An Entire-and-Partial Feature Transfer Learning Approach for Detecting the Frequency of Pest Occurrence

YUH-SHYAN CHEN\textsuperscript{1}, (Senior Member, IEEE), CHIH-SHUN HSU\textsuperscript{2}, (Member, IEEE), AND CHIA-LING LO\textsuperscript{1}

\textsuperscript{1}Department of Computer Science and Information Engineering, National Taipei University, New Taipei 237, Taiwan
\textsuperscript{2}Department of Information Management, Shih Hsin University, Taipei 116, Taiwan

Corresponding author: Chih-Shun Hsu (cshsu@mail.shu.edu.tw)

This research was supported by the Ministry of Science and Technology of Taiwan under Grants MOST 107-2622-E-305-002-CC3 and University System of Taipei Joint Research Program under Grant USTP-NTPU-NTOU-108-01.

ABSTRACT Detecting the frequency of the pest occurrence is always a time consuming and laborious task for agriculture. This paper attempts to solve the problem through the combination of deep learning and pest detection. We propose an entire-and-partial feature transfer learning architecture to perform pest detection, classification and counting tasks which can reach the final goal for detecting the frequency of pest occurrence. In the partial-feature transfer learning, different fine-grained feature map are strengthened by using the weighting scheme of the entire-feature transfer learning. Finally, the cross-layer of the entire-feature network is combined with the multi-scale feature map. The entire-feature transfer learning approach enhances the feature map by creating a shortcut topology for the input and output layers to reduce the gradient disappearance problem which is common to deep networks. The experimental results show that the detection accuracy can be significantly improved and the accuracy can reach 90.2%.

INDEX TERMS Partial-feature transfer learning, cross-layer, frequency of pest occurrence, entire-and partial feature, multi-task learning.

I. INTRODUCTION

Agriculture is indispensable for human life and hence increasing crop yield is important. However, crop yields may reduce production due to pests, which may damage crops or hinder the growth of crops. The method of pest counting used by traditional farmers cannot achieve immediate results. However, if the information on the number of pests cannot be obtained in time, it is impossible to take immediate measurements. Therefore, the relevant information of pests is quite important. To detect the density of pest populations and the number of individual pest species is time consuming and laborious. Due to the current development of deep learning and the rise of machine vision, there are many studies for agriculture applying these techniques and have some excellent results.

Deep learning requires the images of the pests so as to detect the number and class of pests. Some studies have chosen to use pest samples to obtain the dataset of pests, but this approach trains models with limited generalization capabilities, so it is usually only applicable to the environment in which the dataset was originally acquired. The collection method should close to the actual environment and consider the cost effectiveness. The sticky trap is chosen as the data collection method because the pest is easily attracted by color and pheromones. When the pest is stuck on the sticky trap, the camera is used to get the image of the pest, as shown in Fig. 1.

Cho et al. [3] identify whiteflies, mites and thrips by the features of size and color. Gassoumi et al. [8] extract features of images, such as density, aspect ratio and range, to facilitate the classification of pests. Vakilian and Massah [26] use many different classification methods and extract multiple features of pests. Some researches use SVM [27] and artificial neural network [9] and k-nearest neighbor (KNN) [28]. The butterfly recognition based on artificial neural networks is proposed in [10]. An image classification scheme based on the image features is proposed in [5], which is combined with the SVM recognizer. This scheme is mainly based on Haar random...
forest extraction. The traditional machine learning scheme is mainly based on manually extraction. The development of deep learning in recent years does not need to be manually performed. The convolutional neural networks can helps to obtain the main features of the object’s images. The more the deep learning samples are collected, the higher the accuracy becomes.

At present, the machine vision for agriculture with deep learning still faces many challenges. A few factors need to be considered in the actual application environment. The effect of using traditional machine vision is limited. As for the deep learning, the amount of required data is quite high and the collection of certain pest species is not easy to obtain. Therefore, the ability for generalized recognition is limited. Most of the research acquires pest data mainly in the laboratory for manual shooting. The combination of agriculture and technology brings innovation in agriculture, which can increase production capacity and achieve sustainable development. However, agriculture needs to consider various factors, such as climate, pests, crop diseases, weeds and many other influencing factors. Hence, it is not easy to model and accurately detect. The goal of the research is to maintain crop safety and ensure crop quality. The proposed scheme also uses Low-Power Wide-Area (LPWA) communication technology to transmit pest classification results to the cloud server and immediately displays changes in the number of pests on the visual interface which enables the farm staff to take immediate measurement of pests.

In this paper, we propose a framework for pest detection and recognition by localizing and recognizing the image of pests. In order to balance the cost and performance, the proposed scheme embeds the coarse-level and the fine-level feature map to the entire-feature network. In order to make full use of the multi-level feature map of the partial-feature network as the input, the two feature maps of the coarse level and fine level are embedded in the entire-feature network by the weighting scheme. The feature map of the partial-feature network is used to enhance the entire-feature network. This deep network structure is used to enhance pest detection and classification. The main contributions of the proposed scheme are summarized as follows:

- We propose a novel deep learning architecture which combines the partial-feature network and entire-feature network. The coarse and fine level feature map of the partial-feature network are embedded in the entire-feature network by the weighting scheme.
- We create a shortcut topology for the input and output layers to reduce the gradient disappearance problem which is common to deep networks.
- We propose a pest classification and warning system by combining the deep learning, edge computing, and cloud service. The detection result is transmitted by the NB-IoT communication technology.

