Identifying common stored product insects using automated deep learning methods

Chetan Badgujar[a,1], Paul R Armstrong[a], Alison Gerken[a], Lester O. Pordesimo[a], James Campbell[a]

[a]*Stored Product Insect & Engineering Research Unit, USDA-Center of Grain & Animal Health Research, Manhattan, KS 66502*

Abstract

Monitoring for stored product insect pests is a common practice for post-harvest management of stored grain and grain-based commodities, which helps ensure product quality from harvest to final consumer. Current methods of sampling and monitoring can be time-consuming, labor-intensive, expensive and require expertise in insect identification. Therefore, this study aims to develop an image-based automated identification system for common stored product insect species using deep-learning methods. Top-down images of adults of the common stored product insect species *Rhyzopertha dominica*, *Cryptolestes ferrugineus*, *Tribolium castaneum*, *Sitophilus oryzae*, and *Oryzaephilus surinamensis* were acquired and analyzed. Deep learning-based, state-of-the-art Convolutional Neural Networks (CNN) models (ResNet-50, MobileNet-v2, DarkNet-53, and EfficientNet-b0) were fine-tuned with a transfer learning approach to classify the insect species. All models were able to correctly identify the species of the insect with at least 96% accuracy and with few misclassifications. One issue with trained CNNs is that they do not explain the reasoning for the classification and are often called a "black box". Therefore, visualization methods called Gradient-

[1] Corresponding author
Email address: chetan19@ksu.edu (Chetan Badgujar)

weighted Class Activation Mapping (Grad-CAM) were implemented to explore the black box network. The Grad-CAM uses heat maps to highlight the major image features that the network focused on to make insect species predictions. The Grad-CAM verifies the network's prediction and also helps improve network performance. This study contributes to the overall goal of developing a camera-based system for monitoring stored grain insects. The developed system would empower warehouses, flour mills, and other food facilities with a tool to quickly and accurately identify insect species in stored product environments and could be implemented as part of a close to real-time monitoring system.

Keywords: Deep learning, Transfer learning, Insects, Grain storage, Imaging.

---

1. Introduction

Integrated pest management is a holistic approach for stored product insect management that utilizes many tactics to manage insect pests most effectively. Insect monitoring is a critical part of effectively implementing integrated pest management procedures. Effective monitoring requires accurate and timely identification of insect species in order to determine the level of risk and the most appropriate response. Current insect sampling methods used in stored product environments (both in bulk storage such as a grain bin and in food facilities such as mills warehouse) require considerable labor to implement, expertise to accurately identify, and don't provide near real-time information [1]. Traditional monitoring methods for bulk grain can include tactics like a pitfall probe trap which is a long-cylindrical tube with multiple holes where insects enter and fall into a collection cup [2]. Probes are manually monitored by extracting the probe and identifying insects at scheduled intervals of 3-7 days. An electronic version of the pitfall probe trap was developed by Shuman et al. (1996) [3] and later commercialized to count insects and was later refined to identify species [4]. The electronic probe fundamentally uses insect size to

discriminate between species which can lead to difficulty differentiating between the species like *R. dominica* (an internal feeder) and *T. castaneum* (an external feeder) due to similar dimensions. However, it can provide real-time information on insect activity more quickly and easily compared to traditional bulk grain monitoring methods [5].

For other food facilities such as mills, processing plants, warehouses, and retail stores insect monitoring, detection, and identification is primarily by use of traps such as pheromone or kairomone baited traps or light traps, and visual inspection. A visual inspection involves regular and periodic observation of the facility, and the locations where insects are likely to occur. The pheromone or kairomone baited traps include pheromones, synthetic compounds based on the naturally produced pheromones used by the insects as chemical messengers, or kairomones which are food-based volatile compounds, which can attract specific insects to the traps. Light traps use specific wavelengths of light to attract insects. The traps are inspected periodically by professionals, and trapped insects are typically identified in the field by these trained professionals. In some cases, further inspection and analysis of the captured insects need to occur using higher magnification or molecular techniques. These methods are complex, time-consuming, costly, labor-intensive, and often require advanced taxonomic training [6, 7]. It is also not easy to obtain real-time detection of insect activity since traps or inspections typically occur at regular intervals. Hence, there is an urgent need for a rapid, automated system to identify and classify the most common stored product insects to help make pest management decisions.

Recent advancements in sensor and electronics technology have delivered miniature, low-cost cameras and imaging components. The availability of a basic camera and other electronic components makes it increasingly possible for in-situ imaging to be employed to identify insect species. In recent decades, there has been considerable research on the automated identification of agricultural

and biological objects such as fruits [8], vegetables [9], trees [10], trunks, leaves [11, 12], and insects [13, 14, 15]. Deep learning-based models are becoming increasingly popular for image-based data. There are research studies where deep learning-based models were developed to identify stored product insects [16, 17, 18, 19, 20, 21]. In most of these studies, the particular environment of operation or objectives help define the degree of model or image sophistication required to meet the specific needs. However, identifying insect species is relatively challenging because of their minute size, scale (compared to other objects such as grains), the visual complexity of the background, and diversity within the species.

