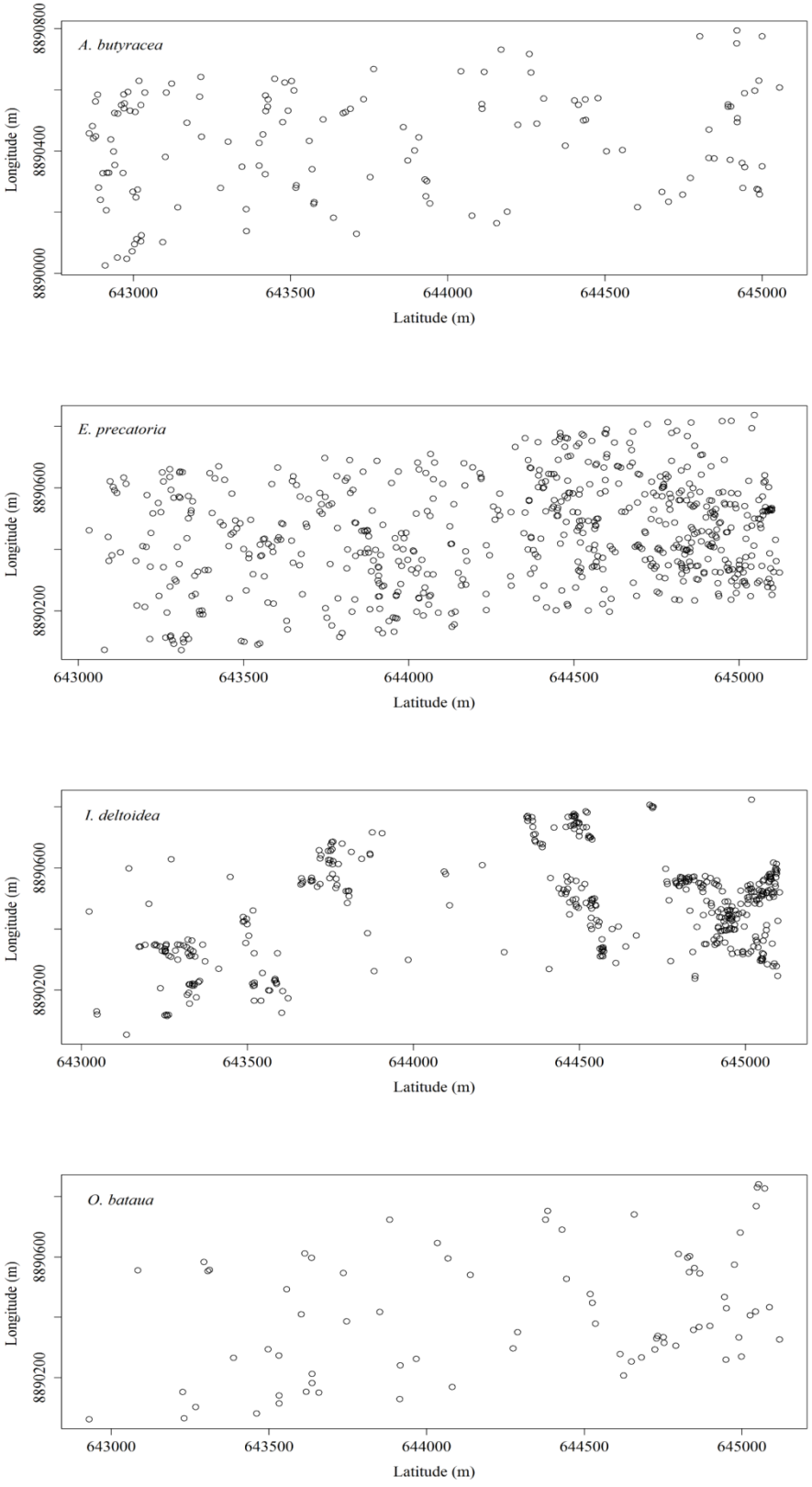


**Figura 20** Disposição das espécies de palmeiras identificadas no presente estudo.

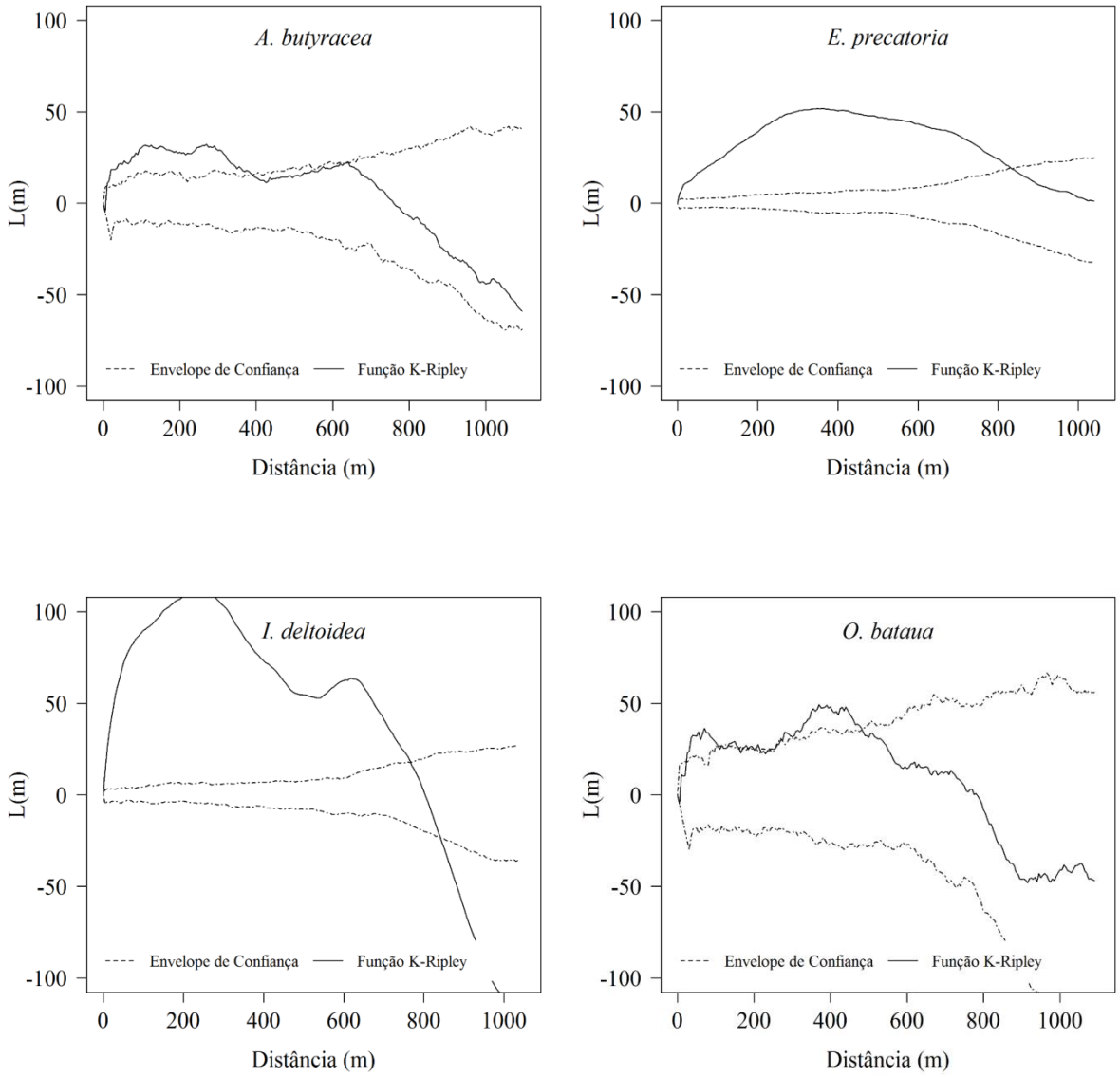


**Figura 21** Análise da distribuição espacial das espécies de palmeiras a partir da função *K* de Ripley. Linhas pontilhadas representam o intervalo de confiança com 1000 simulações.

Como descrito no item 4.11, o ponto inicial da função *K* de Ripley (linha cheia) e do envelope (linhas tracejadas) inicia-se em *L(m)* = 0. O envelope delimita a região de completa aleatoriedade espacial, ou seja, a hipótese de nulidade. Com base nisso, é possível observar na Figura 21 que *A.butyracea*, *E. precatoria* e *O. bataua* apresentam padrão de distribuição espacial agregado em raios até aproximadamente 380 m, 800 m e 450 m de distâncias, respectivamente, passando para aleatório para maiores raios de distâncias. *I. deltoidea* apresenta padrão espacial agregado até 800 m de raio de distância e depois passa para um padrão espacial uniforme.

A estrutura espacial das espécies de palmeiras estudadas é altamente influenciada, principalmente, em sua fase inicial de desenvolvimento, por processos ecológicos como a dispersão de sementes, competição e herbivoria (WRIGHT, 2002; SEIDLER e PLOTKIN, 2006).

Estes padrões de distribuição espacial para as espécies estudadas já eram esperados uma vez que a dispersão espacial das palmeiras está diretamente relacionada ao tipo de dispersão de seus frutos. Além disso, na área de estudo há ocorrência de bambu (*Guadua* spp.) e, de maneira geral, as florestas com presença de bambu, possuem menor riqueza florística e densidade de árvores (FERREIRA, 2014). Embora a ocorrência de bambu possa afetar a densidade das palmeiras, não interferiu na detecção das palmeiras pelo modelo YOLOv4, uma vez que as palmeiras adultas se sobressaem em relação aos tabocais.

Conhecer a densidade e padrão espacial das palmeiras candidatas ao manejo é extremamente importante para quantificação do potencial produtivo de uma floresta e serve como base para o planejamento. A produtividade de palmeiras, muitas vezes, é estimada com base na densidade de indivíduos (ROCHA, 2004; LOPES *et al.*, 2019), todavia quantificar a densidade no campo torna-se inviável do ponto de vista estratégico e operacional. Diante disso, método proposto por este estudo, mostrou ser uma ferramenta eficaz e importante para o mapeamento do potencial produtivo de palmeiras economicamente importantes na Amazônia Ocidental.