The remainder of the paper is organized as follows. Section II describes the related work and motivation. Section III describes the system model, problem formulation and basic idea of the proposed scheme. Section IV describes the proposed entire-and-partial feature transfer learning scheme. Section V provides the experimental results, and the conclusion is finally given in Section VI.

II. RELATED WORKS

This section introduces related works in section II-A and describes the motivation of our research in section II-B.

A. RELATED WORKS

In recent years, the field of deep learning has become a hot and exciting research field. Currently, there are some practical examples for agricultural applications using deep learning. Controlling the number of pests on the farm is very important for farmers. Early farmers spent a lot of human resources and time on counting pests. Some results support farmers to accelerate the statistics of various pests on the farm. A number of studied have been carried out [1], [2], [12], [18] to address the management of pest populations in traditional farms, but they still face a huge challenge in integrating cross-domain knowledge. Hong et al. [29] count
insect by using two methods, which are YOLO and SVM classifier. Deng et al. [4] generate a saliency map using the Natural Statistical Model (SUN) and detect a region of interest (ROI) through the pest image. Redmon et al. [22] propose a new end-to-end object detection scheme called YOLO. This scheme transfers the object detection problem into a regression problem and separates the bounding box and related class probability according to space. Peikari et al. [16] propose the idea of clustering high-density regions in the data space, which is used to enhance the SVM classifier efficacy. Motta et al. [17] extract the characteristics of mosquitoes from images and realize a convolutional neural network model for identifying adult mosquitoes. Ebrahimi et al. [7] use the SVM scheme for image processing, which has a differential kernel function to detect and identify pests that may be found on strawberry plants. Lucero et al. [24] propose the RSC and LIRA neural classifiers. This automatic pest detection system is based on artificial neural networks and is mainly used for beetle detection. Nazri et al. [20] propose a convolutional neural network (CNN) based on the VGG16 architecture. The validity is obtained by insect classification, or the benignity of grayscale images constructed in Euclidean distance map (EDM). Jinhua Liu et al. [14] propose different classifiers generated by different features using BP and nearest neighbor (NN) classifiers for the multi-classifier system (MCS) proposed in the paper. Chen et al. [6] propose a scheme for extracting various insect features by taking advantage of the neural network, which uses a deep learning convolutional network. Rembotski et al. [23] propose a word bag model to convert each of the image blocks into a feature vector. Liu et al. [15] propose a deep learning-based convolutional network and apply to the detection and classification of rice weevil and corn weevil. Shen et al. [25] use a faster R-CNN model to extract areas where insects may present and to classify insects in areas where these insects may present. Murali et al. [19] developed a method to identify and re-identify individual fruit flies, which enhances the current methods of tracking software. Lin et al. [13] propose an optimized Bayesian network (BN) structure learning scheme. The proposed scheme is based on the Wolf Pack algorithm (WPA), which can improve the existing image recognition scheme. Zhu et al. [30] propose a automatic visual perception style transfer scheme. A saliency detection approach is proposed to automatically generate an importance map. This map can guide the style transfer discriminatively across the image space. Qi et al. [21] propose a region of interest extraction scheme based on the maximum difference scheme and morphology. A target recognition scheme based on a deep convolutional neural network is also proposed.

**B. MOTIVATION**

The deep learning convolutional networks have been widely adopted for cognition. However, most of the current research is devoted to the detection of object position and the recognition of objects. Through the CNN model, the image input to the model gets the result of the category to which the image belongs. This process is defined as classification. However, in real-world application scenarios, it is common to identify all objects from a picture and mark the location (the location is called object localization). For current image recognition using deep learning, most research efforts are focused on not only identifying each object in an image, but also identifying the type of object. To do this, the research needs to rely on the object detection algorithm in deep learning, which is also the most prosperous area of deep learning in recent years. The objectives of our classification scheme are described as follows. The main goal of designing classification schemes in deep learning based convolutional neural networks is to achieve high detection accuracy. Because in the cognition based convolutional neural networks, the feature map needs to pass through the pooling layer so as to reduce the feature vectors of the convolutional layer output, while improving the results (not easy to over-fitting), but at the same time the pooling work is often discarded. The positional information of the feature map achieves a certain translation invariance, which means that the precise spatial relative relationship between the image components is lost at the same time, which leads to performance degradation. Therefore, based on the feature level of convolutional networks and the transfer learning of convolutional networks, a multi-task learning based convolutional neural network for partial-feature transfer learning is proposed.

**III. PRELIMINARIES**

The system model, the problem formulation and the basic idea are described in subsections III-A, III-B and III-C, respectively.