Deep learning-based networks are at the forefront of solving a computer vision problem. The deep learning (DL) models process or train on a large number of labeled images and enable the computer to recognize and understand the visual information with a fast, robust, and high prediction accuracy. DL models do not require manual feature engineering but can learn complex patterns or features from raw labeled image data. The DL models have achieved state-of-the-art performance on different computer vision tasks, including image classification, object detection, and image segmentation [22]. However, DL networks are often described as a "black box" because it is difficult to understand how networks make predictions. The DL network includes multiple neurons and interconnected layers trained to learn complex image features. The network parameters are iteratively adjusted during the training, which makes it challenging for humans to comprehend how input is related to output. In order to develop an automated insect image identification system, it is necessary to translate network behavior and prediction into a human-interpretable form so that a person can interpret the network's prediction (output) and understand how specific insect features from input images are mapped to a particular prediction.

The specific objective of this study is to develop automated identification of common stored product insect species using a deep learning method. Automatic insect identification would address a critical challenge of pest management in stored product environments. It would also serve as a decision-making tool that could provide an effective and quick solution for managers in food facilities such as mills and warehouses to identify insects and take necessary management actions. Another goal of the study is to translate the black box network to human-interpretable output by visualization methods. This would answer several questions on how the network is making predictions and would help explore areas of improvement to enhance network performance.

2. Materials and Methods

2.1. Stored Product Insects

The stored product insects used here are commonly observed species in stored grain or flour that are morphologically variable. The following five species were assessed: (1) *Rhyzopertha dominica* (lesser grain borer, LGB), (2) *Cryptolestes ferrugineus* (rusty grain beetle, RGB), (3) *Tribolium castaneum* (red flour beetle, RFB), (4) *Sitophilus oryzae* (rice weevil, RW), and (5) *Oryzaephilus surinamensis* (saw-toothed grain beetle, ST). Two of the insect species (RW and LGB) are primary pests that can feed on intact grains, and the other three species (RGB, RFB, and ST) are secondary pests that impact damaged or processed grains. Adults were selected for identification since early developmental stages (eggs, larvae, and pupae) are challenging to identify based on morphology and are less likely to be detected in a trap or to be highly mobile. Adult insect samples were obtained from laboratory colonies at the USDA-ARS Center of Grain and Animal Health Research, Manhattan, KS, US.

2.1.1. Data Acquisition & Preprocessing

The insect images were captured with a digital camera (Model: SZ-31Mr iHS, Olympus America, Center Valley, PA) at a focal length of 250 mm using a 50 mm lens. The image resolution with this setup was determined to be 83 pixels/mm. Approximately 10-20 insects of the same species were placed on a glass Petri dish (70 mm diameter) that was painted white. The insects were allowed to move on the dish. The insects were replaced after 4-5 images to provide data variability within the species, although the movement of the same insect created significant image variability. Individual insect images were then hand cropped from these larger images for analysis. A total of 100 images were used for each species, which resulted in a total of 500 insect images used. The images were stored and processed as 24-bit bitmap images. Sample images for each insect species are shown in Figure 1. This study followed a top-down view approach for insect image data collection, which may not necessarily be the best approach for a bulked grain system, but does align with how insects will be orientated in systems targeting insects in food facilities such as mills and warehouses and it can generate quick, simple, and consistent images for baseline model analysis.

Figure 1: Sample images for each insect species in the study.

2.1.2. Image Data Augmentation

The DL network requires large, high-quality, and diverse training data. Data augmentation is a powerful and commonly used technique in computer vision to increase the size and diversity of the training dataset. It improves the network's

performance, robustness, and generalization on unseen data [23, 24]. The data augmentation was performed on collected insect images where geometric image transformations such as rotation, scaling, and shearing were randomly applied to the training data alone. Table 1 describes the implemented data augmentation methods. The augmented images simulate the variation and distortion in image acquisition, and sample augmented images generated from the image are shown in Figure 2. Data augmentation prevents model to learn or memorize the exact details of training images which often prevents overfitting.

Table 1: Implemented data augmentation methods.

Transform	**Description**	**Range**
Rotation	Image rotation by angle, Fig 2(b)	-45 to 45°
Scale	Image resize by scale factor, Fig 2(c)	1.2 to 1.5
Shear (vertical)	Vertical image shear by angle, Fig 2(d)	-30 to 30°
Shear (horizontal)	Horizontal image shear by angle, Fig 2(e)	-30 to 30°

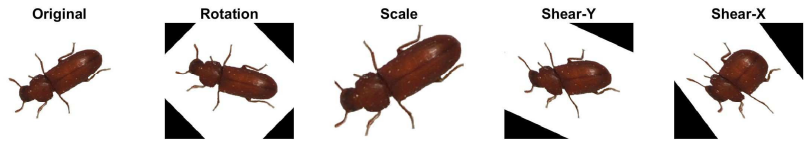

Figure 2: Illustration of data augmentation applied: (a) original sample image, (b) rotation, (c) scale, (d) vertical shear in Y-axis, and (e) horizontal shear in X-axis.

