Deep Learning based Automatic Multi-Class Wild Pest Monitoring Approach using Hybrid Global and Local Activated Features

Abstract—**Specialized control of pests and diseases have been a high-priority issue for agriculture industry in many countries. On account of automation and cost-effectiveness, image analytic based pest recognition systems are widely utilized in practical crops prevention applications. But due to powerless handcrafted features, current image analytic approaches achieve low accuracy and poor robustness in practical large-scale multi-class pest detection and recognition. To tackle this problem, this paper proposes a novel deep learning based automatic approach using hybrid and local activated features for pest monitoring solution. In the presented method, we exploit the global information from feature maps to build our Global activated Feature Pyramid Network (GaFPN) to extract pests' highly discriminative features across various scales over both depth and position levels. It makes changes of depth or spatial sensitive features in pest images more visible during downsampling. Next, an improved pest localization module named Local activated Region Proposal Network (LaRPN) is proposed to find the precise pest objects' positions by augmenting contextualized and attentional information for feature completion and enhancement in local level. The approach is evaluated on our 7-year large-scale pest dataset containing 88.6K images (16 types of pests) with 582.1K manually labelled pest objects. The experimental results show that our solution performs over 75.03% mAP in industrial circumstances, which outweighs two other state-of-the-art methods: Faster R-CNN with mAP up to 70% and FPN mAP up to 72%. Our code and dataset will be made publicly available.**

Index Terms—**Convolutional Neural Network, Pest Monitoring, Global Activated Feature Pyramid Network, Local Activated Region Proposal Network**

I. Introduction

Specialized and effective pest control and monitoring in agriculture is becoming an increasingly serious issue all around the world [1]. The urgent demand for efficiently controlling and inspecting the occurrence of agricultural pests in fields has driven the rapid development of industrial pest prevention solutions and intelligent pest monitoring systems, such as chemical pesticides [2], image analytic systems [3], automatic adjustable spraying device [4], status estimation of wheat plants [5], remote sensing [6], etc. On account of automation and cost-effectiveness, image analytic based pest recognition and monitoring systems are widely utilized in practical crops prevention applications. Typically, these systems install some stationary pest trap devices or facilities in the wild fields for real-time acquisition and transmission of trap images, and then employ advanced image analytic techniques [7], [8], [9], [10] into these images for identification and extraction of pest-associated data in support of intelligent prediction and prevention.

Above advanced image analytic techniques enable abundant success in effective pest detection and recognition of certain types of pest. Yet, utilizing these techniques in designing as well as developing practically useful and robust pest monitoring system is still unsatisfied. The first reason for this problem is that extracted features as pest descriptors are short of sufficient details for tiny and blurred pest objects in 2D static images captured by stationary devices. These pose a fundamental dilemma that it is hard to distinguish small object from the generic clutter in the background. Also, traditional machine learning approaches have been suffering from many limitations such as powerless hand-crafted features and the lack of expert consensus. In addition, most of current systems focus on whole pest image classification rather than detection, which aims to localize and identify each pest instance in the image that is necessary for high-level pest analysis promoting more efficient pest monitoring systems in the wild. Therefore, towards large-scale multi-class pest monitoring, it is highly necessary to develop a novel automatic approach by mining more valuable information as highly discriminative features for pest detection.

Recently, advances in deep learning techniques have led to significantly promising progress in the field of generic object detection, like SSD [11], Faster R-CNN [12], Feature Pyramid Network (FPN) [13] and other extended variants of these networks [14], [15]. Among these approaches, two-stage object detection architectures are the most popular in dealing with practical problems due to higher detection accuracy. In Faster R-CNN, Region-of-Interest (RoI) pooling is used to extract features on a single-scale feature map. But targeting at small object detection, FPN is a better state-of-the-art technique over COCO dataset [16] with mAP up to 56.9%. By building up a multi-scale image pyramid, FPN enables a model to detect all objects across a large range of scales over both positions and pyramid levels. Besides, feature pyramid structure built on Convolutional Neural Network (CNN) has become a wide selection as it covers low-level object features and high-level semantic features simultaneously. This property is particularly useful to tiny object detection like pest detection.

In this context, this paper targets at finding out a practically effective and robust pest monitoring solution by studying the state-of-the-art deep learning methods to solve the problems in current large-scale multi-class pest detection task. As shown in Fig. 1, in our presented method, we firstly construct a CNN based feature pyramid architecture to ensure the pests across various scales could be found, and then propose a Global activated Feature Pyramid Network (GaFPN) for retrieving depth and spatial attention over different levels in the pyramid network. Compared to [12] and [13], this approach, the adjusted network will enable variance or changes of spatial or depth sensitive features in images more visible in the pooling layers. This property will allow some missing features of tiny pests in pooling layers in one level to be redetected by many pyramid levels. Next, an improved pest localization module named Local activated Region Proposal Network (LaRPN) is proposed to find the precise pest objects' positions by augmenting contextualized and attentional information for feature completion and enhancement in local level. Following this idea, we integrate GaFPN and LaRPN into a two-stage convolutional neural network (CNN) approach. It is evaluated over our newly published large-scale pest detection specific image dataset containing 88.6K raw images with 582.1K manually labelled pest objects. The image data were collected in the wild field using mobile camera over 7 years. The experimental results show that our approach achieves over mAP of 75.03%, which outweighs two other state-of-the-art methods [12] with mAP of 70% and [13] mAP of 72%.

The major contributions of this paper are as follows:

(1) A novel two-stage CNN based pest monitoring approach using hybrid global and local activated feature is designed for large-scale multi-class pest dataset. It is implemented as a practically automatic pest monitoring system, which accurately and effectively detects 16 types pest in fields.

(2) The proposed approach introduces two novel global and local activation branches: GaFPN and LaRPN for automatic multi-scale feature extraction and efficient region providing and fine-tuning respectively. Our approach could help recognize and extract discriminative features of tiny objects and accommodate large variations and changes of distribution of tinny objects over images. It benefits the precise measure and prediction of pest in complex circumstances with multi-class insect.

(3) A comprehensive and in-depth experimental evaluation on practical industry level large-scale pest dataset (88.6K images) is provided for verifying the usefulness and robustness of proposed system and approaches. The results show that our approach delivers a mAP of 75.03% over 16 types of pest detection, which outweighs two other state-of-the-art methods: Faster R-CNN [12] with mAP up to 70% and FPN [13] mAP up to 72%.