**A. SYSTEM MODEL**

In this section, the hardware and software framework for counting and classification systems is studied. The edge computing system for pest classification can performs pest counting and classification. The edge computing system for pest classification is a device with classification, counting and transmission functions as shown in Fig. 2. The edge computing system for pest classification uses platforms with technologies such as camera sensors, communication technologies and deep learning to monitor the number of pests on the rendering platform. The architecture of the system platform and components are shown in Fig. 2. The main function of the cloud service system is to train the pest classification model based on deep learning. The pest classification model is transmitted to the edge computing system for pest classification using WiFi communication technology and the edge computing system for pest classification also provides inference work. The cloud service system receives the detection result from the edge computing system. The detection result is transmitted using the NB-IoT communication technology. After the transmission is successful, the cloud service system presents the related information and quantity of the pests through the visual interface and the visual interface provides information of the pests and number of pests.
TABLE 1. Definition of notations.

| Notation | Description |
|----------|-------------|
| $(\hat{X}_i, \hat{Y}_i)$ | The detection and target label. |
| $m$ | Number of outputs in the partial-feature network. |
| $o$ | The output layer in the partial-feature network. |
| $(d_1, d_2)$ | The entire-and-partial ground truth label sets. |
| $\hat{W}$ | The weight of the feature map. |
| $P_{a,b,c}$ | The output layer of the partial-feature network in position $(a, b)$ of the $c$-th channel. |
| $P_c$ | The feature map with the multi-scale filter. |
| $\hat{F}$ | The feature map of the entire-feature network. |
| $\beta$ | The weight control of the coarse-level layer output feature map. |
| $E_f$ | The fine-level feature map combined with the entire and partial feature network. |
| $E_c$ | The coarse-level feature map combined with the entire and partial feature network. |
| $H_i$ | The transformation of the $i$-th building block. |
| $F_i$ | The network map before summation. |
| $F_i$ | An ordinary network map that does not use shortcut modules. |

B. PROBLEM FORMULATION

For the ease of describing the proposed scheme, the notations is defined in Table 1. There are three objective function losses, i.e., pest classification loss, pest location loss and boundary classification loss. The variable $p_i(c)$ is defined as the $i$-th pest classification loss and the variable $\hat{p}_i(c)$ is defined as the $i$-th ground truth box of classification loss. The detection error of the target pest and the sum of squared detection probability errors are shown in equation 1.

$$\begin{align*}
\text{arg min} & \left\{ \sum_{i=0}^{s^2} \sum_{j=0}^{B} \psi_i^j \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
& \quad + \lambda_a \sum_{i=0}^{s^2} \sum_{j=0}^{B} \psi_i^j \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 \right] \\
& \quad + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right\} \\
\text{subject to} & \left\{ \psi_i^j = 1, \quad \text{otherwise} = 0 \right. \\
& \left. 0 \leq \hat{c}_i \leq 1, \quad \forall i, j \right. 
\end{align*}$$

C. BASIC IDEA

The proposed model uses a complete convolutional neural network and a supervised network to extract features from the image and then detect the image through a deep learning model. The partial-feature transfer learning structure includes a feature stream deep network which has a plurality of $m$ outputs. It generates different levels of image feature map for each layer.

The hierarchical output of the multi-level feature map in the extracted background can be changed by changing the hyper-parameter $\phi$. Finally, the partial-feature network output feature map $P_c$ and the corresponding weight $\hat{W}$ perform feedforward work to generate a feed image with multi-level feature map. The coarse and fine level of layers are selected as
feature stream outputs $P_{a,b,c}$ to be merged by the max pooling layer. The deconvolutional layer adjusts the output to the same size and performs an average blending operation. After the feature map is merged, the outputs are fed to the next layer. The feature maps $E_f$ and $E_s$ are combined with the entire- and partial-feature networks, which are inherited by the next layer of the network layer and produces feature output. Fig. 3 shows the comparison of the detection with YOLO model combined classification with SVM classifier [29] and the proposed entire-and-partial feature transfer learning scheme.

IV. THE PROPOSED ENTIRE-AND-PARTIAL FEATURE TRANSFER LEARNING SCHEME

The deep learning model of this paper combines multi-task learning, entire-feature transfer learning and partial-feature transfer learning as shown in Fig. 4. The multi-task learning architecture is used for the mixture of multi-feature in the partial-feature network. The merged feature map provides subsequent coarse and fine level feature map of the deep network via the embedded pipeline. The pest detection network architecture uses a cross layer approach to solve the deep network degradation problem based on the convolutional network YOLO model. The feature map is taken from the previous two layers and is sampled up to 2 samples. The earlier feature map is acquired from the previous network and then the element-wise addition and upsampling features are used. The feature information is acquired from the upsampling features in the early feature map. A few convolution layers are added to the portion of the feature map that handles this combination and detects a double tensor which is twice the size. The box that finally detects the final scale uses the original size. The proposed scheme is split into four phases.

- Entire-and-partial feature learning phase: during this phase, multiple features are put into the partial-feature network. The idea of using multi-task learning with convolutional network is to share parameters of across tasks. The multi-level feature map generated by the partial-feature network is embedded in the entire-feature network.
- Partial-feature transferring phase: during this phase, the entire-feature network and the feature map of the partial-feature network are embedded in the max pooling layer.
of the entire-feature network. The feature map embedding scheme uses a weighting scheme for giving different weights according to different fine-grained features and selecting appropriate channels to join the entire-feature network.

- Entire-feature transferring phase: during this phase, the entire-feature network yields more meaningful feature information from upsampling features and feature map from earlier layer so as to connect the feature map of the previous layer. This operation can enhance the accuracy of the small target detection.
- Pest detection and classification phase: this phase emphasizes on the accuracy of the detection and classification of pests.

