

# Classificação de Imagens de Sementes com *Deep Learning* implementada na plataforma Android

Joel Machado Pires<sup>1</sup>, João Soares Oliveira Neto<sup>1</sup>, Edson Ferreira Duarte<sup>2</sup>,  
Camila Bezerra da Silva<sup>1</sup>, Bárbara Talita Ribeiro Nunes da Silva<sup>1</sup>

<sup>1</sup> Centro de Ciências Exatas e Tecnológicas (CETEC)  
Universidade Federal do Recôncavo da Bahia (UFRB)  
R. Rui Barbosa, Cruz das Almas - BA, 44380-000 - Brasil

<sup>2</sup> Departamento de Botânica, Goiânia, Goiás - Universidade Federal de Goiás

joelpires70@gmail.com, jnetoufrb@gmail.com, efd@ufg.br,  
camilabezerra@ufrb.edu.br, bn8429287@gmail.com

**Abstract.** *In view of the importance of seeds and the growing technological advance, it became possible to implement solutions in botany in the context of pervasive computing. This work applies Deep Learning to seed image classification. It presents some fundamentals of Neural Networks and the tools used to implement an image classification model. The highest accuracy of this model was 94.57%.*

**Resumo.** *Diante da importância de sementes e o crescente avanço tecnológico, tornou-se possível implementações de soluções em botânica no contexto de computação pervasiva. Este trabalho aplica Deep Learning em classificação de imagens de sementes. Nele é apresentado alguns fundamentos de Redes Neurais e as ferramentas usadas para a implementação de um modelo de classificação de imagens. A maior acurácia desse modelo foi de 94,57%.*

## 1. Introdução

O estudo de sementes tem importância tanto ambiental, científica, e mercadológica seja pela busca de soluções de preservação tanto da fauna quanto flora [Carvalho 2006], seja para controle de qualidade [Gazzolla-Neto 2012]. Além disso, a classificação pode ser necessária para catálogo, estudos, separação ou até mesmo contagem [Rauber 2019].

Por outro lado, diante da evolução computacional, em particular na Inteligência Artificial, pode-se destacar a subárea *Deep Learning* para processamento de imagens que está evoluindo de maneira muito rápida [Qiu et al 2018]. Junto com essa evolução, cresce o leque de utilidades, tais como avaliação de qualidade de recursos educacionais na nuvem [Gazzolla 2017]; detecção de pneumonia [Sousa 2018], câncer de mama [Chaves 2019], pólipos, embolia pulmonar, mitóticas [Costa 2017]; e tecnologia para segurança como agentes autônomos [Serafim 2018]. Outra aplicação interessante é o sistema de segurança contra acidentes em carros da empresa de veículos autônomos *Tesla*. Além da *Tesla*, mais empresas estão investindo nessa área [Oberderfer 2018], como a *IBM* e *Google*. Isso reforça a importância de seu estudo e aplicação na atualidade.

Além disso, o uso de dispositivos móveis é bastante amplo. Existem muitas utilidades desses equipamentos, que vêm ganhando espaço em diversas áreas [Lecheta 2013] como educação, através de jogos educativos [Neto 2013] e em pesquisas de campo [Boni 2006].

Este trabalho visa aplicar técnicas de *Deep Learning* usando a ferramenta *Watson Visual Recognition* da IBM para classificação de imagens de sementes. Além disso, desenvolver uma aplicação pervasiva criada para dispositivos móveis Android para a classificação de imagens de sementes.