Nomenclature

$\hat{t}_i$	The ground truth coordinate of bounding box
a	The output of convolution operation and activation function
AP	Average Precision
b_k	The bias of convolution kernel k
C	The number of channels of feature map
FN	False Negatives
FP	False Positives
H	The height of feature map
IoU	Intersection-over-Union
Pr	Precision
Re	Recall
t_i	The predicted coordinate of bounding box
TP	True Positives
W	The width of feature map
W_k	The weight of convolution kernel k

II. Related Work

In agriculture systems, artificial intelligence and machine learning techniques have been widely used in various monitoring tasks. Ruan et al. [17] proposed to mine valuable information from agriculture big data to guide the precise management of apple plant for growers. However, environment information might not be enough for building a mature and comprehensive monitoring system while pest is one of the major risks in agriculture applications. In this case, typical image analytics techniques for pest monitoring focus on the study of object identification, including feature extraction and model training. Early works on insect classification include RGB multispectral analysis [8] and Principle Component Analysis (PCA) algorithm [18]. Then, more valuable and representative features are mined for precise pest recognition such as size, color [19], shape and texture [20]. But these features were too weak to be insensitive to rotation, scale and translation. Thus, Scale-invariant feature transform (SIFT) in modern computer vision techniques are popular to realize rotational variance for pest classification [21]. On the other hand, classifiers are key to achieve better model training performance, such as support vector machine (SVM) [22], k-nearest neighbors (KNN) [23], and Artificial Neural Network (ANN) [24]. While aforementioned approaches achieved success to some extent, their results rely too much on quality of handcrafted features selection. Towards large-scale multi-class insect dataset, one consequence is that within species, extracted descriptors show strong similarity to others. Feature vectors with different species are highly close in feature space to relative variability of their texture, color, shape and so on. It is hard to utilize these approaches in practical pest monitoring applications, since the process of selecting and designing features is laborious and insufficient to represent all aspects of the insects.

Fortunately, the emergence of deep learning techniques has led to significantly promising progress in computer vision techniques that facilitates industrial applications development such as human activity recognition [25], automatic fruit classification [26] and cloud workload prediction [27]. In smart agricultural applications, under the combination with Internet of Things (IoT), various systems are built based on deep learning techniques such as U-Net employed in yellow rust disease monitoring [28]. But the difficulty of remote sensing image capturing limits the real-world applications in this work. In generic image classification and object detection task, CNN has exhibited superior capacities in learning invariance in multiple object categories from large amounts of training data [29]. It enables suggesting object proposal regions in detection process; and extract more discriminative features than hand-engineered features. By detecting locations [12], [14] and fine-tuning [30] general representation to a specific object category, CNNs perform well in object detection. Some two-stages approaches [12] utilizes dense sliding window to find out the possible object regions with low-level cues. They can detect the better proposals and share the weights of convolutional layers with other of detectors. They perform even better than one-stage CNN based approaches with higher accuracy of object detection. The above deep learning methods [11], [12],

[13], [14] have showed great accuracies in many general object detection applications beyond what can be achieved by previous methods [23], [22], [24], but they are often intractable for pest monitoring applications.

Towards large-scale multi-class pest monitoring, deep learning methods need to integrate with other techniques like feature pyramids [13] for improved performance. The experiment results on the Microsoft COCO dataset [16] shows that two-stage object detection framework such as Faster R-CNN is an effective region-based object detector towards general object detection with a mean Average Precision (mAP) up to 42.7% because of region proposals are computed at first stage. But for small object detection, FPN is a better state-of-the-art technique over COCO dataset with mAP up to 56.9% due to the fused low-level object features and high-level semantic features. Despite the fact that Faster R-CNN have showed great accuracies in generic object detection applications, they are often intractable for use in practical real-world small object detection. Taking our targeted pest detection in the wild as an example, designing an effective deep learning approach is extremely difficult due to many constraints: 1) The intuitive features of pest like texture, shape or color, are easily confused with background information 2). Features of tiny pest like rotation, scale and translation, are too weak and insensitive to be recognized. 3). Many deep learning approaches focus on solving classification of different pests, rather than pest detection (localization and counting). 4). Large variations of density distribution and sizes of tiny pests make the activation of some objects even smaller and insensitive with each pooling layer through a deep learning architecture. In order to overcome above obstacles, we attempt to propose a new effective deep learning approach towards large-scale multi-class pest monitoring by using hybrid global and local activated features.

III. Approach Overview

Our proposed approach is a two-stage CNN based pest detection and classification pipeline shown in Fig. 1. Two major stages in this approach are GaFPN for automatic multi-scale feature extraction and LaRPN for generated boxes classification and regression. Under the powerful global and local feature extracted, the output of our system consists of pest localization, classification and severity estimation tasks.

In the first stage of feature extraction, our system relies on traditional CNN backbone by introducing a new global activation feature pyramid network (GaFPN) which is aggregated on each convolutional block for screening and activating depth and spatial information from feature maps outputted by each block. Multi-scale image features extracted from GaFPN are used to re-build the feature maps. This design has two considerations: 1) Sufficient shallow layers enables mining more valuable semantic features for classification. 2) The bottom layers with high spatial information are fully utilized for avoiding some features vanish in deep block.

In the second stage, based on feature maps extracted from stage one, an improved local activated region proposal network (LaRPN) is proposed for providing region proposals and fully connected layers, which are adopted for pest classification and position regression. Different from the standard Region Proposal Network (RPN), we augment local contextualized and attentional information into region proposals for providing more efficient and precise regions.

Finally, we adopt several fully connected layers for the final pest localization and classification results in addition to high-level sematic analysis outputs for pest severity estimation including pest quantity counting and severity prediction. The entire training and inference phase run automatically to achieve effective pest recognition and classification without any human intervention so our method is an end-to-end system.