# 6 CONCLUSÕES

Neste estudo, aplicou-se uma técnica de visão computacional muito conhecida na área da robótica, desenvolvimento de carros autônomos, medicina e segurança, para a identificação e classificação de palmeiras economicamente importantes no sudoeste da Amazônia. O sistema de detecção de objetos implementado é mais uma confirmação da viabilidade em integrar Redes Neurais Convolucionais a imagens *RGB* obtidas por Aeronaves Remotamente Pilotadas de baixo custo.

Ao alterar a composição de um conjunto de dados originais, aplicando técnicas de aumento de dados na base criada para treinamento, observou-se que a precisão média geral do modelo YOLOv4 aumentou de 72,61% para 91,08% quando comparadas ao conjunto dados não aumentados. Além disso, as configurações de treinamento utilizadas permitiram ao modelo aprender as características das quatro espécies de palmeiras estudadas e realizar predições para áreas não vistas pelo modelo com elevada capacidade de generalização (*F1-score* = 0,89).

O YOLOv4 pode ser uma importante ferramenta para o mapeamento das palmeiras em florestas nativas, uma vez que, além de identificar a espécie que cada palmeira pertence, possibilita a obtenção das coordenadas geográficas exatas de cada indivíduo detectado, servindo de suporte para o planejamento e o manejo de produtos florestais não-madeireiros.

A tecnologia apresentada neste estudo poderá facilitar a incorporação de novas áreas ao sistema de manejo de espécies amazônicas de interesse econômico pelas comunidades extrativistas no Acre, as quais serão as maiores beneficiadas. No entanto, são necessárias iniciativas que busquem ampliar o acesso destas comunidades às novas tecnologias, por exemplo, por meio de assessoria técnica, científicas e equipamentos tecnológicos.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS


ANTONIOU, A.; STORKEY, A.; EDWARDS, H. Data augmentation generative adversarial networks. **arXiv preprint arXiv:1711.04340**, 2017.

ANTHONY, M.; HOLDEN, S. B. Cross-validation for binary classification by real-valued functions: theoretical analysis. In: COLT' 98: PROCEEDINGS OF THE ELEVENTH ANNUAL CONFERENCE ON COMPUTATIONAL LEARNING THEORY, 1998, New York. **Anais eletrônicos**… New York: Association for Computing Machinery, 1998. p. 218-229. Doi: 10.1145/279943.279987

ALBERTZ, J. Albrecht Meydenbauer – Pioneer of photogrammetric documentation of the cultural heritage. In: PROCEDIMENTOS DO XVIII. SIMPÓSIO INTERNACIONAL DA CIPA, 2001, Potsdam. **Anais eletrônicos**… Potsdam: CIPA Heritage Documentation, 2002. p. 19-25. Disponível em: <https://www.schweizerbart.de/publications/detail/artno/182024500/Proceedings_of_the_XVIII_International_Symposium_of_CIPA_2001Potsdam_Germany_September_18_21_2001>. Acesso em: 30 jun. 2020.

ALBERTZ, J. A look back 140 Years of "Photogrammetry" Some Remarks on the History of Photogrammetry. **Photogrammetric Engineering & Remote Sensing**, v. 73, n. 5, p. 504-506, 2007.

AL-MASNI, M. A.; AL-ANTARI, M. A.; PARK, J. M.; GI, G.; KIM, T. Y.; RIVERA, P.; VALAREZO, E.; CHOI, M.T.; HAN, S. M.; KIM, T. S. Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system. **Computer methods and programs in biomedicine**, v. 157, p. 85-94, 2018. Doi: 10.1016/j.cmpb.2018.01.017

ALVAREZ-LOAYZA, P.; WHITE JR, J. F.; TORRES, M. S.; BALSLEV, H.; KRISTIANSEN, T.; SVENNING, J. C.; GIL, N. Light converts endosymbiotic fungus to pathogen, influencing seedling survival and niche-space filling of a common tropical tree, Iriartea deltoidea. **PloS one**, v. 6, n. 1, p. e16386, 2011. Doi: 10.1371/journal.pone.0016386

BAKER, W. J.; DRANSFIELD, J. Beyond Genera Palmarum: progress and prospects in palm systematics. **Botanical Journal of the Linnean Society**, v. 182, n. 2, p. 207-233, 2016. Doi: 10.1111/boj.12401

BARRÉ, P.; STÖVER, B. C.; MÜLLER, K. F.; STEINHAGE, V. LeafNet: A computer vision system for automatic plant species identification. **Ecological Informatics**, v. 40, p. 50-56, 2017. Doi: 10.1016/j.ecoinf.2017.05.005

BARRETT, F.; MCROBERTS, R. E.; TOMPPO, E.; CIENCIALA, E.; WASER, L. T. A questionnaire-based review of the operational use of remotely sensed data by national forest inventories. **Remote Sensing of Environment**, v. 174, p. 279-289, 2016. Doi: 10.1016/j.rse.2015.08.029

BERNAL, R.; TORRES, C.; GARCÍA, N.; ISAZA, C.; NAVARRO, J.; VALLEJO, M. I.; GALEANO, G.; BALSLEV, H. Palm management in south america. **The Botanical Review**, v. 77, n. 4, p. 607-646, 2011. Doi: 10.1007/s12229-011-9088-6

BINOTI, M. L. M. S**. Redes neurais artificiais para prognose da produção de povoamentos não desbastados de eucalipto**. 2010. 54 f. Dissertação (Mestrado em Ciência Florestal) – Universidade Federal de Viçosa, Viçosa - MG, 2010.

BINOTI, D. H.; BINOTI, M. L. D. S.; LEITE, H. G.; SILVA, A. Redução dos custos em inventário de povoamentos equiâneos utilizando redes neurais artificiais. **Agrária - Revista Brasileira de Ciências Agrárias**, v. 8, p. 125- 129, 2013. Doi: 10.5039/agraria.v8i1a2209

BISHOP, C. M. **Pattern recognition and machine learning**. New York: Springer, 2006. 738 p.

BOSQUET, B.; MUCIENTES, M.; BREA, V. M. STDnet: Exploiting high resolution feature maps for small object detection. **Engineering Applications of Artificial Intelligence**, v. 91, p. 103615, 2020. Doi: 10.1016/j.engappai.2020.103615

BOCHKOVSKIY, A.; WANG, C. Y.; LIAO, H. Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. **arXiv preprint arXiv:2004.10934**, 2020.

CARRANZA-GARCÍA, M.; TORRES-MATEO, J.; LARA-BENÍTEZ, P.; GARCÍA-

GUTIÉRREZ, J. On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data. **Remote Sensing**, v. 13, n. 1, p. 89, 2021. Doi: 10.3390/rs13010089

CASSEMIRO, G. H. M.; PINTO, H. B. **Composição e processamento de imagens aéreas de alta-resolução obtidas com drone.** 2014. 80 f. Trabalho de Conclusão de Curso (Engenharia Eletrônica) – Faculdade UnB Gama (FGA), Universidade de Brasília (UnB), Brasília, 2014.

BRAGA, J. R. G.; PERIPATO, V.; DALAGNOL, R.; FERREIRA, M. P.; TARABALKA, Y.; ARAGÃO, L. E. O. C.; VELHO, H. F. C.; SHIGUEMORI, E. H.; WAGNER, F. H. Tree Crown Delineation Algorithm Based on a Convolutional Neural Network. **Remote Sensing**, v. 12, n. 8, p. 1288, 2020. Doi: doi:10.3390/rs12081288

BOVI, M. L. A; CASTRO, A. Assaí (Euterpe oleracea, Palmae). In: CLAY, J. W.; CLEMENT, C. R (Org.). **Selected species and strategies to enhance income generation from Amazonian forests**. Rome: FAO, 1993. p.58-67.

CARNEIRO, T.; DA NÓBREGA, R. V. M.; NEPOMUCENO, T.; BIAN, G. B.; DE ALBUQUERQUE, V. H. C.; REBOUCAS FILHO, P. P. Performance analysis of google colaboratory as a tool for accelerating deep learning applications. **IEEE Access**, v. 6, p. 61677-61685, 2018. Doi: 10.1109/ACCESS.2018.2874767

CHEN, C.; LIU, MY.; TUZEL, O.; XIAO, J. R-CNN for Small Object Detection. In: LAI, S. H.; LEPETIT, V.; NISHINO, K.; SATO, Y (Org.). **Computer Vision – ACCV 2016**: **13th Asian Conference on Computer Vision, Taipei, Taiwan.** Springer, Cham, 2017. p. 214-230. Doi: 10.1007/978-3-319-54193-8_14

CHEN, L. C.; ZHU, Y.; PAPANDREOU, G.; SCHROFF F.; ADAM H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: FERRARI, V.; HEBERT, M.; SMINCHISESCU, C.; WEISS, Y (Org.). **Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany**. Springer, Cham, 2018. p. 801-818. Doi: 10.1007/978-3-030-01234-2_49

CHEN, X.; KUNDU, K.; ZHU, Y.; BERNESHAWI, A. G.; MA, H.; FIDLER, S.; URTASUN, R. 3d object proposals for accurate object class detection. In: CORTES, C.;

LAWRENCE, N.; LEE, D.; SUGIYAMA, M.; GARNETT, R (Org.). **Advances in Neural Information Processing Systems 28.** NIPS, 2015. p. 424-432.Disponível em: <https://papers.nips.cc/paper/2015>. Acesso em: 25 out. 2020.

