WACV #1772

# Focal Diversity-Optimized Object Detection Ensembles

Anonymous WACV Algorithms Track submission

Paper ID 1772

## Abstract

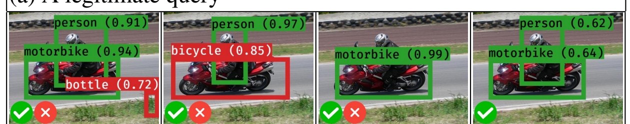
*Object detection ensembles can boost the generalization performance of individual detection models. However, existing ensemble approaches suffer from two weaknesses: (i) a larger number of component models is considered a better ensemble, and (ii) the detection fusion methods for combining results mainly rely on non-maximum suppression (NMS) techniques. This paper presents a focal diversity-optimized object detection ensemble method, coined as ODEN, with three original contributions. First, ODEN introduces the concept of focal object detection diversity to capture the negative correlations among multiple component object detectors. A detection ensemble with a higher focal diversity implies that its component models have higher failure independence and can generalize better than the existing NMS family of ensemble methods. Second, ODEN introduces the focal diversity-optimized ensemble pruning algorithm to produce top-$K$ sub-ensembles from a pool of object detection models to outperform the large ensemble of all models. Third, the ODEN inconsistency solver can resolve three types of inconsistency to combine detection results from multiple object detectors. The joint optimization of focal diversity pruning and robust detection fusion makes the ODEN ensembles outperform the best individual component model and the existing representative ensemble methods. Extensive experiments conducted on three benchmark datasets show that ODEN can improve the detection accuracy of existing ensemble methods by up to $9.33\%$ under benign scenarios and can boost the resilience of object detection against representative adversarial attacks with up to an $82.44\%$ increase in the adversarial robustness.*

## 1. Introduction

Powered by the recent advances in deep neural networks (DNNs), object detection has been widely deployed in numerous applications, such as driving scene understanding [9] and intruder detection [27]. These applications are often mission-critical and hence impose a high demand on DNN-based object detection algorithms to deliver higher accuracy and stronger robustness.

| Member Model Detection | | | ODEN (Ours) |
|---|---|---|---|
| Member 1 | Member 2 | Member 3 | |

(a) A legitimate query




(b) A deceptive query

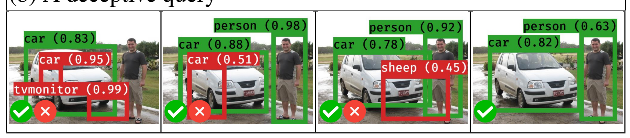



Table 1. Individual object detectors (1st to 3rd columns) can make errors on a given query image due to their inherent weaknesses (a) or evasion attacks (b). The diversity-driven ensemble ensures failure independence and creates opportunities for the inconsistency solver to reconstruct correct detection (4th column).

This paper presents ODEN, a focal diversity-enhanced ensemble framework for real-time object detection to enhance the generalization performance of DNN models for high-quality inference. ODEN consists of two synergistic functional components. First, the focal diversity-optimized ensemble pruning produces sub-ensembles of high focal diversity (high failure independence) and a small ensemble size with a low computational cost. Those sub-ensembles are chosen from a pool of base DNN models using their focal detection diversity scores, having the property that an ensemble with high focal diversity will result in high detection performance. Second, the inconsistency solver produces robust ensemble detection by restoring inconsistent detection results from multiple member models of an ensemble. Unlike the ensemble of single-task learners such as image classifiers [35], object detectors are multi-task learners [22], and ODEN has to deal with inconsistent detection results on all three learning tasks from each ensemble member model: object existence detection, bounding box locations of detected objects, and the classification of detected objects and their confidence scores. These two complementary components strengthen the robustness of object detection, as demonstrated by visual examples in **Table 1**, having an ensemble of three members with high focal diversity.

Focusing on Table 1a, given the same query image from the sensing device (e.g., a camera), each member model can make mistakes due to its imperfect detection performance: member 1 misdetected an extra bottle (1st column), member 2 misclassified the motorbike as a bicycle (2nd column), and member 3 could not recognize the person (3rd column). As the employed ensemble is carefully selected by ODEN with high focal diversity, the high failure independence encourages all members to make errors differently, which creates opportunities for the ODEN inconsistency solver to rectify three levels of inconsistency and reconstruct the correct detection results (4th column). The same idea also applies to evasion attacks [5,15,25,28,32,36] (see Table 1b), which have received much attention as a growing threat to intelligent systems. They generate deceptive queries by injecting human-imperceptible perturbations (note that images displayed in Table 1b are already perturbed by the state-of-the-art attack named TOG [6]) to legitimate queries, aiming to mislead high-quality object detection systems.

The contributions of this paper are as follows. First, we introduce the concept of focal detection diversity to measure the failure independence of member models of an ensemble and propose a focal diversity-optimized ensemble pruning method. Second, we present a robust inconsistency solver to distill disagreeing predictions from member models of an ensemble. We conduct extensive experiments with three popular object detection benchmarks: MS COCO [16], Open Images [14], and PASCAL VOC [8]. Our evaluations show three significant results: (1) Object detection ensembles from ODEN consistently offer high mAP over the best-performing member and improve the ensemble performance by up to 9.33% in mAP compared to the existing representative detection ensemble methods. (2) ODEN can effectively select the top-performing sub-ensembles based solely on their focal diversity scores, demonstrating the importance of our focal diversity-optimized ensemble pruning. (3) ODEN offers high resilience against four state-of-the-art evasion attacks. The source code of ODEN is available at [Anonymized].