IV. Materials and Methods

A. Dataset Setup for Large-scale Multi-Class Pest Detection

To our best knowledge, while there exist some open insect databases released, no existing large-scale datasets that cover multiclass pests in the wild or nature environments are released for study yet. We establish our own dataset for large-scale multi-class pest monitoring by designing an industrial pest capture equipment shown in Fig. 2 This device uses multispectral light trap for attracting various types of pests, where the wavelengths vary with time according to the habit of pests in the day. Meanwhile, HD camera above the tray of this device is set to take pictures at 2592×1944 resolution periodically at 15-second intervals. Pests in the trays were swept away after photographing to avoid images containing 582,170 pests of 16 different types after manual screening to deleting obscure and over-occulted images are used to build our dataset.

Hereafter, images are labeled by agricultural experts with pest categories, localizations and severity. we randomly split entire collected images into 2 subsets for model training and validation respectively at ratio of 9:1, in which training subset could be the 'gold standard' to supervise our model because of labels with expert consensus and validation subset is used to evaluate our system's performance. The statistics of our dataset are provided in Table I.

B. Convolutional Neural Network (CNN) Framework

The approach built on a standard CNN framework is composed of three parts: convolutional layer, activation function and pooling layer. Typically, many combinations of these layers are adopted to extract 3D image features, in which images are input into convolutional layers computed with several convolutional kernels for feature extraction.

Standard convolutional layer takes a set of called convolutional kernels to the input and the output feature map in each subsequent layer are regarded as abstract transformations of image. Generally, for each kernel convolutional kernel k, the forward propagation process of convolution in layer l could be represented by:

$$a_k^l = \sigma(z_k^l) = \sigma(a^{l-1} * W_k^l + b^l) \tag{1}$$

where the a_k^l and a^{l-1} are output of kernel k from layer l and $l-1$. $\sigma(\cdot)$ is ReLU function for non-linear transformation

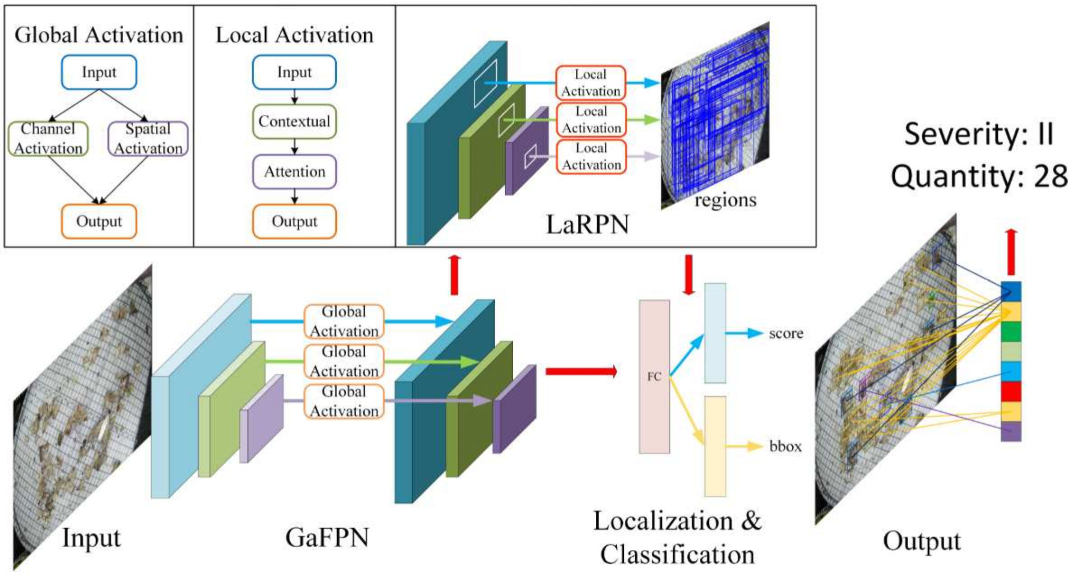


Fig. 1: Workflow of our two-stage CNN based approach

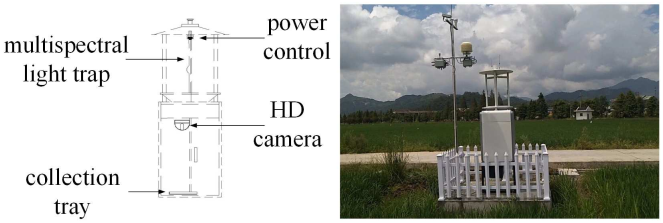


Fig. 2: Pest monitoring equipment in our work

in our approach. $*$indicate the convolution operation. W_k^l and b_k^l represent the convolution kernel and bias in layer l respectively. Therefore, the output convolutional layer could be computed as the sum of outputs from the filterbank:

$$a^l = \sigma(z^l) = \sigma(\sum_{k-1}^{M} z_k^l) = \sigma(\sum_{k=1}^{M} (a_k^l * W_k^l) + b^l) \tag{2}$$

C. Global activated Feature Pyramid Network (GaFPN)

Based on standard CNN architecture, we design our feature extraction network named Global Activated Feature Pyramid Network (GaFPN) whose structure is show in Fig. 3 The motivation of designing feature pyramid is the observation that recognizing pests at vastly different scales in images is challengeable for detectors in single feature map. Thus, we exploit the inherent multi-scale hierarchy of CNN to achieve feature map extraction at various scales to ensure that pests with different sizes are recognized with enough information and avoid missing features of some tiny pests in down-sampling operations. In GaFPN, the powerfully representative information from all convolutional blocks, including high-resolution levels and high-semantic levels, could be featurized to produce a multi-scale pest feature descriptor.

Different from the popular object detection framework FPN [13], our GaFPN makes full use of global information between each convolution block to avoid information loss during downsampling operation. As it is well known, feature maps outputted from CNN layers could be a result of convolutional operation with many kernels consisting of set of kernels. The number of kernels corresponds to be the feature depth and each kernel is learned to extract the specific type of feature such as shape and texture. Therefore, we attempt to make the model to automatically mine the depth activation vector while ignoring the effect of spatial information that could weigh the different kernels so influence the weights of feature maps' depth. As for position activation, the motivation is that limited receptive field of convolution operations lead to powerless features in pests' positions without appropriate supervision. So, we propose a novel supervised mask to learn the spatial activation vector that could activate the position points of objects. Therefore, our GaFPN is proposed to achieve depth and spatial activation in global level that could improve the feature discriminating power of pest objects.

