18º Congresso Interinstitucional de Iniciação Científica – CIIC 2024
27, 28 e 29 de agosto de 2024
ISSN: 2965-2812

# ANÁLISE DOS REGRESSORES GPR, GLM, E K-NEAREST NEIGHBORS PARA ESTIMATIVA DE NÚMERO DE FRUTOS EM LARANJEIRAS.

André L. V. **Almeida**[1]; Kleber X. S. **de Souza**[2]; Sônia **Ternes**[3]; João Camargo **Neto**[4]

Embrapa Agricultura Digital
Av. André Tosello, nº 209 Campus da Unicamp, Barão Geraldo
Caixa Postal: 6041 CEP: 13083-886 – Campinas – SP

**Nº 24602**

***Resumo:*** *Estimar a produção total de frutas é uma tarefa complicada, porém muito importante para a colheita, armazenamento e venda do produto. Atualmente, esse método é feito com a combinação da contagem a mão dos citros e métodos estatísticos. Uma possível solução para o problema é a utilização de programas de identificação visual combinados com modelos de regressão. Neste artigo, buscamos encontrar a eficácia dos regressores GLM, GPR e K-Nearest Neighbors na contagem de laranjas. Dentre os modelos estudados, o que obteve melhores resultados foi a Regressão por Processo Gaussiano.*

***Palavras-chave:*** *Contagem de frutas, modelo de regressão, regressor.*

[1] André L. V. Almeida, Bolsista CNPq (PIBIC): Graduação em Engenharia Agrícola, UNICAMP, Campinas-SP, andrevalerio160920@gmail.com
[2] Kleber X. S. de Souza, Orientador: Pesquisador da Embrapa Agricultura Digital, Campinas-SP, kleber.sampaio@embrapa.br
[3] Sônia Ternes, Pesquisador da Embrapa Agricultura Digital, Campinas-SP, sonia.ternes@embrapa.br
[4] João Camargo Neto, Pesquisador da Embrapa Agricultura Digital, Campinas-SP, Campinas-SP, joao.camargo@embrapa.br

ISSN: 2965-2812

***Abstract:*** *Estimating total fruit production is a complex task, but it is very important for harvesting, storing, and selling the product. Currently, this method is done by combining the manual counting of citrus fruits and statistical methods. A possible solution to the problem is the use of visual identification programs combined with regression models. In this paper, we seek to find the effectiveness of GLM, GPR, and K-Nearest Neighbors regressors in counting oranges. Among the studied models, Gaussian Process Regression obtained the best results.*

***Keywords:*** *Fruit counting, regression models, regressor.*

## 1. INTRODUÇÃO

A agricultura desempenha um papel fundamental na economia brasileira, abrangendo a produção em larga escala de uma variedade de produtos, desde frutas até hortaliças. No segmento de citros, o Brasil destaca-se como um dos principais produtores mundiais, contribuindo com mais da metade da produção global de suco de laranja. Essa posição coloca a laranja como um elemento-chave no mercado agrícola nacional.

Determinar a quantidade de frutas em uma laranjeira é essencial para o planejamento da colheita, armazenamento e comercialização do produto. Tradicionalmente, a contagem manual das frutas tem sido o método predominante, como evidenciado no relatório anual de produção do cinturão citrícola de São Paulo e Triângulo/Sudoeste Mineiro, conduzido pela Fundecitrus em 2023 (Fundecitrus 2023). Embora eficaz, esse procedimento é custoso e demanda trabalho pesado por parte da mão de obra. Diante da necessidade de um processo mais eficiente, torna-se necessário explorar outras abordagens para a estimativa da produção de laranjas.

Uma abordagem que pode ser aplicada seria utilizar métodos automáticos que contam frutas. Este tipo de modelo tende a consumir menos tempo e ter um custo mais baixo do que a contagem manual das laranjas, uma vez que o algoritmo identifica e conta os produtos de maneira eficiente, não necessitando o uso de mão de obra braçal.

Ademais, programas de análise de imagem já existem atualmente, porém quando se trata de contar frutas em plantações, acabam sendo ineficazes. Isto ocorre devido ao fato de parte das frutas estarem localizadas na porção central da árvore, fazendo com que sejam sobrepostas por outras laranjas e fiquem “invisíveis” para o algoritmo. Portanto, decidiu-se experimentar a utilização de modelos estatísticos de regressão, buscando identificar se a estimativa obtida aproximar-se-ia melhor dos valores observados nas plantas. Para isso, neste artigo, três modelos foram avaliados: GPR (Regressão por Processo Gaussiano), GLM (Modelo Linear Generalizado) e KNN (K-Nearest Neighbors).

