Development of AI models for wildlife and bushfire habitat recovery program

Sankaran Iyer[1], Manna Elizabeth Philip[1], Ryan Schoffner[2], Gerasimos Cassis[2], and Arcot Sowmya[1]

[1] School of Computer Science and Engineering, The University of New South Wales, Australia
[2] School of Biological Earth and Environmental Sciences, The University of New South Wales, Australia

Abstract. This work was undertaken to evaluate the regeneration of the priority invertebrate species following the 2019/2020 Black summer wildfires within the northeast forest region of New South Wales, Australia. The goal was to develop an AI-driven application that can be used to identify a range of invertebrate species in the bushfire-affected areas, in order to verify if regeneration has taken place. To achieve this goal, models were built using 9 different state-of-the-art convolutional neural network architectures. Initial evaluation of the models was performed using IP102 and Museum public datasets consisting of 75222 images of 102 distinct species and 63394 images of 291 different species respectively. To identify target species, a Bushfire dataset consisting of 948 images of 14 different species was acquired in house. The best performance was achieved by an ensemble of 5 models built by combining Inception V3 with channel attention blocks using "Squeeze and Excitation" and "Convolution Block Attention" Modules, achieving an accuracy of 65.59% on IP102, which is about 2% less than the best state-of-the-art accuracy, 95% on Museum which is possibly the best result achieved so far, and 93% on the Bushfire dataset.

Keywords: Invertebrate Species · Convolutional Neural Network · Inception V3 · Squeeze and Excitation · Convolution Block Attention

1 Introduction

This study examines the recovery of key invertebrate species following the Black Summer wildfires of 2019/2020 in the northeastern forests of New South Wales, Australia. The field of ecoinformatics has long been concerned with insect identification to better understand their vast numbers, distribution and crucial roles within ecosystems [1][2]. A significant portion of this research has focused on pest detection to enhance agricultural productivity and reduce pesticide overuse[3][4][5].

A variety of techniques, including traditional image processing and machine learning, have been developed for detecting insects; these include monitoring pests on yellow sticky tapes [6], devising automated methods for identifying whitefly, aphid and thrip species in greenhouses [7], employing multiple task sparse representation and multiple kernel learning [8] strategies, and using support vector machines for pest detection [9][10] and quantifying whiteflies on sticky traps [11]. However, these methods are tailored to their specific use cases and cannot be easily generalised for other contexts.

Over the past decade, deep learning (DL) methods, particularly Convolutional Neural Networks (CNN), have gained prominence for their ability to outperform traditional machine learning techniques. CNN have made significant strides, achieving impressive results in object detection, image classification and segmentation, and are widely applied across various fields. The primary advantage of CNN is their capability for automatic feature extraction from data. However, they require extensive labelled datasets. This challenge is mitigated by Transfer Learning (TL), which applies knowledge from one task to a related one [14]. Popular CNN models, pre-trained with ImageNet data, can be adapted for feature extraction or fine-tuned for specific domains. In this study, TL has been extensively used to adapt well-known architectures. Recently, Transformers, which are self-attention-based architectures, have become the preferred models for Natural Language Processing [22] and have been adapted for Computer Vision [22]. The innovations that contributed to ViT's success have been incorporated into the latest CNN evolution, namely ConvNext, proposed by Liu et al [23].

CNN-based methods are widely used for identifying agricultural pests. Teixeira et al.[3] and Li et al.[4] have provided comprehensive surveys on deep learning for insect detection, highlighting TL as a favoured approach. The primary objective of this study is to determine the best model for identifying priority invertebrate species in bushfire-affected areas. We achieve this by fine-tuning pre-trained versions of well-known architectures: VGG16 [15], ResNet50 [18], EfficientNet [20] ,ConvNext [23], ViT [22], MobileNet [21] and Inception V3 [17]. Besides the above 2 other models were experimented with some plugin attention mechanisms on top of Inception V3 as they were proven to produce excellent results namely: Squeeze and Excitation (SE) [24] and Convolution Block Attention Module (CBAM) [25]

2 Materials and Methods

2.1 Datasets

Four different datasets were used in this study:

- **IP102[5]:** This dataset comprises 75,222 images across 102 unique categories, organised into an hierarchical structure. Categories are grouped based on the type of produce affected by the pests.
- **Museum[2]:** Held at the Natural History Museum in London, this dataset features a collection of 63,364 images representing 291 species of ground beetles (Coleoptera: Carabidae).