DENG, J.; DONG, W.; SOCHER, R.; LI, L. J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2009,Miami. **Anais eletrônicos…** Miami: IEEE, 2009. p. 248-255. DOI: 10.1109/CVPR.2009.5206848

DOLLAR, P.; WOJEK, C.; SCHIELE, B.; PERONA, P. Pedestrian detection: An evaluation of the state of the art. **IEEE transactions on pattern analysis and machine intelligence**, v. 34, n. 4, p. 743-761, 2011.Doi: 10.1109/TPAMI.2011.155

D'OLIVEIRA, M. V. N.; REUTEBUCH, S. E.; MCGAUGHEY, R. J.; ANDERSEN, H. E. Estimating forest biomass and identifying low-intensity logging areas using airborne scanning lidar in Antimary State Forest, Acre State, Western Brazilian Amazon**. Remote Sensing of Environment**, v. 124, p. 479-491, 2012. Doi: 10.1016/j.rse.2012.05.014

DOS SANTOS, A. M.; MITJA, D.; DELAÎTRE, E.; DEMAGISTRI, L.; DE SOUZA MIRANDA, I.; LIBOUREL, T.; PETIT, M. Estimating babassu palm density using automatic palm tree detection with very high spatial resolution satellite images. **Journal of environmental management**, v. 193, p. 40-51, 2017. Doi: 10.1016/j.jenvman.2017.02.004

EISERHARDT, W. L.; SVENNING, J. C.; KISSLING, W. D.; BALSLEV, H. Geographical ecology of the palms (Arecaceae): determinants of diversity and distributions across spatial scales. **Annals of Botany**, v. 108, n. 8, p. 1391-1416, 2011. Doi: 10.1093/aob/mcr146

ENE, L. T.; NÆSSET, E.; GOBAKKEN, T.; BOLLANDSÅS, O. M.; MAUYA, E. W.; ZAHABU, E. Large-scale estimation of change in aboveground biomass in miombo woodlands using airborne laser scanning and national forest inventory data. **Remote Sensing of Environment**, v. 188, p. 106-117, 2017. Doi: 10.1016/j.rse.2016.10.046

EVERINGHAM, M.; VAN GOOL, L.; WILLIAMS, C. K.; WINN, J.; ZISSERMAN, A. The pascal visual object classes (voc) challenge. **International journal of computer vision**, v. 88, n. 2, p. 303-338, 2010. Doi: 10.1007/s11263-009-0275-4

FERREIRA, E. J. L. O bambu é um desafio para a conservação e o manejo de florestas no sudoeste da Amazônia. **Ciência e Cultura**, v. 66, n. 3, p. 46-51, 2014. Doi: http://dx.doi.org/10.21800/S0009-67252014000300015

FERREIRA, M. P.; ALMEIDA, D. R. A.; ALMEIDA Papa, D.; MINERVINO, J. B. S.; VERAS, H. F. P.; FORMIGHIERi, A.; SANTOS, C. A. N.; FERREIRA, M. A. D.; FIGUEIREDO, E. O.; FERREIRA, E. J. L.. Individual tree detection and species classification of Amazonian palms using UAV images and deep learning.**Forest Ecology and Management**, v. 475, p. 118397, 2020. DOI:10.1016/j.foreco.2020.118397

FIGUEIREDO, E. O. **Modelagem biométrica para arvores individuais a partir do Lidar em área de manejo de precisão em florestas tropicais na Amazônia Ocidental.** 2014. 86 f. Tese (Doutorado em Ciências de Florestas Tropicais) – Programa de Pós-Graduação em Ciências de Florestas Tropicais, Instituto Nacional de Pesquisas da Amazônia (INPA), Manaus, 2014.

FONTES, J. C.; POZZETTI, V. C. O Uso dos Veículos não Tripulados no Monitoramento Ambiental na Amazônia. **Revista de Direito e Sustentabilidade**, v. 2, n. 2, p. 149-164, 2016.

FREUDENBERG, M.; NÖLKE, N.; AGOSTINI, A.; URBAN, K.; WÖRGÖTTER, F.; KLEINN, C. Large scale palm tree detection in high resolution satellite images using U-Net. **Remote Sensing**, v. 11, n. 3, p. 312, 2019. Doi: 10.3390/rs11030312

GALO, M. L. B. T**. Caracterização Ambiental do Parque Estadual Morro do Diabo através de dados e técnicas de Sensoriamento Remoto: Abordagens utilizando redes neurais artificiais**. 2000. 205 f. Tese (Doutorado em Ciências da Engenharia Ambiental) – Escola de Engenharia de São Carlos (*EESC*), Universidade de São Paulo (USP), São Carlos, 2000.

GHIASI, G.; LIN, T. Y.; LE, Q. V. Dropblock: A regularization method for convolutional networks. In: BENGIO, S.; WALLACH, H.; LAROCHELLE, H.; GRAUMAN, K.; CESA-BIANCHI, N.; GARNETT, R (Org.). **Advances in Neural Information Processing Systems 31.** NeurIPS, 2018. p. 10727-10737. Disponível em: <https://papers.nips.cc/paper/2018>. Acesso em: 25 dez. 2020.

GIRSHICK, R., DONAHUE, J., DARRELL, T., & MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In:**Proceedings of the IEEE conference on computer vision and pattern recognition.** IEEE, p. 580-587, 2014.Doi: 10.1109/CVPR.2014.81

GIRSHICK, R. Fast r-cnn. In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2015, Santiago. **Anais eletrônicos…** Santiago: IEEE, 2015. p. 1440-1448. Doi: 10.1109/ICCV.2015.169

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning: Adaptive Computation and Machine Learning Series.** The MIT Press**,** 2016. 800 p.

GOOGLE. **Colaboratory: Frequently Asked Questions**. Disponível em: <https://research.google. com/colaboratory/faq.html>. Acesso em: 01 dez. de 2020.

HAO, R.; NAMDAR, K.; LIU, L.; HAIDER, M. A.; KHALVATI, F. A Comprehensive Study of Data Augmentation Strategies for Prostate Cancer Detection in Diffusion-weighted MRI using Convolutional Neural Networks. **arXiv preprint arXiv:2006.**01693, 2020.

HASHEMI, M. Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation. **Journal of Big Data**, v. 6, n. 1, p. 98, 2019. Doi: 10.1186/s40537-019-0263-7

HE, K.; ZHANG, X.; REN, S.; SUN, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. **IEEE transactions on pattern analysis and machine intelligence**, v. 37, n. 9, p. 1904-1916, 2015. Doi: 10.1109/TPAMI.2015.2389824

HENDERSON, A.; GALEANO, G. e BERNAL, R. 1995. **Field guide to the palms of the Americas. Pinceton University Press**, New Jersey. 363p.

HOI, S. C.; WU, X.; LIU, H.; WU, Y.; WANG, H.; XUE, H.; WU, Q. Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks. **arXiv preprint arXiv:1511.02462**, 2015.

HUANG, G.; LIU, Z.; VAN DER MAATEN, L.; WEINBERGER, K. Q. Densely connected convolutional networks. In: 2017 IEEE CONFERENCE ON COMPUTER

VISION AND PATTERN RECOGNITION (CVPR), 2017, Honolulu. **Anais eletrônicos…** Honolulu: IEEE, 2017. p. 2261-2269. Doi: 10.1109/CVPR.2017.243

HUANG, G.; LI, Y.; PLEISS, G.; LIU, Z.; HOPCROFT, J. E.; WEINBERGER, K. Q. Snapshot ensembles: Train 1, get m for free. **arXiv preprint arXiv:1704.00109**, 2017.

IBGE (Instituto Brasileiro de Geografia e Estatística). **Potencial Florestal do Estado do Acre (Relatório Técnico)**. Rio de Janeiro: IBGE, 2005. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/livros/liv95899.pdf>. Acesso: 01 nov. 2020.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. **arXiv preprint arXiv:1502.03167**, 2015.

ITAKURA, K.; HOSOI, F. Automatic tree detection from three-dimensional images reconstructed from 360 spherical camera using YOLO v2. **Remote Sensing**, v. 12, n. 6, p. 988, 2020. Doi: 10.3390/rs12060988

KARLIK, B.; OLGAC, A. V. Performance analysis of various activation functions in generalized MLP architectures of neural networks. **International Journal of Artificial Intelligence and Expert Systems**, v. 1, n. 4, p. 111-122, 2011.

KISSLING, W. D.; BALSLEV, H.; BAKER, W. J.; DRANSFIELD, J.; GÖLDEL, B.; LIM, J. Y.; ONSTEIN, R. E.; SVENNING, J. C. PalmTraits 1.0, a species-level functional trait database of palms worldwide. **Scientific Data**, v. 6, n. 178, 2019. Doi: 10.1038/s41597-019-0189-0

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNET classification with deep convolutional neural networks. **Communications of the ACM**, v. 60, n. 6, p. 84-90, 2017. Doi: 10.1145/3065386

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; Haffner, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278-2324, 1998. Doi: 10.1109/5.726791

LESCURE, J. P. Algumas questões a respeito do extrativismo. In: EMPERAIRE, L (Org.). **A floresta em jogo: o extrativismo na Amazônia Central**. São Paulo: UNESP, 2000.

LI, H.; LIN, Z.; SHEN, X.; BRANDT, J.; HUA, G. A convolutional neural network cascade for face detection. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2015, Boston. **Anais eletrônicos…** Boston: IEEE, 2015. p. 5325-5334. Doi: 10.1109/CVPR.2015.7299170

LI, Y.; ZHANG, Y.; YU, J. G.; TAN, Y.; TIAN, J.; MA, J. A novel spatio-temporal saliency approach for robust dim moving target detection from airborne infrared image sequences. **Information Sciences**, v. 369, p. 548-563, 2016. Doi: 10.1016/j.ins.2016.07.042

LI, J.; LIANG, X.; SHEN, S.; XU, T.; FENG, J.; YAN, S. Scale-aware fast R-CNN for pedestrian detection. **IEEE transactions on Multimedia**, v. 20, n. 4, p. 985-996, 2017. Doi: 10.1109/TMM.2017.2759508

LIANG, X.; KANKARE, V.; HYYPPÄ, J.; WANG, Y.; KUKKO, A.; HAGGRÉN, H.; YU, X.; KAARTINEN, H.; JAAKKOLA, A.; GUAN, F.; HOLOPAINEN, M.; VASTARANTA, M. Terrestrial laser scanning in forest inventories. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 115, p. 63-77, 2016. Doi: 10.1016/j.isprsjprs.2016.01.006

LIN, T. Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; ZITNICK, C. L.; DOLLÁR, P. Microsoft coco: Common objects in context. In: FLEET, D.; PAJDLA, T.; SCHIELE, B.; TUYTELAARS, T (Org.). **Computer Vision -- ECCV 2014: 13th European Conference, Zurich, Switzerland**. Springer, Cham, 2014. p. 740-755. Disponível: <https://www.springer.com/gp/book/9783319106014>. Acesso 10 out. 2020.