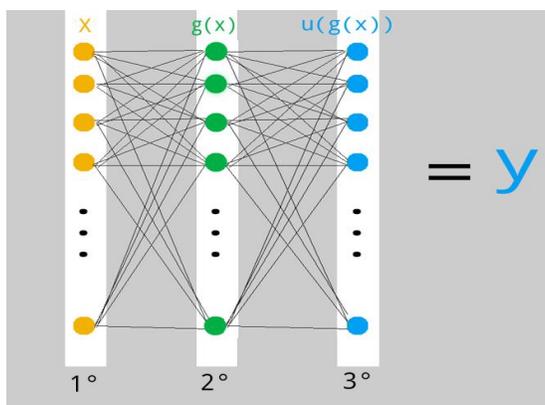
A organização deste artigo está da seguinte maneira: na Seção 2, Revisão Bibliográfica, são apresentados os principais conceitos sobre Redes Neurais e suas técnicas para modelagem e treinamento; em Metodologia, mostra-se como foi desenvolvida a aplicação pervasiva; a seção seguinte, Solução Proposta, apresenta a solução tecnológica proposta e são apresentados os resultados; a seguir, na seção Discussão, são analisados os resultados no contexto de outros trabalhos; e, finalmente, as considerações finais são apresentadas na seção Conclusão.

## 2. Revisão Bibliográfica

Esta seção apresenta os conceitos básicos relacionados às tecnologias que foram utilizadas para a concepção da aplicação de Classificação de Sementes.

### 2.1. Deep Learning

*Deep Learning* é uma subárea da Inteligência Artificial. O estudo deste tema entrou em ascensão após o avanço na tecnologia na área de produção de GPUs (Unidades de Processamentos Gráficos). Com isso, pode-se colocar em prática os estudos sobre Redes Neurais. Rede neural é um modelo matemático constituído por neurônios artificiais baseado na estrutura do cérebro humano para resolver problemas de Computação [Nielsen 2015]. Existem vários desses modelos, entre eles, a *multilayer perceptron* que também pode ser chamada por rede totalmente conectada, como mostra a Figura 1. A Equação 1 mostra que a rede é uma composição de funções [Rauber 2005].



**Figura 1: Rede neural totalmente conectada.**  
Fonte própria: autoria própria.

$$y = u(g(x)) \quad (1)$$

Na segunda camada ( $g(x)$  na equação), chamada de primeira camada profunda, cada circunferência mostrada na Figura 1 representa um neurônio artificial;  $x$  é a entrada, como na imagem, quanto na equação, assim como  $u(g(x))$  é terceira camada e  $y$  é o resultado dessa composição de funções. Saindo dessa visão macroscópica, pode-se concentrar em apenas um neurônio e entender o seu funcionamento. Com isso, seja  $x$  a entrada para um neurônio, a saída para ele é definida por  $y' = f(k(x))$ , onde  $f$  é a função de ativação e  $k$  é uma expressão que define a estrutura desse neurônio, como mostra a Equação 2, onde  $w$  é chamado de peso, que representa o nível de significância de  $x$  para o neurônio; e  $b$  de *bias*, que diz respeito a um limiar para o produto  $x*w$  [Ferneda 2006].

$$k(x) = x*w + b \quad (2)$$

Com os conceitos apresentados, pode-se dizer que a rede mostrada na Figura 1 é formada por uma camada de entrada  $x$ , uma camada profunda  $g(x)$ ; e uma camada de saída  $u(g(x))$ . O resultado da camada profunda já é o resultado  $y$ . Dessa forma, para acrescentar outra camada, basta fazer esse resultado ser a entrada para uma nova camada, que, por sua vez, a saída dela será a final da rede.

Após definir sua estrutura, observa-se que  $w$  e  $b$  são parâmetros como uma incógnita de uma equação que devem ser encontrados. Para isso, foram desenvolvidos métodos de aprendizagem para a rede ajustar esses parâmetros. Esses métodos têm a tese de que é possível encontrar pesos e *bias* caso haja muitas pares  $(x,y)$  para treinar a rede de forma que ela se generalize, a ponto de receber uma nova entrada e retornar a resposta correta. Basicamente, estes métodos consistem em aplicar uma entrada  $x$ , receber a saída  $y$  e comparar com a resposta  $p$  correta a fim de obter o erro associado e corrigir os parâmetros com base nisso [Raubert 2005].