2.2. Deep Learning Networks

An image classification task assigns an image category to an image based on its content. Image classification is challenging since images can vary widely in content, color, size, and orientation. Moreover, there could be a significant overlap between image categories. However, image classification is at the core of computer vision problems and the first step towards more advanced problems such as object detection and segmentation. In recent years, the DL network has

achieved notable results in the image classification task, and Convolutional Neural Networks (CNNs) are the most commonly used type.

CNN is a type of feed-forward neural network that can handle large data efficiently and extract features from the raw images. The standard CNN architecture and working principle are shown in Figure 3. The CNN architecture is made of multiple convolutional, pooling, and fully connected layers that are further categorized into feature extraction and classification task. The insect images are inputs to the feature extraction, where multiple convolutional layers extract spatial features (e.g., edges, textures, and shape) from the image. The pooling layer down samples the feature maps by reducing the number of parameters and making it computationally efficient. The output of feature extraction is flattened and fed to a fully connected layer, which learns to perform a classification task from extracted features. During CNN training, the network adjusts the learnable parameters to minimize the difference between ground truth and predicted labels. The trained network can be used to classify new images. However, CNN is often trained on a large, labeled image dataset, creating a bottleneck for building a new classifier for a new task.

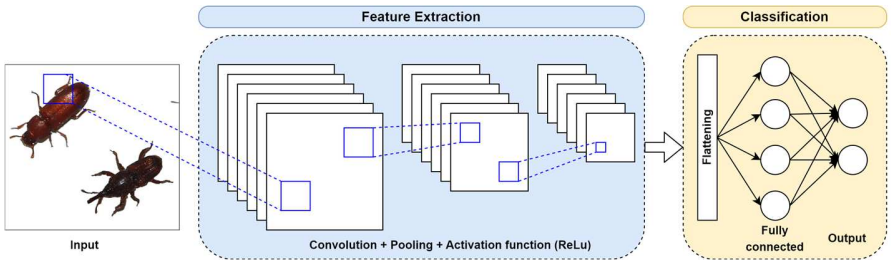

Figure 3: Convolutional Neural Networks (CNNs) architecture for insect image classification. Spatial features from insect images are extracted with convolutional layers. Pooling then reduces the number of parameters (down samples). The output is the trained network for new image input classification.

2.2.1. Transfer Learning

In general, DL networks (specifically CNN) require a relatively large and diverse dataset for training. Generating such datasets is a major constraint for exploiting the potential of DL networks. Therefore, in recent years, the transfer learning approach has become widely used in DL-based computer vision problems [25, 26, 27]. Transfer learning leverages the gained knowledge on one task, trained on a large dataset, to improve the network performance on another task that is new, similar, and has a smaller dataset. In other words, a pretrained CNN is used as a starting point for a new network, and during training, the initial pre-trained layer's weights are frozen while the last few layers are retrained on the new dataset. The reused frozen pre-trained layers (trained on an original large dataset) extract the most general features of images, such as edges, shapes, and textures which benefit in the training of new networks. Transfer learning offers the following benefits: (1) eliminates the dependency of large training data (to train a new model), (2) improves network performance (accuracy and efficiency) on relatively smaller datasets, and (3) faster and easier network training with limited computing resources.

2.2.2. Pre-trained CNN Models

Pre-trained networks are trained on millions of images (e.g., ImageNet database) and can extract powerful and informative features from images. Therefore, it is common to implement a pre-trained network as a starting point and fine-tune a network with a transfer learning method for a new task, which is faster and easier than building and training a new network. In recent decades, pre-trained networks have become very popular and commonly used in computer vision tasks. More than a dozen networks are available that are fast, accurate, and robust for solving varying levels of computer vision tasks. VGG, ResNet, Inception, DarkNet, and efficientNet are the most popular pre-trained CNN models. The pre-trained networks are

distinct and have different characteristics, such as speed, accuracy, and size, which influence network selection for specific tasks.

The pre-trained networks are trained to predict specific tasks, for example, image classification. The initial layers of the network carry the general features learned from the original task. These features can be reused for the new task and can often freeze (weights not updated during the training) to ensure that the general features are not altered during the training. Freezing initial layers can help networks generalize better to new data and significantly speed up network training. Moreover, freezing the initial network layers can prevent overfitting, particularly with a small data set. The weights in the later layers are updated during the training and are more specific to the new task. The network's final layers (learnable convolution and classification layers) are designed to output a specific number of classes relevant to the trained task. The generated insect dataset has five classes; therefore, the final layers of the pre-trained network are replaced to classify new insect images.

There are also several variations of CNNs that can be used for image classification. This study selected the most popular pre-trained network to classify the stored-product insect images. These pre-trained networks are as follows (Table 2): (a) ResNet-50 [28], (b) MobileNet-v2 [29], (c) DarkNet-53 [30], and (d) EfficientNet-b0 [31].

Table 2: The parameters of pre-trained CNN models.

