TrichANet: An Attentive Network for Trichogramma Classification

First Author Name[1][a], Second Author Name[1][b] and Third Author Name[2][c]
[1]*Institute of Problem Solving, XYZ University, My Street, MyTown, MyCountry*
[2]*Department of Computing, Main University, MySecondTown, MyCountry*
{*f_author, s_author*}*@ips.xyz.edu, t_author@dc.mu.edu*

Keywords: Trich Classification, Trich Detection, Multi-Scale Attention

Abstract: *Trichogramma* wasp classification has a significant application in agricultural research, thanks to their massive usage and production in cropping as a bio-control agent. However, classifying these tiny species is a challenging task due to two factors: (i) Detection of these tiny wasps (barely visible with the naked eyes), (ii) Less inter-species discriminative visual features. To combat this, we propose a robust method to detect and classify the wasps from high-resolution images. The proposed method is enabled by a trich detection module that can be plugged into any competitive object detector for improved wasp detection. Further, we propose a multi-scale attention block to encode the inter-species discriminative representation by exploiting the coarse and fine-level morphological structure of the wasps for enhanced wasps classification. The proposed method along with its two key modules is validated in an in-house *Trich* dataset and a classification performance gain of 4% compared to recently reported baseline approaches outlines the robustness of our method.

1 Introduction

Trichogramma (Trich) are one of the smallest parasitic species in the world (<0.5mm), widely used as a biocontrol agent (BCA) to protect crops from pest attacks. They lay and develop their own eggs inside the eggs of harmful insects to trigger the death of harmful ones. For this, *Trichogramma* are produced and used on an industrial scale as an alternative to chemicals in different cropping systems (*i.e., maize fields, tomato-producing greenhouses, etc.*). Thus for optimal pest control, it is essential to analyse their behaviour and movement to ensure proper distribution in crop-fields. But this is difficult for a casual observer in real-world setting due to their minute size. With the recent development of tiny object analysis (Gong et al., 2021), (Lee et al., 2022), (Yang et al., 2022) and Multi-Object Tracking (MOT) (Aharon et al., 2022), (Zhang et al., 2021) in the computer vision domain, a new research direction has opened up to analyse and classify the *Trichogramma* in agricultural research.

To classify the Trich, the essential step lies in detecting these individuals from the observation arena. The detection remains challenging due to (i) Tiny size of the Trich, (ii) Egg patches in the arena as shown in Figure 1. From initial experimentation, it is found that recent popular object detection methods (Ge et al., 2021; Pani et al., 2021) result in many false positives with fewer true positive detection in these challenging scenarios. This failure case is due to the unavailability of a fully annotated dataset, and consequently, the object detectors could not be fine-tuned for the Trich detection task. To combat this, we propose a simple and effective Trich Detection module that is empowered by segmentation of the species from the arena by removing the egg patches and the noise from the background followed by pre-trained object detectors to detect the Trich.

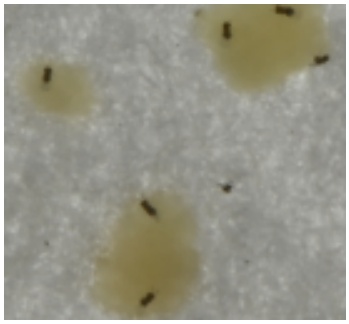

Figure 1: Sample *Trichogramma* captured in a observation arena over egg patches. The insects are in black, while the greenish-yellow patches are the pest eggs.

Upon successful detection of the Trich, the clas-

[a] https://orcid.org/0000-0000-0000-0000
[b] https://orcid.org/0000-0000-0000-0000
[c] https://orcid.org/0000-0000-0000-0000

sification of species remains challenging due to the subtle differences in the spatial cue among the categories. With the recent success of the Vision Transformers (ViTs (Dosovitskiy et al., 2020)) over the ConvNets (He et al., 2016), (Howard et al., 2017), (Krizhevsky et al., 2017), (Simonyan and Zisserman, 2014), (Szegedy et al., 2016), (Tan and Le, 2019), we empirically found that the patch-based relation encoder ViT is capable of providing superior representation than that of ConvNets for various species. However, the performance still remains limited due to the existence of a domain gap between the ViT pre-training (i.e., ImageNet (Deng et al., 2009)) and the target task (Trich classification). Again, due to the limited number of samples in the target dataset, it is non-feasible to fine-tune a high capacity model like ViT. Further, we analyze that there exists a subtle change in spatial cues among the species, which makes the classification more challenging. For this, we propose a Multi-scale Attention (MSA) block to encode the discriminative features between the species by analyzing their features at multiple scales. The discriminability is MSA is encoded by emphasizing on the salient spatial regions on a global scale and suppressing the redundant regions. The proposed MSA block adopts a head-only learning paradigm which is stacked with the ViT feature encoder to train on the target task. We refer our designed network as *TrichANet*, which enables Trichogramma classification in an attentive manner. To validate the robustness of TrichANet, we conduct experiments on an in-house Trich dataset and found that it surpasses the baseline methods by a significant margin.