Contudo, por vezes tem-se problemas em deixar uma rede *multilayer perceptron* com alta capacidade de generalização. Além de problemas como taxa de ajuste tendendo a zero com erro alto, e taxa de erro tendendo a infinito com taxa de erro alto ou baixo. Nesse sentido, sugere-se uma estrutura convolucional. As redes convolucionais tem foco para problemas com imagens e consiste num modelo que usa de extração de características de uma imagem por meio de filtros a fim de determinar sua classificação. A estrutura destas redes pode ser composta por uma camada de entrada, uma camada densa no fim - que é uma rede totalmente conectada - e quantas camadas quanto desejado entre elas, como camadas de convolucionais e resumo. Desse modo, pode-se obter bons resultados mesmo que as figuras estejam em outros contextos, como giradas ou com luminosidade diferente [Qiu et al 2018], esse poder de generalização é muito importante para classificação de imagens de objetos com diferentes cores de fundos. Pensando nessa oportunidade tecnológica, empresas como a *IBM* e *Google* estão desenvolvendo APIs - que são interfaces de programação de aplicações - para

disponibilizar serviços como classificação ou reconhecimento de imagens usando *Deep Learning*.

A *IBM*, por exemplo, fornece uma *API* para classificação de imagens através do serviço *Watson Visual Recognition*. Com esta *API*, é possível fazer treinamento de novos modelos para fazer a classificação de imagens, que pode ser processada a partir da interface de desenvolvimento fornecida gratuitamente pela *IBM*, ou através de cliente *REST* (*Representational State Transfer*), como uma aplicação *mobile* com acesso a internet [Ahmed 2019].

## 2.1. Desenvolvimento *Android*

Uma aplicação *Android* é um programa que conversa com o sistema operacional *Android* de um dispositivo para a execução de tarefas. A plataforma *Android* se destaca (1) pela diversidade de formas de comunicação disponíveis - por *Bluetooth*, *Wireless* ou rede *GSM* (*Global System for Mobile Communications*); (2) por contarem com disponibilizar câmera integrada de fácil manuseio; e (3) por ser um sistema operacional de código aberto e ter a maior base instalada de usuários, [Montoya et al. 2013] produziu um software *Android* para monitoramento e controle de agricultura precisa através de envio de dados remotamente. Além de ter boa conectividade, pode-se usar a câmera deste dispositivo para estudos na área da agricultura, pois permite obter fotos de boa qualidade. Também é importante destacar que as aplicações feitas para ele têm interface de usuário, isso torna o manuseio mais fácil [Kwon e Park 2017].

A programação dos dispositivos *Android* pode ser feita através da Interface de Desenvolvimento (*IDE*) chamada *Android Studio*. Nela, é possível escrever na linguagem Java, que é orientada a objetos e permite a criação de um leque de ferramentas para design, comunicação com a internet, processos assíncronos e acesso à câmera.

Na parte do design, o desenvolvimento pode ser feito com Java e a linguagem de marcação XML (*Extensible Markup Language*), com elas, pode-se definir objetos tais como *ProgressBar* - que é uma barra de progresso; *CropImageView* - interface para edições simples de imagens; *TextView* - exibe textos na tela; e, *ImageButton* e *Button* - botões.

Já para fazer requisições na Web, faz-se necessário implementar um processo paralelo ao da interface de usuário. Para isso, utiliza-se a classe *Thread* do Java. Então, com essas ferramentas, pode-se desenvolver uma aplicação *Android* com interface de usuário para classificação de imagens, por exemplo.

## 3. Metodologia

Para desenvolver a solução, fez-se a revisão bibliográfica de forma a permitir entender alguns dos principais aspectos da parametrização de modelos de redes neurais para classificação de imagens. Em seguida, fez-se a implementação do modelo de classificação de sementes

usando o serviço *Watson Visual Recognition*, e, depois os testes, a fim de avaliar possíveis melhoras nos estudos de classificação de imagens, em especial, de sementes. Depois disso, implementou-se uma aplicação *Android* que utiliza o modelo implementado com o *Watson Visual Recognition*. Para o desenvolvimento dessa aplicação, foi utilizada a interface de desenvolvimento (IDE) *Android Studio* com a linguagem de programação *Java*.

