*Egg classification model*

Matheus Pupp de Araujo Rosa   
Poultry Science Department  
*(Agriculture Department)*  
Auburn University  
Auburn, USA  
mzp0199@auburn.edu

*Abstract*—CNN models are well used for image classification tasks, and as demonstrated by [1] the VGG19 architecture proved to be the most accurate one for the classification of eggs into 4 different categories (Healthy, Dirty, Cracked and Dirty, and Cracked), so I tried to implement the same model architecture to replicate what was done by their team and build a model capable of achieving similar results. Using ChatGPT for code generation I got an accuracy of 87%, which is close to the paper model that achieved 93% so I can say that the implementation was a success, and further applications can be tried next.

Keywords—AI, CNN, Deep Learning, Eggs, Poultry

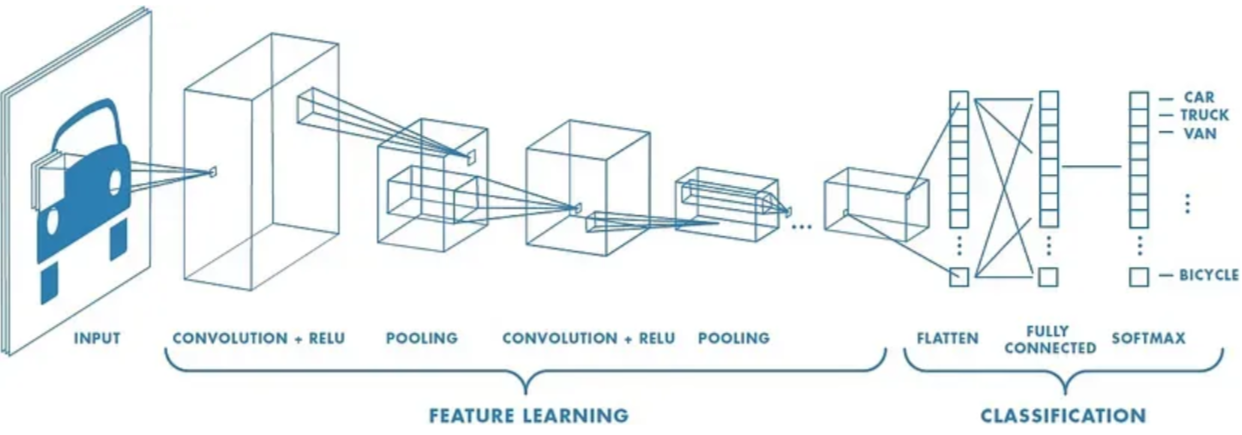
# INTRODUCTION

It is known that cracks in the eggshells can be an entrance point for bacterial contamination, resulting in worse hatching parameters and chicks viability [6]. With that in mind, an extremely important part of the process for incubating eggs is to separate them into different categories, with the purpose of reducing the risk of contamination from one class to another. Many companies usually separate them into Litter Eggs, Clean Nest Eggs and Dirty Nest Eggs, with the intention to minimize the risk of contamination in the incubatory and economic losses with it.

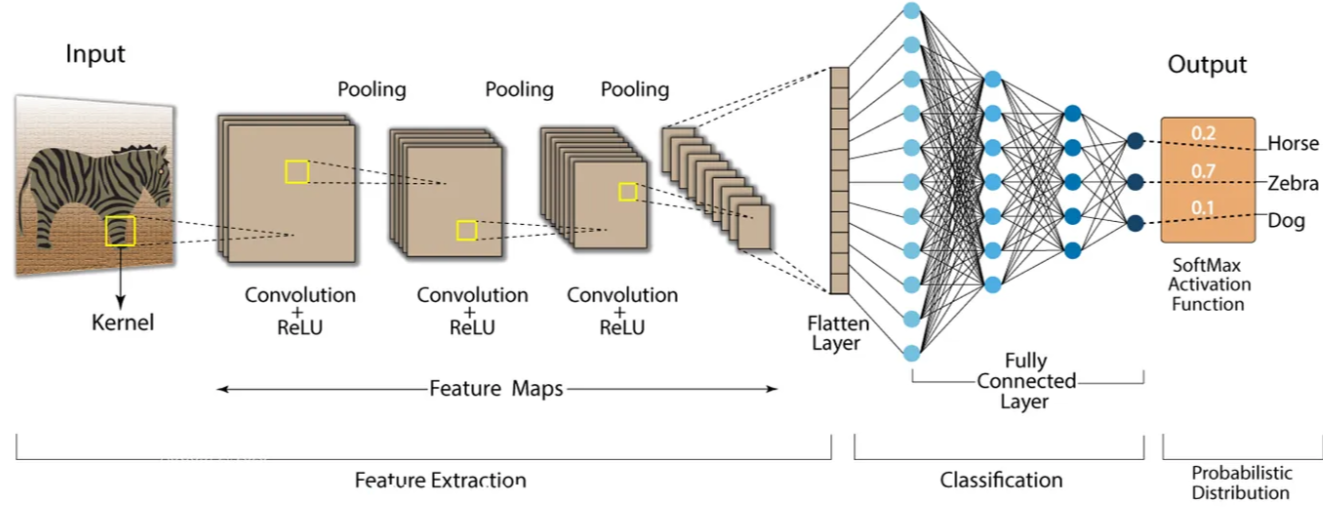
Knowing about the risks of mixing up eggs of different categories in the same incubator, people did manual separation of those eggs, but that can be hard even for well-trained professionals, especially if we try to account for cracks in the eggshells, which can range from easily visible to almost imperceptible. Accounting for that challenge, recent studies showed that it is possible to utilize an artificial intelligence, neural network model to automatically classify those eggs, like it was made by [1]. They compared the accuracy of different convolutional neural networks (CNNs) architectures, such as ResNet34, ResNet50 and VGG19, showing that for this task the most accurate one was VGG19, achieving 93% accuracy after fine-tuning. Other authors explored similar challenges such as the ones observed in [4] and [5].