### A. ENTIRE-AND-PARTIAL FEATURE LEARNING PHASE

The convolutional layers are used in an entire-feature network. The convolutional layers have different kernel sizes because the scales of pests are different and the feature map is extracted from each convolution layer by applying different filter types $\hat{W}$. The main function of using a convolution filter is to extract different features from the image and then generate a feature map. The variable $\hat{E}_l^k$ is defined as the $k$-th feature map of layer $l$ and $b_l^k$ is defined as the $k$-th bias of layer $l$. The convolution operation represents the feature map of the layer which is calculated as equation 4:

$$\hat{E}_l^k = \phi \left( \hat{W}_l^k \ast \hat{E}_{l-1}^k + b_l^k \right)$$

The partial-feature network covers the optimal local sparse structure in a convolutional neural network architecture through locally available dense components. The fixed size filter kernel of the convolutional neural network will not be able to capture the attribute because the spatial correlation of the feature map is multi-level space, which is demonstrated in Fig. 5. Since the sizes of the pests’ images are different, if a single fixed size filter element is used, the appearance of the pest will be ignored. Therefore, using multiple filter kernels to increase the width of the convolutional neural network is the main feature of the partial-feature network. The feature map generated by the partial-feature network corresponding to the multi-level space of the image uses a plurality of filters. These feature maps are combined into one output which is exhibited in Fig. 5.

### FIGURE 4. The entire-and-partial feature transfer learning architecture.

The equations used on the entire-and-partial feature learning phase are described as follows.

The partial-feature learning phase uses a multi-level convolutional network to extract features from pests’ images,
thereby enabling image detection through deep learning models. The partial-feature network utilizes a fully convolutional neural network and supervised learning. The equation is shown as Equation 5.

$$T = (X_n, Y_n), \quad n = 1, \ldots, N,$$

where $X_n$ is the trained data and $Y_n$ is the feature map of the image with the label $X_n$.

The number of partial-feature network output is defined as $m$. The corresponding weights of partial-feature network $w = \{w^1, \ldots, w^m\}$, and $W$ is the corresponding parameter with network. Two levels of feature aggregation are proposed to integrate feature map of different levels into the coarse and fine levels of the classification network which is presented in equation 6.

$$L_o(W, w) = \sum_{i=1}^{m} a_i W_{o_i}(W, w^i)$$

The $m$-th loss function for output of feature map ($m$) function is shown in equation 7.

$$\arg \min \left\{-\gamma \sum_i d_1 \log P(Y_a = 1|X : W, w^m) - (1 - \gamma) \sum_i d_2 \log P(Y_b = 0|X : W, w^m)\right\}$$

According to Fig. 6, the fully connected layer is removed mainly to obtain a fully convolutional network and the fifth pooling layer performs an element-wise operation on each of the convolutional outputs of the downsampling feature map to obtain a composite feature. Each of the element-wise layer is followed by a deconvolution layer for magnifying the feature map size upsampling layer. A cross entropy loss is used after each upsampling layer. All upsampling layers’ outputs are used for concatenation, then a convolution layer is used to perform the feature map fusion, and finally a cross entropy loss layer is used to get the output.

### B. PARTIAL-FEATURE TRANSFERRING PHASE

The feature map generated by the partial-feature network is the input of the entire-feature network, as shown in Fig. 7. The entire-feature learning obtains the feature map from the mixture network and enhance the feature map corresponding to the output layer by using a weighting scheme.

![FIGURE 7. The partial-feature transfer the fine-grained feature map to the entire-feature learning architecture.](image)

The weight matrix constrained by the L1 norm is enhanced by the induced channel selection. By choosing superior invariant features, generalization performance and convergence speed can be improved. The feature map corresponding to the output layer can be enhanced. Equation 8 shows the feature map from the partial-feature network to the convolution layer of the entire-feature network.

$$P_{a,b,c} = \begin{cases} \max \left\{ \hat{F} \otimes (1 + \hat{W}) \otimes P_c \right\} \\ \max \left\{ \hat{F} \otimes (1 + \beta) \otimes P_c \right\} \end{cases}$$

where $\hat{F}$ is the feature map of the entire-feature map, $P_c$ is the feature map of channel $c$ and $\otimes$ indicates that the two matrices have corresponding integrated positions for the flight. Two positive square product combinations are combined to use weights.

The pooling is controlled by the conventional specifications. This pooling of weighted matrices is defined as a regularization pool. In order to achieve sparsity, the target functions of the network training are shown as equation 9:

$$\arg \min \left\{ L = |X_i - Y_i| + \Phi|\hat{W}_F|_2 + \Phi|\hat{W}_P|_1 \right\}$$

The symbol $\Phi$ is the hyper parameter, $X_i$ is the feature map of the convolutional network from input to output, $Y_i$ is the labeled data, $\hat{W}_F$ is the weight of the feature map in the entire-feature network, $\hat{W}_P$ is the weight of the feature map in the partial-feature network. $|\hat{W}_F|_1$ uses L2 regularization and $|\hat{W}_P|_1$ uses non-smooth regularization which can not only prevent overfitting but also enhance the generalization of
the model. The algorithm of the partial-feature transferring phase is shown in Algorithm 1.