LI, W.; DONG, R.; FU, H.; YU, L. Large-scale oil palm tree detection from high-resolution satellite images using two-stage convolutional neural networks. **Remote Sensing**. v. 11, p. 11, Doi: 10.3390/rs11010011

LIPPMANN, R. P. An introduction to computing with neural nets. **IEEE ASSP Magazine**, v.4, n. 2, p. 4-22, 1987. Doi: 10.1109/MASSP.1987.1165576

LITJENS, G.; KOOI, T.; BEJNORDI, B. E.; SETIO, A. A. A.; CIOMPI, F.; GHAFOORIAN, M.; VAN DER LAAK, J. A. W. M.; VAN GINNEKEN, B.;

SÁNCHEZ, C. I. A survey on deep learning in medical image analysis. **Medical image analysis**, v. 42, p. 60-88, 2017. Doi: 10.1016/j.media.2017.07.005

LIU, S.; QI, L.; QIN, H.; SHI, J.; JIA, J. Path aggregation network for instance segmentation. In: 2018 IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2018, Salt Lake City. 2018. **Anais eletrônicos…** Salt Lake City: IEEE, 2018. p. 8759-8768. Doi: 10.1109/CVPR.2018.00913

LIU, W.; ANGUELOV, D.; ERHAN, D.; SZEGEDY, C.; REED, S.; FU, C. Y.; BERG, A. C. Ssd: Single shot multibox detector. In: LEIBE, B.; MATAS, J.; SEBE, N.; WELLING, M (Org.). **Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands**. Springer, Cham, 2016. p. 21-37. Disponível: <https://link.springer.com/book/10.1007/978-3-319-46448-0>. Acesso 02 out. 2020.

LIU, W.; WEN, Y.; YU, Z.; LI, M.; RAJ, B.; SONG, L. Sphereface: Deep hypersphere embedding for face recognition. In: 2017 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2017, Honolulu. **Anais eletrônicos**… Honolulu: IEEE, 2017. p. 6738-6746. Doi: 10.1109/CVPR.2017.713

LIU, Z .; WU, W .; GU, X .; LI, S .; WANG, L .; ZHANG, T. Aplicação da combinação de modelos YOLO e imagens 3D GPR na detecção e manutenção de estradas. **Remote Sensing**,v. 13 , p. 1081, 2021. Doi: 10.3390/rs13061081

LOPES, E.; SOARES-FILHO, B.; SOUZA, F.; RAJÃO, R.; MERRY, F.; RIBEIRO, S. C. Mapping the socio-ecology of Non Timber Forest Products (NTFP) extraction in the Brazilian Amazon: The case of açaí (Euterpe precatoria Mart) in Acre. **Landscape and Urban Planning**, v. 188, p. 110-117, 2019. Doi: 10.1016/j.landurbplan.2018.08.025

LORENZI, H. **Plantas daninhas do Brasil: terrestres, aquáticas, parasitas, tóxicas e medicinais.** 2. ed. Nova Odessa: Plantarum, 2000. 425p.

LOUREIRO, H. A. S.; GUERRA, A. J. T.; ANDRADE, A. G. Contribuição ao estudo de voçorocas a partir do uso experimental de Laser Scanner terrestre e VANT. **Revista Brasileira de Geomorfologia**, v. 21, n. 4, 2020. Doi: http://dx.doi.org/10.20502/rbg.v21i4.1880

MAAS, A. L.; HANNUN, A. Y.; NG, A. Y. Rectifier nonlinearities improve neural network acoustic models. In: DASGUPTA, S.; MCALLESTER, D (Org.). **ICML'13: Proceedings of the 30th International Conference on International Conference on Machine Learning**. JMLR, 2013.

MARTINOT, J. F.; PEREIRA, H. S.; SILVA, S. C. P. Coletar ou Cultivar: as escolhas dos produtores de açaí-da-mata (Euterpe precatoria) do Amazonas. **Revista de Economia e Sociologia Rural**, v. 55, n. 4, p. 751-766, 2017. Doi: 10.1590/1234-56781806-94790550408

MASCARO, J.; DETTO, M.; ASNER, G. P.; MULLER-LANDAU, H. C. Evaluating uncertainty in mapping forest carbon with airborne LiDAR. **Remote Sensing of Environment**, New York, v. 115, n. 12, p. 3770-3774, 2011. Doi: 10.1016/j.rse.2011.07.019

MEYDENBAUER, A.; MEYER, R. **Baukunst in historischen Fotografien**. Fotokinoverlag, 1985.

MERTZLUFFT, C. E.; MADDEN, M.; GOTTDENKER, N. L.; RUNK, J. V.; SALDAÑA, A.; TANNER, S.; CALZADA, J. E.; YAO, X. Landscape disturbance impacts on Attalea butyracea palm distribution in central Panama. **International Journal of Health Geographics**, v. 19, n. 1, p. 1-17, 2020. Doi: 10.1186/s12942-020-00244-y

MIKOŁAJCZYK, A.; GROCHOWSKI, M. Data augmentation for improving deep learning in image classification problem. In: 2018 INTERNATIONAL INTERDISCIPLINARY PHD WORKSHOP (IIPHDW), 2018, Poland. **Anais eletrônico…** Poland: IEEE, 2018. p. 117-122. Doi: 10.1109/IIPHDW.2018.8388338

MISRA, D. Mish: A self regularized non-monotonic neural activation function. **arXiv preprint arXiv:1908.08681**, 2019.

MUBIN, N. A.; NADARAJOO, E.; SHAFRI, H. Z. M.; HAMEDIANFAR, A. Young and mature oil palm tree detection and counting using convolutional neural network deep learning method. **International Journal of Remote Sensing**, v. 40, n. 19, p. 7500-7515, 2019. Doi: 10.1080/01431161.2019.1569282

MONTÚFAR, R.; LAFFARGUE, A.; PINTAUD, J. C.; HAMON, S.; AVALLONE, S.; DUSSERT, S. Oenocarpus bataua Mart.(Arecaceae): Rediscovering a source of high oleic vegetable oil from Amazonia. **Journal of the American Oil Chemists' Society**, v. 87, n. 2, p. 167-172, 2010. Doi: 10.1007/s11746-009-1490-4

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: FÜRNKRANZ, J.; JOACHIMS, T (Org.). ICML'10: Proceedings of the 27th International Conference on International Conference on Machine Learning**.**Madison: Omnipress, 2010. p. 807–814. Disponível em: <https://dl.acm.org/doi/proceedings/10.5555/3104322#secAuthors>. Acesso: 12 nov. 2020.

NEOGI, S. **Exploring Multi-Class Classification with Deep Learning**. Medium, 2020. Disponível em: <https://medium.com/@srijaneogi31/exploring-multi-class-classification-with-deep-learning-239cb42e69bf> Acesso: 13 set. 2020.

NEUBECK, A.; VAN GOOL, L. Efficient non-maximum suppression. In: **18th International Conference on Pattern Recognition (ICPR'06)**. IEEE,v. 3, p. 850-85, 2006. DOI: 10.1109/ICPR.2006.479

NEVALAINEN, O.; HONKAVAARA, E.; TUOMINEN, S.; VILJANEN, N.; HAKALA, T.; YU, X.; HYYPPÄ, J.; SAARI, H.; PÖLÖNEN, I.; IMAI, N. N.; TOMMASELLI, A. M. Individual tree detection and classification with UAV-based photogrammetric point clouds and hyperspectral imaging. **Remote Sensing**, v. 9, n. 3, p. 185, 2017. Doi: 10.3390/rs9030185

NYGREN, A.; LACUNA-RICHMAN, C.; KEINÄNEN, K.; ALSA, L. Ecological, socio-cultural, economic and political factors influencing the contribution of non-timber forest products to local livelihoods: case studies from Honduras and the Philippines. **Small-scale Forest Economics, Management and Policy**, v. 5, n. 2, p. 249-269, 2006.

ONSTEIN, R. E.; BAKER, W. J.; COUVREUR, T. L.; FAURBY, S.; SVENNING, J. C.; e KISSLING, W. D. Frugivory-related traits promote speciation of tropical palms. **Nature ecology & evolution**, v. 1, n. 12, p. 1903-1911, 2017. Doi: 10.1038/s41559-017-0348-7

PAPA, D. de A. Impacto do manejo de precisão em florestas tropicais. **EMBRAPA-Acre-Tese/dissertação** (ALICE), 2018. Disponível em: <http://www.alice.cnptia.embrapa.br/alice/handle/doc/1096083>. Acesso: 05 abr. 2021.