In summary, the key contribution of the work is in three-folds:

1. First, a simple and effective Trich Detection method is proposed to detect the Trich individuals with lower false positives.
2. Second, a generalized TrichANet is proposed in this work for effectively classifying Trich individuals in high-resolution images.
3. To showcase the robustness of each building block in TrichANet, an extensive experimental study is carried out with significant qualitative and quantitative analysis.

2 Related Works

Our work would fall within the domain of tiny object detection and classification. The number of available work in these domains is limited, in spite of the huge scopes.

Tiny Object Detection An interactive object detection module has been reported in the work of Lee *et al.* (Lee et al., 2022), in which user input annotations of some objects are processed and both Late Fusion and Classwise Collated Correlation for the local and glocal context scales for the detection of tiny of various classes and different instances. These are then concatenated channelwise, to obtain the final output detections. In the work of Yang *et al.* (Yang et al., 2022), possible locations of objects are initially predicted from low resolution feature maps, and a sparse feature map from these values are obtained at these regions from the high resolution features. Then, a detector outputs the detections from these feature maps. The entire pipeline is connected in a cascaded manner, to allow for faster and accurate detections. Using a FPN backbone, a customised region proposal network has been proposed in (Qin et al., 2020), to generate rotated proposals at various scales, which are then aligned with the original images. Features are then sampled from it and fed to a network to reduce misalignments and output the final localizations using three detector heads with different structures. In the pipeline reported in (Yi et al., 2021), an U-shaped network with long skip connections are used to obtain four outputs - heat map, offset map, box parameter map and orientation map, from which the center of the bounding boxes are inferred, and the boundary aware vectors are learned, from which the bounding box corners are inferred. Han *et al.* (Han et al., 2021) propose a rotation equivariant architecture to extract rotation equivariant features. They then extract the rotation invariant features by RRoI warping. Thus, their proposed method can extract rotation invariant features in both spatial and orientation dimensions.

Tiny Object Classification A multi-staged module is proposed by Kong *et al.* (Kong and Henao, 2022), in which the first stage generates an attention map from the downscaled input image, from which regions are sampled with replacement. Then, in the second stage, another attention network generates attention maps for each region and samples sub-regions from the regions sampled previously. These sub-regions are finally fed to a feature extractor, and the feature maps are aggregated using the corresponding attention weights, and predictions are obtained using a classification module. The attention networks are also used to sample contrastive examples, during training. A simple, light-weight model for tiny object recognition has been proposed in (Dat et al., 2018). It consists of five convolutional layers, with filters having receptive fields of 3×3, with ReLU activations between each layer, to introduce non-linearity. Batch normal-

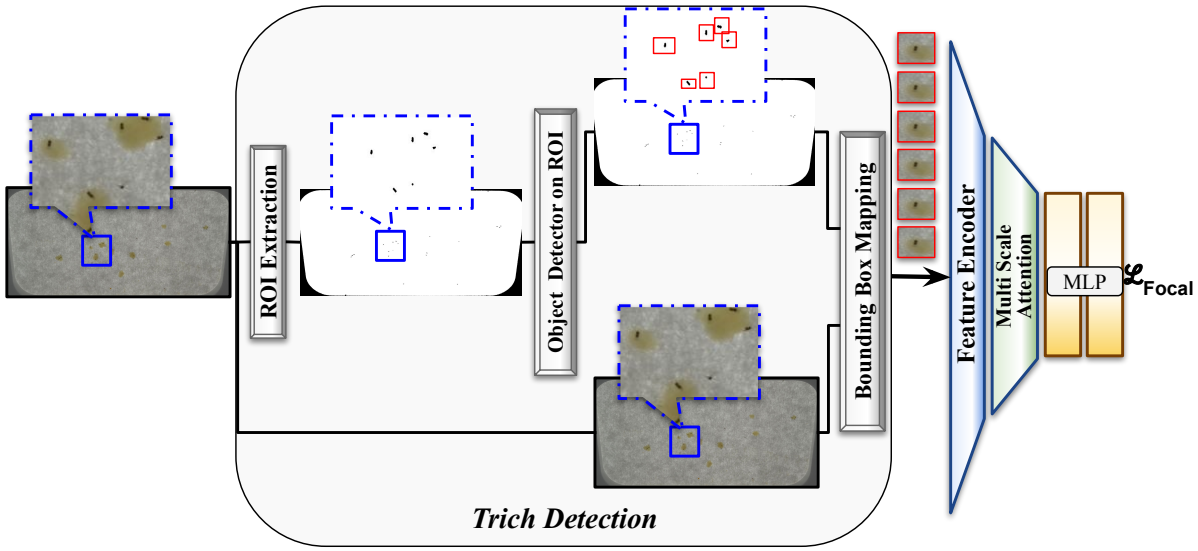

Figure 2: **Proposed Method:** First, pre-processing is done on the input HR images, after which it's broken into patches and passed through a pre-trained YOLOX detector. The detections obtained are then passed through a pre-trained ViT encoder, followed by the proposed MSGA module, and finally a classification head, to obtain the class probability matrix.

ization is also used to speed up training, and dropout is used for regularization.