Fig. 4 shows our intuitive overview of GaFPN structure, in which Global Activation Module (GAM) contains two branches for depth and spatial activation respectively. In the upper branch of depth activation, the 3D feature map with shape of $W \times H \times C$ output from corresponding CNN block is firstly processed by a global pooling layer that averages all the pixels in each channel (depth) and generates a lower dimensional (1D) feature vector ($1 \times 1 \times C$) so the effect of spatial information is eliminated. By taking global pooling, the averaged feature vector describes the global feature in depth level. Next, we apply two sets of fully connected layers with non-linear activation ReLU [31] and Sigmoid following respectively, in which the latter aims to map the feature vector into (0,1). So, the output 1D vector could be learned as depth activation factor in training phase and the final output of depth

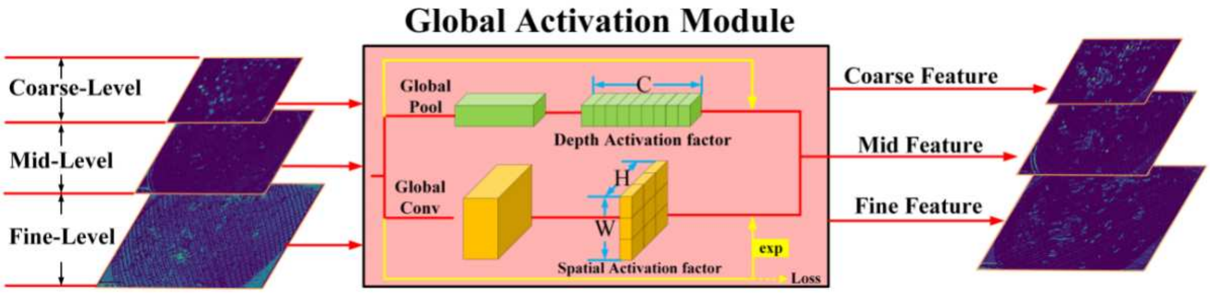


Fig. 3: Structure of Global activated Feature Pyramid Network (GaFPN)

TABLE I: Statistics on Two Subsets for our dataset with training subset and validation subset. For each class, the number of images and objects are shown in this table. Note that because single image may contain objects of several classes, the totals shown in the '#images' columns are not simply the sum of the corresponding columns. (CM: Cnaphalocrocis medinalis, CMw: Cnaphalocrocis medinalis (Walker), MS: Mythimna separate, HA: Helicoverpa armigera, OF: Ostrinia furnacalis, PL: Proxenus lepigone, SL: Spodoptera litura, SE: Spodoptera exigua, SI: Sesamia inferens, AI: Agrotis ipsilon, MB: Mamestra brassicae, HT: Hadula trifolii, HP: Holotrichia parallela, AC: Anomala corpulenta, GO: Gryllotalpa orientalis, AS: Agriotes subrittatus)

Pest name	ID	Training Subset		Validation Subset	
		#images	#objects	#images	#objects
CM	1	6,663	11,663	768	1,332
CMw	2	2,956	7,548	367	914
MS	3	11,280	23,055	1,222	2,741
HA	4	22,854	67,426	2,510	7,143
OF	5	17,586	39,126	1,950	4,190
PL	6	21,675	110,309	2,366	12,200
SL	7	7,301	9,857	782	1,079
SE	8	13,212	25,589	1,403	2,544
SI	9	5,136	7,645	583	830
AI	10	8,952	13,844	992	1,553
MB	11	6,389	9,345	719	1,065
HT	12	11,827	21,051	1,287	2,251
HP	13	8,905	30,792	963	3,460
AC	14	13,765	108,112	1,606	12,141
GO	15	9,632	17,432	1,038	2,056
AS	16	4,756	21,728	546	2,219
total		79,800	524,522	8,870	5,7648

activation module is the broadcast element-wise product of the input 3D feature maps ($W \times H \times C$) and 1D depth activation factor ($1 \times 1 \times C$). In this way, the feature maps are activated in depth.

The second branch of GAM in Fig. 4 is used for activating spatial position that introduces a novel supervised mask to learn a spatial activation vector. Specifically, the spatial activation branch is a segmentation-like branch, in which the supervised mask is obtained by fulfilling 1 into the ground truth positions and 0 into the background areas. In this part, the input feature map with shape of $W \times H \times C$ is input into a 'global convolution' operation that takes kernel to reduce the number of channels to 1 so the output is a $W \times H \times 1$ feature vector, which could ensure the spatial activation vector is learned in spatial level by supervised attention loss. In this method, we adopt pixel-wise sigmoid across entropy as the attention loss. Next, we employ two set of dilated convolution operations [32] with various kernel sizes (i.e. 5×5 and 7×7) that could relieve the drawback of limited receptive field. Similar to depth activation branch, the ReLU and Sigmoid are followed and the output spatial activation factor is learned to be a 2D feature vector whose values are in (0,1). At last, the learned spatial activation factor is fed into exponential operation and then dot with the input 3D feature maps in each position rather than naïve multiplication. In this way, it could maintain more context information while highlight the object information. Thus, our spatial activation could enhance the feature maps in pest objects area and diminish the opposition.

Finally, the output of each block in GaFPN is the sum of two activated feature maps and all of the outputs from blocks will be processed by LaRPN for pest region searching.

D. Local activated Region Proposal Network (LaRPN)

Our proposed system is an improvement on the Region Proposal Network by enhancing the region information in local level during box fine-tuning phase. We called our approach Local activated Region Proposal Network (LaRPN).

The first motivation of 'local activated' is that part of region proposals provided by standard RPN might not cover complete information of target objects. This would result in inaccurate box regression with insufficient features because RoI Align [33] is used to 'crop' the regions into local level from feature maps. To solve this problem, we augment some extra contextual information [15] to ensure enough object features could be considered into box regression. Secondly, the local spatial positions contribute to the pest regions classification because the key feature for precise region might be the fine-grained characteristics such as colors or shapes of pests' wings. Besides, rotational invariance should be ensured when our model is able to be sensitive to local spatial positions of pests.