## CNN

CNN stands for Convolutional Neural Network, which is a type of neural network that tries to imitate what happens in the biological processes of the cortex of animals, commonly used for image classification tasks. They are made up of neurons that have weights and biases, and CNN utilizes a technique called convolution, using a filter that "slides” through the images and extracts features out of them, in at least one of its layers. This produces an activation map, that represents the learned features, which then passes through pooling layers that downsample them to reduce the dimensionality. And the final layer is a dense, fully connected layer that will give us the output for the input image [7].

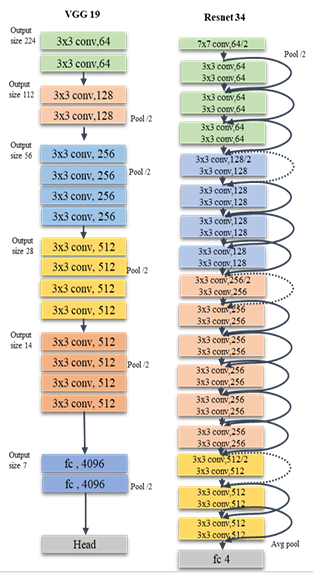
Fig 1. CNN layers example [7]

There are several popular CNN architectures used by researchers, such as AlexNet, VGGNet, ResNet, and Inception.

Fig 2. CNN architecture example [7]

## VGG19

Implemented by Simonyan and Zisserman [3] in 2015, this is a CNN architecture that consists of multiple convolutional layers followed by max-pooling, which captures progressively complex features through its layers [8]. Using transfer learning with pre-trained models has been shown to significantly reduce training time and improve performance, especially in domain-specific problems with limited datasets [3].

  
Fig 3. Structure of the VGG19 and Resnet 34 models [1]

# BACKGROUND

For this project, the work of Moreno and his team on the paper entitled “Poultry Egg Classification System Using Deep Learning” was used as a baseline to what I wanted my model to achieve, and for that I asked the paper team for their dataset, which they kindly gave me access to, since it would not be possible for me to gather by myself all that information in that short amount of time.

I used other papers to better understand different models used for that same task, or similar task, such as [3] and [4], to be able to better wright this report.

Another very important aspect of my project will be my personal experience in the years of the Veterinary Medicine graduation back in Brazil on UFSM (Federal University of Santa Maria), which gave me the opportunity to gather a lot of experience in the poultry field.

# APPROACH

After being able to run the codes for the base (frozen) VGG19 model, with a learning rate of 0.00001, I started running the rest of the codes for the fine-tuning (unfrozen) process. I got some problems with Google Collab and lost the previous outputs and had to re-run the whole base code again, it generated a slightly different output, with the accuracy of the base model going from the 53% (as stated in my midterm report) to ~65%, which should not be a problem. I used a pre-trained VGG19 on ImageNet for the base model.

The images were pre-processed in the same way as for the base model, image size of 224 by 224 as required for the VGG19 architecture [3], and a batch size of 64.

I recompiled the model, which was necessary after the trainable layers, using Adam as the optimizer, categorical cross-entropy for the loss calculation, and accuracy as the metric. The learning rate was set as 1e-5. Then I trained the model again with 5 epochs.

