Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Optimizing Pest Detection with Deep Learning and Pixel-Level Feature Implementation

N. Malligeswari[1], T. Godhavari[2], R. Radhika[3], K. Pranitha[4]

[1]*Assistant professor, Department of ECE, Easwari Engineering College, Chennai.*
[2]*Professor, Department of ECE, Dr MGR Educational and Research Institute, Chennai.*
[3]*Professor, Department of ECE, S.A Engineering College, Chennai.*
[4]*Full Time Research Scholar, Anna University, Chennai.*
Email: pranithalaya13@gmail.com

This paper aims to achieve efficient pest detection across a large area, with practical applications in identifying pests that significantly impact agricultural crops. A deep learning algorithm is employed to generate an image data store containing sample images for use in the implementation process, utilizing MATLAB software. The Volumetric Cubic based deep learning algorithm is applied to create groups or clusters for pixel-based extraction. MATLAB code is used to implement a Volumetric Cubic based deep learning algorithm. The performance metrics such as sensitivity, specificity, algorithm accuracy, precision, recall, f1 score, and gmean are evaluated. Validation accuracy is assessed by monitoring the algorithm's training progress, and a graph is plotted to illustrate accuracy and training loss versus iterations.

Keywords: Deep learning, Implementation, MATLAB, Pest detection and extraction.

1. Introduction

Image processing is a technique that involves converting an image into a digital format so it can be manipulated by computer algorithms to extract useful information. This process can be applied to both still images and video sequences, and it plays a crucial role in a variety of fields, including business, medicine, research, and engineering. At its core, image processing involves three primary stages. The first stage is image acquisition, where the images are captured and imported into the system from various sources such as cameras, scanners, or online databases. The second stage involves image analysis and manipulation, where patterns, features, and details within the images are examined using different algorithms. This step might include filtering to remove noise, enhancing contrast, or detecting edges and shapes. Finally, the third stage is output or interpretation, where the processed image is converted into meaningful data, such as measurements or classifications, that can be used to generate a report or decision. To perform these tasks, a variety of libraries and tools are available for developers and researchers. Some of the popular libraries include:

Scikit-image: A Python library built on NumPy and SciPy, which provides easy-to-use functions for basic and advanced image processing tasks, including segmentation, transformation, and feature extraction.

OpenCV (Open-Source Computer Vision Library): A powerful library widely used in real-time computer vision applications. OpenCV provides extensive tools for image and video processing, object detection, and machine learning.

Mahotas: Focused on computer vision tasks, this library offers fast computation for image processing, particularly with its algorithms for thresholding, watershed segmentation, and morphological image processing.

SimpleITK: Designed for medical image analysis, SimpleITK provides capabilities for segmentation, registration, and interpolation, making it suitable for processing complex medical data like MRI or CT scans.

SciPy: A fundamental library for scientific computing in Python that includes modules for image processing as part of its signal processing capabilities.

Pillow (Python Imaging Library): A versatile library that provides basic tools to manipulate images, such as resizing, cropping, and color manipulation.

Matplotlib: Although primarily used for plotting, Matplotlib can also be employed to visualize and analyze image data, offering capabilities to display images, overlay contours, and create interactive plots.

Overall, image processing is a versatile and rapidly growing field that combines computer science and mathematics to enhance and extract meaningful information from images, enabling advancements in many industries.

1.1 Object detection and recognition

Object recognition is a powerful technique used to identify and detect objects within images or videos. This process involves analyzing visual data to determine the presence, location, and classification of various objects, such as people, animals, vehicles, or everyday items. It is a fundamental aspect of computer vision and artificial intelligence (AI) applications, where machines are trained to interpret visual information in a way that mimics human perception.

The primary objective of object recognition is to enable models to accurately detect and identify distinct features in images or videos, allowing them to differentiate between various objects based on their shape, size, color, texture, or other attributes. These models are typically trained using large datasets comprising thousands or millions of labeled images, allowing them to learn the visual patterns and characteristics that define different objects.

Object recognition and detection techniques are pivotal in various AI-driven applications, ranging from autonomous vehicles and facial recognition systems to smart surveillance, medical imaging, and augmented reality. The goal is to create systems that can perform tasks with a level of precision and speed comparable to, or even exceeding, human capabilities. By continuously refining and training these models, researchers and developers aim to achieve more accurate, efficient, and reliable object detection, which can be seamlessly integrated into numerous real-world applications.