ISSN: 2965-2812

## 2. MATERIAIS E MÉTODOS

Este trabalho utiliza dados coletados pelo Fundecitrus em um projeto conjunto com a Embrapa. Foram coletadas imagens de plantas e de contagem de frutos, os quais foram recortados e anotados pela Embrapa para realizar a contagem automática por algoritmos de visão computacional. Ademais, junto com outras variáveis como o tipo de planta, talhão, dimensões e outros fatores, o número de laranjas é combinado para podermos quantificar a precisão dos regressores.

Obtidos esses dados, seleções de sub-conjuntos e/ou complementos dos mesmos são necessárias algumas transformações e padronizações antes da aplicação dos modelos, bem como definir o processo de validação. Esses detalhes encontram-se nos próximos sub-itens.

### 2.1 Seleção dos dados

Com uma base de dados com o total de 1034 árvores por 30 colunas, foram avaliados a relevância dos dados, sobrando ao final 21 colunas. Dentre as características de cada árvore é importante ressaltar o valor inicial e o valor alvo. Para o valor de início tem-se as colunas “3D2D_A” e “3D2D_B” as quais seriam, respectivamente, a contagem de frutas no lado A e B da planta. Neste caso, havia valores vazios em algumas linhas, sendo estes preenchidos com 0. Já para o valor alvo dos frutos foram “FRUTOS_F1”, “FRUTOS_F2” e “FRUTOS_F3”.

Na ***tabela 1***, temos as colunas da base de dados. A coluna “GRUPO_IDADE” se refere a idade em anos da planta e se é transplantio ou não. “DENSIDADE” se trata da multiplicação de outras 3 colunas de dimensões da planta em metros (“ALTURA”, “LARGURA” e “COMPRIMENTO”). “REGIAO” se diz sobre o município a qual o dado foi retirado, por exemplo “BRO” se trataria sobre a cidade de Brotas-SP. Em “SETOR”, estaríamos tratando sobre qual setor de certa região de plantação de laranjeiras aquela planta se encontra.

**Tabela 1.** Base de dados inicial contendo parte de suas colunas.

| | SETOR | REGIAO | GRUPO_IDADE | TIPO | DENSIDADE | 3D2D_A | 3D2D_B | … |
|---|---|---|---|---|---|---|---|---|
| **0** | NORTE | TMG | 1 | ORIGINAL | 486,06 | 16.0 | 34.0 | … |
| **1** | SUL | PFE | 3 | ORIGINAL | 299,47 | NaN | 29.0 | … |
| **2** | NORTE | TMG | 2 | ORIGINAL | 348,91 | 18.0 | 17.0 | … |

ISSN: 2965-2812

### 2.2 Pré-processamento de dados

Primeiramente, foi necessário transformar algumas colunas em outras ou somar algumas já existentes. No caso do parâmetro “GRUPO_IDADE” e “GRUPO_VARIEDADE”, foram divididas em “IDADE”, “GRUPO” e “TALHÃO”. As colunas “PROFUNDIDADE (MTS)”, “LARGURA (MTS)” e “ALTURA (MTS)” foram multiplicadas entre si e colocadas em “VOLUME”.

Ademais, para o valor alvo foi criado o parâmetro “SOMA_FRUTOS” e somados os valores de “FRUTOS_F1”, “FRUTOS_F2” e “FRUTOS_F3”. Com relação ao valor inicial, foram feitas algumas considerações, valores 0 em alguma das 2 colunas foram substituídas pelo mesmo valor da outra coluna da mesma linha. Isto pode ser visto pela ***figura 2***.

```
for i, valor in dataset['3D2D_A'].items():
    if valor == 0:
        dataset.at[i,'3D2D_A'] = dataset.at[i,'3D2D_B']
```

**Figura 2.** Adequação dos valores iniciais

Após, foram somados os valores de “3D2D_A” e “3D2D_B” em outra coluna “3D2D”. E, a seguir, outras transformações e seleções foram utilizadas, descritas nos próximos sub-itens.