# 2. ODEN Design Overview

## 2.1. Object Detection Ensemble

Given an input image $\boldsymbol{x}$, a $K$-class object detection model $F_i$, parameterized by $\boldsymbol{\theta}$, generates a large number of candidate objects. Each object $\boldsymbol{o}_{i,j} \in F_i(\boldsymbol{x})$ is associated with three perceptual predictions: (i) the estimated objectness $\mathcal{J}_{i,j}$, indicating the probability of the candidate being a real object, (ii) the predicted bounding box $\boldsymbol{b}_{i,j} = (b_{i,j}^{\text{xmin}}, b_{i,j}^{\text{ymin}}, b_{i,j}^{\text{xmax}}, b_{i,j}^{\text{ymax}})$, recorded by the top-left and bottom-right corners of the object in the input image, and (iii) the class probability vector $\boldsymbol{p}_{i,j} = (p_{i,j}^1, p_{i,j}^2, ..., p_{i,j}^K)$ indicating the object classification result with $\ell_{i,j} = \arg\max_{1\le k\le K} p_{i,j}^k$ being the class label and $c_{i,j} = \max_{1\le k\le K} p_{i,j}^k$ being the confidence. The detection result $F_i(\boldsymbol{x})$ on the input image $\boldsymbol{x}$ is finalized by applying confidence thresholding and non-maximum suppression to discard those candidate objects with either low prediction confidence or high overlapping with other candidates.

Based on the three prediction tasks, DNN-based object detection can be formulated as a multi-task learning problem for a given training set $\tilde{\boldsymbol{\mathcal{D}}}$, minimizing the prediction error of (i) objectness $\mathcal{L}_{\text{obj}}$, (ii) bounding boxes $\mathcal{L}_{\text{bbox}}$, and (iii) class labels $\mathcal{L}_{\text{class}}$ of objects, expressed by:

$$\begin{aligned}\mathcal{L}(\tilde{\boldsymbol{\mathcal{D}}}; F_i, \boldsymbol{\theta}) = \ & \mathbb{E}_{(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\mathcal{G}}}) \in \tilde{\boldsymbol{\mathcal{D}}}}[\mathcal{L}_{\text{obj}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\mathcal{G}}}; F_i, \boldsymbol{\theta}) + \\ & \mathcal{L}_{\text{bbox}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\mathcal{G}}}; F_i, \boldsymbol{\theta}) + \mathcal{L}_{\text{class}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\mathcal{G}}}; F_i, \boldsymbol{\theta})],\end{aligned} \tag{1}$$

where $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{\mathcal{G}}}$ denote a training sample and its ground-truth objects respectively. Then, the model parameters $\boldsymbol{\theta}$ of the deep object detector to be optimized are updated iteratively: $\boldsymbol{\theta}^{\text{new}} = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}(\tilde{\boldsymbol{\mathcal{D}}}; F, \boldsymbol{\theta})$ with a learning rate of $\alpha$.

Let $\boldsymbol{F} = \{F_1, ..., F_N\}$ be an ensemble of $N$ object detection models. A query image $\boldsymbol{x}$ sent to the ensemble $\boldsymbol{F}$ will be first dispatched to each of its $N$ member models in parallel and obtain a set of predictions, denoted by $\{F_i(\boldsymbol{x}) | F_i \in \boldsymbol{F}\}$. The problem of an object detection ensemble is to find a detection combination function $\boldsymbol{E}$ that maps the collection of detection sets, one from each member model of the ensemble, to a carefully-constructed set of ensemble-detected objects that are as close as possible to the ground-truth objects $\tilde{\boldsymbol{\mathcal{G}}}$ of the training image $\tilde{\boldsymbol{x}}$ in a training set $\tilde{\boldsymbol{\mathcal{D}}}$, i.e.,

$$\min_{(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\mathcal{G}}}) \in \tilde{\boldsymbol{\mathcal{D}}}} ||\boldsymbol{E}(F_1(\tilde{\boldsymbol{x}}), ..., F_N(\tilde{\boldsymbol{x}})) - \tilde{\boldsymbol{\mathcal{G}}}||, \tag{2}$$

where $||\cdot||$ denotes the difference between the ensemble-detected objects and the ground truth.

## 2.2. Technical Challenges

Given a pool of $N$ object detection models, while one could employ all of them to form a large ensemble of $N$ members, the generalization performance might not be enhanced because some member models could echo the others' decisions and contribute no useful signal for inconsistency evaluation. As to be shown in our experiments in Section 5.1, a large ensemble team does not always provide the best detection accuracy, and hence, we need to first investigate how to find sub-ensembles of strong synergies. With sub-ensembles of size varying from 2 to $N$, we can obtain a total of $2^N - (N+1)$ combinations. The first challenge is determining the top-performing sub-ensembles among the collection of all possible teams. We call this the ensemble selection problem in ODEN.

Unlike an image classifier that outputs one classification prediction for each input image, an object detector outputs

WACV #1772

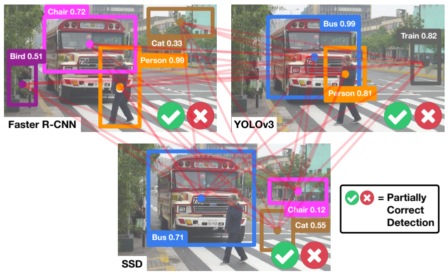



Figure 1. Three object detection models create different partially correct results on the same input image. We need an inconsistency solver to reconstruct correct decisions.

a set of detected objects. As a result, the detection combination algorithm $\boldsymbol{E}$ needs to calibrate the possibly inconsistent detection from multiple member detectors along all three perceptual dimensions for every detected object returned from a member detector of an ensemble. Hence, the open problems include (i) different object detection models may return different numbers of detected objects on the same query image, (ii) different object detectors may return different bounding boxes for the same entity (ground-truth object) with varying locations and sizes, and (iii) for the same ground-truth object, different detectors may return predictions with different confidence scores. **Figure 1** illustrates these open problems by combining detection results from three object detection models. The ensemble takes a query image of a typical driving scene and gets the detection results from three member models: four objects from Faster R-CNN, three from YOLOv3, and three from SSD. The second challenge is to find the resolution of which objects from different models refer to the same entity because bounding boxes almost never align due to their regression nature, and a large number of combinations can be possible (see the red lines) even for those detected objects whose confidence scores are above the threshold.