– **Bushfire invertebrate species original images:** This collection comprises 948 high-resolution images with an aspect ratio of 8600x5700, covering 14 different species as detailed in Table 1. These images were gathered in-house and not publicly available.
– **Bushfire invertebrate species cropped images:** To mitigate computational constraints, this dataset contains cropped versions of the original Bushfire invertebrate images.

Sample images for IP102, Museum and in house samples are presented in Fig 1

(a) IP102 Samples (b) Museum Samples (c) In house Samples

Fig. 1. Sample Images

2.2 Methods

We conducted experiments using transfer learning (TL) to fine-tune pre-trained models with ImageNet weights. The models included VGG16 [15], ResNet50 [18], EfficientNet [20], ConvNext [23], ViT [22], MobileNet [21], and Inception V3 [17]. Fine-tuning was achieved by replacing the classification layer with one specific to our dataset while keeping all other layers frozen, using TensorFlow 2.5 on a workstation running Ubuntu 18.04 with NVIDIA GeForce RTX 2080 Ti GPUs. Training was performed for 80 epochs for each fold using the Adam optimizer, with an initial learning rate of (1×10^{-3}). The learning rate was then allowed to decay by a factor of 0.1 until a minimum level was reached. Early stopping was applied if there was no improvement after a set patience limit.

Additionally, we integrated two attention mechanisms into Inception V3: Squeeze and Excitation (SE) [24], which applies channel attention with minimal computational cost, and the Convolution Block Attention Module (CBAM) [25], which combines channel and spatial attention. Both mechanisms have shown excellent results in previous work. The input images were resized to 224×224 for all models, except for Inception V3 with SE and CBAM, where the input size was set to 299×299.

For all datasets except IP102, which had its own holdout test set of 22,619 images, approximately 20% of the data was allocated for holdout. The remaining data was divided into 5 folds for cross-validation. Five models were built, each using one fold for validation and the other four for training. Hyperparameter

Table 1. Class distribution for the Bushfire dataset

Class Name	Sample Size
Oncophysa Vesiculata Vesiculata	48
Amphistomus Trispiculatus	25
Daerlac Cephalotes	31
Epimixia Vulturna	86
Kirkaldyella Rugosa	60
Epimixia Tropica	62
Eritingis Trivirgata	20
Setocoris Binataphillis	152
Amphistomus Cunninghamensis	25
Amphistomus Primonactus	25
Eritingis Aporema	112
Epimixia Dysmica	172
Epimixia Vittata	36
Woodwardiola Sp	94

tuning was performed on the validation set, and the final evaluation was conducted on the holdout test set. This process was repeated for each fold, resulting in five-fold cross-validation. Finally, the performance on the holdout test set was assessed using an ensemble of the five models built during cross-validation.

3 Results and Discussion

Six different metrics—Sensitivity, Specificity, Precision, Recall, Balanced Accuracy, and Accuracy—were employed to evaluate the performance of the five-fold cross-validation, as well as the ensemble performance of the five models on the holdout test set. Tables 2, 3, 4, and 5 display the five-fold cross-validation results for the IP102, Museum, Bushfire Original, and Bushfire Cropped datasets, respectively. Meanwhile, Tables 6, 7, 8, and 9 present the performance of the ensemble of the five models on the holdout test set for each dataset.

Here are the definitions for each metric: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for a given class(c) using the micro averaging method discussed in [29]:

$$TP = \sum_c TP_c \tag{1}$$

$$TN = \sum_c TN_c \tag{2}$$

$$FP = \sum_c FP_c \tag{3}$$

$$FN = \sum_c FN_c \tag{4}$$

Table 2. Average performance of five-fold cross-validation on the IP102 dataset.