Network	**EfficientNet-b0**	**ResNet-50**	**MobileNet-v2**	**Inception-v3**
Input image size	224×224	224×224	224×224	299×299
Depth (layers)	82	50	53	48
Size (MB)	20	96	13	89
Parameters (MM)	5.3	25.6	3.5	23.9

2.2.3. Network Training & Evaluation

CNN model training is an iterative process where the network weights and biases are adjusted with a backpropagation algorithm to minimize a loss

function. The model training aims to optimize the network performance, and the workflow diagram of network training is shown in Figure 4. The insect dataset included 500 images which were randomly distributed into the train (70%) and validation (30%) sets. The data processing involved pre-processing and augmentation operations. In pre-processing steps, images in the train and validation set were resized to adopt the network input layer, as defined in Table 2. The data augmentation was applied to the training set only (Table 1 and Figure 2). During training, the augmented images help the network prevent overfitting and memorizing the same information. Next, the pre-trained CNN was selected for network fine-tuning, which included: (1) replacing the final classification layer to match the output of the new classification task and (2) freezing initial layers weights (initial ten layers) by setting learning rates to zero. The training options or hyperparameters, such as batch size, learning rate, training algorithm, and data shuffle, were tuned in order to optimize the network performance (Table 3). The regularization methods, such as early stopping, were implemented to prevent network overfitting. Hence, the maximum number of epochs was set to 20. During the training, the network learning curves were closely observed, and the trained network performance was evaluated in terms of network accuracy and confusion matrix on a validation set.

The above process was repeated for the four selected pre-trained networks. The best-performing network was saved and further deployed to classify the insect image label class and class probability. The model training was carried out in MATLAB (The MathWorks Inc., Natick, MA, US) environment with the Deep Learning Toolbox. The network was trained on NVIDIA GeForce GTX 1650 with 4 GB of memory.

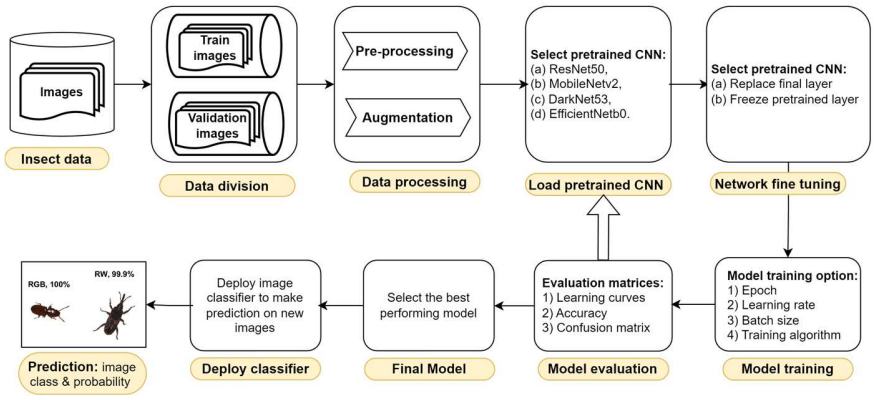

Figure 4: Workflow of automated insect image classification network.

Table 3: Network training parameters used in the study.

Parameters	Description	Value
Algorithm	Stochastic gradient descent with momentum	-
Mini batch	Data subset to update weights & gradient	16 images
Max. epochs	Maximum number of epochs	20
Data shuffle	Train & validation shuffle frequency	1 epoch
Learning rate	Training learning rate	0.0003
Validation freq.	Iterations for network evaluation	21
Validation patience	Early stopping criteria to stop training	5 epochs

2.3. Investigation of Network Predictions

Because the network created is somewhat of a black box, visualization methods were used to understand the inner workings of the network by highlighting the prominent image features that the network focuses on to make predictions. Class activation mapping (CAM) is a visualization method that provides a heat map on CNN predictions. The heat map shows the input image regions that contributed the most to the predicted class. This study used a Grad-CAM (Gradient-weighted Class Activation Mapping), an improved CAM method.

The visualization method was applied, post-training, on the validation images and used only for the best final network. The following steps were used to generate Grad-CAM for CNN:

1. The last convolutional layer of trained CNN was selected because it extracts the high-level features from the input image.
2. Input image was forward passed through the trained network to make a prediction, i.e., class label and class probability.
3. The output gradient was backpropagated with respect to the layer's feature maps. It computes the importance of each feature map in network output.
4. The weighted sum of feature maps was computed to generate a CAM that highlights the most important image regions used for prediction.
5. The Grad-CAM was applied to compute the prediction gradient with respect to feature maps and take a global average. This results in gradient weighted feature maps and generates heatmaps. The heatmap highlights the regions of the input image that contributed to the decision-making process with the higher intensity colors (e.g., reds) indicating higher importance.
6. The heatmap was overlaid on the input image to generate a Grad-CAM visualization.

The mathematical details or equations underlying Grad-CAM visualization were given in Selvaraju et al. (2017) [32]. The human interpretable heatmap generated by visualization methods addresses several questions on network performance, training data, debugging, and verification.