## 3. Focal Diversity-based Ensemble Selection

Given a pool of $N$ base models, we can formulate $\sum_{M=2}^{N}\binom{N}{M} = 2^N-(N+1)$ ensemble teams with the team size $M$ ranging from 2 to $N$. For instance, a 10-model pool leads to $1,013$ teams, and the number of choices jumps exponentially to $1,048,555$ when $N = 20$. In this section, we first introduce the focal detection ensemble diversity measure and then describe a focal diversity-based ensemble selection algorithm, which shows that (i) the top sub-ensembles of high focal diversity are the high-quality ensembles, outperforming the member model with the highest mAP, and (ii) the top sub-ensembles tend to have a smaller committee of highly diverse detectors from the base model pool, which have high failure independence and outperform the largest ensemble of all $N$ models.

### 3.1. Focal Detection Ensemble Diversity

We adopt a focal model paradigm [4,35] for diversity assessment. For each ensemble of size $M$, we consider each of the $M$ member models as a focal model to evaluate the diversity of the ensemble based on the negative samples of the focal model from a validation set. Thus, each ensemble team of size $M$ will have $M$ focal diversity scores, one for each of the $M$ focal models. Finding negative samples of an object detection model is non-trivial because it tends to detect far more objects than those in the ground truth set and it requires a confidence threshold to decide which ones to discard. An inadequate decision on the threshold may result in unnecessary false positives (too low) or false negatives (too high). In light of this, we implement a ranking-based approach for negative sample determination (Algorithm 1 in the appendix), which first sorts the detected objects of the focal model in the descending order of their confidence and finds a one-to-one mapping to the set of ground-truth objects. The approach requires the correctly detected objects to have higher confidence than other irrelevant detection (i.e., no false positives), and all ground-truth objects will be recognized (i.e., no false negatives).

Given an ensemble $\boldsymbol{F}$ of $M$ models ($M \leq N$), i.e., $\boldsymbol{F} = \{F_1, \ldots, F_M\}$, we compute $M$ focal detection diversity scores by considering each member as the focal model. Given a focal model $F_{\text{focal}}$, we obtain a set of negative samples and measure the focal model-based disagreement among the other $M-1$ member models. In our prototype of ODEN, we measure the focal ensemble diversity using the negative sample of the focal model by leveraging the non-pairwise general disagreement defined in [21]. Let $Y$ denote a random variable representing the proportion of models (i.e., $i$ out of $M$) that fail to recognize a random input sample $\boldsymbol{x}$ defined in Algorithm 1. The probability of $Y = \frac{i}{M}$ is denoted as $p_i$. The focal diversity of an object detection ensemble $\boldsymbol{F} = \{F_1, ..., F_{\text{focal}}, ..., F_M\}$ of size $M$ w.r.t. the focal model $F_{\text{focal}}$ is defined as follows:

$$div_{\text{focal}}(\boldsymbol{F}, F_{\text{focal}}) = 1 - \frac{\sum_{i=1}^{M} \frac{i}{M} p_i}{\sum_{i=1}^{M} \frac{i(i-1)}{M(M-1)} p_i}. \tag{3}$$

$div_{\text{focal}}$ is in the range of $[0, 1]$ with the maximum diversity score of 1 when the failure of one member model is accompanied by the correct recognition by the other.

### 3.2. Diversity-based Ensemble Pruning

Given a pool of $N$ base models, say $N = 10$, by choosing $F_1$ as the focal model, we can compare all the sub-ensembles of size $M$ containing $F_1$ as the focal model by their focal diversity scores. For $M = 5$, we have a total of 126 sub-ensembles containing the focal model $F_1$. We can utilize the focal diversity measure $div_{\text{focal}}(\boldsymbol{F}, F_1)$ to partition this set into those sub-ensembles of high focal diversity and those with low diversity and select the top sub-ensembles of highest focal diversity as our recommendation for the top-performing ensemble teams. For a given focal model $F_{\text{focal}}$, we denote $\boldsymbol{\Lambda}_{F_{\text{focal}},M}$ as the set of sub-ensembles of size $M$ containing the focal model $F_{\text{focal}}$. Using Equation 3, we measure the focal ensemble diversity of each sub-ensemble and obtain the diversity-accuracy set, defined by $DA = \{div_{\text{focal}}(\boldsymbol{F}, F_{\text{focal}}), \text{ACC}(\boldsymbol{F})) \mid \boldsymbol{F} \in \boldsymbol{\Lambda}_{F_{\text{focal}},M}\}$, where $\text{ACC}(\cdot)$ returns the mAP using ODEN's detection combination algorithm to be described in Section 4. Each member of the DA set represents a sub-ensemble team of size $M$ containing $F_{\text{focal}}$. To identify those ensembles with high focal diversity, we first define the initial centroid for the cluster with high ensemble diversity using the maximum diversity and the maximum accuracy of all sub-ensembles in the DA set. Similarly, we initialize the second centroid for the cluster with low focal diversity using the minimum focal diversity and the lowest accuracy of ensembles in the DA set. Then, we partition the DA set using a binary clustering algorithm, such as K-Means, with the two specific initial centroids. We use the largest diversity in the cluster with low diversity as the cut-off threshold.