**Algorithm 1 The Partial-Feature Transferring Process**

*Input:* The normalize feature map of partial-feature network $P_c$, The feature map of entire network $\hat{F}$.

*Output:* The fine level feature map $E_f$ combines with entire and partial-feature network $\hat{F} \otimes P_c$. 

1. Input data $a = 0$, $X_0$ is the feature map input to output layer, $Y_i$ is the target ground-truth label.
2. $\hat{W}_p \leftarrow$ L1 regularization.
3. $\hat{W}_F \leftarrow$ L2 regularization.
4. while epoch do
   5. for $a = 1$; $a \leq 2$; $a ++$ do
      6. for $b = 1$; $b \leq 2$; $b ++$ do
         7. if min $|X_i - Y_i| + \Phi |\hat{W}_F| + \Phi |\hat{W}_P|$ then
            8. $L \leftarrow |X_i - Y_i| + \Phi |\hat{W}_F| + \Phi |\hat{W}_P|$
            9. $E_f \leftarrow \begin{bmatrix} \hat{F}_{11} & \hat{F}_{12} & \hat{F}_{13} \\ \hat{F}_{21} & \hat{F}_{22} & \hat{F}_{23} \\ \hat{F}_{31} & \hat{F}_{32} & \hat{F}_{33} \end{bmatrix} \ast$
            10. $(1 + \begin{bmatrix} \hat{W}_{11} & \hat{W}_{12} & \hat{W}_{13} \\ \hat{W}_{21} & \hat{W}_{22} & \hat{W}_{23} \end{bmatrix}) \ast P_c$
        else $a \leftarrow a + 1$
      end for
   end for
   11. $b \leftarrow b + 1$
12. Output: $L/*$ The pooling regularized */
    Output: $E_f/*$ The maximum value of pooling. */

**C. ENTIRE-FEATURE TRANSFERRING PHASE**

The cross-layer method is used in the entire-feature network because depth is very important for CNN recognition results. The deeper the network is, the greater the training complexity will be. The more common reason is because the deeper the network is, the more difficult the gradient descent is to update to the front layer, resulting in parameter updating extremely slow. The general neural network activation function is to make the nonlinear transformation of the input. Every two convolution layers will have a shortcut to connect the input to the back of the network, because the traditional convolution layer will inevitably have data loss when transferring data. The cross layer unit can preserve the integrity of the data to some extent. The entire network only needs to learn the input and output.

Considering the low capacity of the backbone network in feature extraction and the disappearance gradient in the back propagation, a convolutional dense connection structure is adopted. The network improve accuracy by enhancing feature extraction while ensuring maximum information flow in the network.

The connection structure of the entire-feature network is shown in equation 10, in which the feature map of the first l-1 layers are connected together and used as the input to layer l. The variable $H_l$ is defined as the transformation of the l-th building block. The desired output is $H_l(x)$.

$$F_l(x) = H_l(x) - x$$  \hspace{1cm} (10)

The traditional CNN training transformation is based on equation 11 of the l-th building block:

$$H_l(x) = F_l(x) + x$$  \hspace{1cm} (11)

The cross-layer mode connection is shown in equation 13 which introduces neither additional parameters nor computational complexity. Between the ordinary and shortcut topology, the dimensions must be equal to $x$ and $F$ and the linear projection performed by $W_0$ is used to match the dimension’s shortcut connection.

$$y = F(x, W_i) + W_s x$$  \hspace{1cm} (12)
$$y = F(x, W_i) + W_s x$$  \hspace{1cm} (13)

A square matrix $W_s$ is used in equation 13. The identity mapping is sufficient to solve the degradation problem and is economical. Hence, $W_s$ is only used when matching dimensions. $F$ is a shortcut function and its form is flexible. The experiments in this paper involve a function $F$ with three and more layers. However, if $F$ has only one layer, the linear layer is similar to equation 10: $y = W_1 x + x$ and multiple convolutional layers can use the function $F(x, W_i)$. Make two feature map on a channel-by-channel basis to add elements one by one. As shown in Fig. 8, $H$ is the network mapping from the input to the sum with $x$, $F$ is the network mapping before the summation, $F_t$ is the common network mapping without the shortcut connection, we always have to look at the sensitivity of the F mapping.

![The entire-feature transferring phase uses the cross layer method.](image)

**D. PEST DETECTION AND CLASSIFICATION PHASE**

In this phase, the average detection accuracy of multiple overlapping thresholds and the measurement of current test evaluation criteria are emphasized. When the overlap criterion is in the average accuracy, a Non-Maximum
Suppression (NMS) algorithm with a low threshold is applied as shown in Algorithm 2.