3. Results & Discussion

3.1. Deep Learning Network Evaluation

The four pre-trained CNN models were fine-tuned to classify the five common types of stored product insects selected in the study. The network learning curves typically includes the loss (train and validation) and accuracy (train and validation) change for a given number of training epochs or iteration. The learning curves for each network were shown in Figure 5, which explains the network learning performance and provides insight into the training process. The accuracy and loss were computed on the training data batch, and network parameters were optimized to reduce the loss, further improving the network accuracy. During training, the loss decreased consistently with each iteration before plateauing. However, the loss declined rapidly (≤50 iterations) for ResNet50 and Mobilenet-v2. The validation learning curves (loss and accuracy) followed the training learning curves.

The early stopping parameters monitored the network performance and stopped the training when the network performance degraded or was unchanged on the validation set for a predefined number of iterations (105). Therefore, the training of ResNet-50 was stopped after 373 iterations (Figure 5) since the network performance was starting to degrade on a validation set. The early stopping prevented the network overfitting on training data and helped network generalization on new data.

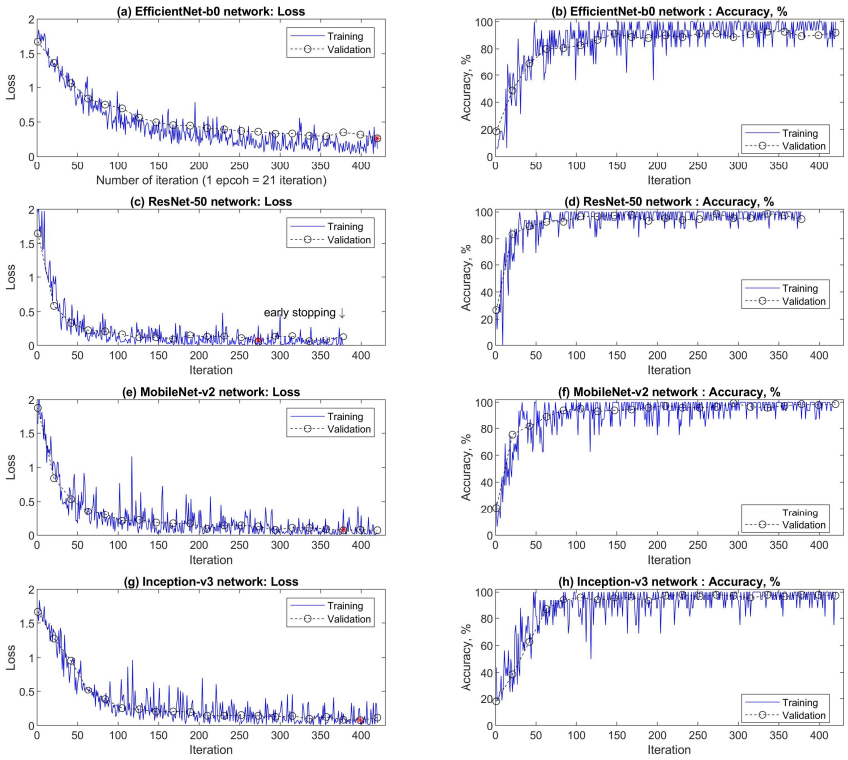

Figure 5: Learning curves of CNN training for insect image classification task. The red point indicates the lowest validation loss.

Table 4: Performance summary of pre-trained CNN in stored product insect identification.

	Loss		**Accuracy**		**Training**		**Overall accuracy (%)**
Network	Train	Val.	Train	Val.	Time	Iteration (Epoch)	
EfficientNet-b0	0.20	0.26	93.45	92.00	8m 24s	420 (20)	97.33
ResNet-50	0.08	0.05	97.32	98.67	5m 14s	273 (13)	99.3
MobileNet-v2	0.06	0.07	98.21	98.67	3m 45s	378 (18)	96.67
Inception-v3	0.10	0.08	96.13	98.00	8m 45s	399 (19)	99.33

The marked red point on the learning curves showed the neural network weights corresponding to the training iteration with the lowest validation loss (Figure 5). These weights were saved and used for making future predictions. The network performance summary was shown in Table 4. The overall classification accuracy was computed on a validation set and ranged between

96.7 to 99.3% for the selected networks. This suggests that the networks performed very well and correctly classified the insect images.

The confusion matrix is another way to investigate the network's performance, and it shows the correct and incorrect network prediction across different image classes. The confusion matrices for the different trained networks were shown in Figure 6. The diagonal entries represented accurate predictions for each category, while off-diagonal entries were misclassified. Each selected pre-trained CNN performed well and accurately classified each insect image class. The confusion matrix also identified the image class the model struggled to predict accurately. For example, the Mobilenet-v2 network was struggling to classify accurately between LGB and RW classes. For future predictions, ResNet-50 was selected since it was relatively fast and accurate for classifying insect images. The sample predicted images classified using the pre-trained Resnet-50 are shown in Figure 7.

