

Aplicando um modelo YOLO para detectar e diferenciar por imagem castas de abelhas melíferas de forma automatizada

Davi Queiroz Albuquerque¹, Antonio Rafael Braga²,
Isac Gabriel Abrahão Bomfim³, Danielo G. Gomes¹

¹Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)
Departamento de Engenharia de Teleinformática
Centro de Tecnologia, Universidade Federal do Ceará (UFC), Fortaleza - CE

²Sistemas de Informação – Campus Quixadá,
Universidade Federal do Ceará (UFC), Quixadá-CE

³Laboratório de Apicultura, Campus Crateús,
Instituto Federal do Ceará (IFCE), Crateús-CE

daviqueiroz2002@alu.ufc.br

[rafaelbraga, danielo]@ufc.br, isac.bomfim@ifce.edu.br

Abstract. *In a honeybee colony (*Apis mellifera* L.) there are three types of caste: queen, worker and drone. Detecting and differentiating them is of paramount importance for the beekeeper, as unusual fluctuation and imbalance in the number and natural proportion among these individuals provide predictions about events that can negatively impact the welfare and production of the colony. In this article, we apply the concept of Digital Image Processing, through the YOLO object detector, to differentiate drones and workers of honeybees in order to provide subsidies for the advancement of precision beekeeping. Thorough cross-validation, the chosen architecture correctly recognized and classified most of the bees present in the images, obtaining values of $mAP@50$ above 94%. Furthermore, even with an unbalanced dataset and most of the bees being worker bees, the proposed model was able to find and classify most drones, reflecting recall values above 85%.*

Resumo. *Em uma colônia de abelhas melíferas (*Apis mellifera* L.) há três tipos de casta: rainha, operária e zangão. Detectá-las e diferenciá-las é de suma importância para o apicultor, pois a flutuação e o desbalanço fora do normal no número e na proporção natural entre indivíduos fornecem previsões sobre eventos que podem impactar negativamente o bem-estar e a produção da colônia. Neste artigo, aplicamos o conceito de Processamento Digital de Imagens, através do detector de objetos YOLO, para diferenciar entre operárias e zangões de abelhas melíferas a fim de fornecer subsídios para o avanço da apicultura de precisão. Por meio de validação cruzada, a arquitetura escolhida reconheceu e classificou corretamente a maioria das abelhas presentes nas imagens, obtendo valores de $mAP@50$ acima de 94%. Além disso, mesmo com um conjunto de dados desbalanceados e a maior parte das abelhas sendo de operárias, o modelo proposto conseguiu encontrar e classificar a maioria dos zangões, refletindo nos valores de recall acima de 85%.*

1. Introdução

Diante das mais variadas notícias e pesquisas que relatam as ameaças que as abelhas vêm sofrendo, e como isso pode impactar na vida da humanidade [Watanabe 2013, Potts et al. 2016, Soroye et al. 2020], a comunidade científica busca maneiras de contribuir para a sobrevivência e melhoria do bem-estar e da produção desses valiosos insetos, inclusive aplicando conceitos e tecnologias computacionais, como os já adotados na Apicultura de Precisão [Hadjur et al. 2022, Braga et al. 2020, Debauche et al. 2018].

A abelha melífera (*Apis mellifera* L.) é a espécie de abelha mais criada pelo homem, e suas colônias organizam-se socialmente em três castas: operária, zangão e rainha, as quais diferem em formato, tamanho e função. Cada uma destas castas tem sua importância específica na manutenção de uma colônia. As duas castas que podem ser observadas com mais frequência na entrada (alvado) ou fora da colmeia são as operárias e os zangões. As operárias realizam as mais diversas tarefas ao longo de sua vida, exercendo o papel de faxineiras, nutrizes, engenheiras entre outras atividades internas do ninho até alcançar as funções de guardas e campeiras, as quais já podem ser vistas entrando e saindo da colmeia [Kevan et al. 2007]. Já os zangões são os machos da colônia, cuja contribuição com as tarefas no ninho é inexistente; na verdade, eles demandam muito mais alimento do que as operárias e sua função na vida se limita a realizar um único acasalamento com uma rainha virgem para, assim, transmitir e perpetuar parte dos genes da colônia, e em seguida morrer [Winston 2003, Seeley 2006].