Motivated by these observations, we propose an improvement of standard RPN named LaRPN to take contextual and attentional information into consideration to locally activate region proposals derived from RPN, whose structure is shown in Fig. 4. There are three steps in our LaRPN. Firstly, apply the

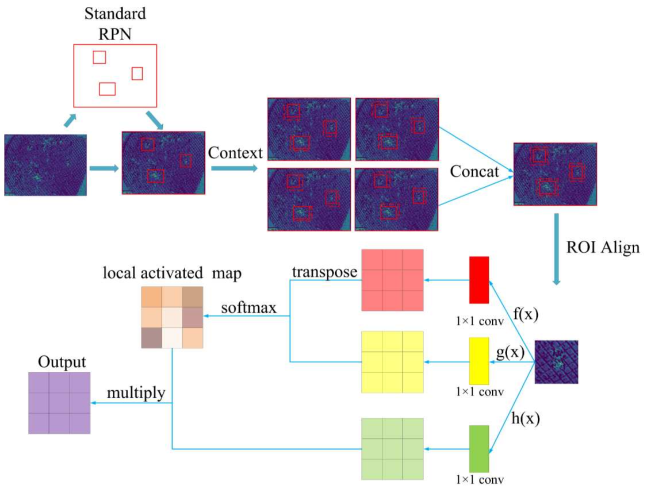


Fig. 4: Structure of Local activated Region Proposal Network (LaRPN)

standard RPN referenced by [12] in each output from GAM in GaFPN with our assigned anchors associated with every specific scale of feature pyramid structure. The aspect ratio for our anchors is set to be 1:1.5 because most of pests in our dataset are approximately square. During training phase, the anchors with Intersection-over-Union (IoU) to ground truth more than 0.7 are regarded as preliminary pest regions. Next, we expand these positive regions to be 1.5 times larger in four different directions to ensure the contextual regions could cover more complete information. And the enriched pest regions are mapped to feature maps and processed by RoI Align to be features. Thirdly, we introduce self-attention mechanism [34] with softmax activation function [35] to obtain the local attention vector in spatial level. Therefore, the relationships among different positions of pests could be learned and the output is multiplication of regions and spatial activated map. Finally, the output is used for pest classification and box finetuning.

E. Training and Evaluation

We use large-scale pest dataset for training and validating our proposed approach. Different loss functions are selected as supervisory indicators for pest localization, classification and estimation training. A number of evaluation metrics were built to access performance of our system on these tasks.

Pest Localization: Pest localization is a task to predict bounding boxes for each input image. To measure the performance of localization, we pay more attention on the positioning accuracy rather than categories of boxes. Therefore, we employ box regression loss as the criterion for pest localization task during training phase. Among various regression losses, we select smooth L1 loss as the loss function which is the combination of L1 and L2 norm so the gradient near 0 is smoother:

$$Loss_L = \sum_{i\in\{x,y,w,h\}} \begin{cases} 0.5(t_i-\hat{t}_i)^2 & \left|t_i-\hat{t}_i\right| < 1, \\ \left|t_i-\hat{t}_i\right| - 0.5 & \left|t_i-\hat{t}_i\right| \geq 1. \end{cases} \tag{3}$$

In this loss function, a region could be characterized by $\{t_x, t_y, t_w, t_h\}$ in which $\{t_x, t_y\}$ are the upper-left coordinates of boxes and $\{t_w, t_h\}$ are the width and height. Thus, t_i and $\hat{t}_i$ represent the ground truth and localized bounding boxes respectively.

In terms of metrics, binary precision and recall are chosen to evaluate the pest localization performance. During testing phase, the regions are predicted into two categories: non-background and background, in which non-background (positive) samples are the regions with overlap more than 0.7 with the ground truth bounding boxes while the other regions are background (negative). The Precision (Pr) and Recall (Re) are calculated by:

$$\mathrm{Pr} = \frac{\#\mathrm{TP}}{\#\mathrm{TP} + \#\mathrm{FP}} \tag{4}$$

$$\mathrm{Re} = \frac{\#\mathrm{TP}}{\#\mathrm{TP} + \#\mathrm{FN}} \tag{5}$$

in which $\#TP$, $\#FP$ and $\#FN$ represent the number of True Positive, False Positive and False Negative samples respectively so the Precision measures the samples that are incorrectly detected while higher Recall indicates the lower misdetection rate.

Furthermore, Average Precision (AP) for binary pest localization is applied as a comprehensive evaluation metric to fuse the Precision and Recall together. In localization task, the AP is computed by the integration of Precision-Recall (PR) curve:

$$AP_L = \int_0^1 \mathrm{Pr} d\mathrm{Re} \tag{6}$$

Pest Classification: while localizing pest objects in images, we classify each bounding box into the corresponding category. Different from binary classification in LaRPN (foreground or background), the bounding boxes are classified into 16 types that are the major categories of pests we target to monitor in our approach. In this task, we use multi-class cross-entropy loss for this pest classification problem:

$$Loss_C = \sum_{i=1}^{N_{cls}} -y_i \log(\hat{y}_i) \tag{7}$$

where N_{cls} represents the number of pest categories (in our task, N_{cls} = 16). y_i and $\hat{y}_i$ indicate the truth label and predicted category respectively. From the perspective of evaluation metrics for pest classification, AP value [16] is updated for different categories and we combine localization and classification validation methods together. Thus, in our system, we calculate APs for 16 categories based on the corresponding PR curve as:

$$AP(c) = \int_0^1 \mathrm{Pr(c)} d\mathrm{Re(c)} \tag{8}$$

In addition, the final metric for pest classification task, mAP is obtained by taking the mean of APs with all the classes:

$$\mathrm{m}AP = \frac{1}{N_{cls}} \sum AP(c) \tag{9}$$

Pest severity estimation: the high-level task, pest severity estimation targets at predicting the severity of pest occurrence from the input image. According to agricultural experts' consensus, the severities are divided into 5 levels from 'general' to 'serious' that describes the occurrence of pests in the field, so the images are labeled to I-V by experts after image acquisition. In the process of pest severity prediction, the input features are the combined results from localization and classification tasks above. In terms of encoding method, we adopt a variant of one-hot encoder to transform the pest detection results into N_{cls}-dimensional vector, where each element in this vector indicates the number of detected pests with corresponding category. In this input vector, we only focus on the quantity of detected pests from each category rather than their positions.