Multi-Object Tracking The authors of (Cao et al., 2022) have proposed using the momentum of the object in the association stage, developing a pipeline with less noise and more robustness in occlusion and erratic motion. They also add a separate observation term in the association cost, and also include a recovery module to search for lost objects around the time of their last detection. Motion and appearance information have been combined, along with camera-motion compensation and a new Kalman filter state vector for better box localization, in the tracker reported in (Aharon et al., 2022). They also present a new method to fuse IoU and ReID's cosine-distance for better association between detections and tracklets. Instead of simply discarding the detection boxes with score below the pre-determined threshold, the authors in (Zhang et al., 2021) propose tracking these detection boxes by association. The similarities with tracklets are analysed for low score detection boxes, to recover true objects are remove background detections. In order to improve from DeepSORT (Wojke et al., 2017), the authors of (Du et al., 2022) by using a new appearance feature extractor and a newer backbone architecture, to extract much more discriminating features. Also, the feature bank is replaced with a feature extraction strategy. Camera motion compensation is also added in the motion branch, and the vanilla Kalman algorithm is replaced with the NSA Kalman algorithm. Finally, thr assignment problem is solved with both appearance and motion information.

3 TrichANet

The overview of the *TrichANet* is shown in Figure 2. It can be seen that *TrichANet* has four modules *i.e. (i) Trich Detection, (ii) Feature Encoder, (iii) Multi-Scale Attention and (iv) MLP Head*, that are sequentially executed to achieve *Trichogramma* classification. A detailed description of each module is presented in the following subsections.

3.1 Trich Detection

The primary goal of this module is to detect the tiny trich individuals present in the observatory arena. The proposed trich detection module comprising of three steps (*i.e., ROI Extraction, Object Detector, BBOX Mapping*) is presented in Figure 2.

ROI Extraction: It enables the effective segmentation of *Trich* wasps from the observatory arena by eliminating the egg patches and background noise, as shown in the Figure 3. For this, first a dynamic thresholding operation is performed on the input images using the Otsu Algorithm with a fixed offset. It can be visualized from Figure 3 that the output of otsu thresholding results in a few noises along with the wasps. From analysis, we found that the noises are structurally smaller than the wasps. Thus, to eliminate them completely while preserving the original

wasps' structure, a spatial contact-expand operation is applied on the binary images. The spatial contact-expand operator is essentially three consecutive morphological erosion operation to eliminate the noise in the image plane followed by three consecutive dilation operations to regain the original size of trich wasps. For both erosion and dilation a 5×5 structuring element is used.

Object Detector on ROI: A pre-trained object detector YOLOX is considered here to detect the trich on the ROI extracted binary images. Further, to ease in the detection, the high-resolution 8256×5504 binary images are split into 224×224 patches. For each patch, YOLOX is applied parallely to obtain the detection bounding boxes in the binary images. The resultant of this step is the set of n (Where, $n =$ no. of wasps present in observatory arena) bounding boxes that precisely detect the tiny wasps in the ROI extracted binary image.

RGB Trich Extraction: Since there exists no structural distinction between the *trich* wasps categories, it is non-trivial to classify the binary detected wasps. For this, we aim at extracting the RGB trich patches by mapping the bounding box coordinates obtained from ROI images to that of original RGB images. So, for a given image I containing n trich wasps, this step outputs a trich image map $I_T \in \mathbb{R}^{n \times h \times w}$, where $h = h_1, h_2, \ldots, h_n$ and $w = w_1, w_2, \ldots, w_n$. Since the shape of the bounding boxes are non-identical, a `NULL-padding` operation is performed across h and w to maintain the homogeneity in the bounding box shape while preserving the original trich resolution. Next, the resultant I_T is considered in the following steps to classify the individual trich into pre-defined set of categories.

3.2 Feature Encoder

In order to categorize the trich wasps, at first, feature extraction is carried out for each trich wasp image in I_T from a feature encoder. With the recent popularity of the Vision Transformers (ViTs) over the ConvNets and due to its patch-based relation encoding, giving rise to superior representation than ConvNets, we adopt a pre-trained ViT as the feature encoder. Since there exists a scarcity of sufficient samples in trich classification task, we found that it is non-trivial to train or fine-tune ViT. For this, the pre-trained weights from the ImageNet dataset for ViT are considered for extracting D dimensional feature representation for each trich wasp. So for a given $I_T \in \mathbb{R}^{n \times h \times w}$, the feature encoder outputs a feature map $F \in \mathbb{R}^{n \times D}$.