O monitoramento do fluxo de entrada e saída dessas castas pode fornecer informações valiosas ao apicultor, dando estimativas do desenvolvimento e saúde da colônia, bem como predições sobre eventos que poderiam impactar negativamente a produtividade da colônia, e, conseqüentemente, economicamente o apicultor. No tocante aos zangões, por exemplo, um grande aumento em sua população durante o auge do período produtivo pode sinalizar que a colônia está superpopulosa e próxima a enxamear (reprodução natural de uma colônia que pelo desenvolvimento de uma nova rainha e divisão da população forma mais uma outra) [Couto and Couto 2006]. Outro sinal de alerta com esse excesso de zangões seria a perda da rainha, já que uma colônia sem sua rainha pode se tornar “zanganeira”, *i.e.* colônias que apresentam operárias que assumem o papel da rainha, mas por incapacidade fisiológica só conseguem pôr ovos que darão origem a zangões que, por consumirem muitos recursos e não contribuirão com as tarefas do ninho, levam a colônia ao enfraquecimento e morte, a não ser que haja uma urgente intervenção por parte do apicultor [Wiese 2005].

Por outro lado, técnicas de aprendizagem de máquina (e.g. redes neurais convolucionais (*Convolutional Neural Networks* - CNNs) e *deep learning*) em visão computacional e no Processamento Digital de Imagens (PDI) têm sido aplicadas com êxito para automatizar a identificação e contagem de insetos e proporcionando uma grande economia de tempo em comparação à execução manual [De Cesaro Júnior and Rieder 2020]. No caso particular das abelhas, o uso de CNNs e *deep learning* tem se mostrado eficaz no reconhecimento tanto de abelhas nativas [Leocádio et al. 2021] quanto de abelhas melíferas [Barros et al. 2021, Bilik et al. 2021].

A partir deste contexto, neste artigo aplicamos uma rede neural profunda do tipo YOLO (vide Seção 2.2) para detectar e diferenciar de forma automatizada as operárias e os zangões de abelhas melíferas no alvado das colmeias.

2. Material e Método

2.1. Base de imagens

Para os experimentos, todas as imagens utilizadas são da plataforma *BeeStream*¹ [Tashakkori et al. 2021] da *Appalachian State University* (Appstate), a qual disponibiliza gravações públicas da entrada (alvado) de colmeias localizadas no campus da Appstate e permite escolher a câmera, a data e o horário da filmagem, recursos que levaram a plataforma a ser o alicerce para construção da base de imagens. A fim de maximizar as amostras de zangões, que apresentam-se naturalmente em menor quantidade [Seeley 2006], 07 dos vídeos escolhidos, responsáveis por 342 imagens distintas, os quais foram gerados entre os dias 4 e 7 de abril de 2021 (início da primavera no Hemisfério Norte) e no período da tarde, entre 13h e 15h, intervalo tido na literatura como o de maior fluxo de entrada e saída de zangões durante seus voos de orientação (voos curtos que precedem o de acasalamento) [Winston 2003] além de outras 22 fotos sem identificação da data de obtenção. Assim, a base de imagens foi composta por 364 cenas de resolução 640×480 , contendo uma ou mais abelhas em diferentes poses e condições ambientes (e.g. luminosidade e localização no alvado, o que afeta a visibilidade dos indivíduos).

No contexto de detecção de objetos com YOLO, a arquitetura escolhida trabalha fundamentalmente com caixas delimitadoras (*bounding boxes*), as quais descrevem onde os alvos do reconhecimento estão nas imagens, bem como a classe as quais pertencem. Usamos o software *LabelImg*² cuja interface gráfica é intuitiva, além de exportar as anotações no formato aceito pela YOLO, descrevendo as seguintes características da caixa: classe, coordenadas (x, y) do seu centro, largura e altura, seguindo o sistema de orientação da Figura 1. Ao fim dessa etapa, 3463 abelhas foram rotuladas, sendo 2826 operárias e 637 zangões. Nas imagens da Figura 2 estão presentes algumas amostras de imagens presentes no *dataset*, juntamente com as *bounding boxes* (em vermelho, se a abelha é uma operária, ou azul, se é um zangão) definidas pelos autores no processo de rotulagem, ressaltando que essas marcações das caixas, tal como mostradas, não fazem parte das imagens originais usadas para o treinamento e teste do modelo e são meramente ilustrativas com relação à posição e à casta dos indivíduos de interesse nas cenas.