#### 2.2.1 Transformação OneHotEncoder

Como nem todos os valores da base de dados são numéricos, é necessário utilizar o OneHotEncoder (Géron 2019), onde valores em “string” são transformados em indicadores 0 ou 1. Por exemplo, a coluna “TIPO” possui o valor ORIGINAL, após o processamento será criada uma outra coluna chama “TIPO_ORIGINAL” mostrando apenas valores 1 para as plantas que forem do tipo original e 0 para as que não forem.

#### 2.2.2 Padronização dos dados

A discrepância nos dados pode ser prejudicial para a convergência dos modelos, exemplo disso são valores que vão de 0-10.000 em uma coluna e em outra de 1-10. Com isso, é necessário aplicar a padronização dos valores numéricos (Géron 2019).

#### 2.2.3 Filtro de dados 30%

Como na coleta de dados pode haver problemas, algumas plantas podem ficar com poucas laranjas visíveis. Neste caso, é necessário aplicar o filtro de 30%, onde valores de início são descartados caso sejam menores que 30% de valores alvo.

Após todo o processo descrito acima, foram deletadas algumas colunas já não necessárias, restando um total de 5 colunas. Utilizando o OneHotEncoder adicionaram-se mais 34 colunas, tendo no final 39 para serem aplicadas nos modelos de regressão.

ISSN: 2965-2812

Para todo modelo de regressão é necessário separar os dados em treino e teste. Neste estudo não foi diferente, tendo 80% dos dados separados em dados de treino e 20% em treino. Além disso, o mesmo foi feito para o valor alvo “SOMA_FRUTOS”, separado em teste e treino.

### 2.3 Modelos de regressão utilizados

Neste trabalho, estamos avaliando três modelos de regressão, sendo eles Regressão por Processo Gaussiano, Modelo Linear Generalizado e K-Nearest Neighbors. Os modelos foram implementados em Python utilizando como plataforma o Jupyter-Notebook. As bibliotecas utilizadas foram SciKit-Learn, Keras e Scipy.

No problema de regressão, buscamos um modelo que relacione a variável de interesse com as previsoras. Para avaliar a qualidade do modelo, usamos o Erro Absoluto Médio (MAE), a Raiz do Erro Quadrático Médio (RMSE), e o Coeficiente de Determinação ($R^2$).

#### 2.3.1 GPR (Regressão por Processo Gaussiano)

O modelo GPR baseia-se na ideia de que pontos de dados próximos no espaço de entrada provavelmente terão valores de saída semelhantes (Duvenaud et al. 2021) . Utilizando uma função kernel, que define a similaridade entre dois pontos de dados, o modelo prevê o valor de saída para um novo ponto de entrada com base nos valores de saída dos pontos de dados de treinamento próximos.

Os parâmetros definidos no modelo podem ser vistos pela ***figura 3***, sendo eles a função kernel, alpha e normalize_y. No kernel, foram utilizados o ConstantKernel (C), que é padrão na construção do modelo GPR, e o RBF para medir a similaridade entre os pontos, ambos com valores (1e-4, 1e4). Alpha tem a função de regularização e tratamento de ruído nos dados de treinamento. O parâmetro normalize_y é utilizado para normalizar a variável resposta ('y') antes de ajustar o modelo.

```
kernel = C(1.0, (1e-4, 1e4)) * RBF(1.0, (1e-4, 1e4))
modelGPR = GaussianProcessRegressor(kernel=kernel, alpha=0.2, normalize_y=True)
```

**Figura 3.** Configuração para construção do modelo GPR.

Para os testes no modelo GPR e dos seguintes, cada parâmetro foi testado individualmente levando em conta a melhor convergência do modelo, por exemplo quando testado o parâmetro “alpha”, os outros parâmetros se mantiveram iguais aos do modelo acima. Os testes podem ser vistos na ***Tabela 2.***

ISSN: 2965-2812

**Tabela 2.** Testes com diferentes parâmetros e seus valores no regressor GPR.

| Parametros GPR | Valor | Global $R^2$ | $R^2$ 30% | Global MAE | MAE 30% | Global RMSE | RMSE 30% |
|---|---|---|---|---|---|---|---|
| **Alpha** | 0.1 | 0,53 | 0,78 | 193,45 | 150,74 | 265,38 | 196,79 |
| | 0.2 | 0,6 | 0,8 | 178,13 | 144,17 | 242,67 | 188,33 |
| | 0.3 | 0,62 | 0,79 | 175,24 | 146,85 | 236,12 | 191,08 |
| | 0.5 | 0,62 | 0,77 | 178,01 | 153,85 | 237,73 | 200,13 |
| | 0.8 | 0,61 | 0,74 | 178,72 | 160,61 | 239,15 | 213,48 |
| | 1 | 0,61 | 0,73 | 179,02 | 162,53 | 239,95 | 218,06 |
| **Normalize_y** | True | 0,6 | 0,8 | 178,13 | 144,17 | 242,67 | 188,33 |
| | False | -0,09 | 0,16 | 274,55 | 239,62 | 403,48 | 383,84 |