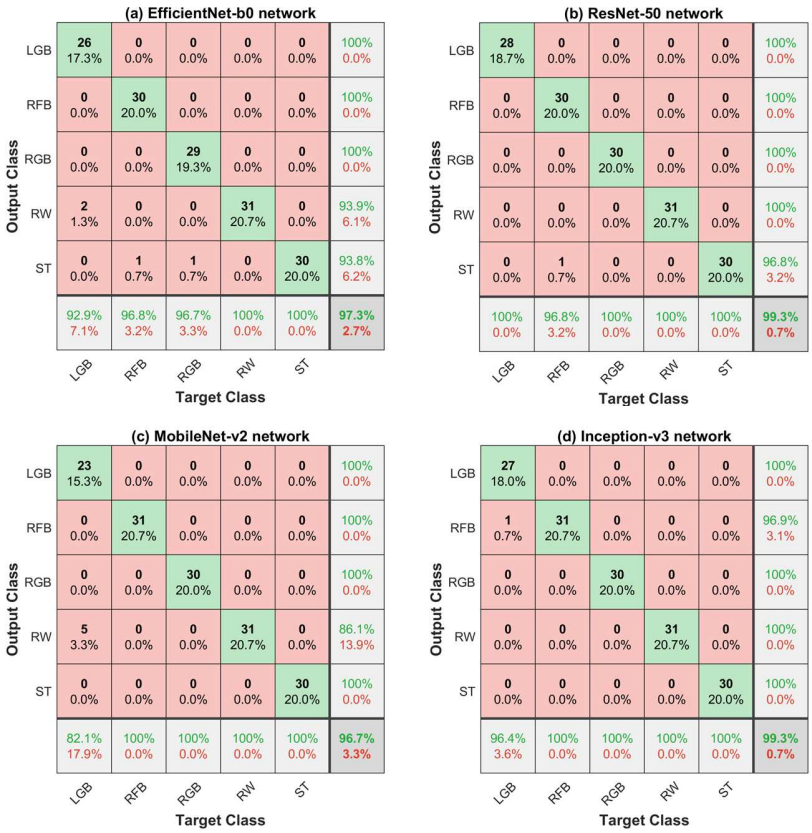

Figure 6: Confusion chart of a pre-trained network (validation set image). Diagonal entries (green) were correctly classified. Off-diagonal entries (red) were misclassified.

Figure 7: Sample predicted insect image classification by Resnet-50 with prediction probability.

3.2. Network Predictions Analysis

Grad-CAM visualization method was used for exploring the black box network. The generated sample output for each insect class was shown in Figure

8. The heat map showed the image region used for making the prediction. The different parts of insects were highlighted or identified by the network to distinguish between the insects. Although the networks may appear to have missed certain key insect features, such as the snout on an RW, most of the key features highlighted by the Grad-CAM analysis focused on the body shape of the insects and the abdominal and thoracic connections that differ between the insects studied here. For example, the RW is highly convex in body shape with elliptical-shaped pits on the thorax (pronotal disc), save for a smooth strip down the center, with dull coloration on the thorax and elytra (wing covers) [33, 34]. The network also focused on the more continuous body shape from the abdomen to the thorax in the RW. For the RFB, the oval shape of the lower body was a distinguishing feature, along with the lengthwise ridges of the elytra. Between the elytra is also a longitudinal ridge that helped distinguish RFB from the other insects [33, 34].

For LGB, there are two or three areas that the Grad-CAM highlighted as important to insect identification which are also important morphological characteristics: the blunt-end of the elytra, the pitted elytra, and the square scutellum (at the top of the elytra near the thorax). The more cylindrical shape of the body also lends to a key feature of the LGB and also helps distinguish the pitted elytra from the RW [33, 34]. Key features for ST identification were the classic saw-toothed pattern on the thorax (pronotum) along with three longitudinal ridges on the thorax [33, 34]. The connection between the thorax and abdomen was quite thin as well, which was also highlighted in the Grad-CAM output. Finally, the flattened shape of the RGB, along with a raised longitudinal line on the pronotum, were identified as key features [33, 34]. The head was also another key feature, with a more flattened beak-like shape and long, thin antennae near the eyes.

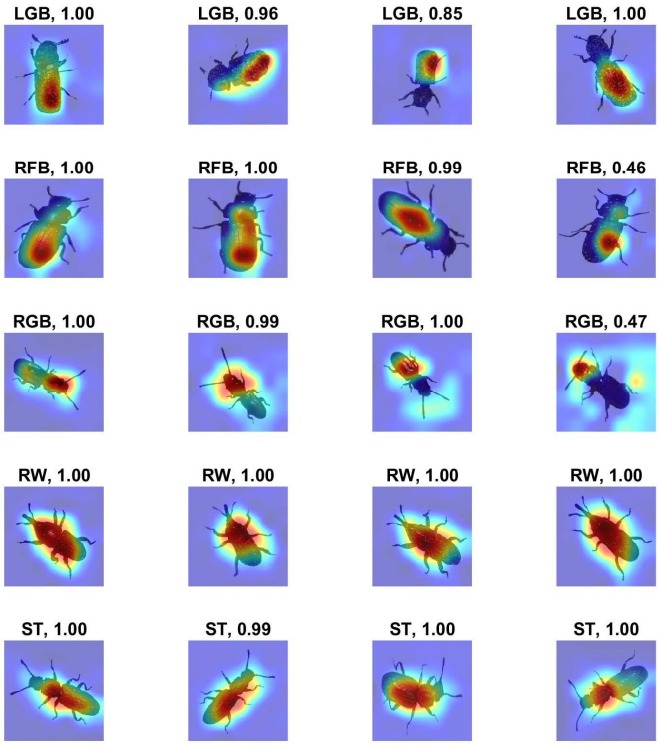

Figure 8: The network visualization with Grad-CAM and generated output for sample images (The number denotes the class prediction probability).