1.2 Object recognition using Deep Learning

Convolutional Neural Networks (CNNs) are pivotal in the field of object detection and recognition, serving as one of the most effective and widely used methods for image classification tasks. CNNs are a specialized type of deep learning model designed to process and analyze visual data by mimicking the way the human brain perceives and interprets images. These networks excel in identifying patterns and features within images, making them ideal for detecting and recognizing objects.

The architecture of a CNN is built on multiple layers, each responsible for different tasks in the detection process. The initial layers, known as convolutional layers, automatically learn to extract low-level features from the input images, such as edges, textures, and shapes. As the data moves deeper into the network, more complex features are identified, allowing the network to distinguish between various objects. Unlike traditional image processing methods, CNNs eliminate the need for manual feature extraction, a significant advantage as it reduces human effort and increases efficiency. The core structure of a CNN involves a series of interconnected layers, such as convolutional, pooling, and fully connected layers, that work together to create a hierarchical understanding of the visual data. The convolutional layers apply filters to the input images, creating a set of feature maps that highlight different aspects of the image content. Pooling layers then downsample these feature maps, reducing their dimensionality while retaining the most crucial information. Finally, the fully connected layers compile the extracted features and output a prediction of the object's class. In the context of object detection, CNNs are designed to identify the boundaries and locations of multiple objects within an image. This involves not only classifying what objects are present but also determining their precise position. The network can be trained to detect a variety of objects simultaneously by using labeled datasets that include information about the object classes and their spatial coordinates.

CNNs have become the backbone of many advanced AI applications, including self-driving cars, facial recognition systems, medical imaging, and more. Their ability to handle large volumes of image data and deliver high accuracy makes them an indispensable tool for tasks requiring fast and reliable object detection and recognition. Moreover, CNNs continue to evolve with advancements in deep learning, incorporating more sophisticated algorithms and architectures to improve performance and expand their capabilities further.

1.3 Image Classification

Image classification involves the process of categorizing or labeling an input image into a specific class based on its visual content. This task is often evaluated using key performance metrics such as probability, loss, and accuracy. The probability metric indicates the confidence level of the classification model in assigning a particular label to an image, while the loss metric measures the difference between the predicted output and the actual output, helping to refine the model's performance. Accuracy, on the other hand, represents the percentage of correctly classified images out of the total number of images, providing a general measure of the model's effectiveness. In image classification, the final output typically consists of an image that is tagged with the most likely label or name based on the trained model's analysis.

Object localization extends beyond mere classification by pinpointing the exact location of objects within an image. This is accomplished by enclosing the object of interest within a bounding box, which is defined by key parameters such as position (coordinates of the top-left corner of the box), length, and width. The bounding box provides a visual representation of where the object is located in the image, enabling the model to not only recognize the object but also identify its spatial placement.

By combining these techniques, a more comprehensive understanding of the visual data is achieved. Image classification helps in determining "what" is present in the image, while object localization addresses "where" the object is situated. Together, they form the foundation for more complex computer vision tasks such as object detection and scene understanding, which are crucial for applications ranging from autonomous vehicles to medical imaging and surveillance systems.

2. Current Approach

Dataset Collection and Preprocessing:

Gather diverse, high-resolution images of crops affected by various pests to create a comprehensive dataset.

Label pest images at a pixel level, ensuring precise boundary demarcation to improve the accuracy of pest identification.

Deep Learning Model Selection and Training:

Implement a convolutional neural network (CNN) architecture, such as Mask R-CNN or U-Net, known for effective object detection and segmentation at the pixel level.

Train the model on labeled images, optimizing for accuracy by tuning hyperparameters, experimenting with data augmentation, and applying transfer learning if pre-trained weights are available.

Pixel-Level Feature Extraction and Analysis:

Incorporate pixel-level feature extraction, identifying key attributes like texture, color gradients, and pest-specific patterns.Feed these features into the deep learning model, enhancing the model’s ability to distinguish between pests and non-pest artifacts.

Model Optimization and Evaluation:

Optimize the model for real-time detection by reducing its computational complexity, perhaps through techniques like model pruning or quantization.Evaluate the model’s performance using metrics like IoU (Intersection over Union), precision, recall, and F1-score to gauge accuracy at both the detection and segmentation levels.