In pest severity estimation task, we build consequent two FC layers for feature extraction and softmax predictor for severity estimation. As criterion, we employ a weighted multi-class cross-entropy loss defined as:

$$Loss_E = \sum\nolimits_{i=1}^{N} -\lambda_i y_i \log(\hat{y}_i) \tag{10}$$

where λ_i is parameter to weight the loss function which measures the risk of different misclassification samples. We define the risk parameter λ_i as the difference between predicted severity and truth severity. As for evaluation, we consider total accuracy as evaluation metric for pest estimation task.

V. Experimental Results and Discussion

A. Implementation Details

We use Inception [36] and ResNet50 [37] as CNN backbones to train our pest monitoring model and also build some experiments to evaluate the performance of our system. During the training phase, The proposed method and other state-of-the-art approaches are trained via back-propagation and stochastic gradient descent (SGD), with momentum 0.9 [38] and initialize learning rate to 0.001 that will be dropped by 10 at 8th and 11th epoch followed by [39]. The mini-batch size is set to 4 in training phase. In terms of weight initialization, we adopt transfer learning [40] that copy the CNN backbones' weights pre-trained on ImageNet dataset [41]. In order to avoid over-fitting problem, we utilize early-stopping strategy [42] to select the best training iteration. We conduct our experiments using two GTX1080Ti GPUs with 12GB memory. The performance of our approach is evaluated on our built dataset across multiple tasks: pest localization, classification and severity estimation.

B. Pest Localization Task

For pest localization task, we present the experimental results in Table II, in which we compare our method with two state-of-the-art approaches Faster R-CNN [12] and FPN [13] that are the base detectors we attempt to improve using our proposed techniques. Because localization task is evaluated on regions accuracy alone, the AP$_L$ does not take categories into consideration. As it can be observed, our proposed method could dramatically surpass the localization performance of Faster R-CNN using different CNN backbones for feature extraction, which achieves 5.12% and 4.87% AP$_L$ improvement. Besides, compared with another feature pyramid method FPN, our system could also obtain a slight improvement in pest localization task. Among these results of our method, the best performance occurs in ResNet50 backbone which achieves localization accuracy with 83.61% AP$_L$.

TABLE II: Pest Localization Results AP$_L$

CNN Backbone	Method	AP$_L$
Inception [36]	Faster R-CNN [12]	74.99%
	FPN [13]	76.65%
	Ours	80.11%
ResNet50 [37]	Faster R-CNN [12]	78.74%
	FPN [13]	80.29%
	Ours	83.61%

It is interesting to note the detailed pest localization performance between our approach and other state-of-the-art methods in Fig. 5 which shows the PR curve of various networks. Obviously, our proposed global and local activated approach outperforms Faster R-CNN by a large margin and improves FPN slightly. This improvement could be contributed to two reasons. Firstly, our method with GaFPN applies a pyramid feature extraction architecture and localize pests' regions on multi-level feature maps that could help precisely find pests positions on various scales, which is also evidence from AP$_L$ values of our method in Fig. 6. Secondly, holding global activation factors by our presented global activated features for enhancing the depth and spatial information in global level makes it easier to localize pests' positions because of much more remarkable features between foreground and background.

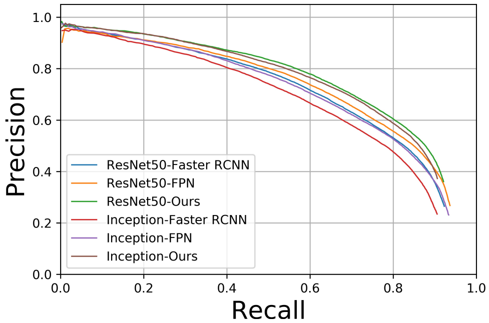


Fig. 5: Precision-Recall curve for pest localization

C. Pest Classification Task

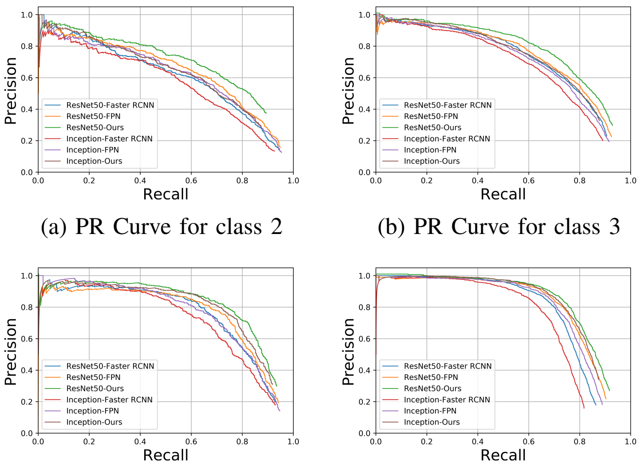
For pest classification task, we show the experimental results in Table III that presents the AP for 16 pest categories performed by our method and other state-of-the-art models. Observed from Table III, having pest localization information associated with the predicted bounding boxes to pests, our method could achieve more accurate pest recognition on these classes. It is obvious that our approach could significantly outperform Faster R-CNN in pest classification over almost all the pest categories under Inception as CNN backbone. The homologous phenomenon occurs in that using ResNet50 network with 3.28% mAP improvement. In addition, our approach could also largely improve mAP compared to another feature pyramid object detection structure FPN. This gain is largely due to our LARPN's ability to introduce the contextual and local activated information before fully connected layers for pest classification, which is helpful to sufficiently learn the features of pests in local level.

TABLE III: Pest Classification Task Results AP value (%). FR-CNN indicates Faster R-CNN method.