Algorithm 2 The NMS Algorithm

```
Input: B = \{b_1, \ldots, b_N\} is the initial detection boxes,
S = \{s_1, \ldots, s_N\}, includes parallelism detection scores, the NMS threshold is N_t
Output: The set of final detections D. The score of detection boxes S
1 D ← \{\} /*normalize */
2 while B ≠ ∅ do
3     m ← s
4     M ← b_m
5     D ← D ∪ M
6     B ← B - M
7     for b_i in B do
8         if iou(M, b_i) ≥ N_t then
9             B ← B - b_i
10            S ← S - s_i
11            s_i ← s_i f(iou(M, b_i))
12        return D, S
```

The detection frame b_i may be very close to the object, so the detection threshold O_t is called here (in 0.7 overlap range). When b_i is low, N_t suppression occurs because the score is slightly lower than M (M is not covered). This case increases the threshold of the overlap criterion. Therefore, suppressing all nearby low N_t detection box will increase the miss rate. In addition, when O_t is low, using a high N_t will increase the false positive, thereby reducing average accuracy of multiple thresholds. In this case, the increase in true positive will be much lower. The increase in false positive increases the true positive because the number of objects is usually much smaller than the number of ROIs produced by the detector. The scoring function is shown in equation 14:

\[
s_i = \begin{cases} \hat{s}_i, \text{iou}(M, b_i) < N_t \\ \hat{s}_i (1 - \text{iou}(M, b_i) \geq N_t), \text{iou}(M, b_i) \geq N_t \end{cases}
\]

(14)

The detection score above the threshold N_t will be attenuated by the above function to a linear function that overlaps with M. The detection frames far from M will not be affected and larger penalties will be assigned to the closer detection frames.

The labelled data needs to pass the model in order to be trained. Suppose the image have divided into a grid of size 3 × 3 and there are total 6 classes which set to classify the objects. For each grid cell, the label \(Y_i\) will be an eleven dimensional vector. Here, \(\hat{P}_i(c)\) defines whether an object is present in the grid or not (it is the probability) \(\hat{c}_1, \hat{c}_i, \hat{v}_i, \hat{h}_i\) specify the bounding box if there is an object \(\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{c}_5, \hat{c}_6\) represent the classes. So, if the object is a \(\hat{c}_1, \hat{c}_1\) vector element will be 1 and \(\hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{c}_5, \hat{c}_6\) vector element will be 0, and so on. The classification approach selects the first grid from the above example: Since there is no object in this grid, \(\hat{P}_i(c)\) will be zero which means that it does not matter what \(\hat{c}_i, \hat{v}_i, \hat{h}_i, \hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{c}_4, \hat{c}_5, \hat{c}_6\) contain as there is no object in the grid.

V. EXPERIMENTAL RESULTS

This section describes the environment configuration, model parameter settings, and experimental results. In the execution environment, we divided all the functions into six parts. The six parts are smart farm sensor, edge computing platform, edge computing, edge link, deep learning server, and real-time equipment monitoring. The cloud service receives detections of the pest classification from the edge computing system and sends detections using NB-IoT communication technology. After the transmission is successful, the cloud service displays the information and quantity of the six pests through a visual interface, and the visual interface provides information of the pest information and the number of pests.

The GPU used in the deep learning server is Nvidia GTX 1080 Ti. The TensorFlow version 1.5.0, the operating system Ubuntu 16.04, and the programming language python version 3.6 are used to implement the 106-layer DNN architecture. 14000 sampling data sets are divided into 8000 training sets and 6000 data sets. The learning rate is set as 0.001, momentum is set as 0.9, step size is set as 4000, decay is set as 0.0005 and the iteration is set as 70,000. The proposed system uses three algorithms, namely the partial-feature transferring, the entire-feature transferring, and the entire-and-partial feature transferring algorithms. The red line represents the proposed schemes, the green line represents the baseline scheme, and the blue line represents the detection based on YOLO model and classification with SVM scheme. The YOLO-SPP (You Only Look Once-Spatial pyramid Pooling) [11] end to end model is adopted as our baseline scheme.

The performance metrics to be observed are defined as follows.

- Mean average precision (mAP): mean average precision is the mean of the average precision, where \(\text{Precision} = \frac{TP}{TP+FP}\). TP denotes true positive (the prediction is positive and the results is also positive) and FP denotes false positive (the prediction is positive but the results is negative).
- Recall: \(\text{Recall} = \frac{TP}{TP+FN}\), where FN denotes false negative (the prediction is negative but the results is positive).
- Training loss: the training losses of the pest bonding box and the pest classification are considered.
- Class accuracy: the accuracy of classification.

A. MEAN AVERAGE PRECISION (mAP)

Fig. 10 shows the experimental results of the effect of iteration (ranging from 0 to 7000), epoch (ranging from 0 to 80), batch (ranging from 1 to 16) and training instances per class on the mean average precision. Fig. 10(a) shows the impact of the iteration to the mean average precision. As the iteration increases, the mean average precision also increases. When the times of iteration is greater than 40000, the mean average
FIGURE 9. The environment deployment.

FIGURE 10. (a) The mean average precision vs. iteration. (b) The mean average precision vs. epoch. (c) The mean average precision vs. batch. (d) The mean average precision vs. training instances per class.