### 2.3.2 GLM (Modelo Linear Generalizado)

O GLM é um modelo de regressão que permite com que variáveis de resposta que não seguem uma distribuição normal possam ser modeladas (Hilbe 1994). Essa generalização é feita na regressão linear, permitindo que a relação entre variável de entrada e a variável alvo seja modelada por duas funções: função link e distribuição da família exponencial.

```
modelGLM = sm.GLM(train_labels, train_features_with_intercept,
                  family=sm.families.Gaussian(),
                  links=sm.families.links.identity)
```

**Figura 4.** Configuração para construção do modelo GLM.

Antes de compilar este modelo, é necessário incluirmos um termo de intercepto, o qual é usado para capturar a média da variável alvo quando todas as variáveis iniciais são iguais a zero.

Nos parâmetros deste tipo de modelo, é necessário usar sempre uma distribuição de Family acompanhado de uma função link. A família refere-se à distribuição probabilística da variável alvo e determina a função de link conveniente para transformar a relação linear da variável inicial de maneira adequada para modelar a variável alvo. Neste caso, o arranjo da family que convergiu melhor foi a gaussian junto com a função de link identity.

Os testes para o modelo GLM podem ser vistos pela ***Tabela 3.***

ISSN: 2965-2812

**Tabela 3.** Testes com o parâmetro “Family” e seus valores no regressor GLM.

| Parametros GLM | Valor | Global $R^2$ | $R^2$ 30% | Global MAE | MAE 30% | Global RMSE | RMSE 30% |
|---|---|---|---|---|---|---|---|
| **Family** | Gaussian | 0,58 | 0,71 | 186,95 | 167,62 | 249,31 | 227,42 |
| | Binomial | -1,73 | -1,55 | 507,81 | 523,5 | 637,56 | 671,32 |
| | Gamma | -27,79 | -25,42 | 448,01 | 492,46 | 2068,71 | 2160,37 |
| | Tweedie(var-power=0) | 0,54 | 0,56 | 197,28 | 212,03 | 260,17 | 277,41 |
| | Poisson | 0,45 | 0,49 | 198,46 | 204,27 | 285,92 | 299,6 |

### 2.3.3 KNN (K-Nearest Neighbors)

O modelo de regressão KNN é um modelo mais simples e sem fase de treinamento explícita, armazenando todos os exemplos de treinamento e fazendo previsões baseadas na proximidade das variáveis durante sua fase de teste (Wan Nurazwin Syazwani et al. 2021) . A configuração usada pode ser vista na ***figura 5.***

```
knn_regressor = KNeighborsRegressor(n_neighbors=10, weights='distance',
                                    algorithm='kd_tree', metric='chebyshev')
```

**Figura 5.** Configuração para construção do modelo KNN

Os parâmetros utilizados e seus testes podem ser vistos na ***Tabela 4.*** Neste modelo foram usados os parâmetros: n_neighbors (número de vizinhos), weights, algorithm e metric. Em n_neighbors, o número de vizinhos mais próximos é definido, ou seja, os pontos de dados mais próximos do ponto que está sendo predito. Para weights, são especificados o quão diferentes vizinhos contribuem para a predição final, sendo colocados “pesos”. Ademais, algorithm especifica o tipo de algoritmo usado para encontrar os k vizinhos mais próximos. Finalmente, o parâmetro metric específica a métrica de distância usada para determinar a proximidade entre os pontos de dados.

Assim como no modelo GPR, os testes foram feitos para cada parâmetro levando em conta a melhor configuração do modelo.