PÉREZ, F.; GRANGER, B. E. IPython: a system for interactive scientific computing. **Computing in science & engineering**, v. 9, n. 3, p. 21-29, 2007. Doi: 0.1109/MCSE.2007.53

PEREZ MALLA, C. U.; VALDES HERNANDEZ; M. D. C.; RACHMADI, M. F.; KOMURA, T. Evaluation of enhanced learning techniques for segmenting ischaemic stroke lesions in brain magnetic resonance perfusion images using a convolutional neural network scheme. **Frontiers in neuroinformatics**, v. 13, p. 33, 2019. Doi: 10.3389/fninf.2019.00033

PÉREZ-RODRÍGUEZ, L. A.; QUINTANO, C.; MARCOS, E.; SUAREZ-SEOANE, S.; CALVO, L.; FERNÁNDEZ-MANSO, A. Evaluation of Prescribed Fires from Unmanned Aerial Vehicles (UAVs) Imagery and Machine Learning Algorithms. **Remote Sensing**, v. 12, n. 8, p. 1295, 2020. Doi: 10.3390/rs12081295

PINARD, M. Impacts of stem harvesting on populations of Iriartea deltoidea (Palmae) in an extractive reserve in Acre, Brazil. **Biotropica**, p. 2-14, 1993. Doi: 10.2307/2388974

PIOTROWSKI, A. P.; NAPIORKOWSKI, J. J. A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling. **Journal of Hydrology**, v. 476, p. 97-111, 2013. Doi: 10.1016/j.jhydrol.2012.10.019

POWERS, D. M.W. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. **arXiv preprint arXiv:2010.16061**, 2020.

PUTTEMANS, S.; VAN BEECK, K.; GOEDEMÉ, T. Comparing boosted cascades to deep learning architectures for fast and robust coconut tree detection in aerial images. In: IMAI, F.; TREMEAU, A.; BRAZ, J. (Org.). **Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - (Volume 5), in Funchal, Madeira, Portugal.** 2018. p. 230-241.Doi: 10.5220/0006571902300241

RAHMAN, M. A.; WANG, Y. Optimizing intersection-over-union in deep neural networks for image segmentation. In: BEBIS, G.; BOYLE, R.; PARVIN, B.; KORACIN, D.; PORIKLI, F.; SKAFF, S.; ENTEZARI, A.; MIN, J.; IWAI, D.; SADAGIC, A.; SCHEIDEGGER, C.; ISENBERG, T (Org.). **Advances in Visual Computing: 12th International Symposium, Las Vegas, 2016, Proceedings, Part II**. Springer, Cham, 2016. p. 234-244. Doi: 10.1007/978-3-319-50835-1_22

RAMACHANDRAN, P.; ZOPH, B.; LE, Q. V. Searching for activation functions. **arXiv preprint arXiv:1710.05941**, 2017.

REDMON, J; FARHADI, A. Yolov3: An incremental improvement. **arXiv preprint arXiv:1804.02767**, 2018.

REDMON, J., DIVVALA, S., GIRSHICK, R., e FARHADI, A. You only look once: Unified, real-time object detection. **arXiv:1506.02640v5**, 2016.

REED, R.; MARKSII, R. J. **Neural smithing: supervised learning in feedforward artificial neural networks**. Mit Press, 1999.

REN, S., HE, K., GIRSHICK, R., SUN, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In: **IEEE Transactions on Pattern Analysis and Machine Intelligence.** IEEE, v. 39, n. 6, p. 1137 – 1149, 2016. Doi: 10.1109/TPAMI.2016.2577031

REX, F. E.; SILVA, C. A.; DALLA CORTE, A. P.; KLAUBERG, C.; MOHAN, M.; CARDIL, A.; SILVA, V. S.; ALMEIDA, D. R. A.; GARCIA, M.; BROADBENT, E. N.; VALBUENA, R.; STODDART, J.; MERRICK, T.; HUDAK, A. T. Comparison of Statistical Modelling Approaches for Estimating Tropical Forest Aboveground Biomass Stock and Reporting their Changes in Low-intensity Logging Areas using Multi-temporal LiDAR Data. **Remote Sensing**, v. 12, n. 9, p. 1498, 2020. 10.3390/rs12091498

RIPLEY, B. D. Tests of randomness for spatial point patterns. **Journal of the Royal Statistic Society**, v. 41, p. 368-374, 1979. Doi: 10.1111/j.2517-6161.1979.tb01091.x

RIZZINI, C. T. Tratado de fitogeografia do Brasil: aspectos ecológicos, sociológicos e florísticos. Rio de Janeiro: Âmbito Cultural Edições Ltda. 1997.

REGO, J. F. Amazônia: do extrativismo ao neoextrativismo. **Ciência Hoje**, v. 25, n. 147, p. 62-65, 1999.

ROCHA, E. Potencial ecológico para o manejo de frutos de açaizeiro (Euterpe precatória Mart.) em áreas extrativistas no Acre, Brasil. **Acta amazónica**, v. 34, n. 2, p. 237-250, Manaus, 2004. Doi: http://dx.doi.org/10.1590/S0044-59672004000200012

RODRIGUES, T. E.; SILVA, J. M. L.; CORDEIRO, D. G.; A. GOMES, T. C. A.; CARDOSO JÚNIOR, E. Q. **Caracterização e classificação dos solos do campo experimental da Embrapa Acre, Rio Branco, Estado do Acre**. Belém: Embrapa Amazônia Oriental, 2001. 45 p.

SALMAN, S.; LIU, X. Overfitting mechanism and avoidance in deep neural networks. **arXiv preprint arXiv:1901.06566**, 2019.

SEIDLER, T. G.; PLOTKIN, J. B. Seed dispersal and spatial pattern in tropical trees. **PLoS Biology**, v. 4, n. 11, p. e344, 2006. Doi: 10.1371/journal.pbio.0040344

SELVARAJ, M. G.; VERGARA, A.; MONTENEGRO, F.; RUIZ, H. A.; SAFARI, N.; RAYMAEKERS, D.; OCIMATI, W.; NTAMWIRA, J.; TITS, L.; OMONDI, A. B.; BLOMME, G. Detection of banana plants and their major diseases through aerial images and machine learning methods: A case study in DR Congo and Republic of Benin. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 169, p. 110-124, 2020. Doi: 10.1016/j.isprsjprs.2020.08.025

SHAFRI, H.Z.M.; HAMDAM, N.; SARIPAN, M.I. Semi-automatic detection and counting of oil palm trees from high spatial resolution airborne imagery. **Int. J. Remote Sensing**, v 32,n.8, p. 2095-2115, 2011.Doi:10.1080/01431161003662928

SHANLEY, P.; MEDINA, G. **Frutíferas e plantas úteis na vida amazônica**. Belém: CIFOR, Amazon, 2005. 300 p.

SILVA, A. G. P.; GÖRGENS, E. B.; CAMPOE, O. C.; ALVARES, C. A.; STAPE, J. L.; RODRIGUEZ, L. C. E. Assessing biomass based on canopy height profiles using airborne laser scanning data in eucalypt plantations. **Scientia Agricola**, v. 72, n. 6, p. 504-512, 2015. Doi: 10.1590/0103-9016-2015-0070

SILVA, C. A.; HUDAK, A. T.; VIERLING, L. A.; LOUDERMILK, E. L.; O'BRIEN, J. J.; HIERS, J. K.; JACK, S. B.; GONZALEZ-BENECKE, C.; LEE, H.; FALKOWSKI, M. J.; KHOSRAVIPOUR, A. Imputation of individual longleaf pine (Pinus palustris Mill.) tree attributes from field and LiDAR data. **Canadian Journal of Remote Sensing**, v. 42, n. 5, p. 554-573, 2016. Doi: 10.1080/07038992.2016.1196582

SILVA, G. M. da. Aspectos Florísticos e Fitossociológicos de Palmeiras (arecaceae) em Florestas Com e Sem Bambu (Guadua Spp.) na Apa do Igarapé São Francisco, Acre. In**:** XX JORNADA DE INICIAÇÃO CIENTIFICA PIBIC INPA - CNPq/FAPEAM**,** 2012, Manaus. **Anais eletrônicos...** Manaus:INPA–CNPq/FAPEAM, 2012.

SOBRINHO, M. F. O.; CORTE, A. P. D.; VASCONCELLOS, B. N.; SANQUETTA, C. R.; REX, F. E. Uso de Veículos Aéreos Não Tripulados (VANT) para mensuração de processos florestais. **ENCICLOPÉDIA BIOSFERA, Centro Científico Conhecer**, v. 15, n. 27, p. 117–129, 2018. Doi: 10.18677/EnciBio_2018A80

SRESTASATHIERN, P.; RAKWATIN, P. Oil palm tree detection with high resolution multi-spectral satellite imagery. **Remote Sensing**, v. 6, n. 10, p. 9749-9774, 2014. Doi: 10.3390/rs6109749

SUN, Yi; WANG, X.; TANG, X. Deep learning face representation by joint identification-verification. **arXiv preprint arXiv:1406.4773**, 2014.

SUN, Y.; LIANG, D.; WANG, X.; TANG, X. Deepid3: Face recognition with very deep neural networks. **arXiv preprint arXiv:1502.00873**, 2015.

SZEGEDY, C.; VANHOUCKE, V.; IOFFE, S.; SHLENS, J.; WOJNA, Z. Rethinking the inception architecture for computer vision. In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2016, Las Vegas. **Anais eletrônico…** Las Vegas: IEEE, 2016. p. 2818-2826. Doi: 10.1109/CVPR.2016.308.

TER STEEGE, H.; HENKEL, T. W.; HELAL, N.; MARIMON, B. S.; MARIMON-JUNIOR, B. H.; HUTH, A.; GROENEVELD, J.; SABATIER, D.; SOUZA COELHO, L.; ANDRADE LIMA FILHO, D.; *et al*. Rarity of monodominance in hyperdiverse Amazonian forests. **Scientific reports**, v. 9, n. 1, p. 1-15, 2019. Doi: 10.1038/s41598-

019-50323-9

TAN, M.; LE, QUOC V. Efficientnet: Rethinking model scaling for convolutional neural networks. CHAUDHURI, K.; SALAKHUTDINOV, R (Org.). **Proceedings of the 36th International Conference on Machine Learning, 2019, Long Beach, California, USA.** PMLR, 2019. p. 6105-6114. Disponível em: <http://proceedings.mlr.press/v97/>. Acesso: 06 jan. 2021.