Model	Sensitivity mean $\pm$ SD	Specificity mean $\pm$ SD	Precision mean $\pm$ SD	Balanced Accuracy mean $\pm$ SD	F1 Score mean $\pm$ SD	Accuracy mean $\pm$ SD
ResNet50v2	**59.42 ± 0.1**	**99.6 ± 0**	**59.42 ± 0.1**	**79.51 ± 0.05**	**59.42 ± 0.1**	**59.42 ± 0.1**
VGG16	53.21 ± 0.19	99.54 ± 0	53.21 ± 0.19	76.31 ± 0.1	53.21 ± 0.19	53.21 ± 0.19
EfficientNetV2L	17.48 ± 0.63	99.18 ± 0.01	$17.48 \pm .63$	58.33 ± 0.32	17.48 ± 0.63	17.48 ± 0.63
ConvNeXTLarge	50.5 ± 0.53	99.51 ± 0.01	50.5 ± 0.53	75.01 ± 0.27	50.5 ± 0.53	50.5 ± 0.53
MobileNetV2	**57.38 ± 0.27**	**99.58 ± 0**	**57.38 ± 0.27**	**78.48 ± 0.14**	**57.38 ± 0.27**	**57.38 ± 0.27**
Vitb32	44.36 ± 0.24	99.45 ± 0	44.36 ± 0.24	79.91 ± 0.12	44.36 ± 0.24	44.36 ± 0.24
InceptionV3	**57.41 ± 0.46**	**99.56 ± 0.04**	**57.41 ± 0.46**	**78.49 ± 0.23**	**57.41 ± 0.46**	**57.41 ± 0.46**
SEInceptionV3	**59.56 ± 0.66**	**99.60 ± 0.01**	**59.56 ± 0.66**	**79.58 ± 0.33**	**59.56 ± 0.66**	**59.56 ± 0.66**
CBAMInceptionV3	**59.21 ± 1.93**	**99.59 ± 0.02**	**59.21 ± 1.93**	**79.4 ± 0.98**	**59.21 ± 1.93**	**59.21 ± 1.93**

Table 3. Average performance of five-fold cross-validation on the Museum dataset.

Model	Sensitivity mean $\pm$ SD	Specificity mean $\pm$ SD	Precision mean $\pm$ SD	Balanced Accuracy mean $\pm$ SD	F1 Score mean $\pm$ SD	Accuracy mean $\pm$ SD
ResNet50v2	**65.11 ± 0.16**	**99.88 ± 0**	**65.11 ± 0.16**	**82.49 ± 0.01**	**65.11 ± 0.16**	**65.11 ± 0.16**
VGG16	**64.05 ± 0.55**	**99.88 ± 0.01**	**64.05 ± 0.55**	**81.97 ± 0.28**	**64.05 ± 0.55**	**64.05 ± 0.55**
EfficientNetV2L	6.38 ± 0.031	99.67 ± 0.02	6.38 ± 0.031	53.03 ± 0.16	6.38 ± 0.031	6.38 ± 0.031
ConvNeXTLarge	23.45 ± 0.54	99.73 ± 0.01	23.45 ± 0.54	61.49 ± 0.23	23.45 ± 0.54	23.45 ± 0.54
MobileNetV2	**57.94 ± 0.28**	**99.86 ± 0**	**57.94 ± 0.28**	**77.90 ± 2.17**	**57.94 ± 0.28**	**57.94 ± 0.28**
Vitb32	43.76 ± 1.54	99.80 ± 0.01	43.76 ± 1.54	72.53 ± 2.09	43.76 ± 1.54	43.76 ± 1.54
InceptionV3	**64.26 ± 0.39**	**99.88 ± 0.00**	**64.26 ± 0.39**	**82.10 ± 0.39**	**64.26 ± 0.39**	**64.26 ± 0.39**
SEInceptionV3	**93.42 ± 0.08**	**99.98 ± 0.00**	**93.42 ± 0.08**	**97.18 ± 1.4**	**93.42 ± 0.08**	**93.42 ± 0.08**
CBAMInceptionV3	**93.24 ± 0.97**	**99.98 ± 0.00**	**93.24 ± 0.97**	**96.61 ± 0.48**	**93.24 ± 0.97**	**93.24 ± 0.97**

Table 4. Average performance of five-fold cross-validation on the Bushfire Original dataset.