Por fim, como a técnica de validação cruzada foi utilizada, a base de imagens foi dividida aleatoriamente em 5 subconjuntos mutuamente exclusivos, de forma que a cada rodada de estimação dos parâmetros a razão entre as quantidades de treinamento e teste fosse de, aproximadamente, 80/20, apresentando na Tabela 1 a seguir como as classes estão distribuídas em cada uma dessas partes.

2.2. YOLOv5

A rede neural profunda YOLO, cujo nome é a sigla em inglês de *You Only Look Once* (Você Só Olha Uma Vez, em tradução direta), é um detector de objetos que apresenta como principal diferencial a combinação entre desempenho e acurácia, proporcionando portanto uma detecção rápida e eficiente [Redmon et al. 2016]. A rede recebe como entrada a imagem na qual estão os objetos de interesse e retorna como saída as detecções seguindo o sistema de anotações descritos anteriormente, bem como o nível de confiança das mesmas.

¹<https://cs.appstate.edu/beemon/beestream/>

²<https://github.com/tzutalin/labelImg>



Figura 1. Sistema de coordenadas da imagem utilizado pela arquitetura YOLO com os sentidos de crescimento dos eixos x e y. Fonte: *BeeStream* e [Jocher et al. 2020] (adaptada).



(a) Original.

(b) Com as *bounding boxes*.

Figura 2. (a) Amostra da imagem original, presente na base e utilizada para treinamento e validação, e (b) da mesma com representação visual das *bounding boxes* presentes nos arquivos de anotações. Fonte: *BeeStream*.

Tabela 1. Distribuição das classes em cada subconjunto da base de imagens.

Subconjunto	Imagens	Operárias	Zangões
1	73	574	132
2	73	547	120
3	73	589	129
4	73	561	126
5	72	555	130

Neste artigo utilizamos a implementação de código aberto YOLOv5 *by Ultralytics* [Jocher et al. 2020], constituída por 5 modelos que diferem no tamanho da rede neural e

que possuem integração nativa com o framework *PyTorch*³. Para os experimentos, foi utilizada a variação *Small* (YOLOv5s) ao invés da *Nano* (YOLOv5n), a mais rápida, pois ambas apresentam tempos de inferência próximos mas com uma diferença significativa a favor da primeira nas métricas [Jocher et al. 2020], sendo portanto ideal para um sistema eficaz de monitoramento em tempo real.

A arquitetura do modelo escolhido foi estudada por [Yan et al. 2021], dada a ênfase nas 3 subdivisões principais da rede: o (i) *Backbone*, responsável por extrair características (*features*) da imagem, (ii) a *Neck*, a qual trata essas *features* para a etapa de detecção, agregando e combinando as mesmas e (iii) a *Detect* (também chamada de *Head*), onde a detecção é feita de fato, que possui como saída um vetor contendo os dados das detecções feitas (descrições das caixas delimitadoras) e, com isso, a imagem com a localização dos objetos de interesse é formada. Na Figura 3 está representado o fluxo de dados entre essas 3 subdivisões, desde a entrada da imagem até a saída com as detecções.

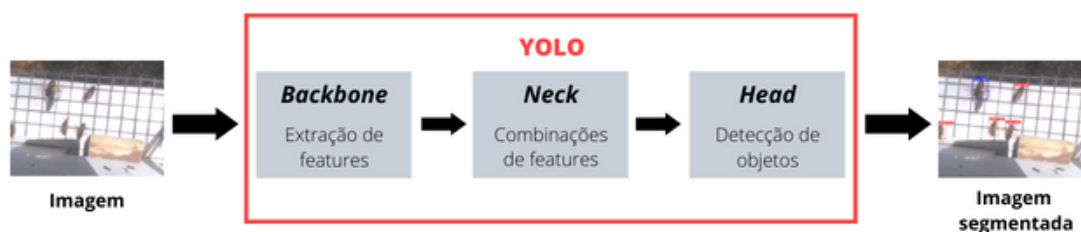


Figura 3. Fluxo de dados em um detector de objetos, destacando o papel de cada subdivisão do modelo. Fonte: [Kubera et al. 2022] (adaptada).

2.3. Treinamento

A etapa de treinamento foi realizada na plataforma online *Google Colab*⁴, a qual disponibiliza uma máquina virtual com placa de vídeo dedicada (GPU), que nesse caso foi uma NVIDIA Tesla K80 12GB. O ambiente de execução ainda é composto por um processador Intel Xeon 2.20GHz 64 bits e 12GB de memória RAM rodando no sistema operacional Ubuntu 18.04 LTS.