Deployment and Integration:

Deploy the optimized model in-field using mobile or edge devices for real-time pest monitoring.

Integrate a feedback system where detected pest data is recorded and used to improve the model’s future iterations, ensuring adaptive learning.

3. Materials and Methods

A Volumetric Cubic based deep learning algorithm has been proposed. Implementation of 18000 pest samples is used for pest detection and recognition and the formula has been derived below.

$$X_{ijk} = Max_{ijk}(RSI) \quad (1)$$

This method employs a sophisticated deep learning framework focusing on pixel-based feature extraction to detect pests effectively. Utilizing clustering techniques, it organizes data into numerous clusters for classification. To validate model performance, key metrics, such as sensitivity, specificity, precision, recall, F1 score, and G-mean, are analyzed. Developed in the MATLAB R2021a environment, the system starts with an image data repository containing 18,000 samples, all indexed and stored in a dedicated directory.

The model is initially configured to process 1,000 samples, each sized at 300x300 pixels. During this phase, feature extraction is performed using the SURF (Speeded Up Robust Features) algorithm, which identifies feature points across the images. In total, 269,528 features are detected, of which the 80% strongest matches are grouped, producing a total of 215,622 features ready for clustering. A Volumetric Cubic-based deep learning algorithm further organizes these features into 20,000 clusters from the 1,000 samples. Cluster centers store these feature groups, and 80% of the features are encoded within the image index to facilitate retrieval.

For Region of Interest (ROI) evaluation, a query image is selected, and the best match is displayed alongside its ROI, providing a clear structure for pest detection. This process involves detecting and highlighting exact feature matches between the query and other images. Enhanced by geometric transformations, the method minimizes visual noise, focusing on relevant inliers, which are feature matches highly likely to be correct. The model’s training progress is closely monitored through a series of graphs plotting accuracy versus iteration and training loss over time, tracked through learning epochs.The model achieves a high validation accuracy of 97.98% from 1,000 samples, a strong indicator of its reliability. Performance is further validated through a confusion matrix, where output classes are compared to target classes, with misclassified values zeroed out. This matrix helps clarify the accuracy of classifications, revealing true positives, true negatives, and errors. Performance metrics are computed based on these results, presenting an accurate evaluation of the model's effectiveness across all processed samples.