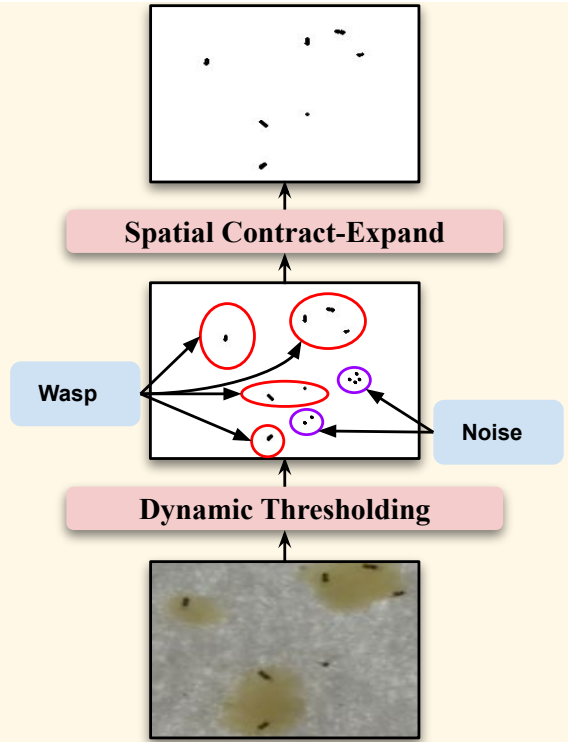

Figure 3: **ROI Extraction:** First, dynamic thresholding is performed to dissociate the foreground (*i.e., wasps*) from the background (*i.e., the arena and eggs*) from the original image. Next, a spatial contraction-expansion operation is performed in the binary image to remove the additional noises.

3.3 Multi-Scale Attention Module

In order to obtain discriminative representation among the wasp categories, it's necessary to capture the changes in coarse and fine-grained spatial features. Coarse level feature variations can be encoded by 1D convolution operation with higher receptive field (RF) (*i.e.* RF = 5). In contrast, variations in fine-grained features can be encoded by 1D convolution operation with lower receptive field(*i.e.* RF = 3). So to effectively encode the change in coarse and fine-grained spatial cues for discriminative representation, a multi-scale attention module (MSAM) is proposed.

As shown in Figure 4, MSAM inputs the feature map $F_i \in \mathbb{R}^{1 \times D}$ of i^{th} trich (where $i = 1, 2, \ldots, n$) extracted from the feature encoder. Next, MSAM projects the F_i to three latent space (*i.e* key (K), query (Q), and value (V)) with multiple feature scales. The multiple feature scales are achieved by varying the RF (*i.e* RF=3,5) of the latent projection to encode the coarse and fine-grained spatial cues. In MSAM, K and Q latent projections have similar RF=5, which is obtained from two sequential Conv1D layers. But, V

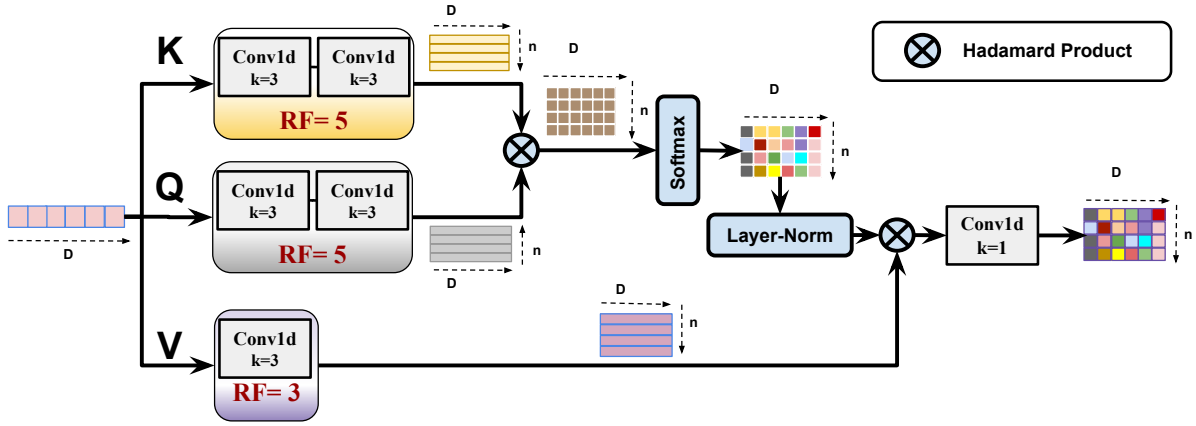

Figure 4: **Proposed Multi-Scale Attention Module:** It generates a multi-scale attentive feature map by projecting the input feature map to different convolutional receptive fields (*i.e.* RF=3,5). Here RF=5 projection encodes the coarse-level contextual feature, whereas RF=3 projection provides the fine-grained cue. Hence an attentive feature map generated from such a coarse-fine encoding ensures to capture discriminability in spatial cues across species categories.