Em cada subdivisão, o conjunto de treino teve aproximadamente 80% das imagens da base completa e o restante para validação dos pesos obtidos. Como a distribuição das classes é desbalanceada, em todas as 5 rodadas foram feitas 100 épocas de treinamento com iterações formadas por lotes de 16 imagens, mantendo os outros parâmetros como os padrões do detector YOLO. Alguns hiperparâmetros significantes foram mantidos no estado padrão disponível no repositório, como (i) o tamanho da entrada do modelo em 3 canais (RGB) de dimensões 640x640, (ii) o otimizador *SGD*, (iii) o limiar de *Non-maximal supression* (NMS), responsável por evitar detecções múltiplas para um mesmo objeto [Redmon et al. 2016], com valor de 0,45 e (iv) o limiar de confiança da detecção em 0,25.

2.4. Avaliação do modelo

As métricas de avaliação da arquitetura passam pelo cálculo da Interseção sobre a União (*IoU*, do inglês *Intersection over Union*), a qual indica o quão bem as caixas delimita-

³<https://pytorch.org>

⁴<https://colab.research.google.com>

doras preditas encaixam-se nas reais indicadas no processo de rotulagem, cujo cálculo é demonstrado intuitivamente na Figura 4. Ainda nesse contexto, define-se um limiar L para definir se aquela inferência é tratada, dentro da classe a qual pertence, como Verdadeiro Positivo (VP), se $IoU \geq L$, Falso Positivo (FP), se $IoU < L$, ou Falso Negativo (FN), quando não há predição naquela área. Com isso, o *recall* (Equação 1), a precisão (Equação 2) e a *Average Precision* (AP), que é a área sob a curva *Precisão x Recall* [Hui 2018], de cada classe são calculados, bem como a *mean Average Precision* (mAP), a média das APs, para aquele limiar L .

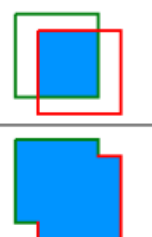
$$IoU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{área de sobreposição}}{\text{área da união}}$$


Figura 4. Como é realizado o cálculo da Interseção sobre a União (IoU) dada a caixa real e a predita. Fonte: [Padilla et al. 2020].

$$Recall = \frac{VP}{VP + FN} \quad (1)$$

$$Precisão = \frac{VP}{VP + FP} \quad (2)$$

Nesse trabalho, dois limiares foram utilizados: $L = 0,50$ e $L = 0,75$, os quais são mais utilizados [Padilla et al. 2020] para analisar um detector de objetos no conjunto de imagens *COCO* [Lin et al. 2014], referência no âmbito de visão computacional, visando entender se não ocorre grande perda do poder de detecção com a mudança do patamar de confiança. Além disso, como a distribuição das classes é desbalanceada, outra importante métrica para avaliar os resultados do treinamento é o *recall* da casta zangão, visto que isso indicará se o algoritmo conseguiu reconhecer bem os indivíduos da classe minoritária. Vale ressaltar que os valores de precisão e *recall* a serem listados estão relacionados a $L = 0,50$, valor padrão para o cálculo no repositório público da arquitetura YOLOv5.

3. Resultados e discussão

Os resultados obtidos para cada subconjunto de índice v , com $v \in \{1, 2, 3, 4, 5\}$, utilizado para validação e o restante das imagens para treinamento estão expostos na Tabela 2, destacando os maiores valores obtidos para as métricas de avaliação listadas anteriormente na Seção de Metodologia e utilizando a notação $AP@x$ para indicar o cálculo da AP com $L = 0.01 \cdot x$.

Inspecionando a Tabela 2, é possível inferir o bom desempenho da arquitetura escolhida para a problemática levantada ao longo desse trabalho. Em todas as subdivisões do treinamento, os valores de $mAP@50$ ficaram acima de 0,94 (94%) e, mesmo aumentando o limiar de IoU para calcular o $mAP@75$, este apresentou-se maior que 76%, com máximo de 82%, revelando que o modelo conseguiu de maneira eficiente encontrar as

Tabela 2. Métricas obtidas ao final das 100 épocas de treinamento.