Model	Sensitivity mean $\pm$ SD	Specificity mean $\pm$ SD	Precision mean $\pm$ SD	Balanced Accuracy mean $\pm$ SD	F1 Score mean $\pm$ SD	Accuracy mean $\pm$ SD
ResNet50v2	**88.06 ± 2.95**	**99.08 ± 0.23**	**88.06 ± 2.95**	**93.57 ± 1.59**	**88.06 ± 2.95**	**88.06 ± 2.95**
VGG16	**86.24 ± 2.2**	**$98.94 \pm 0,17$**	**86.24 ± 2.2**	**92.59 ± 1.19**	**86.24 ± 2.2**	**86.24 ± 2.2**
EfficientNetV2L	31.29 ± 8.12	94.71 ± 0.63	31.29 ± 8.12	63 ± 4.37	31.29 ± 8.12	31.29 ± 8.12
ConvNeXTLarge	73.77 ± 2.23	97.98 ± 0.17	73.77 ± 2.23	85.87 ± 1.2	73.77 ± 2.23	73.77 ± 2.23
MobileNetV2	**89.15 ± 1.03**	**99.18 ± 0.08**	**89.15 ± 1.03**	**94.27 ± 0.56**	**89.15 ± 1.03**	**89.15 ± 1.03**
Vitb32	20.97 ± 4.38	93.92 ± 0.34	20.97 ± 4.38	57.44 ± 2.36	20.97 ± 4.38	20.97 ± 4.38
InceptionV3	**88.28 ± 2.45**	**99.30 ± 0.36**	**88.28 ± 2.45**	**93.69 ± 1.32**	**88.28 ± 2.45**	**88.28 ± 2.45**
SEInceptionV3	**90.97 ± 3.62**	**99.30 ± 0.28**	**90.97 ± 3.62**	**95.14 ± 1.95**	**90.97 ± 3.62**	**90.97 ± 3.62**
CBAMInceptionV3	**90.86 ± 3.06**	**99.30 ± 0.23**	**90.86 ± 3.06**	**95.08 ± 1.65**	**90.86 ± 3.06**	**90.86 ± 3.06**

Table 5. Average performance of five-fold cross-validation on the Bushfire Cropped dataset.

Model	Sensitivity mean $\pm$ SD	Specificity mean $\pm$ SD	Precision mean $\pm$ SD	Balanced Accuracy mean $\pm$ SD	F1 Score mean $\pm$ SD	Accuracy mean $\pm$ SD
ResNet50v2	**86.56 ± 3.49**	**98.96 ± 0.27**	**86.56 ± 3.49**	**92.76 ± 1.88**	**86.56 ± 3.49**	**86.56 ± 3.49**
VGG16	**88.92 ± 1.64**	**$99.15 \pm 0,13$**	**88.92 ± 1.64**	**99.04 ± 0.88**	**88.92 ± 1.64**	**88.92 ± 1.64**
EfficientNetV2L	31.94 ± 2.91	94.67 ± 0.23	31.94 ± 2.91	63.35 ± 1.56	31.94 ± 2.91	31.94 ± 2.91
ConvNeXTLarge	76.02 ± 1.8	98.16 ± 0.14	76.02 ± 1.8	87.09 ± 0.97	76.02 ± 1.8	76.02 ± 1.8
MobileNetV2	**88.07 ± 2.09**	**99.08 ± 0.16**	**88.07 ± 2.09**	**93.57 ± 1.13**	**88.07 ± 2.09**	**88.07 ± 2.09**
Vitb32	21.08 ± 4.58	93.93 ± 0.35	21.08 ± 4.58	57.5 ± 2.47	21.08 ± 4.58	21.08 ± 4.58
InceptionV3	**88.39 ± 1.05**	**99.11 ± 0.08**	**88.39 ± 1.05**	**93.75 ± 0.57**	**88.39 ± 1.05**	**88.39 ± 1.05**
SEInceptionV3	**91.29 ± 3.43**	**99.33 ± 0.26**	**91.29 ± 3.43**	**95.31 ± 1.85**	**91.29 ± 3.43**	**91.29 ± 3.43**
CBAMInceptionV3	**89.25 ± 3.63**	**99.17 ± 0.28**	**89.25 ± 3.63**	**94.21 ± 1.96**	**89.25 ± 3.63**	**89.25 ± 3.63**

Table 6. Average performance of an ensemble of 5 models on the holdout test set for the IP102 dataset.