Pest ID	Inception [36]			ResNet50 [37]		
	FR-CNN [12]	FPN [13]	Ours	FR-CNN [12]	FPN [13]	Ours
1	51.62	60.24	63.91	57.12	62.13	64.24
2	56.26	61.00	63.11	59.70	62.96	65.98
3	64.27	67.33	68.85	69.75	70.16	73.96
4	80.74	82.10	84.37	83.73	82.82	85.68
5	65.65	69.73	74.39	70.17	71.22	76.48
6	65.36	68.45	69.38	68.60	68.98	70.32
7	63.09	63.30	66.76	68.39	69.46	70.43
8	45.31	49.70	53.43	48.57	53.47	54.19
9	69.93	71.17	76.57	72.56	72.91	77.81
10	75.55	76.27	79.33	79.92	80.58	81.13
11	50.71	51.74	57.04	54.45	57.35	64.02
12	63.17	66.78	68.17	66.26	69.20	71.15
13	77.48	83.31	85.38	84.94	85.18	86.65
14	79.43	86.93	88.51	87.86	88.03	88.76
15	89.81	89.77	90.27	89.93	89.97	90.31
16	69.13	72.51	76.46	73.38	74.37	79.40
mean	66.72	70.02	72.87	70.96	72.42	75.03

Apart from mAP results, there are obvious differences within classes that can be seen in Table III. Specifically, pest #8 seems to be the most difficult to be categorized on these pre-calculated regions with lowest AP value while almost all the models could classify pest #15 well even using shallow CNN backbone. This can be explained by that the pests in the 'easy' class hold up a large number of training examples, which help reduce difficulty to classify them comparing Table III and Table I. Even though, the amount of data might not be the main factor affecting performance of our approach, where pest #16 still could be categorized with a large AP value (more than 78% in our method) even if there are only 4,756 training images containing pests of this class. Therefore, our method could largely overcome the sample limitation and imbalance problem with a great improvement.

Fig. 6 illustrates some of PR curves in our experiments. Note that only four classes PR curves are shown here due to the space limitation. As it is shown, precision could keep a high value with the recall increasing in various models. Especially, our approach using different CNN backbones could obtain a larger precision and recall compared to Faster R-CNN, which indicates that it could effectively reduce false positive rate as well as misdetections rate. Concretely speaking, pest #2 is relatively difficult to classify so the PR curve for this class is further away from the point (1,1). In addition, PR curve for pest #16 represents that it is hard to obtain a high recall value but could get satisfied precision value so this curve signifies that our system could make sure that almost all the detected insects of this class are correct but might not detect all of the insects. Furthermore, among these illustrated PR curves, our system performs best on class #3 that maintains high precision in addition to recall simultaneously.



(a) PR Curve for class 2

(b) PR Curve for class 3

(c) PR Curve for class 9

(d) PR Curve for class 16

Fig. 6: Illustration of PR curves in our experiments. Note that only 4 curves are shown here for limited space.

D. Pest severity estimation Task

For pest severity estimation, our method regards this task as a classification problem so we achieve severity estimation based on the encoded results outputted from previous pest localization and classification tasks. So we compare our severity estimation predictor with the state-of-the-art CNN based models that estimate severity by softmax classifier using the whole image as input. Table IV illustrates the comparable results in our experiments. As it is shown, our method could beat theses CNN approaches with approximately 2% classification accuracy improvement due to the prior information from detected pests.

TABLE IV: Pest severity estimation Task Results Accuracy

CNN Backbone	Method	Accuracy
Inception [36]	Softmax [35] ours	80.5% 83.0%
ResNet50 [37]	Softmax [35] ours	84.9% 86.9%

E. Result Visualization

We visualize part of the pest monitoring results in Fig. 7 that fuses localization, recognition and severity estimation tasks together. These results are outputted by our system based on ResNet50 backbone. The environments of input images from top to bottom are more and more complicated. As it can be seen, our method could achieve multi-class pest localization and recognition under both simple and complicated environments and provide the predicted severity estimation, despite the intractable challenges such as noisy image and tiny objects. Some feature maps outputted from 2 middle blocks with FPN (left) and our method (right) using ResNet50 are visualized in Fig. 8. It is found that, the feature maps in our system diminish the highlights of non-objects and focus more attention on pest regions with lighter activation points. Therefore, our method could perform better on pest detection and progressively learn the pests' features well.

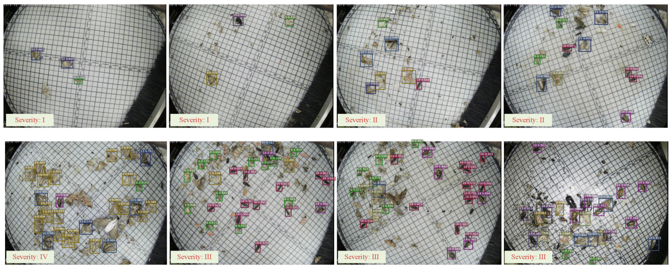
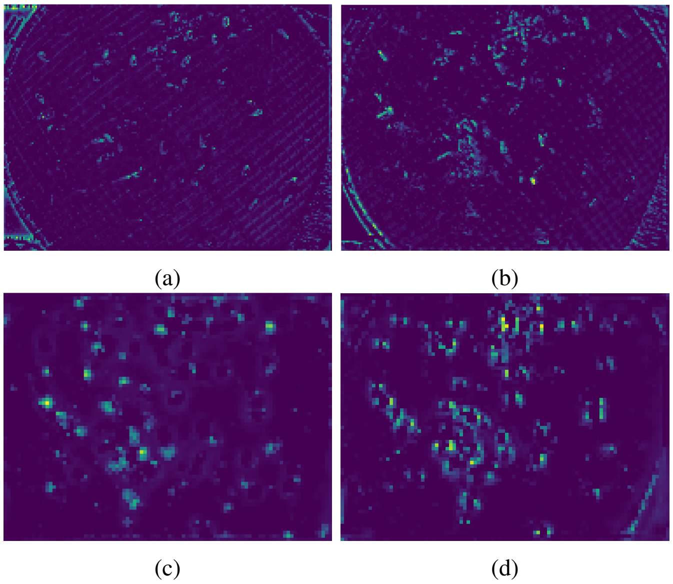


Fig. 7: Examples of pest monitoring results demonstration



(a)

(b)

(c)

(d)

Fig. 8: Part of feature maps generated by FPN (left) and our method. (a) From shallow block by FPN. (b) From shallow block by our method. (c) From deep block by FPN. (d) From deep block by FPN.

F. Future Work

Despite that we develop a novel deep learning based system for pest monitoring task in the field and achieve a successful performance in our dataset, there are several limitations of our method that could be improved in future smart agriculture innovation. Firstly, the unbalanced data structure could be alleviated in the next work. Specifically, due to the difficulty in capturing pests of some rare categories in our pest monitoring equipment, our system tends to identity an unknown pest into the common species, which might improve the risk of inaccurate pest severity warning. Besides, it is necessary to achieve the real-time pest image recognition and detection performance in our system, in which current inference time might be an important factor that limits the advances in agricultural applications. Therefore, future work would target at solving the problem of unbalanced dataset and focus on developing real-time automatic pest monitoring system.