	Precisão	Recall	AP@50	AP@75
v = 1				
Operária	0.91	0.92	0.96	0.74
Zangão	0.89	0.88	0.92	0.77
Geral	0.90	0.90	0.94	0.76
v = 2				
Operária	0.92	0.93	0.97	0.77
Zangão	0.86	0.89	0.93	0.78
Geral	0.89	0.91	0.95	0.77
v = 3				
Operária	0.93	0.91	0.95	0.75
Zangão	0.94	0.85	0.96	0.79
Geral	0.94	0.88	0.96	0.77
v = 4				
Operária	0.94	0.94	0.97	0.79
Zangão	0.97	0.87	0.97	0.86
Geral	0.96	0.90	0.97	0.82
v = 5				
Operária	0.93	0.92	0.95	0.73
Zangão	0.93	0.87	0.94	0.80
Geral	0.93	0.90	0.95	0.76

abelhas nas imagens e classificá-las corretamente em cada casta independente de quais ramificações foram utilizadas para treinamento.

Além disso, apesar dos zangões apresentarem-se como a classe minoritária, o *recall* dessa casta obteve valores acima de 0.85 (85%), alcançando até mesmo 89% quando $v = 2$. Em suma, o modelo treinado conseguiu detectar efetivamente os zangões nas imagens, contando ainda com uma precisão elevada (superando 86%), indicando que a maioria das abelhas classificadas nessa casta são realmente zangões. Vale salientar que esse número de zangões bem menor do que o de operárias é natural em uma colônia. O investimento de uma colônia para produção de zangões é alto, pois este não contribui nas atividades internas do ninho. Isso faz com que essa produção seja nula ou mínima (normalmente de zero a um pouco mais de 400 machos adultos em um dado momento) [Couto and Couto 2006] e sazonal, elevando-se, naturalmente, apenas no período em ir ocorrer a enxameao da colnia.

A Figura 5 apresenta algumas deteces feitas com os pesos extraidos na etapa em que $v = 4$, modelo esse que apresentou melhores resultados para $mAP@50$ e $mAP@75$ em conjunto com um *recall* da casta zango prximo ao valor mximo para essa mtrica, configurando portanto os melhores parmetros obtidos para a rede neural.



Figura 5. Detecções utilizando o modelo obtido para $v = 4$. Para cada detecção, tem-se a *bounding box* (em vermelho as operárias e em azul os zangões), a casta (operária ou zangão) e a confiança da detecção.

4. Conclusão

A principal contribuição e originalidade deste artigo é um método para detectar e diferenciar, por imagem, e de forma automatizada e precisa as operárias e os zangões de abelhas melíferas. Nessa perspectiva, entende-se que a solução apresentada é inédita e expressiva, uma vez que foi introduzida uma abordagem baseada em aprendizado profundo não somente para reconhecer abelhas em imagens, mas também classificá-las de acordo com a casta a qual pertencem. O uso do detector de objetos YOLOv5, conhecido pela sua alta velocidade de inferência, revelou-se uma boa estratégia e base para um sistema de distinção mais amplo de castas em tempo real e um importante avanço para a apicultura de precisão.

Na qualidade de possíveis trabalhos futuros, sugerimos o balanceamento da base de imagens através da técnica de *oversampling* da classe minoritária e a adição da classe das rainhas, visando desenvolver um classificador completo das castas de abelhas melíferas. Uma outra possibilidade é estender este trabalho para outras espécies de abelhas sociais de interesse zootécnico.

Agradecimentos

O presente artigo foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Danielo G. Gomes agradece o suporte financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) através dos processos 432585/2016-8 e 310317/2019-3.

Referências

Barros, C., Freitas, E. D., Braga, A. R., Bomfim, I. G., and Gomes, D. (2021). Aplicando redes neurais convolucionais em imagens para reconhecimento automatizado de abelhas melíferas (*Apis mellifera* L.). In *Anais do XII Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais*, pages 19–28, Porto Alegre, RS, Brasil. SBC. <https://doi.org/10.5753/wcama.2021.15733>.