For each sub-ensemble of $M$ member models, each of the $M$ models will be used as a focal model once, and thus it will have $M$ focal diversity scores. For example, the ensemble $F_{1,2,3}$ (i.e., a team with $F_1$, $F_2$, and $F_3$ as members) has three focal diversity scores: one in $\boldsymbol{\Lambda}_{F_1,3}$ with $F_1$ as the focal model, one in $\boldsymbol{\Lambda}_{F_2,3}$ with $F_2$ as the focal model, and the third one in $\boldsymbol{\Lambda}_{F_3,3}$ with $F_3$ as the focal model. Let $HDEnsSet_{F_{\text{focal}},M,\boldsymbol{F}}$ be the partition of the sub-ensembles of size $M$ with high focal diversity for a given focal model $F_{\text{focal}}$. We can use an affirmative or unanimous vote to determine if an ensemble $\mathcal{E}$ of $M$ models should be chosen as the recommended ensemble by our focal diversity-based ensemble selection algorithm. Using the unanimous voting scheme (intersection), an ensemble $\mathcal{E}$ is selected if $\mathcal{E} \in \bigcap_{i=1}^{N} HDEnsSet_{F_i^{\text{focal}},M,\boldsymbol{F}}$. Using affirmative voting (union), an ensemble $\mathcal{E}$ is selected if $\mathcal{E} \in \bigcup_{i=1}^{N} HDEnsSet_{F_i^{\text{focal}},M,\boldsymbol{F}}$. Affirmative voting is used as the default in the prototype of ODEN.

# 4. Robust Detection Combination

Having an ensemble of diverse object detectors is not sufficient. An effective combination algorithm plays a crucial role in complementing one member with others and offers strong robustness. ODEN combines object detection results from each member model of an ensemble through three tiers of perceptual calibrations: *First*, it examines all the detected objects and partitions them into class-based groups identifying which objects produced by different member models refer to the same entity. *Second*, it examines each detection group to perform bounding box (BBox) calibration to produce the ensemble prediction of the bounding box. *Third*, it generates the confidence score for each ensemble prediction through group-based confidence calibration with the ensemble size and the fine-grained detection consistency. **Figure 2** illustrates the workflow of the three-phase ensemble detection calibration.

## 4.1. Candidate Detection Grouping

The goal of candidate detection grouping is to perform entity resolution: It determines whether two detected objects from different member models refer to the same entity and thus are associated based on (i) whether they are detected with the same class label and (ii) whether their BBoxes overlap significantly. The pseudocode is provided in Algorithm 2 in the appendix.

Given a set of detection results from each of the $N$ member models in an ensemble, we first partition all detected objects by their class label and sort the detected objects of each class $\ell$ in the descending order of their prediction confidence scores and produce a sorted list of detected objects for each class $\ell$, denoted by $\boldsymbol{\mathcal{G}}_\ell$. Second, we further partition the sorted list $\boldsymbol{\mathcal{G}}_\ell$ into different groups. Each corresponds to the same entity in the ground truth. Concretely, we first find the detected object with the highest confidence in $\boldsymbol{\mathcal{G}}_\ell$ and use it as the anchor prediction for the first group. Then, we choose the next detected object $\boldsymbol{o}_j \in \boldsymbol{\mathcal{G}}_\ell$ and assign it to a group $\boldsymbol{\gamma}$ if it satisfies the following conditions: (i) the model detecting the object $\boldsymbol{o}_j$ has not yet contributed any detected object to the group $\boldsymbol{\gamma}$, and (ii) there is a significant overlapping between the detected object $\boldsymbol{o}_j$ and those already in the group $\boldsymbol{\gamma}$. This process repeats until all detected objects in the partition $\boldsymbol{\mathcal{G}}_\ell$ are examined and added to a group. In ODEN, we introduce a system-supplied threshold $\mathcal{T}_{\text{IOU}}$ (e.g., $0.50$) and define the significant overlapping by checking if the overlapping measured by the intersection over union (IOU) is larger than the threshold $\mathcal{T}_{\text{IOU}}$. To compare overlapping between the $\boldsymbol{o}_j$ and those already in the group $\boldsymbol{\gamma}$, we we generate the representative BBox of the group $\boldsymbol{\gamma}$ by averaging all BBoxes of the detected objects in the group, weighted by their confidence scores and measure the overlapping with it. We call it the weighted averaging approach, denoted as $\beta_{WA}(\boldsymbol{o}_j, \boldsymbol{\gamma})$:

$$\beta_{WA}(\boldsymbol{o}_j, \boldsymbol{\gamma}) = \text{IOU}(\boldsymbol{b}_j, \sum_{\boldsymbol{o}_r \in \boldsymbol{\gamma}} \frac{\boldsymbol{b}_r c_r}{\sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} c_i}). \quad (4)$$

If $\boldsymbol{o}_j \in \boldsymbol{\mathcal{G}}_\ell$ has a significant overlapping with the group $\boldsymbol{\gamma}$ and the detector detecting $\boldsymbol{o}_j$ has not yet made any contribution to the group $\boldsymbol{\gamma}$, then we add $\boldsymbol{o}_j$ to the group. Otherwise, we will create a new group with $\boldsymbol{o}_j$ as the anchor detection. The Phase 1 detection grouping repeats for each class until all detected objects from the $N$ member models of an ensemble have been evaluated. The final result of Phase 1 is a list of groups, denoted by $\boldsymbol{\Gamma}$, where each group $\boldsymbol{\gamma} \in \boldsymbol{\Gamma}$ contains a set of detected objects of the same class label, each