#### 4. Solução Proposta

Para criar o modelo de classificação de imagens de sementes, foram necessárias imagens para treinamento e testes. Foram utilizadas imagens de 13 classes de sementes catalogadas no Laboratório de Análise de Sementes do Centro de Ciências Agrárias, Ambientais e Biológicas (CCAAB), da Universidade Federal do Recôncavo da Bahia (UFRB). Cada semente tem nome e identificação numérico registrados em seus reservatórios. Foram tiradas de 10 a 30 fotos de cada semente. A Figura 2 mostra as sementes com identificação 126 e 133. A semente 126 é uma classe e a semente 133 é outra, cada classe representa uma espécie com 10 a 30 fotos.

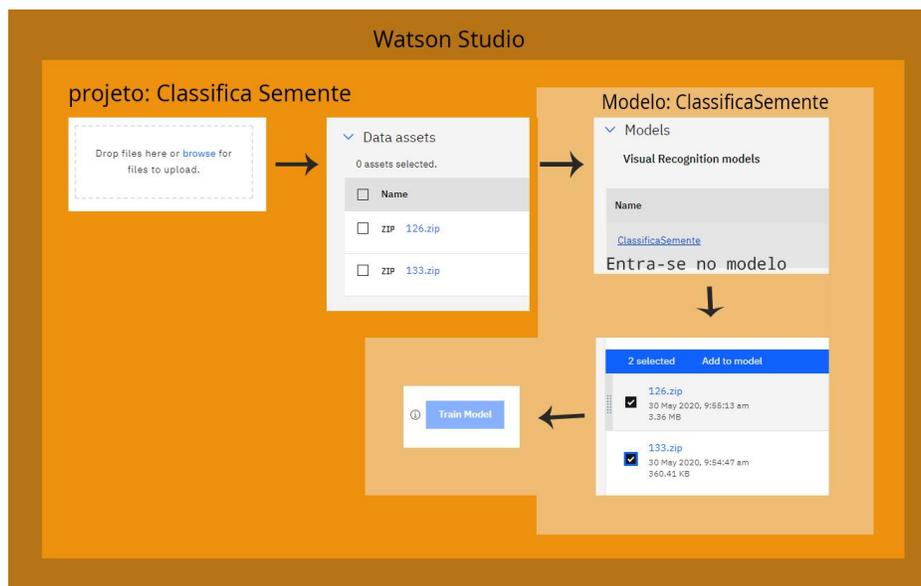


**Figura 2.** Imagens das classes 126 e 133, que são dois conjuntos de imagens de sementes com identificação 126 e 133, respectivamente.

Fonte: autoria própria.

Como a retirada das sementes mostradas na Figura 2 do laboratório onde elas estavam armazenadas não foi permitida, escolheu-se tirar fotos delas em superfícies coloridas para ter a melhor generalização de fotos possível nessas condições. Como medida de pré-processamento para o treinamento, foram geradas novas imagens a partir de recortes para cada cor de fundo, giro, zoom. Parte dessas imagens são retiradas para teste do modelo de classificação das imagens.

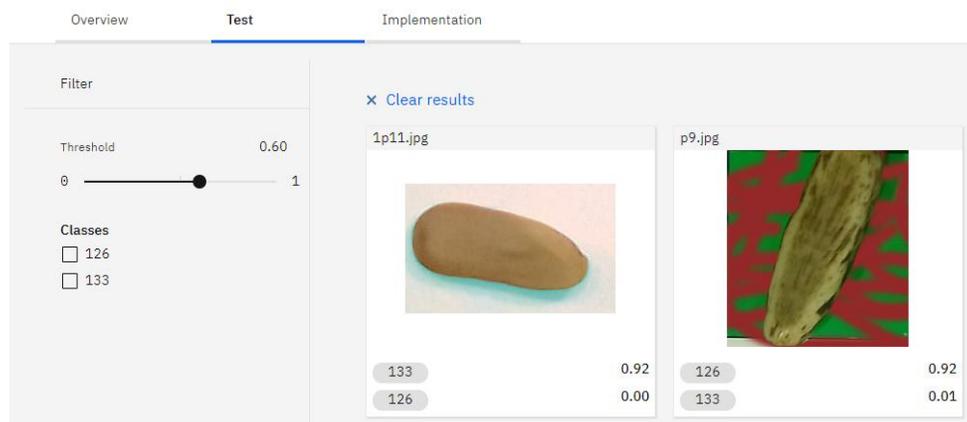
Iniciou-se então o treinamento do modelo usando o serviço *Watson Visual Recognition*.