TURIAN, J.; BERGSTRA, J.; BENGIO, Y. Quadratic features and deep architectures for chunking. In: OSTENDORF, M.; COLLINS, M.; NARAYANAN, S.; OARD, D. W.; VANDERWENDE, L (Org.). **Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers**. Boulder, Colorado: Association for Computational Linguistics, 2009. p. 245-248. Disponível em: <https://www.aclweb.org/anthology/N09-2000/>. Acesso: 10 jan. 2021.

URIBE, A.; VELÁSQUEZ, P.; MONTOYA, M. Ecología de poblaciones de Attalea butyracea (Aracaceae) en un área de bosque seco tropical (Las Brisas, Sucre, Colombia). **Actualidades Biológicas**, v. 23, n. 74, p. 33-39, 2001.

VELOSO H. P. Sistema fitogeográfico. In: IBGE (Ed.). **Manual técnico da vegetação brasileira**. Série Manuais Técnicos em Geociências, v.1. Brasília: IBGE, p.8-38. 1992.

VENTURIERI, A.; SANTOS, J. R. Técnicas de Classificação de Imagens para Análise de Cobertura Vegetal. In: ASSAD, E. D.; SANO, E. E (Org.). **Sistema de Informação Geográfica: Aplicações na Agricultura**. Brasília: Embrapa – SPI/Embrapa- CPAC, 1998. p. 351-371.

VISSER, M. D.; MULLER-LANDAU, H. C.; WRIGHT, S. J.; RUTTEN, G.; JANSEN, P. A. Tri-trophic interactions affect density dependence of seed fate in a tropical forest palm. **Ecology letters**, v. 14, n. 11, p. 1093-1100, 2011. Doi: 10.1111/j.1461-0248.2011.01677.x

ORMISTO, J. Palms as rainforest resources: how evenly are they distributed in Peruvian Amazonia? **Biodiversity & Conservation**, v. 11, n. 6, p. 1025-1045, 2002.

WANG, C. Y.; LIAO, H. Y. M.; WU, Y. H.; CHEN, P. Y.; HSIEH, J. W.; YEH, I. H. CSPNet: A new backbone that can enhance learning capability of cnn. In: HE, K.; FAN, H.; WU, Y.; XIE, S.; GIRSHICK, R (Org.). **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops**. 2020. p. 390-391.Disponívelem:<https://openaccess.thecvf.com/content_CVPRW_2020/html/w28/Wang_CSPNet_A_New_Backbone_That_Can_Enhance_Learning_Capability_of_CVPRW_2020_paper.html>. Acesso: 09 dez. 2021.

PEREZ, L.; WANG, J. The effectiveness of data augmentation in image classification using deep learning. **Convolutional Neural Networks Vis. Recognit**, v. 11, 2017. Disponível em: < https://arxiv.org/abs/1712.04621>. Acesso: 02 mar. 2021.

TING K.M. Confusion Matrix. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning and Data Mining. **Springer**, Boston, MA, 2017. Doi: 10.1007/978-1-4899-7687-1_50

WEINSTEIN, B. G.; MARCONI, S.; BOHLMAN, S.; ZARE, A.; WHITE, E. Individual tree-crown detection in RGB imagery using semi-supervised deep learning neural networks. **Remote Sensing**, v. 11, n. 11, p. 1309, 2019. Doi: 10.3390/rs11111309

WRIGHT, S. J.; DUBER, H. C. Poachers and forest fragmentation alter seed dispersal, seed survival, and seedling recruitment in the Palm Attalea butyracea, with implications for tropical tree diversity 1. **Biotropica**, v. 33, n. 4, p. 583-595, 2001. Doi: 10.1111/j.1744-7429.2001.tb00217.x

WRIGHT, J. S. Plant diversity in tropical forests: a review of mechanisms of species coexistence. **Oecologia**, v. 130, n. 1, p. 1-14, 2002. Doi: 10.1007/s004420100809

WU, Y.; SUI, Y.; WANG, G. Vision-based real-time aerial object localization and tracking for UAV sensing system. **IEEE Access**, v. 5, p. 23969-23978, 2017. Doi: 10.1109/ACCESS.2017.2764419

Wu, W., Z., J.; F., H.; Li, W.; Yu, L.Cross-Regional Oil Palm Tree Detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. **Anais eletrônico**…IEEE. p. 56-57, 2020. Doi:10.1109/CVPRW50498.2020.00036

WULDER, M. A.; WHITE, J. C.; NELSON, R. F.; NÆSSET, E.; ØRKA, H. O.; COOPS, N. C.; HILKER, T.; BATER, C. W.; GOBAKKEN, T. Lidar sampling for large-area forest characterization: A review. **Remote Sensing of Environment**, v. 121, p. 196-209, 2012. Doi: 10.1016/j.rse.2012.02.001

XIAO, J.; HAYS, J.; EHINGER, K. A.; OLIVA, A.; TORRALBA, A. Sun database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2010, San Francisco. **Anais eletrônico…** San Francisco: IEEE, 2010. p. 3485-3492. Doi: 10.1109/CVPR.2010.5539970

YAMASHITA, R.; NISHIO, M.; DO, R. K. G.; TOGASHI, K. Convolutional neural networks: an overview and application in radiology. **Insights into imaging**, v. 9, n. 4, p. 611-629, 2018. Doi: 10.1007/s13244-018-0639-9

YAO, Z.; CAO, Y.; ZHENG, S.; HUANG, G.; LIN, S. Cross-iteration batch normalization. **arXiv preprint arXiv:2002.05712**, 2020.

ZHANG, C.; BENGIO, S.; HARDT, M.; RECHT, B.; VINYALS, O. Understanding deep learning requires rethinking generalization. **arXiv preprint arXiv:1611.03530**, 2016.

ZHANG, Y.; BAI, Y.; DING, M.; GHANEM, B. Multi-task Generative Adversarial Network for Detecting Small Objects in the Wild. **International Journal of Computer Vision**, p. 1-19, 2020. Doi: 10.1007/s11263-020-01301-6

ZHENG, J.; LI, W.; XIA, M.; DONG, R.; FU, H.; YUAN, S. Large-scale oil palm tree detection from high-resolution remote sensing images using faster-rcnn. In: IGARSS 2019 - 2019 IEEE INTERNATIONAL GEOSCIENCE AND REMOTE SENSING SYMPOSIUM, Yokohama, 2019. **Anais eletrônico…** Yokohama: IEEE, p. 1422-1425. Doi: 10.1109/IGARSS.2019.8898360

ZHENG, J.; FU, H.; LI, W.; WU, W.; YU, L.; YUAN, S.; TAO, W. Y. W.; PANG, T. K.; KANNIAH, K. D. Growing status observation for oil palm trees using Unmanned Aerial Vehicle (UAV) images. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 173, p. 95-121, 2021. Doi: 10.1016/j.isprsjprs.2021.01.008

ZHENG, Z. *et al.* Distance-IoU loss: Faster and better learning for bounding box regression. arXiv 2019. **arXiv preprint arXiv:1911.08287**, 2020.

ZHONG, Z.; ZHENG, L.; KANG, G.; LI, S.; YANG, Y. Random Erasing Data Augmentation. **Proceedings of the AAAI Conference on Artificial Intelligence,** v. 34, n. 07, p. 13001-13008, 2020. Doi: 10.1609/aaai.v34i07.7000

ZHU, X. X.; TUIA, D.; MOU, L.; XIA, G. S.; ZHANG, L.; XU, F.; FRAUNDORFER, F. Deep learning in remote sensing: A comprehensive review and list of resources. **IEEE Geoscience and Remote Sensing Magazine**, v. 5, n. 4, p. 8-36, 2017. Doi: 10.1109/MGRS.2017.2762307