There was no need for image augmentation to be applied, the dataset already contained the augmented data. Training and validation accuracies are monitored to detect underfitting or overfitting. All experiments were conducted on Google Colab using GPU acceleration. The troubleshooting was done with Gemini and ChatGPT AI tools assistance.

# CODE REUSE

There was no code reusage on this project, all the codes were generated by ChatGPT.

# EXPERIMENT

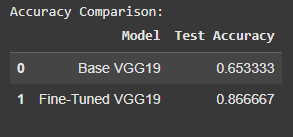
I started trying to run the codes ChatGPT gave me, and at each error that happened I stopped and used at first Google Colab’s integrated Gemini to troubleshoot, until the error was not happening anymore and the output seemed to be right, if Gemini could not help me with it, then I tried ChatGPT.

My dataset was kindly given by the team of the paper [1], and consists of a Google Drive folder with 3 different folders for the training, validation and test sets, containing a total of 1500 images of eggs that passed through a black box chamber with a light source underneath for the photo, and separated into 4 different categories (Healthy, Dirty, Cracked and Dirty, and Cracked), which were the ones that the model needed to classify at the end. These images passed through a process of data augmentation before being handed to me, as this was the method utilized in [1] to have a better number of samples.

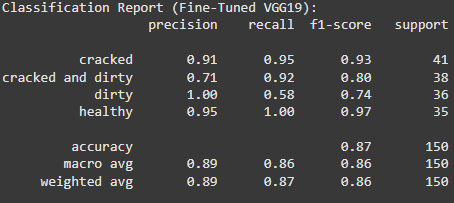
My evaluation parameters for the model, were the same as before for the base VGG19, them being based on [1]:

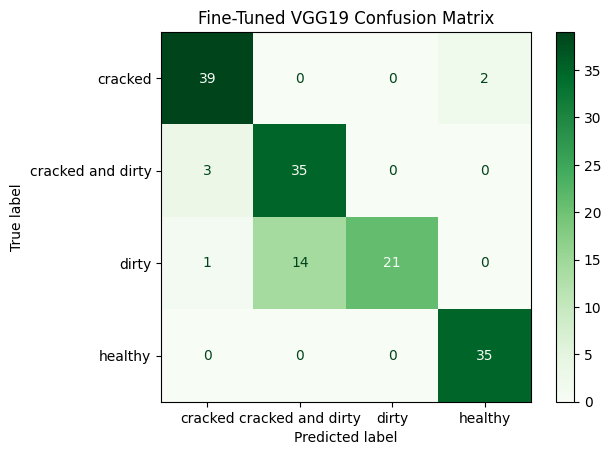
* Accuracy - the condition or quality of being true, correct, or exact;
* Precision - proportion of correct predictions out of all predicted instances for a class. High precision means few false positives;
* Recall - proportion of correct predictions out of all actual instances for a class. High recall means few false negatives;
* F1-score - harmonic mean of precision and recall. Useful when class imbalance exists.

As for the results themselves, I was able to reach an accuracy of ~87%, which was a good increase when compared with ~65% of the base model as per seem in Fig. 4, and also getting really close to the 93% that was achieved in the base paper [1].

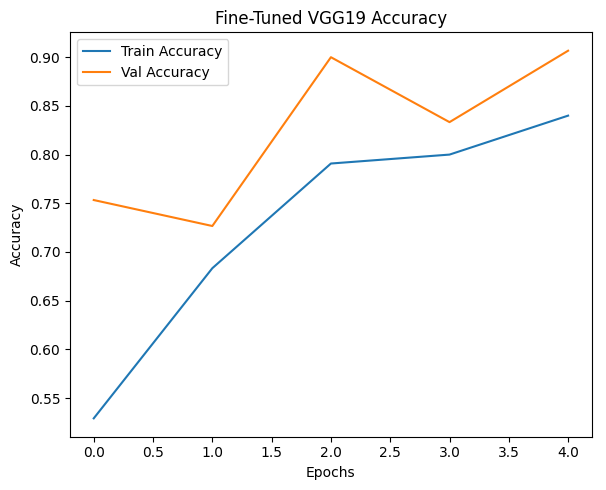
Fig 4. Accuracy comparison between base and fine-tuned VGG19 models