**Figura 3. Passos para criação e treinamento do modelo na IDE Watson Studio.**  
 Fonte: autoria própria.

Na Figura 3, é mostrado o processo de criação, seleção de imagens - que deve-se ter feito o upload antes - e treinamento do modelo usando a *IDE Watson Studio*. Cada cor nesta imagem representa um nível de hierarquia no sentido de sequência.

O modelo feito com a *API Watson Visual Recognition* foi promissor e obteve média 92,31% de acerto para os testes iniciais como mostrado na Figura 4.



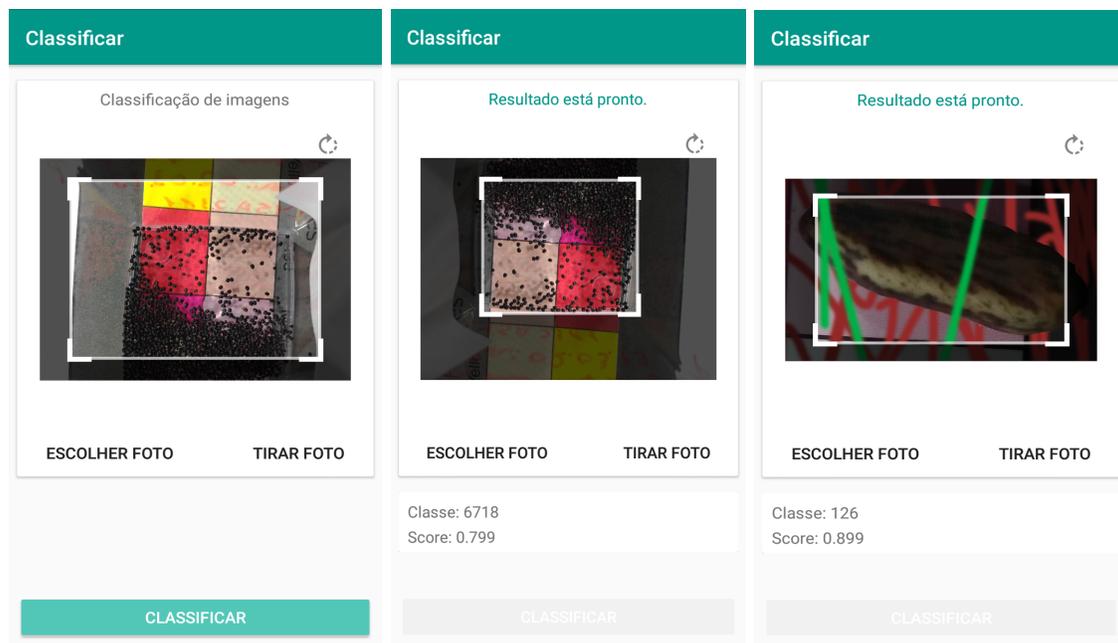
**Figura 4. Testes iniciais com o modelo Watson Vision Recognition.**  
 Fonte: autoria própria.

Para a implementação no *Android* é preciso definir a classe (em *Java*) que define o objeto de resultado do modelo gerado pela *API* usada, nela terá as informações de

identificação e score. Daí, define-se um processo assíncrono, nesse caso, utilizou-se a classe *Thread* do *Java*, para fazer o envio de imagem e receber o resultado de classificação.

Já no desenvolvimento da interface, tenta-se defini-la para melhor experiência e usabilidade. Com a informação de que se deve submeter uma imagem mais resumida possível para classificação, isto é, que a semente tome a maior parte do espaço da imagem, implementou-se um cortador de imagens. Como mostrado na Figura 6 (A), quando o usuário selecionar a imagem para classificação, se deparará com uma interface de edição de imagens (corte e giro) parecida com a de aplicativos conhecidos, como o *Whatsapp*. Assim, espera-se que o usuário selecione a área que compreenda a semente.