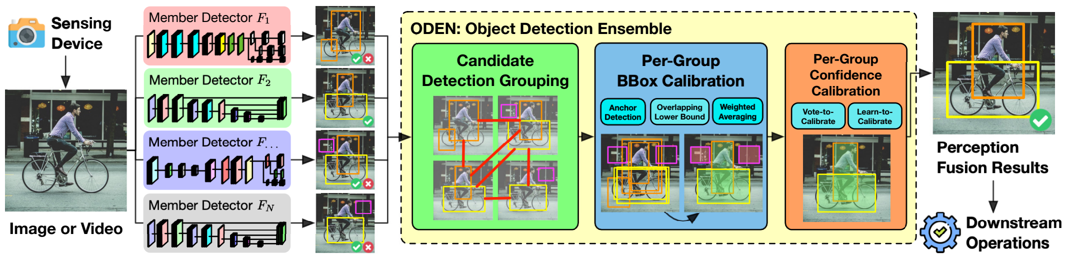



Figure 2. The three-phase ensemble detection calibration framework in ODEN.

from a different member, and all recognize the same entity.

## 4.2. Per-Group BBox Calibration

The second phase of the ODEN inconsistency solver takes the list $\mathbf{\Gamma}$ of groups from Phase 1 and performs per-group-based bounding box calibration. Recall that although different detectors often generate different bounding boxes and different confidence scores for their detection, all detected objects in each group $\boldsymbol{\gamma} \in \mathbf{\Gamma}$ have the same class label and correspond to the same entity. To generate the ensemble detection results, each characterizes the *delegate* object representing a group, we need to compute the exact bounding box (location and size) and the confidence for the ensemble detection by aggregating the BBoxes and the different confidence scores of the detected objects in each group in addition to the existence of the object of class $\ell$. The former is carried out by group-based BBox calibration in Phase 2, and the latter is performed by group-based confidence calibration in Phase 3 in Section 4.3.

Based on how the group is composed, several approaches can be employed to calibrate the bounding boxes of each group $\boldsymbol{\gamma} \in \mathbf{\Gamma}$. If we use the anchor detection for grouping in Phase 1 (i.e., $\beta_{anchor}$), we can return the bounding box $\boldsymbol{b}_{anchor(\boldsymbol{\gamma})}$ of the anchor as the calibrated BBox. Alternatively, if we use the overlapping lower bound $\beta_{LB}$ or the weighted averaging $\beta_{WA}$ for grouping in Phase 1, we can compute the BBox of the delegate object by aggregating the bounding boxes of all detected objects in the group, each is weighted by the confidence of the corresponding detection. Formally, the bounding box $\hat{\boldsymbol{b}}$ of the delegate object is computed as follows:

$$\hat{\boldsymbol{b}} = \left( \frac{\sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} b_i^{\text{xmin}} c_i}{\sum_{\boldsymbol{o}_j \in \boldsymbol{\gamma}} c_j}, \frac{\sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} b_i^{\text{ymin}} c_i}{\sum_{\boldsymbol{o}_j \in \boldsymbol{\gamma}} c_j}, \frac{\sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} b_i^{\text{xmax}} c_i}{\sum_{\boldsymbol{o}_j \in \boldsymbol{\gamma}} c_j}, \frac{\sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} b_i^{\text{ymax}} c_i}{\sum_{\boldsymbol{o}_j \in \boldsymbol{\gamma}} c_j} \right) \tag{5}$$

The confidence-weighted calibration of the bounding boxes incorporates both the estimated location and size of each bounding box and how certain the estimation is from each corresponding member. We use this approach as the default in our prototype of ODEN.

Recall that for an $N$-member ensemble, the goal of the ensemble detection combination method is to combine the detection results of the $N$ member models to generate the ensemble detection results. Let $\hat{d} = [\hat{\boldsymbol{b}}, \hat{\ell}, \hat{c}]$ be an ensemble detection result, representing the detected object of class $\hat{\ell}$ with bounding box $\hat{\boldsymbol{b}}$ and detection confidence $\hat{c}$. According to the detection grouping in Phase 1, every group has a set of the detected objects of one specific class. Upon the completion of Phase 2, for each group $\boldsymbol{\gamma} \in \mathbf{\Gamma}$, we also generated the bounding box $\hat{\boldsymbol{b}}$ of the delegate object representing the group. The final step is to compute the confidence for each ensemble detection result $\hat{d}$, which is the focus of Phase 3.

## 4.3. Per-Group Confidence Calibration

For a given ensemble $\boldsymbol{F}$ of $N$ models, upon completing the first two phases of the detection combination, we obtain the list $\mathbf{\Gamma}$ of groups, and for each group $\boldsymbol{\gamma} \in \mathbf{\Gamma}$, we have the class label $\hat{\boldsymbol{\ell}}$ and the bounding box $\hat{\boldsymbol{b}}$ for the delegate object representing the group. An intuitive approach to computing the confidence $\hat{\boldsymbol{c}}$ for the delegate object of each group is to take the average of the confidence scores of the detected objects in the group $\boldsymbol{\gamma}$: $\hat{c} = \frac{1}{|\boldsymbol{\gamma}|} \sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} c_i$, where $c_i$ is the confidence of the detected object $\boldsymbol{o}_i$ in the group $\boldsymbol{\gamma}$. However, this approach does not consider the votes from different member models of the ensemble and can work poorly when the member models generate fake detection. Recall Figure 1, all three models produce at least one fabricated object (e.g., YOLOv3 incorrectly returns a train). These fake objects do not overlap with one another, and each of them will form a single-object group. If we use group-based averaging for the confidence calibration, these fake objects will be kept by the ensemble detection with high confidence (e.g., 0.82 for the train).