B. RECALL

Fig. 11 shows the experimental results of the effect of iteration, epoch, batch and training instances per class on the recall. Fig. 11(a) shows the impact of the iteration to the recall. As the iteration increases, the recall also increases. When the times of iteration is greater than 40000, the recall becomes stable. Fig. 11(b) shows the impact of the epoch to the recall. As the epoch increases, the recall also increases. Fig. 11(c) shows the impact of the batch to the recall. As the batch increases, the recall also increases. When the epoch is greater than 13, the recall becomes stable. Fig. 11(d) shows the impact of the training instances per class to the recall. Among the 6 classes of pests, the recall of the muscomorpha is the highest, followed by the bactrocera dorsalis, liriomyza trifolii, thyssanoptera, plutella xylostella, and phyllotreta striolata. Overall, the recall of the proposed entire-and-partial feature transferring scheme is the highest, followed by the entire feature transferring scheme, the partial feature transferring scheme, the base line model scheme, and the detection based on YOLO model and classification with SVM scheme because the proposed entire-and-partial feature transferring scheme can enhance the feature map and reduce the gradient disappearance. The proposed entire-and-partial feature transferring scheme can achieve about 90% average precision.
on YOLO model and classification with SVM scheme. The proposed entire-and-partial feature transferring scheme can achieve more than 85% recall rate.

C. TRAINING LOSS

The training loss is calculated as follows. First, each detected frame with IOU and all ground truth losses are calculated and the loss is the confidence error of the calculated background. If a threshold is less than this value, the detected frame is to be marked as a background and needs to be calculated as an object confidence error. And then the coordinate error is calculated and calculates the difference between the ground truth box and the detected width. Finally, calculates each part with the loss value that matches its ground truth, the error including coordinate, confidence and classification errors.

The matching principle works as follows. First determine which cell the center point falls on and then calculate the the previous box with IOU value and the base fact of the cell. When calculating the IOU value, the coordinates are not considered, only the shape is considered, so the previous frame and the ground truth point are offset to the same position and then the corresponding IOU value is calculated. The previous box with the largest IOU value matches the ground truth and the corresponding detection box is used to detect the ground.

Fig. 12 shows the experimental results of the effect of iteration, epoch, batch and training instances per class on the training loss. Fig. 12(a) shows the impact of the iteration to the training loss. As the iteration increases, the training loss decreases. Fig. 12(b) shows the impact of the epoch to the training loss. As the epoch increases, the training loss decreases. When the epoch is greater than 10, the training loss becomes stable. Fig. 12(d) shows the impact of the training instances per class to the training loss. Overall, the training loss of the proposed entire-and-partial feature transferring scheme is the lowest, followed by the entire feature transferring scheme, the partial feature transferring scheme, the base line model scheme, and the detection based on YOLO model and classification with SVM scheme. The proposed entire-and-partial feature transferring scheme can achieve less than 1% of the training loss.

D. CLASS ACCURACY

Fig. 13 shows the experimental results of the effect of iteration, epoch, batch and training instances per class on the class accuracy. Fig. 13(a) shows the impact of the iteration to the class accuracy. As the iteration increases, the class accuracy also increases. When the times of iteration is greater than 50000, the class accuracy becomes stable. Fig. 13(b) shows the impact of the epoch to the class accuracy. As the epoch increases, the class accuracy also increases. When the epoch is greater than 60, the class accuracy becomes stable. Fig. 13(c) shows the impact of the batch to the class accuracy. As the batch increases, the class accuracy also increases. When the epoch is greater than 12, the class accuracy becomes stable. Fig. 13(d) shows the impact of the training instances per class to the class accuracy. Among the 6 classes of pests, the class accuracy of the muscomorpha is the highest, followed by the bactrocera dorsalis, liriomyza trifolii, plutella xylostella, thysanoptera, and phyllotreta striolata. Overall, the class accuracy of the proposed entire-and-partial feature transferring scheme is the highest, followed by the entire feature transferring scheme, the partial feature transferring scheme, the base line model scheme, and the detection based
Y.-S. Chen et al.: Entire-and-Partial Feature Transfer Learning Approach for Detecting the Frequency of Pest Occurrence

FIGURE 13. (a) The class accuracy of iterations. (b) The class accuracy of epoch. (C) The class accuracy of batch. (d) The class accuracy of training instances per class.

on YOLO model and classification with SVM scheme. The proposed entire-and-partial feature transferring scheme can achieve more than 90% class accuracy.

E. CUMULATIVE DISTRIBUTION FUNCTION (CDF)

Fig. 14 (a), (b), (c), (d) show the CDF of the average precision, recall, training loss, and class accuracy respectively. As shown in Fig. 14(a), more than 70% of the average precision of the proposed entire-and-partial feature transferring scheme is higher than 80%. As shown in Fig. 14(b), more than 75% of the recall of the proposed entire-and-partial feature transferring scheme is higher than 75%. As shown in Fig. 14(c), all the training loss of the proposed entire-and-partial feature transferring scheme is less than 0.7%. As shown in Fig. 14(d), more than 85% of the class accuracy of the proposed entire-and-partial feature transferring scheme is higher than 75%. Overall, the proposed entire-and-partial feature transferring scheme performs the best, followed by the entire feature transferring scheme, the partial feature transferring scheme, the base line model scheme, and the detection based on YOLO model and classification with SVM scheme.

VI. CONCLUSIONS

This paper attempts to solve the pests detection problem through the combination of deep learning and pest detection and thus a lot of manpower is saved. We propose an entire-and-partial feature transfer learning network architecture. In the partial-feature network, different fine grained feature maps are used to enhance the entire feature network, and the cross layer method of the entire-feature network enhances the input of the network layer near the output layer. The feature map creates a shortcut topology for the input and output layers to reduce the gradient disappearance problem common to deep networks. The experimental results show that the detection accuracy can be significantly improved especially for smaller pests.