# 8 ANEXOS

## 8.1 Anexo I

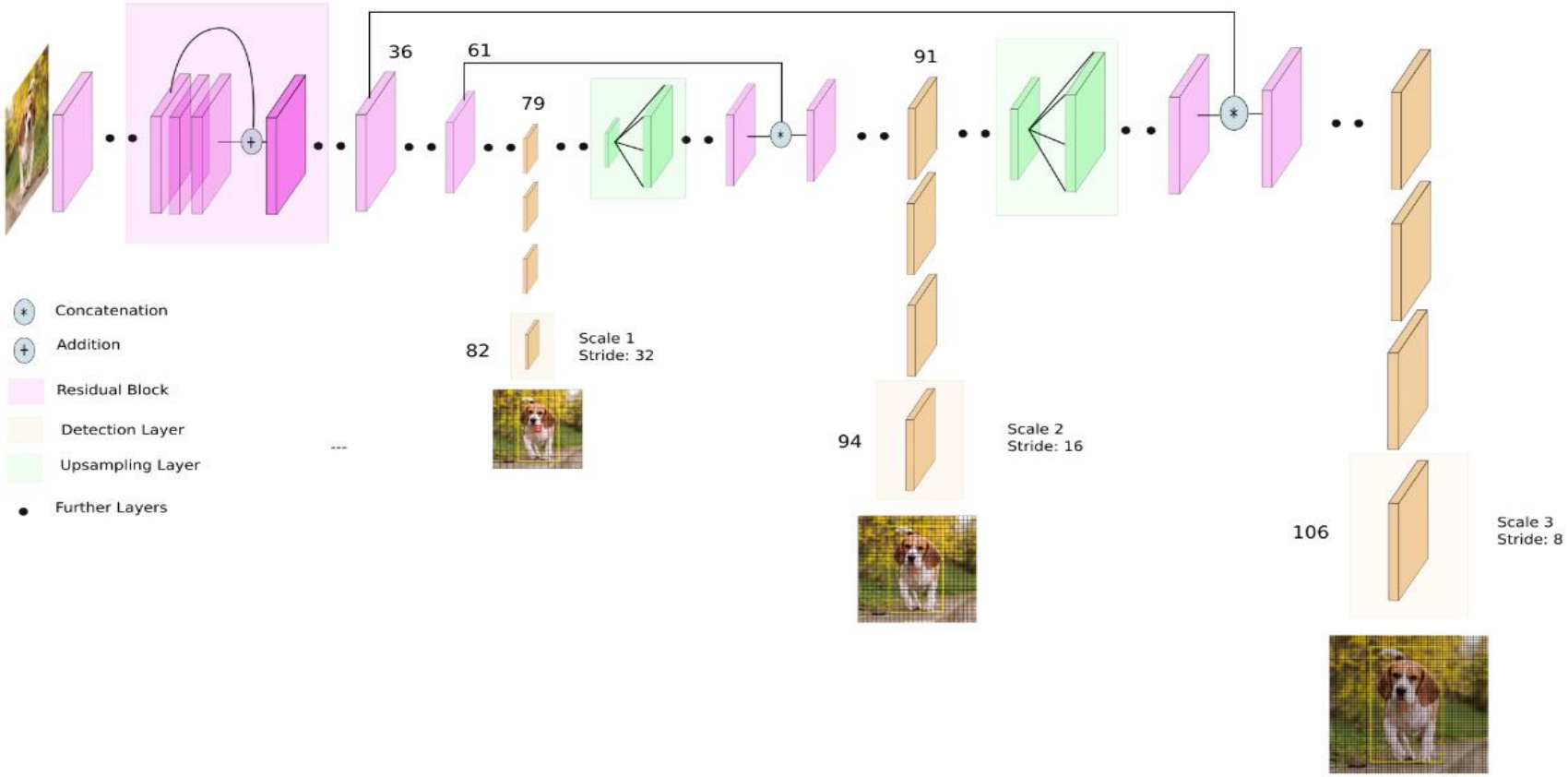


**Figura 22** Arquitetura YOLOv3

## 8.2 Anexo II

**Tabela 7** Estrutura e configuração completa do YOLOv4.

| | Camada | Filtros | Tamanho/Stride | Entrada | | Saída | Ativação | BFLOPS | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | conv | 32 | 3 x 3/ 1 | 416 x 416 x 3 | -> | 416 x 416 x 32 | mish | 0.299 | BF |
| 1 | conv | 64 | 3 x 3/ 2 | 416 x 416 x 32 | -> | 208 x 208 x 64 | mish | 1.595 | BF |
| 2 | conv | 64 | 1 x 1/ 1 | 208 x 208 x 64 | -> | 208 x 208 x 64 | mish | 0.354 | BF |
| 3 | route | 1 | | | -> | 208 x 208 x 64 | | | |
| 4 | conv | 64 | 1 x 1/ 1 | 208 x 208 x 64 | -> | 208 x 208 x 64 | mish | 0.354 | BF |
| 5 | conv | 32 | 1 x 1/ 1 | 208 x 208 x 64 | -> | 208 x 208 x 32 | mish | 0.177 | BF |
| 6 | conv | 64 | 3 x 3/ 1 | 208 x 208 x 32 | -> | 208 x 208 x 64 | mish | 1.595 | BF |
| 7 | Shortcut Layer: 4, wt = 0, wn = 0, outputs: 208 x 208 x 128 0.003 BF | | | | | | linear | | |
| 8 | conv | 64 | 1 x 1/ 1 | 208 x 208 x 64 | -> | 208 x 208 x 64 | mish | 0.354 | BF |
| 9 | route | 8 2 | | | -> | 208 x 208 x128 | | | |
| 10 | conv | 64 | 1 x 1/ 1 | 208 x 208 x 128 | -> | 208 x 208 x 64 | mish | 0.709 | BF |
| 11 | conv | 128 | 3 x 3/ 2 | 208 x 208 x 64 | -> | 104 x 104 x 128 | mish | 1.595 | BF |
| 12 | conv | 64 | 1 x 1/ 1 | 104 x 104 x 128 | -> | 104 x 104 x 64 | mish | 0.177 | BF |
| 13 | route | 11 | | | -> | 104 x 104 x 128 | | | |
| 14 | conv | 64 | 1 x 1/ 1 | 104 x 104 x 128 | -> | 104 x 104 x 64 | mish | 0.177 | BF |
| 15 | conv | 64 | 1 x 1/ 1 | 104 x 104 x 64 | -> | 104 x 104 x 64 | mish | 0.089 | BF |
| 16 | conv | 64 | 3 x 3/ 1 | 104 x 104 x 64 | -> | 104 x 104 x 64 | mish | 0.797 | BF |
| 17 | Shortcut Layer: 14, wt = 0, wn = 0, outputs: 104 x 104 x 128 000.1 BF | | | | | | linear | | |
| 18 | conv | 64 | 1 x 1/ 1 | 104 x 104 x 64 | -> | 104 x 104 x 64 | mish | 0.089 | BF |
| 19 | conv | 64 | 3 x 3/ 1 | 104 x 104 x 64 | -> | 104 x 104 x 64 | mish | 0.797 | BF |
| 20 | Shortcut Layer: 17, wt = 0, wn = 0, outputs: 104 x 104 x 64 000.1 BF | | | | | | linear | 0.001 | BF |
| 21 | conv | 64 | 1 x 1/ 1 | 104 x 104 x 64 | -> | 104 x 104 x 64 | mish | 0.089 | BF |
| 22 | route | 21 12 | | | -> | 104 x 104 x 128 | | | |
| 23 | conv | 128 | 1 x 1/ 1 | 104 x 104 x 128 | -> | 104 x 104 x 128 | mish | 0.354 | BF |
| 24 | conv | 256 | 3 x 3/ 2 | 104 x 104 x 128 | -> | 52 x 52 x 256 | mish | 1.595 | BF |
| 25 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 256 | -> | 52 x 52 x 128 | mish | 0.177 | BF |
| 26 | route | 24 | | | -> | 52 x 52 x 256 | | | |
| 27 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 256 | -> | 52 x 52 x 128 | mish | 0.177 | BF |
| 28 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.089 | BF |
| 29 | conv | 128 | 3 x 3/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.797 | BF |
| 30 | Shortcut Layer: 27, wt = 0, wn = 0, outputs: 52 x 52 x 128 000.0 BF | | | | | | linear | | |
| 31 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.089 | BF |
| 32 | conv | 128 | 3 x 3/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.797 | BF |
| 33 | Shortcut Layer: 30, wt = 0, wn = 0, outputs: 52 x 52 x 128 000.0 BF | | | | | | linear | | |
| 34 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.089 | BF |
| 35 | conv | 128 | 3 x 3/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.797 | BF |
| 36 | Shortcut Layer: 33, wt = 0, wn = 0, outputs: 52 x 52 x 128 000.0 BF | | | | | | linear | | |
| 37 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.089 | BF |
| 38 | conv | 128 | 3 x 3/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.797 | BF |
| 39 | Shortcut Layer: 36, wt = 0, wn = 0, outputs: 52 x 52 x 128 000.0 BF | | | | | | linear | | |

Continua...

| | Camada | Filtros | Tamanho/Stride | Entrada | | Saída | | BFLOPS | |
|---|---|---|---|---|---|---|---|---|---|
| 40 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.089 | BF |
| 41 | conv | 128 | 3 x 3/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.797 | BF |
| 42 | Shortcut Layer: 39, wt = 0, wn = 0, outputs: 52 x 52 x 128 000.0 BF | | | | | | linear | | |
| 43 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.089 | BF |
| 44 | conv | 128 | 3 x 3/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.797 | BF |
| 45 | Shortcut Layer: 42, wt = 0, wn = 0, outputs: 52 x 52 x 128 000.0 BF | | | | | | linear | | |
| 46 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.089 | BF |
| 47 | conv | 128 | 3 x 3/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.797 | BF |
| 48 | Shortcut Layer: 45, wt = 0, wn = 0, outputs: 52 x 52 x 128 000.0 BF | | | | | | linear | | |
| 49 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.089 | BF |
| 50 | conv | 128 | 3 x 3/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.797 | BF |
| 51 | Shortcut Layer: 48, wt = 0, wn = 0, outputs: 52 x 52 x 128 000.0 BF | | | | | | linear | | |
| 52 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 128 | mish | 0.089 | BF |
| 53 | route | 52 25 | | | -> | 52 x 52 x 256 | | | |
| 54 | conv | 256 | 1 x 1/ 1 | 52 x 52 x 256 | -> | 52 x 52 x 256 | mish | 0.354 | BF |
| 55 | conv | 512 | 3 x 3/ 2 | 52 x 52 x 256 | -> | 26 x 26 x 512 | mish | 1.595 | BF |
| 56 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 512 | -> | 26 x 26 x 256 | mish | 0.177 | BF |
| 57 | route | 55 | | | -> | 26 x 26 x 512 | | | |
| 58 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 512 | -> | 26 x 26 x 256 | mish | 0.177 | BF |
| 59 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.089 | BF |
| 60 | conv | 256 | 3 x 3/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.797 | BF |
| 61 | Shortcut Layer: 58, wt = 0, wn = 0, outputs: 26 x 26 x 256 000.0 BF | | | | | | linear | | |
| 62 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.089 | BF |
| 63 | conv | 256 | 3 x 3/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.797 | BF |
| 64 | Shortcut Layer: 61, wt = 0, wn = 0, outputs: 26 x 26 x 256 000.0 BF | | | | | | linear | | |
| 65 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.089 | BF |
| 66 | conv | 256 | 3 x 3/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.797 | BF |
| 67 | Shortcut Layer: 64, wt = 0, wn = 0, outputs: 26 x 26 x 256 000.0 BF | | | | | | linear | | |
| 68 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.089 | BF |
| 69 | conv | 256 | 3 x 3/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.797 | BF |
| 70 | Shortcut Layer: 67, wt = 0, wn = 0, outputs: 26 x 26 x 256 000.0 BF | | | | | | linear | | |
| 71 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.089 | BF |
| 72 | conv | 256 | 3 x 3/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.797 | BF |
| 73 | Shortcut Layer: 70, wt = 0, wn = 0, outputs: 26 x 26 x 256 000.0 BF | | | | | | linear | | |
| 74 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.089 | BF |
| 75 | conv | 256 | 3 x 3/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.797 | BF |
| 76 | Shortcut Layer: 73, wt = 0, wn = 0, outputs: 26 x 26x 256 000.0 BF | | | | | | linear | | |
| 77 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.089 | BF |
| 78 | conv | 256 | 3 x 3/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.797 | BF |
| 79 | Shortcut Layer: 76, wt = 0, wn = 0, outputs: 26 x 26 x 256 000.0 BF | | | | | | linear | | |
| 80 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.089 | BF |
| 81 | conv | 256 | 3 x 3/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.797 | BF |
| 82 | Shortcut Layer: 79, wt = 0, wn = 0, outputs: 52 x 52 x 128 000.0 BF | | | | | | linear | | |