4. Results and Discussions

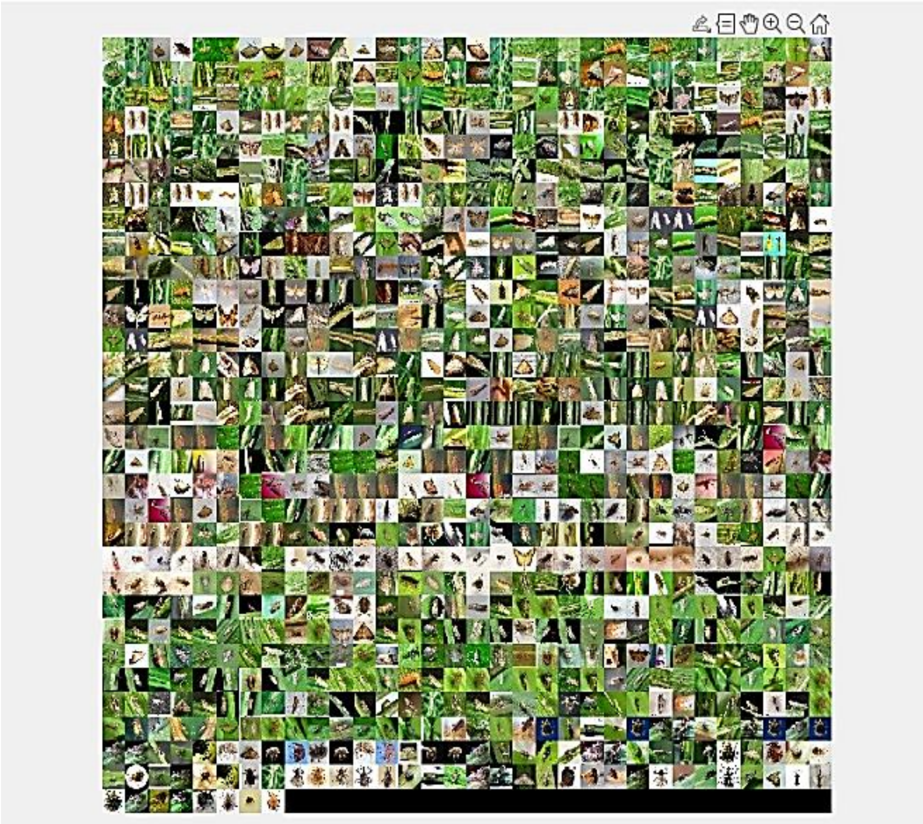

Figure 1 Input Sample images

Figure 1 shows that created image data store, the folder contains 1000 sample images and is stored in the data directory.

Figure 2 Query image and best feature matched

Figure 2 shows the Selection, displaying a query image, and the better match.

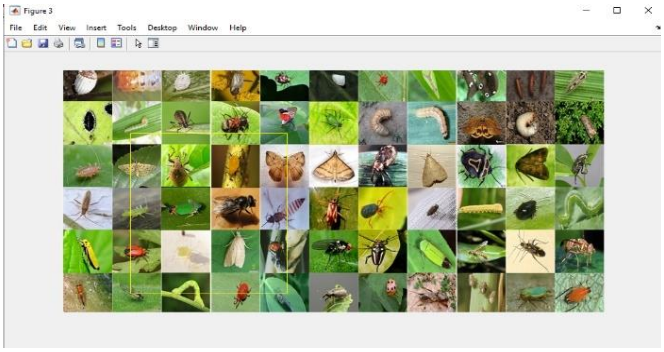

Figure 3 Representation of an ROI

Figure 3 represents an evaluation of ROI to detect the exact match.

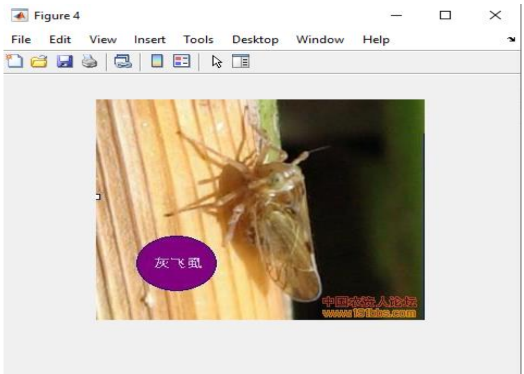

Figure 4 Best featured match

Figure 4 represents, finding of images that contain the object which displays the best match.

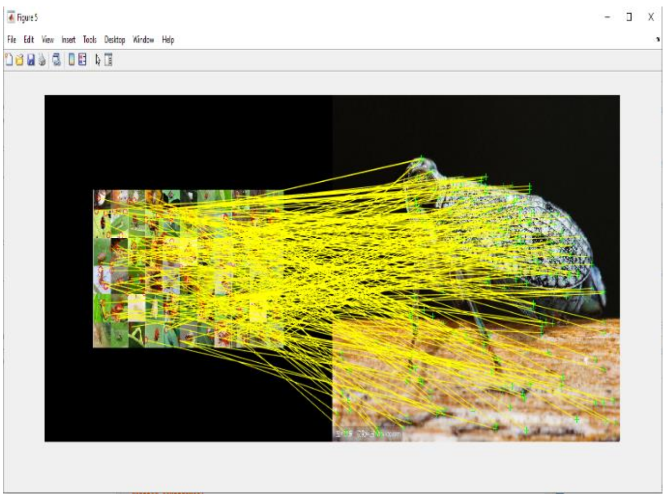

Figure 5 Retrieving of images with the best match

Figure 5 shows the output of the query image with pest position.