REFERENCES

[1] A. T. Nieuwenhuizen, J. Hemming, and H. K. Suh, “Detection and classification of insects on stick-traps in a tomato crop using faster R-CNN,” in Proc. Netherlands Conf. Comput. Vis., Eindhoven, The Netherlands, Jan. 2018, pp. 1–4.

[2] C. Xie, R. Wang, J. Zhang, P. Chen, W. Dong, R. Li, T. Chen, and H. Chen, “Multi-level learning features for automatic classification of field crop pests,” Comput. Electron. Agricult., vol. 152, pp. 233–241, Sep. 2018.

[3] J. Cho, J. Choi, M. Qiao, C. W. Ji, and H. Y. Kim, “Automatic identification of whiteflies, aphids and thrips in greenhouse based on image analysis,” Int. J. Math. Comput. Simul., vol. 1, no. 1, pp. 46–53, 2007.

[4] L. Deng, Y. Wang, Z. Han, and R. Yu, “Research on insect pest image detection and recognition based on bio-inspired methods,” Biosystems Eng., vol. 169, pp. 139–148, May 2018.

[5] W. Ding and G. Taylor, “Automatic moth detection from trap images for pest management,” Comput. Electron. Agricult., vol. 123, pp. 17–28, Apr. 2016.

[6] D. Xia, P. Chen, B. Wang, J. Zhang, and C. Xie, “Insect detection and classification based on an improved convolutional neural network,” Sensors, vol. 18, no. 12, p. 4169, 2018.

[7] M. A. Ebrahimi, M. H. Khoshtaghaza, S. Minaei, and B. Jamshidi, “Vision-based pest detection based on SVM classification method,” Comput. Electron. Agricult., vol. 137, pp. 52–58, May 2017.

[8] H. Gassoumi, N. R. Prasad, and J. J. Ellington, “Neural network-based approach for insect classification in cotton ecosystems,” in Proc. Int. Conf. Intell. Technol., Bangkok, Thailand, 2000, pp. 1–7.

[9] S.-H. Kang, S.-H. Song, and S.-H. Lee, “Identification of butterfly species with a single neural network system,” J. Asia-Pacific Entomol., vol. 15, no. 3, pp. 431–435, Sep. 2012.

[10] Y. Kaya and L. Kayci, “Application of artificial neural network for automatic detection of butterfly species using color and texture features,” Vis. Comput., vol. 30, no. 1, pp. 71–79, Jan. 2014.

[11] K.-J. Kim, P.-K. Kim, Y.-S. Chung, and D.-H. Choi, “Performance enhancement of YOLOv3 by adding prediction layers with spatial pyramid pooling for vehicle detection,” in Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS), Auckland, New Zealand, Nov. 2018, pp. 1–6.
Yuh-Shyan Chen (Senior Member, IEEE) received the B.S. degree in computer education from National Taiwan Normal University, Taiwan, in 1990, the M.S. degree in computer science from National Taiwan University, Taiwan, in 1992, and the Ph.D. degree in computer science from National Central University, Taiwan, in 2004. He joined the Faculty of the Department of Information Management, Nanya Institute of Technology, Taiwan, as an Instructor, in August 1996, and became an Associate Professor, since 2008. His recent research topics include wireless communications, mobile computing, and next-generation personal communication systems. He is a Senior Member of the IEEE Communication Society and the Phi Tau Phi Society. His paper won the 2001 IEEE 15th IICON-15 Best Paper Award. He was a recipient of the 2005 Young Scholar Research Award from National Chung Cheng University, Taiwan. He served as the Editor-in-Chief for the International Journal of Ad Hoc and Ubiquitous Computing, an Associate Editor-in-Chief for Sensors (section for sensor networks), and the Regional Editor (Asia and Pacific) for IET Communications.

Chih-Shun Hsu (Member, IEEE) received the B.S. degree in computer science from National Taiwan Normal University, Taipei, Taiwan, in June 1988, and the M.S. and Ph.D. degrees in computer science and information engineering from National Central University, Taiwan, in 1991 and 1996, respectively. He joined the Department of Computer Science and Information Engineering, National Taipei University, Taiwan, in August 2006, and has been a Distinguished Professor, since 2018. His recent research topics include wireless communications, mobile computing, and next-generation personal communication systems. He is a Senior Member of the IEEE Communication Society and the Phi Tau Phi Society. His paper won the 2001 IEEE 15th IICON-15 Best Paper Award. He was a recipient of the 2005 Young Scholar Research Award from National Chung Cheng University, Taiwan. He served as the Editor-in-Chief for the International Journal of Ad Hoc and Ubiquitous Computing, an Associate Editor-in-Chief for Sensors (section for sensor networks), and the Regional Editor (Asia and Pacific) for IET Communications.

Chia-Ling Lo received the B.S. degree from the Department of Computer Science and Communication Engineering, Providence University, Taiwan, in June 2017, and the M.S. degree from the Department of Computer Science and Information Engineering, National Taipei University, Taiwan, in August 2019. Her research interests include wireless communications and mobile computing. She became the Editor-in-Chief of the International Journal of Vehicle Information and Communication Systems, in September 2016.