ISSN: 2965-2812

**Tabela 4.** Testes com diferentes parâmetros e seus valores no regressor KNN.

| Parâmetros KNN | Valor | Global $R^2$ | $R^2$ 30% | Global MAE | MAE 30% | Global RMSE | RMSE 30% |
|---|---|---|---|---|---|---|---|
| **N_neighbors** | 10 | 0,61 | 0,73 | 184,36 | 160,54 | 240,82 | 217,94 |
| | 50 | 0,56 | 0,67 | 194,53 | 180,31 | 256,72 | 239,34 |
| | 100 | 0,55 | 0,61 | 199,46 | 201,43 | 258,22 | 263,61 |
| | 500 | 0,39 | 0,24 | 244,23 | 276,43 | 300,31 | 364,29 |
| **Weights** | uniform | 0,55 | 0,73 | 196,58 | 161,8 | 256,14 | 218,48 |
| | distance | 0,61 | 0,73 | 184,36 | 160,54 | 240,82 | 217,94 |
| **Algorithm** | ball_tree | 0,55 | 0,57 | 198,72 | 211,84 | 259,17 | 276,99 |
| | kd-tree | 0,61 | 0,73 | 184,36 | 160,54 | 240,82 | 217,94 |
| | brute | 0,54 | 0,71 | 199,25 | 174 | 260,32 | 224,39 |
| **Metric** | euclidean | 0,49 | 0,68 | 208,59 | 173,74 | 275,16 | 239,17 |
| | manhattan | 0,43 | 0,62 | 220,56 | 183,63 | 290,3 | 258,73 |
| | chebyshev | 0,61 | 0,73 | 184,36 | 160,54 | 240,82 | 217,94 |
| | minkowski | 0,49 | 0,68 | 208,59 | 173,74 | 275,16 | 239,17 |

### 2.4 Validação cruzada

Após os testes feitos em cada modelo para encontrar uma boa configuração para convergência, é necessário utilizarmos a validação cruzada para definir o melhor modelo para cada regressor.

A validação cruzada (cross validation) é uma técnica que avalia a capacidade de generalização de um modelo de machine learning. Seu objetivo principal é testar o desempenho de um modelo em dados não vistos, garantindo que o modelo se comportará bem com dados reais e de teste e evitando o overfitting (quando o modelo se acomoda demais aos dados de treinamento).

No caso, os dados foram divididos em 10 partes aproximadamente iguais chamadas de folds, tendo que as nove primeiras partes são para treinamento e a décima para avaliação dos modelos. Para verificar a superioridade de um modelo a outro, utilizamos o *teste de hipótese estatístico* (King e Zisserman 2019). É suposto que a ideia de os dados seguirem uma distribuição gaussiana não seja real, já que a distribuição gaussiana acontece em conjuntos de dados maiores que 30 elementos e com variância conhecida. Portanto, utilizamos a estatística-T para os 10 folds, com valor-p de 0,05. O valor de 5% ou menor, é possível concluir que os modelos são estatisticamente diferentes.

ISSN: 2965-2812

## 3. RESULTADOS E DISCUSSÃO

Utilizando as 1036 observações do banco de dados sobre as laranjeiras, foi possível determinar entre os três modelos de regressão deste estudo qual melhor se enquadra para determinação do número de laranjas em um pé de laranja. Para treinamento foram utilizadas 829 observações e para teste, 207.

Como parte da validação cruzada, os modelos foram treinados 10 vezes, tendo a parte de teste separada dos 10 folds. A principal métrica utilizada para avaliar o desempenho dos modelos foi o coeficiente de determinação ($R^2$), pois mede a proporção da variância na variável resposta explicada pelo modelo. Sem o filtro de 30%, os valores de $R^2$ nos modelos se mantiveram parecidos, sendo o menor o GLM ($R^2 = 0,58$) seguido do GPR ($R^2 = 0,6$) e KNN ($R^2 = 0,61$)

Ressaltando o médio erro absoluto (MAE) e a raiz do erro quadrático médio (RMSE), temos que o GPR (MAE = 178,13) apresentou o melhor valor, seguido do KNN (MAE = 184,36) e GLM (MAE= 186,95). Já para o RMSE, o KNN (RMSE = 240,82) se destacou, seguido do GPR (RMSE = 242,67) e GLM (RMSE = 249,31).

Aplicando o filtro de 30% dos dados, o resultado acaba sendo um pouco diferente. No estudo, o modelo GLM apresentou $R^2 = 0,71$ e o KNN $R^2 = 0,73$. Porém, o regressor GPR teve uma grande melhora, indo de um $R^2 = 0,6$ para um $R^2 = 0,8$.