Model	Sensitivity %	Specificity %	Precision %	Balanced Accuracy %	F1 Score %	Accuracy %
ResNet50v2	**63.38**	**99.64**	**63.38**	**81.51**	**63.38**	**63.38**
VGG16	57.04	99.57	57.04	78.31	57.04	57.04
EfficientNetV2L	17.99	98.39	17.99	58.59	17.99	17.99
ConvNeXTLarge	53.99	99.54	53.99	76.77	53.99	53.99
MobileNetV2	**61.63**	**99.62**	**61.63**	**80.63**	**61.63**	**61.63**
Vitb32	49.64	99.5	49.64	74.57	49.64	49.64
InceptionV3	**61.38**	**99.62**	**61.38**	**80.5**	**61.38**	**61.38**
SEInceptionV3	**65.32**	**99.66**	**65.32**	**82.49**	**65.32**	**65.32**
CBAMInceptionV3	**65.69**	**99.66**	**65.69**	**82.68**	**65.69**	**65.69**

Table 7. Average performance of an ensemble of 5 models on the holdout test set for the Museum dataset.

Model	Sensitivity %	Specificity %	Precision %	Balanced Accuracy %	F1 Score %	Accuracy %
ResNet50V2	**69.6**	**99.9**	**69.6**	**84.75**	**69.6**	**69.6**
VGG16	**67.84**	**99.89**	**67.84**	**83.67**	**67.84**	**67.84**
EfficientNetV2L	6.67	99.68	6.67	53.17	6.67	6.67
ConvNeXTLarge	25.66	99.74	25.66	62.7	25.66	25.66
MobileNetV2	62.18	99.87	62.18	81.02	62.18	62.18
Vitb32	47.02	99.82	47.02	73.42	47.02	47.02
InceptionV3	**67.91**	**99.89**	**67.91**	**83.9**	**67.91**	**67.91**
SEInceptionV3	**95.45**	**99.98**	**95.45**	**97.72**	**95.45**	**95.45**
CBAMInceptionV3	**95.63**	**99.98**	**95.63**	**97.81**	**95.63**	**95.63**

Table 8. Average performance of an ensemble of 5 models on the holdout test set for the Bushfire Original dataset.

Model	Sensitivity %	Specificity %	Precision %	Balanced Accuracy %	F1 Score %	Accuracy %
ResNet50V2	**90.86**	**99.3**	**90.86**	**95.08**	**90.86**	**90.86**
VGG16	**88.71**	**99.13**	**88.71**	**93.92**	**88.71**	**88.71**
EfficientNetV2L	18.28	93.71	18.28	56	18.28	18.28
ConvNeXTLarge	76.88	98.22	76.88	87.55	76.88	76.88
MobileNetV2	**93.01**	**99.46**	**93.01**	**96.24**	**93.01**	**93.01**
Vitb32	23.12	94.09	23.12	58.6	23.12	23.12
InceptionV3	**90.86**	**99.3**	**90.86**	**95.08**	**90.86**	**90.86**
SEInceptionV3	**93.55**	**99.5**	**93.55**	**96.53**	**93.55**	**93.55**
CBAMInceptionV3	**93.01**	**99.46**	**93.01**	**96.24**	**93.01**	**93.01**

Table 9. Average performance of an ensemble of 5 models on the holdout test set for the Bushfire Cropped dataset.