has RF=3 that is obtained from a single Conv1D layer. The kernel size (k) in all the convolution layers of K, Q, and V is set to 3 and the number of convolution filters applied is N. Now `hadamard product` is applied between the $D \times N$ dimensional output feature map obtained from K and Q followed by a *softmax* activation to generate the attention mask (A). Further, the attention mask (A) is normalized and multiplied with V by a `hadamard product`. The resultant is then applied to a single Conv1D layer with k=1 to obtain the $D \times N$ dimensional attentive feature map (F_A). Subsequently, F_A is `flattened` and passed to the MLP head for classification.

3.4 MLP Head

The MLP head consists of two MLP layers of decreasing number of hidden units. The last layer is activated with *softmax* activation to obtain the class probabilities.

3.5 Network Optimization

The proposed TrichANet is end-to-end trainable excluding the trich detection and feature encoder blocks. In order to optimize TrichANet, the Focal Loss function is used as formulated below,

$$\mathcal{L}_{focal} = -\alpha_t(1-p_t)^{\gamma} log(p_t) \quad (1)$$

where, p_t is a measure of prediction accuracy, and thus, the loss is decreased for examples that are predicted better. This is to ensure that the model doesn't over-focus on the 'easier' examples, and thus, produce a skewed confusion matrix, the Focal Loss was used as the objective function to ensure greater weightage for 'harder' samples.

4 Experiments

4.1 Dataset

The experiments are conducted on a in-house *Trich* dataset that comprises of 518 number of high-resolution raw images belonging to two trich categories (*i.e.* TB, TE). Out of 518 raw images, 454 and 64 raw images are from TB and TE categories respectively. The images collected in *Trich* dataset follow a definite image acquisition step as discussed below.

Image Acquisition: The image acquisition is done with a NIKON © Z7 camera which captures the images with high-resolution (*i.e.* 8256×5504) but at a low frequency (*i.e.* a shot is taken in every 10 seconds for 5 minutes). The settings were: ISO 250; diaphragm aperture F/22; shutter speed 1/160. In order to ensure sufficient lighting, the observation arenas were placed on a LED plate and surrounded by a lightbox. A ventilation system was installed to avoid overheating, resulting in a temperature of 28 ± 0.8 °C.

Train-Test Protocol: For a trivial train-test protocol, first, the Trich Detection method is applied in the raw images. From this, a total of 10659 trich species were obtained out of which 9558 and 1101 belong to the TB and TE classes respectively. Then, we follow 94-6% split for train-test. Thus, the training dataset consists of 10000 images, of which 8976 images belong to the 'TB' class and 1024 images belong to the 'TE' class. The testing set consists of 659 images, of which 582 images belong to the 'TB' class and 77 images belong to the 'TE' class. This split was done randomly. After the nine-fold augmentation of the 'TE'

class, the training set consists of 18192 samples, of which the number of samples belonging to the 'TB' class remains the same as before, but the number of samples belonging to the 'TE' class is now 9216. The testing set also remains the same as before, in order to maintain the fairness of comparison.

4.2 Evaluation Metric

For robust evaluation in trich classification, we use Accuracy, Precision, and Recall as the evaluation metric which is from the confusion matrix. The evaluation metrics can be formulated as :

$$Accuracy = \frac{1}{m}\sum_{r=1}^{m}\frac{TP_r + TN_r}{TP_r + TN_r + FP_r + FN_r} \quad (2)$$

$$Precision = \frac{1}{m}\sum_{r=1}^{m}\frac{TP_r}{TP_r + FP_r} \quad (3)$$

$$Recall = \frac{1}{m}\sum_{r=1}^{m}\frac{TP_r}{TP_r + FN_r} \quad (4)$$

where, m, TP, TN, FP and FN represent the batch size, True-Positives, True-Negatives, False-Positives and False negatives respectively.

4.2.1 Implementation Details:

The model is implemented in the PyTorch framework using a Nvidia GTX 2080 Ti GPU with 32 GB memory. The Adam Optimization Algorithm is used for minimizing the loss function. An adaptive learning scheme is employed to efficiently decay the learning rate whenever necessary during training. Initially, the learning rate is set to 0.0002 which is decayed by a factor of 2 as the loss curve starts oscillating around a local minima for 3 consecutive epochs. The network is trained for a total of 20 epochs for a batch size of 8, using Adam Optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, since the number of trainable parameters and images are both less in number, and thus, the model would fit within that many epochs.