4. Conclusions

In this study, the most popular pre-trained CNNs were fine-tuned to classify the five species of stored grain insect species. The fine-tuned networks performed very well on the collected insect image dataset, and the prediction accuracy ranged between 96 to 99%. The network accurately distinguished between insects that geometrically share common features, such as ST and RGB, by highlighting more distinguishing features, such as the head shape for RGB and the thoracic sawtooth patterns on the ST. Using a pre-trained network model as a

starting point dramatically reduces the model development time and even performs well on a relatively small data set. The network visualization methods demonstrate the network prediction outcome and are a great tool for improving network performance.

Overall, the study provides an excellent preliminary evaluation of one method for in-situ monitoring of insects in stored grain. This automated framework would serve as a fundamental building block for the decision-making tool for pest management in food facilities. Moreover, with more refinement and research, this can be easily integrated into a more advanced automated insect detection platform. As a next step, will need to evaluate other stored product insect species, some of which are very morphologically similar to the species already evaluated. Most food facilities have a wide range of insects that can be observed, and how the decision-making tool works with morphologically similar species or novel species needs to be considered. Other potential limitations of this approach will also need to be addressed in the future. A miniaturized, low-cost imaging system of similar or greater pixel resolution needs to be developed and integrated into a monitoring system. The instrument must deal with a dirty environment; both the background and the insects themselves may have particulate material on them, and the system will need to provide suitable lighting for image capture. There may also be a problem with multiple counts of the same insect if they are able to move in and out of the field of view.

Declaration of Competing Interest

There is no conflict of interest to describe regarding this manuscript.

Acknowledgement

We thank Oak Ridge Institute of Science and Education for fellowship assistance. The research was supported in part by appointment to the Agricultural Research Service (ARS) Research Participation Program

administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and the U.S. Department of Agriculture (USDA). ORAU manages ORISE under DOE contract number DE-SC0014664. All opinions expressed in this paper are the author's and do not necessarily reflect the policies and views of USDA, DOE, or ORAU/ORISE. Mention of trade names or commercial products in this publication is solely for the purpose of providing specific information and does not imply recommendation or endorsement by the U.S. Department of Agriculture. USDA is an equal-opportunity employer and provider.

CRediT

Chetan Badgujar: Conceptualization, Methodology, Data collection, Data curation, Software, Writing- Original draft preparation. Paul Armstrong.: Visualization, Investigation, Supervision, Writing- Original draft preparation. Alison Gerken.: Data curation, model building, and Validation, Supervision, Reviewing, and Editing. Lester O.Pordesimoa.: Supervision, Reviewing and Editing. James Campbel.: Conceptualization, Reviewing, and Editing.

References

[1] Hagstrum DW, Athanassiou CG. Improving Stored Product Insect Pest Management: From Theory to Practice. Insects 2019;10(10):332. doi:10.3390/insects10100332.

[2] Epsky ND, Morrill WL, Mankin RW. Traps for Capturing Insects. Dordrecht: Springer Netherlands. ISBN 978-1-4020-6359-6; 2008, p. 3887–901. doi:10.1007/978-1-4020-6359-6_2523.

[3] D. Shuman , J. A. Coffelt , D. K. Weaver . A Computer-based Electronic

Fall-through Probe Insect Counter for Monitoring Infestation in Stored Products. Transactions of the ASAE 1996;39(5):1773–80. doi:10.13031/2013.27653.

[4] Shuman D, Weaver DK, Larson RG. Performance of an Analytical, Dual Infrared-Beam, Stored-Product Insect Monitoring System. Journal of Economic Entomology 2005;98(5):1723–32. doi:10.1093/jee/98.5.1723.

[5] Flinn PW, Opit GP, Throne JE. Predicting Stored Grain Insect Population Densities Using an Electronic Probe Trap. Journal of Economic Entomology 2009;102(4):1696–704. doi:10.1603/029.102.0438.

[6] Mueller DK, VanRyckeghem A. Chapter 13 - pheromones for stored product protection. In: Heaps JW, editor. Insect Management for Food Storage and Processing (Second Edition); second edition ed. American Associate of Cereal Chemists International; AACC International Press; 2006, p. 153–64. doi:https://doi.org/10.1016/B978-1-891127-46-5.50018 -0.

[7] Yalcin H. Vision-based automatic inspection of insects in pheromone traps. In: 2015 Fourth International Conference on Agro-Geoinformatics. 2015, p. 333–8. doi:10.1109/Agro-Geoinformatics.20 15.7248113.

[8] Sa I, Ge Z, Dayoub F, Upcroft B, Perez T, McCool C. Deepfruits: A fruit detection system using deep neural networks. Sensors 2016;16(8). doi:10.3390/s16081222.