Continua

| Camadas | | Filtros | Tamanho/ Stride | Entrada | | Saída | | BFLOPS | |
|---|---|---|---|---|---|---|---|---|---|
| 83 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 256 | mish | 0.089 | BF |
| 84 | route | 83 56 | | | -> | 26 x 26 x 512 | | | |
| 85 | conv | 512 | 1 x 1/ 1 | 26 x 26 x 512 | -> | 26 x 26 x 512 | mish | 0.354 | BF |
| 86 | conv | 1024 | 3 x 3/ 2 | 26 x 26 x 512 | -> | 13 x 13 x 1024 | mish | 1.595 | BF |
| 87 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 1024 | -> | 13 x 13 x 512 | mish | 0.177 | BF |
| 88 | route | 86 | | | -> | 13 x13 x 1024 | | | |
| 89 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 1024 | -> | 13 x 13 x 512 | mish | 0.177 | BF |
| 90 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 512 | mish | 0.089 | BF |
| 91 | conv | 512 | 3 x 3/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 512 | mish | 0.797 | BF |
| 92 | Shortcut Layer: 89, wt = 0, wn = 0, outputs: 13 x 13 x 512 000.0 BF | | | | | | linear | | |
| 93 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 512 | mish | 0.089 | BF |
| 94 | conv | 512 | 3 x 3/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 512 | mish | 0.797 | BF |
| 95 | Shortcut Layer: 92, wt = 0, wn = 0, outputs: 13 x 13 x 512 000.0 BF | | | | | | linear | | |
| 96 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 512 | mish | 0.089 | BF |
| 97 | conv | 512 | 3 x 3/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 512 | mish | 0.797 | BF |
| 98 | Shortcut Layer: 95, wt = 0, wn = 0, outputs: 13 x 13 x 512 000.0 BF | | | | | | linear | | |
| 99 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 512 | mish | 0.089 | BF |
| 100 | conv | 512 | 3 x 3/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 512 | mish | 0.797 | BF |
| 101 | Shortcut Layer: 98, wt = 0, wn = 0, outputs: 13 x 13 x 512 000.0 BF | | | | | | linear | | |
| 102 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 512 | mish | 0.089 | BF |
| 103 | route | 102 87 | | | -> | 13 x 13 x 1024 | | | |
| 104 | conv | 1024 | 1 x 1/ 1 | 13 x 13 x 1024 | -> | 13 x 13 x 1024 | mish | 0.354 | BF |
| 105 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 1024 | -> | 13 x 13 x 512 | leaky | 0.177 | BF |
| 106 | conv | 1024 | 3 x 3/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 1024 | leaky | 1.595 | BF |
| 107 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 1024 | -> | 13 x 13 x 512 | leaky | 0.177 | BF |
| 108 | max | | 5 x 5/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 512 | | 0.002 | BF |
| 109 | route | 107 | | | -> | 13 x 13 x512 | | | |
| 110 | max | | 9 x 9/1 | 13 x 13 x 512 | -> | 13 x 13 x 512 | | 0.007 | BF |
| 111 | route | 107 | | | -> | 13 x 13 x 512 | | | |
| 112 | max | | 13 x 13/1 | 13 x 13 x512 | -> | 13 x13 x 512 | | | |
| 113 | route | 112 110 108 107 | | | -> | 13 x 13 x 2048 | | | |
| 114 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 2048 | -> | 13 x 13 x 512 | leaky | 0.354 | BF |
| 115 | conv | 1024 | 3 x 3/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 1024 | leaky | 1.595 | BF |
| 116 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 1024 | -> | 13 x 13 x 512 | leaky | 0.177 | BF |
| 117 | conv | 256 | 1 x 1/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 256 | leaky | 0.044 | BF |
| 118 | upsample | | 2 x | 13 x 13 x 256 | -> | 26 x 26 x 256 | | | |
| 119 | route | 85 | | | -> | 26 X 26 X 5256 | | | |
| 120 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 512 | -> | 26 x 26 x 256 | leaky | 0.177 | BF |
| 121 | route | 120 118 | | | -> | 26 X 26 X 512 | | | |
| 122 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 512 | -> | 26 x 26 x 256 | leaky | 0.177 | BF |
| 123 | conv | 512 | 3 x 3/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 512 | leaky | 1.595 | BF |
| 124 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 512 | -> | 26 x 26 x 256 | leaky | 0.177 | BF |
| 125 | conv | 512 | 3 x 3/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 512 | leaky | 1.595 | BF |

Continua...

| | Camadas | Filtros | Tamanho/Stride | Entrada | | Saída | BFLOPS | | |
|---|---|---|---|---|---|---|---|---|---|
| 126 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 512 | -> | 26 x 26 x 256 | leaky | 0.177 | BF |
| 127 | conv | 128 | 1 x 1/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 128 | leaky | 0.044 | BF |
| 128 | upsample | | 2 x | 26 x 26 x 128 | -> | 52 x52 x128 | | | |
| 129 | route | 54 | | | -> | 52 x 52 x 256 | | | |
| 130 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 256 | -> | 52 x 52 x 128 | leaky | 0.177 | BF |
| 131 | route | 130 128 | | | -> | 52 x 52 x 256 | | | |
| 132 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 256 | -> | 52 x 52 x 128 | leaky | 0.177 | BF |
| 133 | conv | 256 | 3 x 3/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 256 | | 1.595 | BF |
| 134 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 256 | -> | 52 x 52 x 128 | leaky | 0.177 | BF |
| 135 | conv | 256 | 3 x 3/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 256 | leaky | 1.595 | BF |
| 136 | conv | 128 | 1 x 1/ 1 | 52 x 52 x 256 | -> | 52 x 52 x 128 | leaky | 0.177 | BF |
| 137 | conv | 256 | 3 x 3/ 1 | 52 x 52 x 128 | -> | 52 x 52 x 256 | leaky | 1.595 | BF |
| 138 | conv | 30 | 1 x 1/ 1 | 52 x 52 x 256 | -> | 52 x 52 x 30 | linear | 0.042 | BF |
| 139 | yolo | | | | | | | | |
| [yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.20 | | | | | | | | | |
| 140 | route | 136 | | | -> | 52 x 52 x 128 | | | |
| 141 | conv | 256 | 3 x 3/ 2 | 52 x 52 x 128 | -> | 26 x 26 x 256 | leaky | 0.399 | BF |
| 142 | route | 141 126 | | | | 26 x 26 x 512 | | | |
| 143 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 512 | -> | 26 x 26 x 256 | leaky | 0.177 | BF |
| 144 | conv | 512 | 3 x 3/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 512 | leaky | 1.595 | BF |
| 145 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 512 | -> | 26 x 26 x 256 | leaky | 0.177 | BF |
| 146 | conv | 512 | 3 x 3/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 512 | leaky | 1.595 | BF |
| 147 | conv | 256 | 1 x 1/ 1 | 26 x 26 x 512 | -> | 26 x 26 x 256 | leaky | 0.177 | BF |
| 148 | conv | 512 | 3 x 3/ 1 | 26 x 26 x 256 | -> | 26 x 26 x 512 | leaky | 1.595 | BF |
| 149 | conv | 30 | 1 x 1/ 1 | 26 x 26 x 512 | -> | 26 x 26 x 30 | linear | 0.021 | BF |
| 150 | yolo | | | | | | | | |
| [yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.1 | | | | | | | | | |
| 151 | route | 147 | | | | 26 x 26 x 256 | | | |
| 152 | conv | 512 | 3 x 3/ 2 | 26 x 26 x 256 | -> | 13 x 13 x 512 | leaky | 0.399 | BF |
| 153 | route | 152 116 | | | -> | 13 x 13 x 1024 | | | |
| 154 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 1024 | -> | 13 x 13 x 512 | leaky | 0.177 | BF |
| 155 | conv | 1024 | 3 x 3/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 1024 | leaky | 1.595 | BF |
| 156 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 1024 | -> | 13 x 13 x 512 | leaky | 0.177 | BF |
| 157 | conv | 1024 | 3 x 3/ 1 | 13 x 13 x 512 | -> | 13 x 13 512 | leaky | 1.595 | BF |
| 158 | conv | 512 | 1 x 1/ 1 | 13 x 13 x 1024 | -> | 13 x 13 x 512 | leaky | 0.177 | BF |
| 159 | conv | 1024 | 3 x 3/ 1 | 13 x 13 x 512 | -> | 13 x 13 x 1024 | leaky | 1.595 | BF |
| 160 | conv | 30 | 1 x 1/ 1 | 13 x 13 x 1024 | -> | 13 x 13 x30 | linear | 0.01 | BF |
| 161 | yolo | | | | | | | | |
| [yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.05 | | | | | | | | | |

BFLOPS (BF) = Biliões de Operações de Ponto Flutuante por Segundo; *max* = *Max Polling*, conv = camada convolucional; route = recursos de camadas inicias; *upsample* = aumento da amostra por interpolação bilinear; *ShortcutLayer* = camada atalho, é uma conexão de salto.