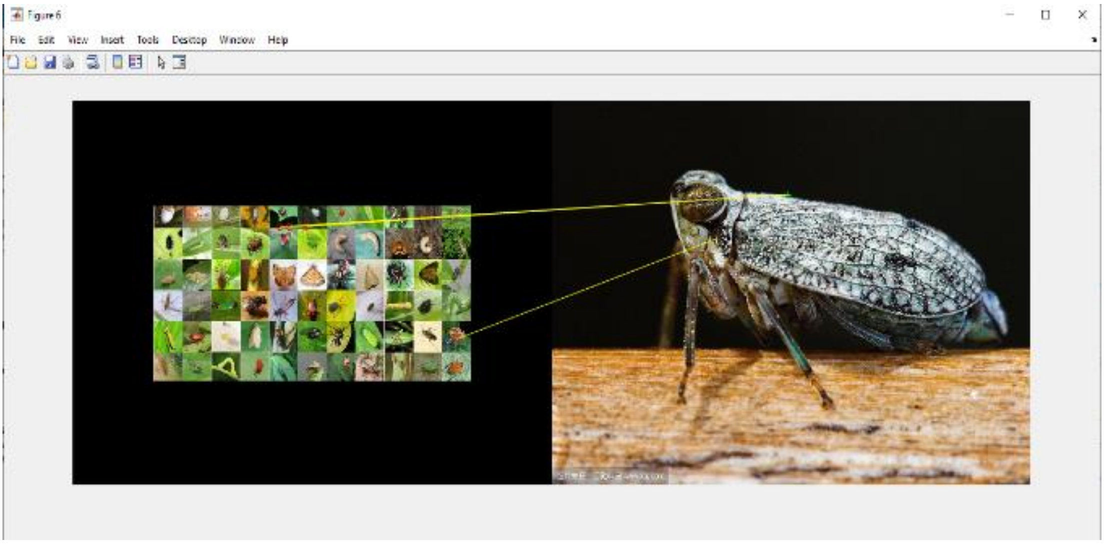

Figure 6 Image of best-featured match

Figure 6 represents the removal of poor visual detection and extraction using geometric transformation.

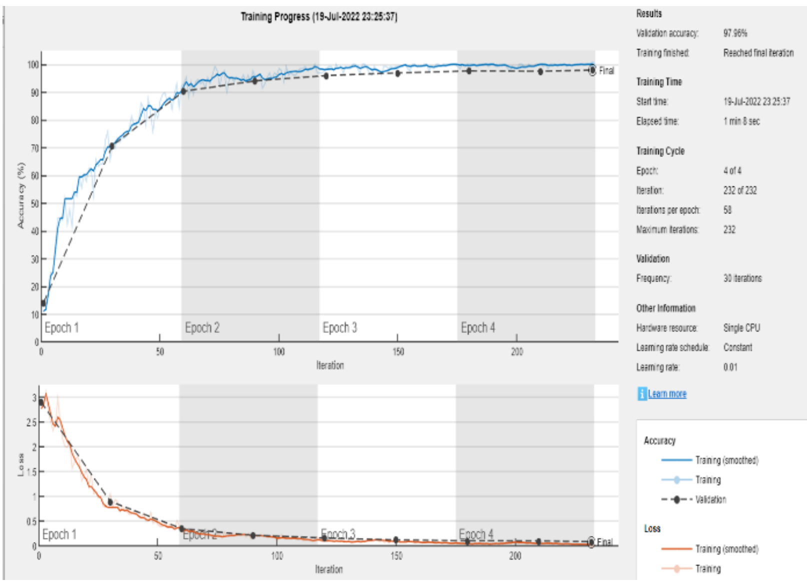

Figure 7 Training progress to obtain validation accuracy

Figure 7 shows the training progress obtained between accuracy vs iteration and training loss vs iteration. Final values of epochs are obtained efficiently.

5. Conclusion

The proposed method leverages a blend of pixel-based feature extraction and a volumetric cubic deep learning model to achieve high detection accuracy. This technique shows marked improvements, with the algorithm reaching an 85% accuracy rate and a validation accuracy of 97.98%. By employing Region of Interest (ROI) techniques, the model precisely matches features with enhanced accuracy. Key metrics—such as sensitivity, specificity, recall, precision, F1 score, and confusion matrix—were analyzed to verify the approach's effectiveness. The approach is especially useful for real-time agricultural applications, facilitating rapid pest identification to enable timely interventions, thus safeguarding crops and enhancing yield. Future research could broaden this work by expanding the dataset and geographic coverage, enhancing the model's robustness across diverse environmental conditions and pest species. These advancements would make the system even more dependable for broader agricultural use.