[9] Suryawanshi Y, Patil K, Chumchu P. VegNet: Dataset of vegetable quality images for machine learning applications. Data in Brief 2022;45:108657. doi:10.1016/j.dib.2022.108657.

[10] Grondin V, Fortin JM, Pomerleau F, Giguere P. Tree detection and diameter estimation based on deep learning. Forestry: An International Journal of Forest Research 2022;96(2):264–76. doi:10.1093/forestry/cpac043.

[11] Karlekar A, Seal A. SoyNet: Soybean leaf diseases classification. Computers and Electronics in Agriculture 2020;172:105342. doi:10.1016/j.compag.2020.105342.

[12] Badgujar C, Mansur H, Flippo D. Pest-infested Soybean Leaf Image Classification with Deep Learning Techniques for Integrated Pest Management (IPM). In: ASABE AIM 2022 Houston, Texas July 17-20, 2022. doi:10.13031/aim.202201096.

[13] Kirkeby C, Rydhmer K, Cook SM, Strand A, Torrance MT, Swain JL, et al. Advances in automatic identification of flying insects using optical sensors and machine learning. Scientific Reports 2021;11(1):1555. doi:10.1038/s41598-021-81005-0.

[14] Li W, Yang Z, Lv J, Zheng T, Li M, Sun C. Detection of Small-Sized Insects in Sticky Trapping Images Using Spectral Residual Model and Machine Learning. Frontiers in Plant Science 2022;13.

[15] Teixeira AC, Ribeiro J, Morais R, Sousa JJ, Cunha A. A Systematic Review on Automatic Insect Detection Using Deep Learning. Agriculture 2023;13(3):713. doi:10.3390/agriculture13030713.

[16] Shen Y, Zhou H, Li J, Jian F, Jayas DS. Detection of stored-grain insects using deep learning. Computers and Electronics in Agriculture 2018;145:319– 25. doi:10.1016/j.compag.2017.11.039.

[17] Wu L, Liu Z, Bera T, Ding H, Langley DA, Jenkins-Barnes A, et al. A deep learning model to recognize food contaminating beetle species based on elytra fragments. Computers and Electronics in Agriculture 2019;166:105002. doi:10.1016/j.compag.2019.105002.

[18] Shi Z, Dang H, Liu Z, Zhou X. Detection and Identification of stored Grain Insects Using Deep Learning: A More Effective Neural Network. IEEE Access 2020;8:163703–1. doi:10.1109/ACCESS.2020.3021830.

[19] Barboza da Silva C, Silva AAN, Barroso G, Yamamoto PT, Arthur V, Toledo CFM, et al. Convolutional Neural Networks Using Enhanced Radiographs for Real-Time Detection of *Sitophilus zeamais* in Maize Grain. Foods 2021;10(4):879. doi:10.3390/foods10040879.

[20] Li J, Su Y, Cui Z, Tian J, Zhou H. A Method to Establish a Synthetic Image Dataset of Stored-Product Insects for Insect Detection. IEEE Access 2022;10:70269–78. doi:10.1109/ACCESS.2022.3188282.

[21] Mendoza QA, Pordesimo L, Neilsen M, Armstrong P, Campbell J, Mendoza PT. Application of Machine Learning for Insect Monitoring in Grain Facilities. AI 2023;4(1):348–60. doi:10.3390/ai4010017.

[22] Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data 2021;8(1):53. doi:10.1186/s40537-021-00444-8.

[23] Shorten C, Khoshgoftaar TM. A survey on Image Data Augmentation for Deep Learning. Journal of Big Data 2019;6(1):60. doi:10.1186/s40537-0 19-0197-0.

[24] Yang S, Xiao W, Zhang M, Guo S, Zhao J, Shen F. Image Data Augmentation for Deep Learning: A Survey. Arxiv 2022;doi:10.48550/ARXIV.220 4.08610.

[25] Lu J, Behbood V, Hao P, Zuo H, Xue S, Zhang G. Transfer learning using computational intelligence: A survey. Knowledge-Based Systems 2015;80:14–23. doi:10.1016/j.knosys.2015.01.010.

[26] Weiss K, Khoshgoftaar TM, Wang D. A survey of transfer learning. Journal of Big Data 2016;3(1):9. doi:10.1186/s40537-016-0043-6.

[27] Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, et al. A Comprehensive Survey on Transfer Learning. Proceedings of the IEEE 2021;109(1):43–76. doi:10.1109/JPROC.2020.3004555.

[28] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016, p. 770–8. doi:10.1109/CVPR.2016.90.

[29] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018, p. 4510–20. doi:10.1109/CVPR.2018.00474.

[30] Redmon J. Darknet: Open-source neural networks in c. 2013.

[31] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016, p. 2818–26. doi:10.1109/CVPR.2016.308.

[32] Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV). 2017, p. 618–26. doi:10.1109/ICCV.2017.74.

[33] Gorham JR. Insect and mite pests in food: An illustrated key. 1991. URL: https://handle.nal.usda.gov/10113/CAT91961681.

[34] USDA . Stored Grain Insect Reference. 2016.