O resultado da aplicação *Android* é mostrado na Figura 6 (a), (b) e (c).



**Figura 6 (a), (b), (c). Interface da aplicação *Android* de classificação de sementes.**  
**Fonte: autoria própria.**

Por fim, fez-se uma série de testes mostrado na Tabela 1 com seus respectivos resultados.

**Tabela 1. Resultado dos testes. Fonte própria.**

| Testes |                         |            |                       |                  |
|--------|-------------------------|------------|-----------------------|------------------|
| Classe | Quantidade de Q. Testes | Q. Acertos | Porcentagem de acerto | Média de Certeza |
| 126    | 15                      | 15         | 100.00%               | 81.47%           |
| 133    | 15                      | 15         | 100.00%               | 92.87%           |
| 159    | 11                      | 11         | 100.00%               | 69.27%           |
| 515    | 10                      | 10         | 100.00%               | 93.3%            |
| 411    | 16                      | 13         | 81.25%                | 87.13%           |
| 5625   | 26                      | 23         | 88.47%                | 86.54%           |
| 6718   | 16                      | 12         | 75.00%                | 75.94%           |
| 6905   | 15                      | 14         | 93.33%                | 87.49%           |
| 6909   | 50                      | 31         | 62.00%                | 60.02%           |
| 6965   | 13                      | 13         | 100.00%               | 94.15%           |
| 6967   | 12                      | 12         | 100.00%               | 92.5%            |
| 6975   | 45                      | 45         | 100.00%               | 91.93%           |
| 7466   | 23                      | 23         | 100.00%               | 94.57%           |

Como mencionado previamente antes, foram retiradas imagens aleatoriamente do conjunto total de imagens (originais e geradas). Para testes, as imagens retiradas não são usadas para treinamento. Na Tabela 1, a quantidade de teste é justamente a quantidade de imagens usadas para classificação de teste. A porcentagem de acerto é dada pela razão entre a quantidade de acertos e quantidade de testes. A média de certeza é a média aritmética do *score* de cada classificação em determinada imagem.

## 5. Discussão

Quanto aos resultados, foram com alta precisão, a maior foi de 94,57%. Para ter esse resultado, todo processamento de treinamento é feito na nuvem e pode ser feito por qualquer computador, com o limite de 200MB/mês ou 100 imagens/mês.

De maneira semelhante, mas, voltado para uma aplicação mais específica, [De Souza Júnior 2018] criou modelos usando a biblioteca *keras* (*Biblioteca de Redes Neurais*) para um sistema de classificação de sementes de soja para a avaliação do vigor usando *Deep Learning*. Neste trabalho, são apresentados os resultados dos modelos de classificação, onde a maior precisão foi de 78,01%. [Singh, Singh and Kaur 2018] mostraram várias técnicas para classificação de imagens de sementes, entre estas, uma usando vários algoritmos para extrair 131 características de imagens como o *LBP* (*local binary pattern*), *GLCM* (*gray level co-*

*occurrence matrix*), *LSP (local similarity pattern)*, *LSN (local similarity numbers)*, *GLRM (gray level run length matrix)*. Por outro lado, pensando no desenvolvimento com uso de smartphones, [Kwon e Park 2017] apresentaram 17 trabalhos que usam smartphones em aplicações na agricultura, comida e meio ambiente. Entre elas, é citada uma aplicação *Android* que quantifica clorofila em folhas de milho usando a câmera de um smartphone.

Já no presente trabalho, fez-se um modelo de classificação de sementes para a plataforma *Android*, usando elementos de interface de usuário, de modo que pessoas que não têm muito conhecimento na área poderão criar e treinar seus próprios modelos ou atualizar. Neste modelo, são extraídas características diversas, desde primitivas, como bordas e formas em geral, até características de alto nível, que são a combinação das primitivas. A facilidade de implementar a *API* fornecida para cliente *REST* tornou possível a implementação de uma aplicação *Android* que faz uso do modelo criado. Essa aplicação fornece mobilidade e facilidade de uso.