- Bilik, S., Kratochvila, L., Ligocki, A., Bostik, O., Zemcik, T., Hybl, M., Horak, K., and Zalud, L. (2021). Visual diagnosis of the varroa destructor parasitic mite in honeybees using object detector techniques. *Sensors*, 21(8). <https://doi.org/10.3390/s21082764>.
- Braga, A. R., G. Gomes, D., Rogers, R., E. Hassler, E., M. Freitas, B., and A. Cazier, J. (2020). A method for mining combined data from in-hive sensors, weather and apiary inspections to forecast the health status of honey bee colonies. *Computers and Electronics in Agriculture*, 169:105161. <https://doi.org/10.1016/j.compag.2019.105161>.
- Couto, R. H. N. and Couto, L. A. (2006). *Apicultura: manejo e produtos*. Funep Jaboticabal.
- De Cesaro Júnior, T. and Rieder, R. (2020). Automatic identification of insects from digital images: A survey. *Computers and Electronics in Agriculture*, 178:105784. <https://doi.org/10.1016/j.compag.2020.105784>.
- Debauche, O., Moulat, M. E., Mahmoudi, S., Boukraa, S., Manneback, P., and Lebeau, F. (2018). Web monitoring of bee health for researchers and beekeepers based on the internet of things. *Procedia Computer Science*, 130:991–998. The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops. <https://doi.org/10.1016/j.procs.2018.04.103>.
- Hadjur, H., Ammar, D., and Lefèvre, L. (2022). Toward an intelligent and efficient beehive: A survey of precision beekeeping systems and services. *Computers and Electronics in Agriculture*, 192:106604. <https://doi.org/10.1016/j.compag.2021.106604>.
- Hui, J. (2018). map (mean average precision) for object detection. Disponível em: <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>. Acesso em: 03 mar. 2022.
- Jocher, G., Nishimura, K., Mineeva, T., and Vilariño, R. (2020). Yolov5 by ultralytics. Disponível em: <https://github.com/ultralytics/yolov5>.
- Kevan, P. G. et al. (2007). *Bees, biology & management*. Enviroquest.
- Kubera, E., Kubik-Komar, A., Kurasiński, P., Piotrowska-Weryszko, K., and Skrzypiec, M. (2022). Detection and recognition of pollen grains in multilabel microscopic images. *Sensors*, 22(7). <https://doi.org/10.3390/s22072690>.
- Leocádio, R., Segundo, A., Souza, J., Galaschi-Teixeira, J., Souza, P., and Pessin, G. (2021). Detecção de abelhas nativas em colmeias em campo utilizando visão computacional. In *Anais do XII Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais*, pages 59–68, Porto Alegre, RS, Brasil. SBC.
- Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312. <https://doi.org/10.48550/arXiv.1405.0312>.
- Padilla, R., Netto, S. L., and da Silva, E. A. B. (2020). A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242. <https://doi.org/10.1109/IWSSIP48289.2020.9145130>.

- Potts, S. G., Imperatriz Fonseca, V., Ngo, H. T., Biesmeijer, J. C., Breeze, T. D., Dicks, L., Garibaldi, L. A., Hill, R., Settele, J., Vanbergen, A. J., et al. (2016). *Summary for policymakers of the assessment report of the Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services on pollinators, pollination and food production*. Intergovernmental SciencePolicy Platform on Biodiversity and Ecosystem Services.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.
- Seeley, T. (2006). *Ecologia da Abelha: um estudo de adaptação na vida social (tradução de CA Osowski)*. LTDA, Porto Alegre.
- Soroye, P., Newbold, T., and Kerr, J. (2020). Climate change contributes to widespread declines among bumble bees across continents. *Science*, 367(6478):685–688. <https://doi.org/10.1126/science.aax8591>.
- Tashakkori, R., Hamza, A. S., and Crawford, M. B. (2021). Beemon: An iot-based beehive monitoring system. *Computers and Electronics in Agriculture*, 190:106427. <https://doi.org/10.1016/j.compag.2021.106427>.
- Watanabe, M. E. (2013). Pollinators at Risk: Human activities threaten key species. *BioScience*, 64(1):5–10. <https://doi.org/10.1093/biosci/bit012>.
- Wiese, H. (2005). *Apicultura: novos tempos*. Agrolivros Porto Alegre.
- Winston, M. (2003). *A biologia da abelha. Tradução de Carlos A. Osowski*. Magister. Porto Alegre.
- Yan, B., Fan, P., Lei, X., Liu, Z., and Yang, F. (2021). A real-time apple targets detection method for picking robot based on improved yolov5. *Remote Sensing*, 13(9). <https://doi.org/10.3390/rs13091619>.