5 Results and Discussion

5.1 Results on Wasp Detection

Since there are no ground truths available for the detection sub-task, a purely quantitative analysis has been performed to evaluate the performance of the proposed detection pipeline, to obtain the bounding boxes, from the RGB images.

The existing architecture, TrichTrack (Pani et al., 2021), consists of a YOLOv5 detector, which has been trained iteratively over two stages on a dataset similar to ours.

Table 1: Comparison Table of detection results. The metric used is average number of detections per image.

Model	**Avg. # of BBox** (Actual = 20)
TrichTrack (Pani et al., 2021)	2
Ours	**21**

As can be observed from Table 1, our proposed detection algorithm vastly over-performs from TrichTrack. Considering that the number of wasps present per images in 20, our proposed architecture detects an average of 21 individuals per image, whereas the existing model detects only 2 individuals per image.

5.2 Results of Wasp Classification

Table 2: Comparison Table of classification performance using different attention blocks. Here, 'mAcc', 'Prec' and 'Rec' refer to mean Accuracy, Precision and Recall metrics. The attention modules tested are: Squeeze-and-Excite (SE) module (Hu et al., 2018), Non-Local (NL) module (Wang et al., 2018), Multi-scale adaptation of conventional Non-Local block (NL*) and the proposed Multi-scale Attention (MSA) module

Model	**mAcc**	**TB**		**TE**	
		Prec	**Rec**	**Prec**	**Rec**
ViT-SE (Hu et al., 2018)	0.91	0.94	0.96	0.65	0.51
ViT-NL (Wang et al., 2018)	0.90	**0.97**	0.91	0.54	**0.78**
ViT-NL*	0.90	0.96	0.92	0.55	0.73
ViT-MSA	**0.93**	0.95	**0.97**	**0.71**	0.64

For the task of classifying the wasp species, the proposed methodology was compared with other popular attentive enhancements, namely the Squeeze-and-Excite (SE) module (Hu et al., 2018), the Non-Local (NL) block (Wang et al., 2018) and Multi-scale adaptation of the conventional Non-Local block. It can be observed from Table 2 that the proposed method outperforms the other attentive enhancements in the classification task, compared using the standard classification metrics.

The SE block is often used as an attentive enhancement for channels, and has been utilized on the one-dimensional feature map obtained from the encoder, to augment or suppress the respective features. The NL block is often a popular choice as an attentive enhancement, since it is capable of capturing long-range dependencies from feature maps, as opposed to purely convolution-based attentive enhancements, which are only able to capture local short-range dependencies. Thus, it was also tested out as the attentive enhancement in our pipeline. Since, the NL block was insufficient as-is to suitably enhance the classification performance, we tried to increase

the receptive fields of the three branches- query and key to 5 and value to 3, and also adding a layer-norm module. Finally, we change the structure of the Multi-scale NL block by changing the matrix-multiplication to a Hadamard product, resulting in the MSA block, which resulted in improved classification performance.The results have also been visualized, with the use of confusion matrices, in Figure 5.

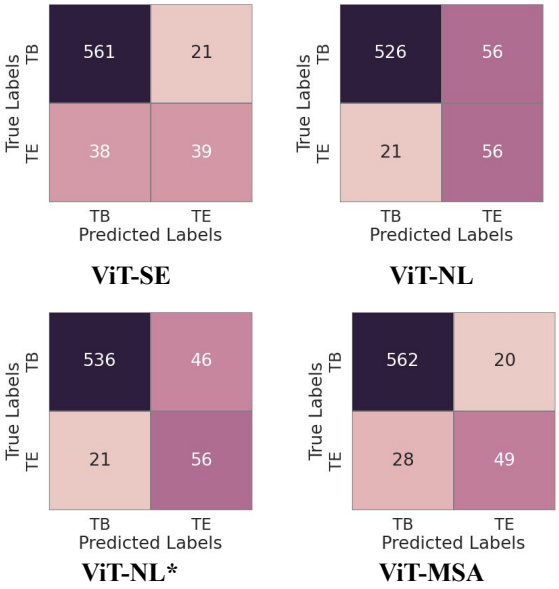

Figure 5: Comparative Analysis of the confusion metrics obtained from experiments with various popular attentive enhancements and the proposed attention module, results of which have been tabulated in Table 2. The values are on the testing dataset, which consists of 582 samples belonging to the 'TB' class and 77 samples belonging to the 'TE' class.

6 Experiments

6.1 Wasp Individual Detection

An ablation study was performed on the employed pre-processing techniques in data preparation - thresholding and C-E. The results have been tabulated in Table 3. The metric used was the average number of detections per image, since a specific number of insects (here, 20) are present in each image, irrespective of the species.