## 6. Conclusão

Diante dos resultados apresentados neste artigo, conseguiu-se aplicar *Deep Learning* para classificação de imagens de sementes, e implementar uma aplicação *mobile* para a plataforma *Android*. Pensando nisso, pode-se melhorar os resultados no contexto de mobilidade, como implementar uma Rede Neural usando a Biblioteca *Keras* de forma que seja possível o uso da aplicação *Android* mesmo *off-line* em um trabalho futuro.

## Referências

- Ahmed, Imtihan, et al. "Custom visual recognition model with Watson studio." Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering. 2019.
- Boni, Gabriel N., et al. "Desenvolvimento de um Software de Coleta de Dados para Pesquisas de Campo Através de Dispositivos Móveis." X Congresso Brasileiro de Informática em Saúde. 2006.
- Carvalho, Leticia Renata de, Edvaldo Aparecido Amaral da Silva, and Antonio Claudio Davide. "Classificação de sementes florestais quanto ao comportamento no armazenamento." Revista Brasileira de Sementes 28.2 (2006): 15-25.
- Chaves, Esdras de Lima. "Detecção de câncer de mama por meio de imagens infravermelhas utilizando Redes Neurais Convolucionais." (2019).
- Costa, Marly Guimarães Fernandes. "Redes Neurais Convolucionais na Saúde." Journal of Health Informatics 9.4 (2017).
- De Souza Júnior, Marcelo et al. Rumo à Melhoria de Produtividade e Sustentabilidade Agrícola por meio da Classificação Automática do Vigor de Sementes de Soja. In: Anais do XLV Seminário Integrado de Software e Hardware. SBC, 2018.

- Ferneda, Edberto. Redes neurais e sua aplicação em sistemas de recuperação de informação. *Ciência da Informação*, v. 35, n. 1, 2006.
- Gazzola, Murilo. "Um Método para Avaliação Automática da Qualidade de Recursos Educacionais Abertos usando Deep Learning." *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. Vol. 28. No. 1. 2017.
- Gazolla-Neto, Alexandre, et al. "Rastreabilidade aplicada à produção de sementes de soja." *Informativo ABRATES, Londrina* 22.2 (2012): 20-24.
- Kwon, Ojun; Park, Tusan. Applications of smartphone cameras in agriculture, environment, and food: A review. *Journal of Biosystems Engineering*, v. 42, n. 4, p. 330-338, 2017.
- Lecheta, Ricardo R. *Google Android-3ª Edição: Aprenda a criar aplicações para dispositivos móveis com o Android SDK*. Novatec Editora, 2013.
- Montoya, Francisco G. et al. A monitoring system for intensive agriculture based on mesh networks and the android system. *Computers and Electronics in Agriculture*, v. 99, p. 14-20, 2013.
- Neto, José Francisco Barbosa, and Fernando de Souza da Fonseca. "Jogos educativos em dispositivos móveis como auxílio ao ensino da matemática." *RENOTE-Revista Novas Tecnologias na Educação* 11.1 (2013).
- Nielsen, Michael A. *Neural Networks and Deep Learning*. Determination Press, 2015.
- Oberderfer Júnior, Réges Eduardo. "Deep learning for boolean matching." (2018).
- Rauber, Jonathan Terhorst. "Sistema de visão computacional aplicado à contagem de sementes de soja movimentadas por uma transportadora de sucção pneumática." (2019).
- Rauber, Thomas Walter. "Redes neurais artificiais." *Universidade Federal do Espírito Santo* (2005).
- Qiu, Zhengjun, et al. "Variety identification of single rice seed using hyperspectral imaging combined with convolutional neural network." *Applied Sciences* 8.2 (2018): 212.
- Serafim, Paulo Bruno de Sousa. "Avaliação da competição no treino de agentes autônomos com Aprendizado Profundo por Reforço em jogos de Tiro em Primeira Pessoa." (2018).
- Singh, Simranjeet, Parminder Singh, and Avinash Kaur. "A Survey on Image Processing Techniques for Seeds Classification." *2018 4th International Conference on Computing Sciences (ICCS)*. IEEE, 2018.
- Sousa, Gabriel Garcez Barros. "Deep learning para a detecção e classificação de pneumonia por radiografias do tórax." (2018).