Além da métrica do $R^2$, é importante ressaltar o médio erro absoluto (MAE) e a raiz do erro quadrático médio (RMSE). Assim como o $R^2$, a tendência de valores no MAE se manteve, com o GPR sendo o melhor (MAE = 144,17), seguido do KNN (MAE = 160,54) e GLM (MAE = 167,62). O mesmo ocorreu com a RMSE, o modelo GPR (RMSE = 188,33) se destacou, seguido pelo KNN (RMSE = 217,94) e GLM (RMSE = 227,42).

**Tabela 5.** Resultado dos três modelos aplicados ao mesmo dataset. As colunas com 30% são referentes aos dados filtrados com o filtro de 30% e as colunas “Global” são sobre o todo o dataset.

| Algorithm | Global $R^2$ | $R^2$ 30% | Global MAE | MAE 30% | Global RMSE | RMSE 30% |
|---|---|---|---|---|---|---|
| GPR | 0,6 | 0,8 | 178,13 | 144,17 | 242,67 | 188,33 |
| GLM | 0,58 | 0,71 | 186,95 | 167,62 | 249,31 | 227,42 |
| KNN | 0,61 | 0,73 | 184,36 | 160,54 | 240,82 | 217,94 |

ISSN: 2965-2812

## 4. CONCLUSÃO

Neste estudo, analisamos três modelos de regressão aplicados à contagem de frutos em pés de laranja, tendo o número de laranjas sido identificado por programas de identificação visual. Dentre os regressores testados, o que apresentou melhor convergência foi o de Regressão por Processo Gaussiano (GPR) com $R^2$ = 0,8, considerando que, por parte dos dados, pelo menos 30% das frutas estavam visíveis. Além disso, os resultados para as outras métricas foram superiores no modelo GPR. Os outros dois regressores também apresentaram resultados próximos, como o K-Nearest Neighbors (KNN) e o Modelo Linear Generalizado (GLM) que mostraram, respectivamente, $R^2$ = 0,73 e $R^2$ = 0,71. Ademais, concluiu-se que a performance de qualquer regressor está diretamente ligada à qualidade das imagens e dos dados coletados.

## 5. AGRADECIMENTOS

Agradeço ao CNPq pela oportunidade de fazer este artigo e pela bolsa concedida. Também agradeço a meu orientador Kleber X. S. de Souza por me guiar e ajudar neste trabalho e agradecemos ao Fundecitrus pelo fornecimento das imagens e dados das plantas obtidos por ocasião da derriça.

## 6. REFERÊNCIAS

Cerqueira, L. M. de Souza, K. X. S., Ternes, S., and Camargo Neto, J. (2020). ***Usando a rede neural faster-rcnn para identificar frutos verdes em pomares de laranja.*** In CONGRESSO INTERINSTITUCIONAL DE INICIAÇÃO CIENTíFICA, Campinas, SP. Embrapa

Duvenaud, D., Rippel, O., Adams, R., & Ghahramani, Z. (2021). ***An Introduction to Gaussian Process Models***, páginas 3-22

Fundecitrus (2023). ***Relatório de atividades: junho 2022/maio 2023***. Technical report, Fundo de Defesa da Citricultura (Fund for Citrus Protection) – Fundecitrus.

Géron, A. (2019). **Hands-on Machine Learning with Scikit-Learn, Keras, and Tensor-Flow: Concepts, Tools, and Techniques to Build Intelligent Systems.** O'Reilly Media, Incorporated.

Hilbe, J. M. (1994). ***Generalized Linear Models*****,** *The American Statistician* 48

King, A. P. e Zisserman, A. (2019). ***Statistics for Biomedical Engineers and Scientists.*** Academic Press, primeira edição.

Maldonado, W. and Barbosa, J. C. (2016). ***Automatic green fruit counting in orange trees using digital images. Computers and Electronics in Agriculture***, 127:572–581. Informática Agropecuária.

Wan Nurazwin Syazwani R., Muhammad Asraf H., Megat Syahirul Amin M.A., & Nur Dalila K.A. (2021). ***Automated image identification, detection and fruit counting of top-view pineapple crown using machine learning.*** Alexandria Engineering Journal, 61(2), 1265-1276.

Wulfsohn, D., Zamora, F. A., Téllez, C. P., Lagos, I. Z., and García-Fiñana, M. (2012). ***Multilevel systematic sampling to estimate total fruit number for yield forecasts*****.** Precision Agriculture, 13:256–280.