One solution to this problem is to aggregate the confidence scores of all the detected objects in the group $\boldsymbol{\gamma}$ normalized by the ensemble size $N$ as $\hat{c} = \frac{1}{N} \sum_{\boldsymbol{o}_i \in \boldsymbol{\gamma}} c_i$. This approach can be viewed as a refinement of the group-based averaging method by adding the weight $\frac{|\boldsymbol{\gamma}|}{N}$. If the group $\boldsymbol{\gamma}$ contains the detected objects from only a few member models, the ensemble detection should be assigned low confidence, reflecting that the delegate object representing the group is less likely to correspond to a real entity compared to another group supported by a larger number of member models. This ensemble vote normalized method will effectively reduce the confidence for those single-object groups or the groups supported by only a few member models.

The third approach is *learn-to-calibrate*, which trains a model for confidence calibration using the validation data. It is motivated by the observation that a group having the detected objects of high confidence and high overlapping with their bounding boxes is more likely to correspond to a real entity compared to a group having objects of low confidence and with marginally overlapping bounding boxes. Instead of manually examining these statistics for all the groups on each input image, in order to define the per-group confidence calibration rules, the *learn-to-calibrate* approach will first perform feature extraction for each group $\gamma$ to distill useful perceptual properties from the group. Let $V_c$ denote the confidence vector of $N$ elements for group $\gamma$, each element denotes the confidence of the detected object from a member model in the ensemble. Similarly, let $V_{IOU}$ denote the IOU vector of the group with $N$ elements, each element denotes the overlapping between the BBox of each detected object in the group $\gamma$ and the BBox of the delegate object representing the group. Zero confidence and IOU are assigned if a member does not contribute any detected object to the group. We define the features extracted for the group $\gamma$ as the concatenation of these two vectors: $\Theta(\gamma, \boldsymbol{F}) = V_c || V_{\mathrm{IOU}}$. To learn how to calibrate the confidence of the delegate object representing the group $\gamma$, we next train a model to estimate the probability of a given group corresponding to a real entity in the ground truth, i.e., $P(\textsc{Real} = \textsc{True} | \Theta(\gamma, \boldsymbol{F}))$. We employ logistic regression to estimate such a probability distribution and compute the calibrated confidence $\hat{c}$:

$$\hat{c} = \frac{\sum_{\boldsymbol{o}_i \in \gamma} c_i}{N(1 + \exp(-(\boldsymbol{W}\Theta(\gamma, \boldsymbol{F}) + b)))}, \quad (6)$$

where the parameters $\boldsymbol{W}$ and $b$ are learned using a validation set. The *learn-to-calibrate* is used as the default.

## 5. Experimental Evaluation

We conduct extensive experiments on three object detection benchmarks: (i) MS COCO [16], (ii) Open Images [14], and (iii) PASCAL VOC [8]. **Table 2** summarizes the seventeen base models used in our experiments, including their mAP [8], the best-performing model in each dataset (the 2nd to the last row), and the average mAP of each base model pool (the last row). We compare ODEN with three popular methods for object detection fusion: non-maximum weighted (NMW) [40], soft non-maximum suppression (Soft-NMS) [2], and non-maximum suppression (NMS) [19]. Detailed setup is given in the appendix.

### 5.1. Benign Detection Performance Analysis

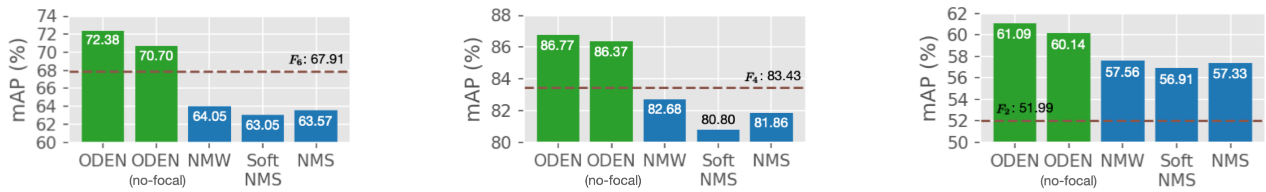
We first evaluate ODEN under benign scenarios with no adversaries. **Figure 3** compares ODEN with non-maximum weighted (NMW), soft non-maximum suppression (Soft-NMS), and non-maximum suppression (NMS) in terms of benign mAP on three vision benchmarks. ODEN refers to

| | MS COCO | | Open Images | | PASCAL VOC | |
|---|---|---|---|---|---|---|
| | Model | mAP | Model | mAP | Model | mAP |
| $F_1$ | SSD300-R | 52.47 | CRCNN | 50.60 | FRCNN | 67.37 |
| $F_2$ | SSD300-V | 46.70 | RetinaNet | 51.99 | SSD300 | 76.11 |
| $F_3$ | SSD512-R | 57.67 | CRCNN-FPN | 50.55 | SSD512 | 79.83 |
| $F_4$ | SSD512-V | 55.81 | MRCNN | 49.14 | YOLOv3-D | 83.43 |
| $F_5$ | SSD512-M | 42.70 | FRCNN | 45.28 | YOLOv3-M | 71.84 |
| $F_6$ | YOLOv3-D | 67.91 | - | - | - | - |
| $F_7$ | YOLOv3-M | 60.20 | - | - | - | - |
| Best | YOLOv3-D | 67.91 | RetinaNet | 51.99 | YOLOv3-D | 83.43 |
| Avg. | - | 54.78 | - | 49.51 | - | 75.72 |

Table 2. A summary of base models for three benchmark datasets in our experimental evaluation.