Model	Sensitivity %	Specificity %	Precision %	Balanced Accuracy %	F1 Score %	Accuracy %
ResNet50V2	**92.47**	**99.42**	**92.47**	**95.95**	**92.47**	**92.47**
VGG16	**90.32**	**99.26**	**90.32**	**94.79**	**90.32**	**90.32**
EfficientNetV2L	33.33	94.87	33.33	64.1	33.33	33.33
ConvNeXTLarge	78.49	98.35	78.49	88.42	78.49	78.49
MobileNetV2	**94.62**	**99.59**	**94.62**	**97.11**	**94.62**	**94.62**
Vitb32	28.49	94.5	28.49	61.5	28.49	28.49
InceptionV3	**90.86**	**99.3**	**90.86**	**95.08**	**90.86**	**90.86**
SEInceptionV3	**95.7**	**99.67**	**95.7**	**97.68**	**95.7**	**95.7**
CBAMInceptionV3	**93.01**	**99.46**	**93.01**	**96.24**	**93.01**	**93.01**

The measures Sensitivity, Specificity, Precision, Balanced Accuracy, F1 Score and Accuracy are defined as follows: Sensitivity also known as Recall measures the proportion of actual positives that are correctly identified by the model.

$$Sensitivity\ (also\ known\ as\ Recall) = \frac{TP}{TP + FN} \tag{5}$$

Specificity measures the proportion actual negatives that are correctly identified by the model.

$$Specificity = \frac{TN}{TN + FP} \tag{6}$$

Precision measures the proportion of positive predictions that are actually correct.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

Balanced Accuracy is the average of Sensitivity and Specificity, providing a more balanced measure when dealing with imbalanced datasets.

$$Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2} \tag{8}$$

F1 Score is the harmonic mean of Precision and Sensitivity (Recall), providing a single metric that balances both. It's particularly useful when there is a need to balance the trade-off between Precision and Sensitivity.

$$F1\ Score = \frac{2TP}{2TP + FP + FN} \tag{9}$$

Accuracy measures the overall proportion of correct predictions made by the model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

As explained in [29], Micro Average Precision = Micro Average Sensitivity = Micro Average F1 Score = Accuracy.

As shown in the tables, Inception V3 with SE and CBAM consistently outperformed the other models. Inception V3, ResNet50V2, MobileNetV2 and VGG16 also demonstrated relatively good performance.

On the IP102 dataset, the proposed approach achieved an accuracy of 65.69%, which is close to the 67.13% accuracy reported by Ayan et al. [26], and it outperformed the accuracies of 61.93% by Nanni et al. [28], 55.24% by Ren et al. [27], and 49.5% by Wu et al.[5] . The performance on the Museum dataset was excellent, achieving 95.63%, which is significantly higher than the 51.9% achieved by Hansen et al. [2]. An improvement could be to evaluate an ensemble of the best-performing models or a weighted ensemble of all models, which is planned for future work.