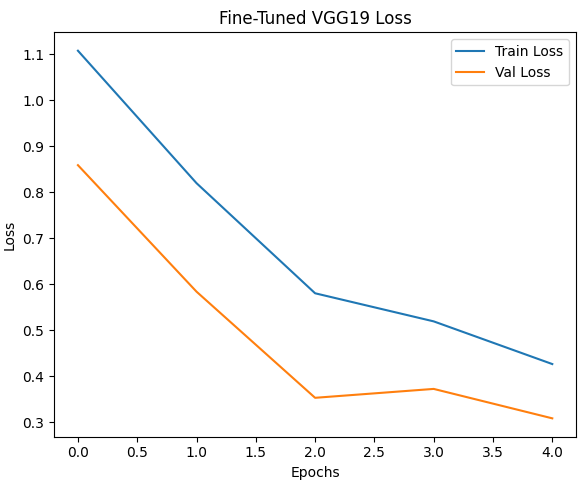
Got a weighted average of precision, recall and F1-scores parameters of, respectively, 0.89, 0.87 and 0.86 as shown in Fig. 5. This indicates that the model was able to correctly classify the eggs with few false positives as seen in the confusion matrix of Fig. 6, it is thorough and balanced, since performance across classes is stable, even though the classifications on the “dirty” class got more wrong classifications going to the “cracked and dirty” class, which make sense in a way because the classes have really similar features and part of them are actually the same as the “dirty” class so the confusion may be caused by that, possibly an even better feature extraction should be done on it in the future to try accounting for that, but I think it will not be a problem for now.

Fig 5. Classification report (Fine-Tuned VGG19)

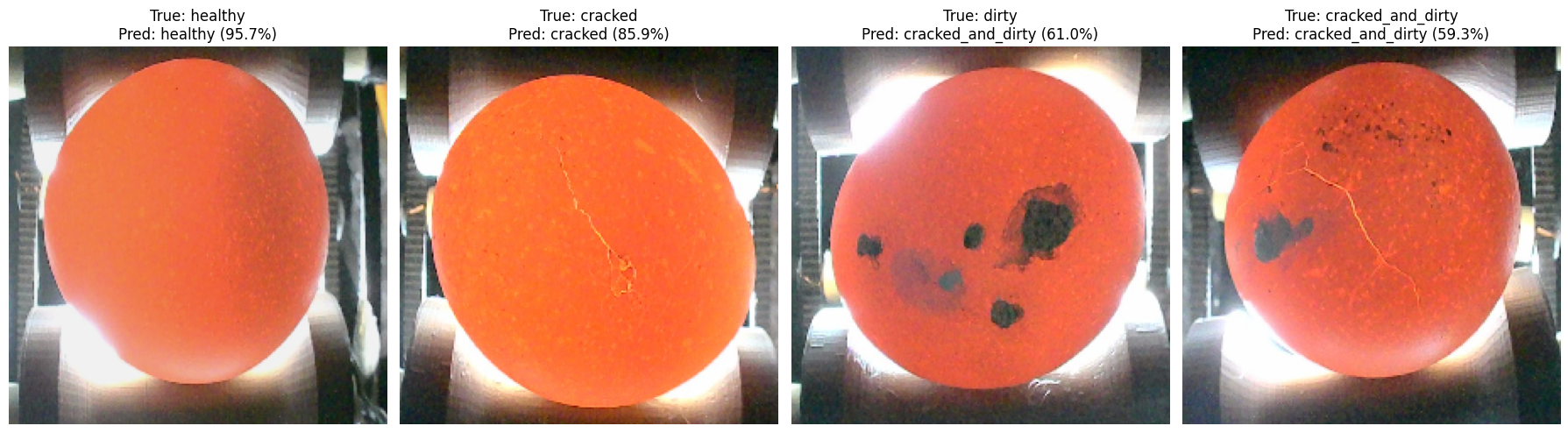
Fig 6. Fine-Tuned VGG19 Confusion Matrix

As we can observe from Fig. 7 and Fig. 8, the validation set outperformed the training set in terms of accuracy and loss, giving further support to the theory that the model worked as it should, without signs of overfitting.

Fig 7. Accuracy plot comparison between training and validation sets of Fine-Tuned VGG19

Fig 8. Loss plot comparison between training and validation sets of Fine-Tuned VGG19

I also did a sample test using the model to check how it was performing in the classification of some egg images of the test set with the results being observed on Fig. 9. I also tried to run the model using a random cracked egg image to see if it would be able to generalize to that and as seen in Fig. 10, the model seems to have a good ability to generalize and classify even eggs that are not from the dataset.