Importance of Dynamic Thresholding:

Initially, dynamic thresholding is performed on the input RGB images using a threshold value obtained from the Otsu Algorithm, offset by a fixed empirical value of -10. This not only makes the insects more visible, but also suppresses the eggs and other noise present in the images. Its efficacy can be observed from Table 3, where it reduces the number of detections per image from 60 to 57.

Table 3: Ablation Study on Pre-processing techniques. Here, 'Thresh' refers to the dynamic thresholding operation, and 'S-C-E' refers to the Spatial Contract-Expand operation. The average number of bounding boxes detected across all images is used as the metric.

Pre-processing		Avg. # of BBox
Thresh	S- C-E	(Actual = 20)
✗	✗	60
✓	✗	57
✓	✓	21

Model	Precision	Recall	F1
Swin	0.682	0.765	0.719
Swin + Scene branch	0.715	0.789	0.750
Swin + Part branch	**0.735**	0.783	0.749
DECO	0.731	**0.841**	**0.756**

Importance of Spatial Contract-Expand:

After dynamic thresholding, the spatial contract-expand algorithm is applied on the images, before being broken into patches. The 'Contraction' half almost completely eliminates all noise from the images - eggs or background - which just thresholding alone failed to remove. Then, the 'Expansion' half brings the insects back to their original sizes, since they have been reduced by 'Contraction'. Its efficacy is obvious from Table 3, since it reduces the average number of detections drastically - from 57 to 21, bringing it very close to the true value of 20 insects per image.

6.2 Wasp Species Classification

6.2.1 Experiments with the encoder architecture

As can be observed from Table 4 that the transformer encoder outperforms the convolution based encoders by a large margin, in the metrics used for comparative purposes. Even from the obtained confusion matrix for each model, as shown in Figure 6, it can be inferred that the used encoder architecture succeeds in classifying both species more accurately than the two baseline encoders compared with.

6.2.2 Experiments on the classification pipeline

The proposed classification pipeline incorporates a pre-trained and frozen ViT encoder as it is not feasible to train the encoder with a limited size of dataset.

Table 4: Comparison Table of classification performance, with various encoder architectures, with the same classification head and objective function. Here, 'mAcc', 'Prec' and 'Rec' refer to mean Accuracy, Precision and Recall metrics. The encoders compared with are a ResNet18 and an EfficientNet b7. ViT is pre-trained Vision Transformer, with a trainable classification head.

Model	mAcc	TB		TE	
		Prec	Rec	Prec	Rec
ResNet18 (He et al., 2016)	0.76	0.96	0.75	0.30	0.79
EfficientNet_b7 (Tan and Le, 2019)	0.62	0.96	0.59	0.21	**0.82**
ViT (Dosovitskiy et al., 2020)	**0.89**	**0.96**	**0.91**	**0.51**	0.69

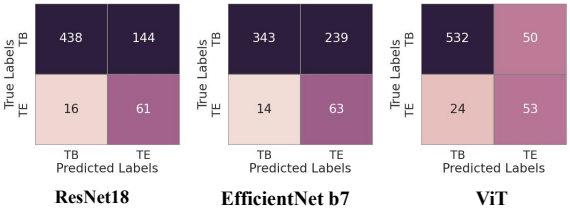

Figure 6: Comparative Analysis of the confusion metrics obtained from the encoders experimented with, as tabulated in Table 4. The values are on the testing dataset, which consists of 582 samples belonging to the 'TB' class and 77 samples belonging to the 'TE' class.

Thus, external attentive enhancements is performed by MSA module to enhance the extracted feature map. MSA helps to encode the coarse-fine features to better differentiate between the two classes. Its efficacy can be observed from Table 5 and Figure 7, irrespective of the objective function used.

Importance of Focal Loss: The classification accuracy was further boosted, by the use of Focal Loss as the objective function. Focal Loss gives more weight to samples that have low classification accuracy, and thus, focuses more on the 'hard' samples, as opposed to the vanilla Cross-Entropy Loss, which treats each sample equally. As can be observed from Table 5, it provides a boost in performance to the baseline model and a bigger boost in accuracy to the complete model.