References

1. Gerovichev, A., et al., High Throughput Data Acquisition and Deep Learning for Insect Ecoinformatics. Journal **Frontiers in Ecology and Evolution**, 2021, 9, DOI: https://doi.org/10.3389/fevo.2021.600931
2. Hansen, O.L.P., et al., Species-level image classification with convolutional neural network enables insect identification from habitus images. **Ecology and Evolution**, 2020. 10(2): p. 737-747, DOI: https://doi.org/10.1002/ece3.5921
3. Teixeira, A.C., et al., A Systematic Review on Automatic Insect Detection Using Deep Learning. **Agriculture**, 2023. 13(3): p. 713,DOI: https://doi.org/10.3390/agriculture13030713
4. Li, W., et al., Classification and detection of insects from field images using deep learning for smart pest management: A systematic review. **Ecological Informatics**, 2021. 66: p. 101460, DOI: https://doi.org/10.1016/j.ecoinf.2021.101460
5. Wu, X., et al. IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition. **2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. 2019.
6. Rustia, D.J.A. and T.-T. Lin, An IoT-based wireless imaging and sensor node system for remote greenhouse pest monitoring. **Chemical Engineering Transactions**, 2017. 58: p. 601-606, DOI: http://dx.doi.org/10.3303/CET1758101
7. Xia, C., et al., Automatic identification and counting of small size pests in greenhouse conditions with low computational cost. **Ecological Informatics**, 2015. 29: p. 139-146, DOI: https://doi.org/10.1016/j.ecoinf.2014.09.006
8. Xie, C., et al., Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning. **Computers and electronics in agriculture**, 2015. 119: p. 123-132, DOI: https://doi.org/10.1016/j.compag.2015.10.015
9. More, S. and M. Nighot, AgroSearch: A Web Based Search Tool for Pomegranate Diseases and Pests Detection Using Image Processing, **Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies**. 2016, Association for Computing Machinery: Udaipur, India. p. Article 44, DOI: 10.1145/2905055.2905102
10. Ebrahimi, M.A., et al., Vision-based pest detection based on SVM classification method. **Computers and electronics in agriculture**, 2017. 137: p. 52-58, DOI: https://doi.org/10.1016/j.compag.2017.03.016
11. Qiao, M., et al., Density estimation of Bemisia tabaci (Hemiptera: Aleyrodidae) in a greenhouse using sticky traps in conjunction with an image processing system. **Journal of Asia-pacific Entomology - J ASIA-PAC ENTOMOL**, 2008. 11: p. 25-29, DOI: http://dx.doi.org/10.1016/j.aspen.2008.03.002
12. Goodfellow, I., Y. Bengio, and A. Courville, Deep Learning. 2016.
13. Krizhevsky, A., I. Sutskever, and G. Hinton, ImageNet classification with deep convolutional neural networks. **Communications of the ACM**, 2017. 60(6): p. 84-90, DOI: https://doi.org/10.1145/3065386
14. Soria Olivas, E., Handbook of research on machine learning applications and trends. **algorithms, methods and techniques**. 2010, Hershey, PA: Hershey, PA : Information Science Reference.
15. Simonyan, K. and A. Zisserman Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv e-prints, 2014, DOI: https://doi.org/10.48550/arXiv.1409.1556
16. Szegedy, C., et al. Going deeper with convolutions. **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. 2015, DOI: https://doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298594
17. Szegedy, C., et al., Rethinking the Inception Architecture for Computer Vision. 2015, DOI: https://doi.org/10.48550/arXiv.1512.00567
18. He, K., et al. Deep Residual Learning for Image Recognition. **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. 2016, DOI: https://doi.org/10.1109/CVPR.2016.90
19. Chollet, F., Xception: **Deep Learning with Depthwise Separable Convolutions**. 2017, DOI: https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.195
20. Tan, M. and Q.V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. 2020, DOI: http://dx.doi.org/10.48550/arXiv.1905.11946
21. Howard, A.G., et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 2017. arXiv:1704.04861 DOI: 10.48550/arXiv.1704.04861.
22. Dosovitskiy, A., et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. 2020. arXiv:2010.11929, DOI: https://doi.org/10.48550/arXiv.2010.11929
23. Liu, Z., et al. A ConvNet for the 2020s. **2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. 2022, DOI: https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01167
24. Hu, J., et al., Squeeze-and-Excitation Networks. **2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition**, 2018, DOI: https://doi.org/10.1109/CVPR.2018.00745
25. Woo, S., et al. CBAM: Convolutional Block Attention Module. 2018. arXiv:1807.06521 DOI: 10.48550/arXiv.1807.06521, DOI: https://doi.org/10.48550/arXiv.1807.06521
26. Ayan, E., H. Erbay, and F. Varçın, Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks.**Computers and electronics in agriculture**, 2020. 179: p. 105809, DOI: https://doi.org/10.1016/j.compag.2020.105809
27. Ren, F., W. Liu, and G. Wu, Feature Reuse Residual Networks for Insect Pest Recognition. IEEE Access, 2019. 7: p. 122758-122768, DOI: http://dx.doi.org/10.1109/ACCESS.2019.2938194
28. Nanni, L., G. Maguolo, and F. Pancino, Insect pest image detection and recognition based on bio-inspired methods. **Ecological informatics**, 2020. 57: p. 101089, DOI: https://doi.org/10.1016/j.ecoinf.2020.101089
29. Grandini, M., E. Bagli, and G. Visani Metrics for Multi-Class Classification: an Overview. 2020. arXiv:2008.05756 DOI: 10.48550/arXiv.2008.05756.