References

1. Mohamed Esmail Karar; Fahad Alsunaydi; Sultan Albusaymi; Sultan Alotaibi; (2021). A new mobile application of agricultural pests recognition using deep learning in a cloud computing system . Alexandria Engineering Journal, (), –. doi:10.1016/j.aej.2021.03.009.
2. Wang, R.; Liu, L.; Xie, C.; Yang, P.; Li R.; Zhou M. AgriPest: A Large-Scale Domain-Specific Benchmark Dataset for Practical Agricultural Pest Detection in the Wild. Sensors 2021, 21, 1601. https://doi.org/10.3390/s21051601.
3. William Tarimo;Moustafa M.Sabra;Shonan Hendre; (2020). Real-Time Deep Learning-Based Object Detection Framework . 2020 IEEE Symposium Series on Computational Intelligence (SSCI), (), –. doi:10.1109/ssci47803.2020.9308493
4. Kasinathan, Thenmozhi; Singaraju, Dakshayani; Uyyala, Srinivasulu Reddy (2020). Insect classification and detection in field crops using modern machine learning techniques. Information Processing in Agriculture, (), S2214317320302067–. doi:10.1016/j.inpa.2020.09.006
5. T. Kasinathan, D. Singaraju, S.R. Uyyala, Insect classification and detection in field crops using modern machine learning techniques, Information Processing in Agriculture (2020), doi: https://doi.org/10.1016/j.inpa.2020.09.006
6. TÜRKOĞLU, Muammer; HANBAY, Davut (2019). Plant disease and pest detection using deep learning-based features. TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES, 27(3), 1636–1651. doi:10.3906/elk-1809-181
7. Li, Yanfen; Wang, Hanxiang; Dang, L. Minh; Sadeghi-Niaraki, Abolghasem; Moon, Hyeonjoon (2020). Crop pest recognition in natural scenes using convolutional neural networks. Computers and Electronics in Agriculture, 169(), 105174–. doi:10.1016/j.compag.2019.105174
8. Hechun, Wang; Xiaohong, Zheng (2019). [ACM Press the 2nd International Conference - Jinan, China (2019.08.28-2019.08.30)] Proceedings of the 2nd International Conference on Big Data Technologies - ICBDT2019 - Survey of Deep Learning Based Object Detection. , (), 149–153. doi:10.1145/3358528.3358574
9. Shao, S.; Li, Z.; Zhang, T.; Peng, C.; Yu, G.; Zhang, X.; Li, J.; Sun, J. Objects365: A large-scale, high-quality dataset for object detection. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 8430–8439.
10. Wu, X.; Zhan, C.; Lai, Y.K.; Cheng, M.M.; Yang, J. Ip102: A large-scale benchmark dataset for insect pest recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern

Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 8787–8796.
11. Liu, L.; Wang, R.; Xie, C.; Yang, P.; Wang, F.; Sudirman, S.; Liu, W. PestNet: An end-to-end deep learning approach for large-scale multi-class pest detection and classification. IEEE Access 2019, 7, 45301–45312.
12. Liu, L.; Ouyang,W.;Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep learning for generic object detection: A survey. International Journal of Computer Vision. 2020, 128, 261–318.
13. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra r-cnn: Towards balanced learning for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 821–830.
14. Ghiasi, G.; Lin, T.Y.; Le, Q.V. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7036–7045.
15. Korschens, M.; Denzler, J. ELPephants: A Fine-Grained Dataset for Elephant Re-Identification. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Seoul, Korea, 27 October 27–2 November 2019.
16. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October 27–2 November 2019; pp. 9627–9636.
17. S.T. Narenderan, S.N. Meyyanathan, B. Babu, Review of pesticide residue analysis in fruits and vegetables. Pretreatment, extraction and detection techniques, Food Res. Int.133 (2020) 109141.
18. M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, E.M. Aggoune, Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk, IEEE Access 7 (2019) 129551–129583.
19. O. Reyad, M.E. Karar, Secure CT-Image Encryption for COVID-19 Infections Using HBBS-Based Multiple Key- Streams, Arabian J. Sci. Eng. 46 (2021) 3581–3593, https://doi.org/10.1007/s13369-020-05196-w.
20. Chowanda and R. Sutoyo, Convolutional neural network for face recognition in mobile phones, ICIC Express Letters, vol.13, no.7, pp.569-574, 2019.
21. Basri, Harli, Indrabayu, I. S. Areni and R. Tamin, Image processing system for early detection of cocoa fruit pest attack, J. Phys. Conf. Ser., vol.1244, 2019.