Fig 9. Egg images classifications using one image from the test set given by [1]

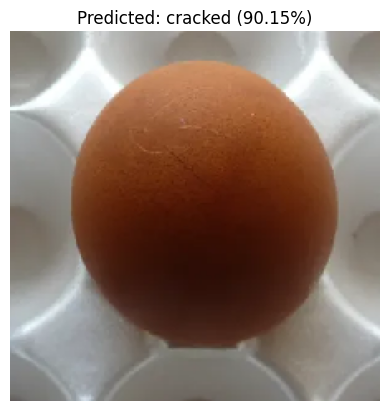


Fig 10. Egg image with a crack found online and classified using the implemented VGG19 model

One thing I noticed after having problems in the connection and losing the run on Google Colab was that, after I ran the training again to have the variables to work with in the later steps of the code, the evaluation parameters improved, the results I show in this report are from the second re-run of the codes, but after I ran a third time, to get to use a sample to test the model, the parameters actually improved again, which got me curious as to why that behavior was happening. In this third run I made the model reach an accuracy of 92.67%, pretty much the same as the original paper [1] researchers made it reach. I also changed some of the parameters, dropout from 0.5 to 0.6 and learning rate from 0.0001 to 0.00001, and adding a weight parameter for each class to balance, because the test images I was running the model with were being classified to the wrong class consistently, the model run with similar accuracy and parameters as for the second run, but I realized that the error was that I was preprocessing the test egg images in a different format then what I trained the model with at first.

All experiments were conducted on Google Colab using GPU acceleration.

# CONCLUSION

The model had a final accuracy of ~87%, and good evaluation parameters values, which make possible the assumption that the model I implemented was sufficiently successful in accomplishing its designated task. The final evaluation parameters give support for the idea that the model can classify the eggs in the different categories proposed, having a good generalization overall.

From implementing this model and writing this report, I was able to better understand what it truly means to work with artificial intelligence. It gave me a more profound vision of how a model operates and how I can play around with its parameters for achieving better results.

Looking ahead, I plan to apply what was learned in this class and with this project in ways that can be useful to my laboratory and to my personal research as well, specifically looking into oocysts counts (and possibly identification/differentiation) under the microscope, and also necrotic enteritis evaluations.

# DIVISION OF LABOR

I made everything by myself with the help of ChatGPT for generating the codes and the help of ChatGPT and Gemini with troubleshooting errors.

This project was made individually.

##### ACKNOWLEDGEMENT

I would like to give my sincere thanks to the professor of the class COMP6600, Dr. Aakur, for the opportunity and support throughout this semester, my orientator, Dr. Ruediger Hauck for the support and counseling, and Auburn University for the opportunity.

Participating in this class was an excellent experience and opened my eyes to the possibilities of AI in my field of research and I am willing to put them into practice!

##### REFERENCES

1. Sánchez, J. M. P., Moreno, L. A. O., Rodríguez, J. L. R., & Corro, I. D. M. (2023). Poultry Egg Classification System Using Deep Learning (pp. 1–6). *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE. <https://doi.org/10.1109/HORA58378.2023.10156776>
2. Dunkley C. S. A Dozen Egg Abnormalities: HOW THEY AFFECT EGG QUALITY, UGA Cooperative Extension Circular 1255, April 2022. Available at: < <https://secure.caes.uga.edu/extension/publications/files/pdf/C%201255_1.PDF>>
3. Simonyan K., Zisserman A. VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION, Published as a conference paper at ICLR, 2015. <https://doi.org/10.48550/arXiv.1409.1556>.
4. Victor Massaki Nakaguchi, R.M. Rasika D. Abeyrathna, Tofael Ahamed. Development of a new grading system for quail eggs using a deep learning-based machine vision system, Computers and Electronics in Agriculture, Volume 226, 2024, 109433, ISSN 0168-1699. <https://doi.org/10.1016/j.compag.2024.109433>.
5. Çelik A., Tekin El. Classification of Hatchery Eggs Using a Machine Learning Algorithm Based on Image Processing Methods: A Comparative Study, Brazilian Journal of Poultry Science, e-ISSN: 1806-9061 2024 / v.26 / n.2 / 001-012. <http://dx.doi.org/10.1590/1806-9061-2023-1882>.
6. KHABISI, M. M, SALAHI, A, & MOUSAVI, S. N (2012). The influence of egg shell crack types on hatchability and chick quality. Turkish Journal of Veterinary & Animal Sciences 36 (3): 289-295. <https://doi.org/10.3906/vet-1103-20>
7. Modi, P. Convolutional Neural Networks for Dummies, Medium, 2023. Available at: <https://medium.com/@prathammodi001/convolutional-neural-networks-for-dummies-a-step-by-step-cnn-tutorial-e68f464d608f>. Accessed: 07/23/2025.