our ensemble with inconsistency solver and focal diversity ensemble pruning turned on. The team with the highest focal diversity is $F_{1,3,4,6,7}$ for MS COCO, $F_{1,2,3,4}$ for PASCAL VOC, and $F_{1,2,3,5}$ for Open Images. To provide a zoom-in comparison of ODEN with NMW, SoftNMS, and NMS, which use the entire base model pool as the ensemble, we also include ODEN (no-focal), which is the version of ODEN that has the inconsistency solver but does not use focal diversity-optimized ensemble pruning. Instead, the entire pool of the base models is used as the ensemble team. We make two observations. First, both ODEN and ODEN (no-focal) significantly outperform existing approaches for all benchmark datasets, and both provide better generalization performance than the best-performing base model in the pool. Second, compared to ODEN (no-focal), we show that the generalization performance of ODEN can be further strengthened by combining the detection inconsistency solver with the focal diversity ensemble pruning. **Table 3** provides two visual examples to compare ODEN (the 4th column) with three existing baselines: NMW, SoftNMS, and NMS (the 5th to 7th columns). We use the same ensemble team of $F_{2,3,4}$ on PASCAL VOC for a fair comparison. It shows their effectiveness in resolving detection inconsistency when combining partially correct decisions from individual member models (the 1st to 3rd columns).

**Figure 4** shows a quantitative comparison with the same team, where NMS and SoftNMS perform worse than the best member ($F_5$) with an mAP of 83.43%, and ODEN reaches an ensemble mAP of 86.62%, having a 3.19% improvement. Such an observation can be made consistently across all ensemble teams, meaning that ODEN can reach detection quality higher than other approaches given the same ensemble. For each dataset and its corresponding base model pool, we evaluate all ensemble teams with at least two members, resulting in 120 ensembles for MS COCO, 26 ensembles for Open Images, and 26 ensembles for PASCAL VOC. **Figure 5** reports the ensemble mAP of all teams by comparing ODEN with three existing representative detection combination methods. *First*, among the 172 teams across three datasets, ODEN (red) consistently outperforms




(a) MS COCO (b) PASCAL VOC (c) Open Images

Figure 3. ODEN outperforms three representative detection ensemble methods in benign mAP and the best-performing base model in the respective pool marked by the horizontal line.

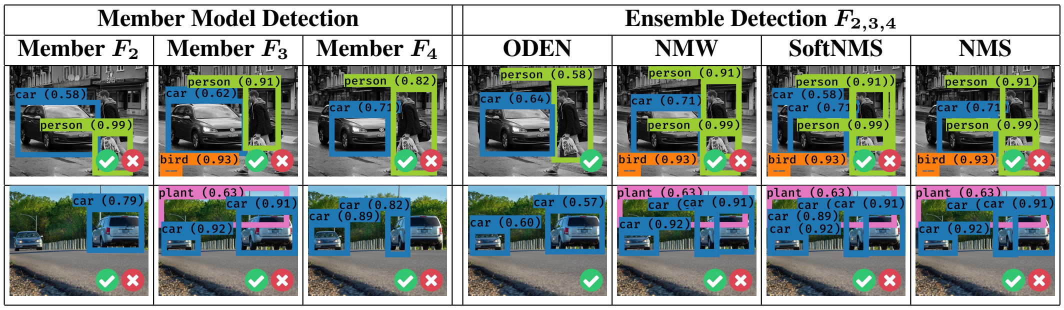



Table 3. Detection results on two test images by three member models and four ensemble methods using the same ensemble team $F_{2,3,4}$. ODEN inconsistency solver successfully removes false positives.

| **Ensemble** | $F_{1,2,3,4,5,6,7}$ | $F_{1,2,3,4,6,7}$ | $F_{1,3,4,6,7}$ | $F_{1,3,6,7}$ | $F_{1,4,6}$ |
|---|---|---|---|---|---|
| **mAP** | 70.70% | 71.32% | 72.38% | 72.19% | 71.69% |
| **mAP Gain** | 0% | +0.62% | +1.68% | +1.49% | +0.99% |
| **Best M.** | $F_6$(67.91%) | $F_6$(67.91%) | $F_6$(67.91%) | $F_6$(67.91%) | $F_6$(67.91%) |
| **Best M. Gain** | +2.79% | +3.41% | +4.47% | +4.28% | +3.78% |
| **Team Size** | 7 | 6 | 5 | 4 | 3 |
| **Cost** | 100% | 86% | 71% | 57% | 43% |

(a) MS COCO

| **Ensemble** | $F_{1,2,3,4,5}$ | $F_{1,2,3,5}$ | $F_{1,2,3}$ |
|---|---|---|---|
| **mAP** | 60.14% | 61.09% | 60.33% |
| **mAP Gain** | 0% | +0.95% | +0.19% |
| **Best M.** | $F_2$(51.99%) | $F_2$(51.99%) | $F_2$(51.99%) |
| **Best M. Gain** | +8.15% | +9.10% | +8.34% |
| **Team Size** | 5 | 4 | 3 |
| **Cost** | 100% | 80% | 60% |

(b) Open Images

Table 4. The teams selected by ODEN in MS COCO and Open Images. The 4th and 6th rows compare the mAP gains of using the selected ensembles compared to the ensemble composed of all base models and the best mAP member model. The last two rows show that the higher mAP of sub-ensembles can be achieved with smaller ensemble team size and lower execution cost.

the three existing schemes (NMW in blue, Soft-NMS in green, and NMS in orange) by a large margin. The improvement can be as large as 9.14% on MS COCO, 4.58% on Open Images, and 6.05% on PASCAL VOC. *Second*, the three existing representative methods for combining multiple detections (i.e., NMW, Soft-NMS, and NMS) behave similarly in terms of the ensemble mAP performance for different teams, with NMW performing slightly better than NMS and Soft-NMS being the worst among the three with a marginally lower mAP for all three datasets.