VI. Conclusion

This paper proposes a novel deep learning approach using hybrid global and local activated features for automatic pest monitoring in industrial equipment to simultaneously perform three key tasks: localization, classification and severity estimation. Our method successfully realizes efficient and automatic feature extraction with global activated feature pyramid GaFPN structure. Furthermore, we present local activation to enhance position-sensitive features of pest boxes by LaRPN for powerful regions proposal. Under our enriched stationary pest dataset captured by our designed pest monitoring equipment, our method has outperformed the state-of-the-art methods in pest localization, classification and severity estimation tasks. Future work will consider developing more efficient deep learning architecture for real-time pest monitoring.

References

[1] G. D. Santangelo, "The impact of fdi in land in agriculture in developing countries on host country food security," *Journal of World Business*, vol. 53, no. 1, pp. 75–84, 2018.

[2] B. L. Bures, K. V. Donohue, R. M. Roe, and M. A. Bourham, "Non-chemical dielectric barrier discharge treatment as a method of insect control," *IEEE transactions on plasma science*, vol. 34, no. 1, pp. 55–62, 2006.

[3] H. Liu, S.-H. Lee, and J. S. Chahl, "A multispectral 3-d vision system for invertebrate detection on crops," *IEEE Sensors Journal*, vol. 17, no. 22, pp. 7502–7515, 2017.

[4] R. Berenstein and Y. Edan, "Automatic adjustable spraying device for site-specific agricultural application," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 641–650, 2017.

[5] S. B. Sulistyo, W. L. Woo, and S. S. Dlay, "Regularized neural networks fusion and genetic algorithm based on-field nitrogen status estimation of wheat plants," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 103–114, 2016.

[6] J. Luo, W. Huang, J. Zhao, J. Zhang, C. Zhao, and R. Ma, "Detecting aphid density of winter wheat leaf using hyperspectral measurements," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 2, pp. 690–698, 2013.

[7] W. Ding and G. Taylor, "Automatic moth detection from trap images for pest management," *Computers and Electronics in Agriculture*, vol. 123, pp. 17–28, 2016.

[8] I. Zayas and P. W. Flinn, "Detection of insects in bulkwheat samples with machine vision," *Transactions of the ASAE*, vol. 41, no. 3, p. 883, 1998.

[9] J. Cho, J. Choi, M. Qiao, C.-w. Ji, H.-y. Kim, K.-b. Uhm, and T.-s. Chon, "Automatic identification of whiteflies, aphids and thrips in greenhouse based on image analysis," *Red*, vol. 346, no. 246, p. 244, 2007.

[10] C. Wen, D. E. Guyer, and W. Li, "Local feature-based identification and classification for orchard insects," *Biosystems engineering*, vol. 104, no. 3, pp. 299–307, 2009.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[13] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[14] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[15] P. Hu and D. Ramanan, "Finding tiny faces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 951–959.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[17] J. Ruan, H. Jiang, X. Li, Y. Shi, F. T. Chan, and W. Rao, "A granular ga-svm predictor for big data in agricultural cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6510–6521, 2019.

[18] P. J. Weeks, M. A. O'Neill, K. Gaston, and I. Gauld, "Species–identification of wasps using principal component associative memories," *Image and Vision Computing*, vol. 17, no. 12, pp. 861–866, 1999.

[19] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 761–769.

[20] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 2008, pp. 722–729.

[21] L. O. Solis-Sánchez, R. Castañeda-Miranda, J. J. García-Escalante, I. Torres-Pacheco, R. G. Guevara-González, C. L. Castañeda-Miranda, and P. D. Alaniz-Lumbreras, "Scale invariant feature approach for insect monitoring," *Computers and electronics in agriculture*, vol. 75, no. 1, pp. 92–99, 2011.

[22] N. Larios, B. Soran, L. G. Shapiro, G. Martinez-Munoz, J. Lin, and T. G. Dietterich, "Haar random forest features and svm spatial matching kernel for stonefly species identification," in *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 2624–2627.

[23] X.-L. Li, S.-G. Huang, M.-Q. Zhou, and G.-H. Geng, "Knn-spectral regression lda for insect recognition," in *2009 First International Conference on Information Science and Engineering*. IEEE, 2009, pp. 1315–1318.

[24] Y. Kaya and L. Kayci, "Application of artificial neural network for automatic detection of butterfly species using color and texture features," *The visual computer*, vol. 30, no. 1, pp. 71–79, 2014.

[25] Q. Zhu, Z. Chen, and Y. C. Soh, "A novel semisupervised deep learning method for human activity recognition," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 3821–3830, 2018.

[26] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, "Automatic fruit classification using deep learning for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1027–1034, 2018.

[27] Q. Zhang, L. T. Yang, Z. Yan, Z. Chen, and P. Li, "An efficient deep learning model to predict cloud workload for industry informatics," *IEEE transactions on industrial informatics*, vol. 14, no. 7, pp. 3170–3178, 2018.

[28] J. Su, D. Yi, B. Bu, Z. Mi, C. Liu, X. Xu, X. Xu, L. Guo, and W. Chen, "Aerial visual perception in smart farming: Field study of wheat yellow rust monitoring," *IEEE Transactions on Industrial Informatics*, 2020.

[29] P. Li, Z. Chen, L. T. Yang, Q. Zhang, and M. J. Deen, "Deep convolutional computation model for feature learning on big data in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 790–798, 2017.

[30] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[31] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[32] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 1451–1460.

[33] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[34] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *arXiv preprint arXiv:1805.08318*, 2018.

[35] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[38] L. Bottou, "Stochastic gradient descent tricks," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 421–436.

[39] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[40] H. K. Suh, J. IJsselmuiden, J. W. Hofstee, and E. J. van Henten, "Transfer learning for the classification of sugar beet and volunteer potato under field conditions," *Biosystems engineering*, vol. 174, pp. 50–65, 2018.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[42] R. Caruana, S. Lawrence, and C. L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *Advances in neural information processing systems*, 2001, pp. 402–408.