7 Conclusion

In this work, we propose a combined detection-classification pipeline to handle the detection of very tiny wasps from high-resolution images and classify them into species based on very subtle spatial cues. Our pipeline consists of a light yet effective data preparation module to extract the ROIs (i.e., the wasp inidividuals) from the high-resolution images. We have also obtained SOTA results in the classification subtask, as compared to other existing methods. This

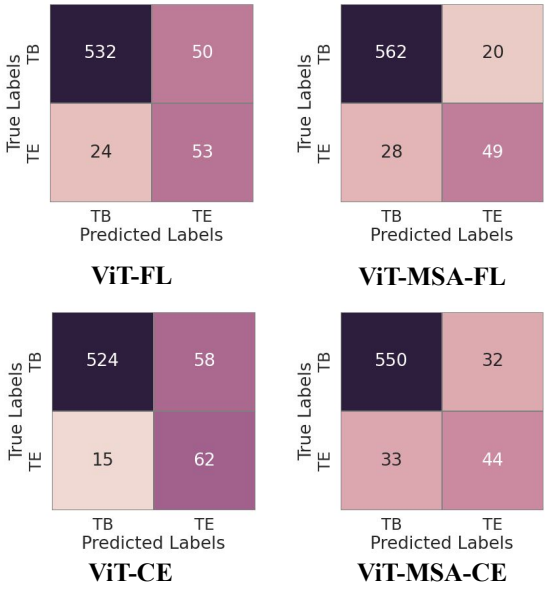

Figure 7: Comparative Analysis of the confusion metrics obtained with the baseline model and with the addition of the proposed attention module. The experimental values from the two objective functions experimented with have also been visually demonstrated, and have been tabulated in Table 5. The values are on the testing dataset, which consists of 582 samples belonging to the 'TB' class and 77 samples belonging to the 'TE' class.

can be attributed to our classification pipeline, especially the MSA block, which can extract subtle visual cues to distinguish between wasp individuals. As evident from the results reported, this is a robust pipeline which can be used in other similar wasp detection and classification problems, or tiny object detection and classification problems in general.

REFERENCES

Aharon, N., Orfaig, R., and Bobrovsky, B.-Z. (2022). Botsort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*.

Cao, J., Weng, X., Khirodkar, R., Pang, J., and Kitani, K. (2022). Observation-centric sort: Rethinking sort for robust multi-object tracking. *arXiv preprint arXiv:2203.14360*.

Dat, T., Nguyen, V.-T., and Tran, M.-T. (2018). Lightweight deep convolutional network for tiny object recognition. pages 675–682.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is

Table 5: Comparison Table of classification performance, with two objective functions. Here, 'mAcc', 'Prec' and 'Rec' refer to mean Accuracy, Precision and Recall metrics. The models tested are: pre-trained ViT encoder with two linear layers as classification head; and a pre-trained ViT encoder with the Multiscale Attention, and two linear layers as classification head.

Model	**Focal Loss**					**Cross-Entropy Loss**				
		TB		**TE**			**TB**		**TE**	
	mAcc	**Prec**	**Rec**	**Prec**	**Rec**	**mAcc**	**Prec**	**Rec**	**Prec**	**Rec**
ViT+MLP	0.89	**0.96**	0.91	0.51	**0.69**	0.88	**0.97**	0.90	0.52	**0.81**
ViT-MSA+MLP	**0.93**	0.95	**0.97**	**0.71**	0.64	**0.90**	0.94	**0.95**	**0.58**	0.57

worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Du, Y., Song, Y., Yang, B., and Zhao, Y. (2022). Strongsort: Make deepsort great again. *arXiv preprint arXiv:2202.13514*.

Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2021). Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.

Gong, Y., Yu, X., Ding, Y., Peng, X., Zhao, J., and Han, Z. (2021). Effective fusion factor in fpn for tiny object detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1160–1168.

Han, J., Ding, J., Xue, N., and Xia, G.-S. (2021). Redet: A rotation-equivariant detector for aerial object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2786–2795.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141.

Kong, F. and Henao, R. (2022). Efficient classification of very large images with tiny objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2384–2394.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90.

Lee, C., Park, S., Song, H., Ryu, J., Kim, S., Kim, H., Pereira, S., and Yoo, D. (2022). Interactive multi-class tiny-object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14136–14145.

Pani, V., Bernet, M., Calcagno, V., van Oudenhove, L., and Bremond, F. F. (2021). TrichTrack: Multi-Object Tracking of Small-Scale Trichogramma Wasps. In *AVSS 2021 - 17th IEEE International Conference on Advanced Video and Signal-based Surveillance*, Virtual, United States.

Qin, R., Liu, Q., Gao, G., Huang, D., and Wang, Y. (2020). Mrdet: A multi-head network for accurate oriented object detection in aerial images. *arXiv preprint arXiv:2012.13135*.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR.

Wang, X., Girshick, R., Gupta, A., and He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Wojke, N., Bewley, A., and Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE.

Yang, C., Huang, Z., and Wang, N. (2022). Querydet: Cascaded sparse query for accelerating high-resolution small object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13668–13677.

Yi, J., Wu, P., Liu, B., Huang, Q., Qu, H., and Metaxas, D. (2021). Oriented object detection in aerial images with box boundary-aware vectors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2150–2159.

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., and Wang, X. (2021). Bytetrack: Multi-object tracking by associating every detection box. *arXiv preprint arXiv:2110.06864*.