**Table 4** gives the top-$k$ sub-ensembles with the highest diversity scores identified by ODEN on MS COCO and Open Images. The 2nd column shows the teams using all available models in the respective pool (i.e., the ODEN (no-focal) in Figure 3). In such cases, the detection mAP reaches 70.70% on MS COCO and 60.14% on Open Images. Ensembles with a smaller size can lead to a higher mAP than the ensemble composed of all base models. For example, the 5-member ensemble $F_{1,3,4,6,7}$ on MS COCO achieves an mAP of 72.38%, which is +4.47% higher than the best member model and +1.68% higher than the ensemble using all seven models, while the cost of ensemble execution is only 71% compared with the ensemble using all base models. Similar observations can be made in the other two datasets.

## 5.2. Defensibility Under Evasion Attacks

We conduct experiments on PASCAL VOC using four state-of-the-art evasion attacks: TOG [6], UEA [32], RAP [15], and DAG [31]. We compare ODEN with three ensemble defense methods (NMW, SoftNMS, and NMS) and adversarial training (AdvDetTrain) [38]. We report the comparison results in **Table 5**. $F_1$ (i.e., FRCNN) is the victim model. We make three observations. First, ODEN outperforms the other three ensemble approaches and the representative adversarial training defense under all four evasion attacks and benign scenarios (2nd column). Second, all five ensemble methods significantly outperform the adversarial training defense under all four evasion attacks and in benign scenarios. Third, the ensemble methods NMW, Soft-NMS, and NMS suffer severely under TOG evasion attack with a low mAP of 13.41~17.56%, showing its poor defensibility. In comparison, AdvDetTrain offers slightly better defensibility under TOG attack (from 2.64% to 34.07%), but the benign mAP drops significantly from 67.37% to

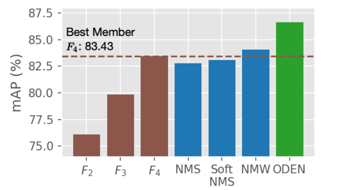
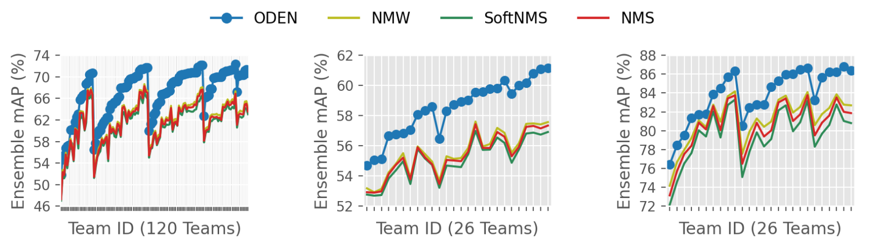



Figure 4. ODEN improves mAP over the best-performing member.




(a) MS COCO (b) Open Images (c) PASCAL VOC

Figure 5. Ensemble mAP comparisons for all possible teams with at least two members. With the same ensemble, ODEN always achieve an ensemble mAP higher than the other approaches.

| | Benign mAP (%) | Attack mAP (%) | | | |
|---|---|---|---|---|---|
| | | TOG | UEA | RAP | DAG |
| **(a) No Protection** | | | | | |
| $F_1$: FRCNN | 67.37 | 2.64 | 18.07 | 4.78 | 3.56 |
| **(b) Protected** | | | | | |
| ODEN | **86.77** | **81.47** | **58.97** | **84.67** | **86.00** |
| NMW [40] | 82.98 | 17.56 | 54.64 | 75.65 | 76.29 |
| SoftNMS [2] | 82.23 | 13.41 | 53.29 | 76.67 | 76.11 |
| NMS [19] | 82.15 | 16.86 | 54.08 | 75.02 | 76.01 |
| AdvDetTrain [38] | 35.99 | 34.07 | 17.67 | 35.60 | 35.58 |

Table 5. Defensibility comparison under four evasion attacks on PASCAL VOC.

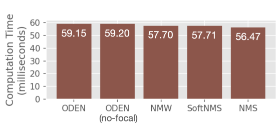



Figure 6. Computation time analysis for detecting objects on an image.

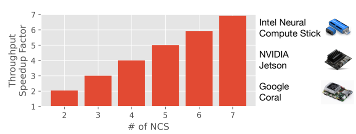



Figure 7. The NCS [12] speedup factor for running ODEN on the edge.

35.99%. We provide the visualization of the defensibility of ODEN against all four evasion attacks in the appendix.

### 5.3. Computation Time Analysis

We compare the average time spent to detect one query image on PASCAL VOC in **Figure 6** using ODEN, ODEN (no-focal), NMW, SoftNMS, and NMS. This includes the model inference and detection combination time in milliseconds. Even though ODEN uses the focal diversity-optimized ensemble, which is $F_{1,2,3,4}$, instead of the ensemble of all five detectors in the base model pool like the other approaches, the computation time is comparable. This is because all ensemble methods can run with parallel execution of all member models [34], as shown in **Figure 7** with Intel Neural Compute Stick 2 [12] on an edge node, demonstrating the increased throughput. The computation time is dominated by the slowest model (i.e., FRCNN), which takes 55.56 milliseconds to compute. Comparatively, the time spent on the ensemble detection inconsistency solver is negligible: 3.60 milliseconds by ODEN, 3.65 milliseconds by ODEN (no-focal), 2.15 milliseconds by NMW, 2.16 milliseconds by SoftNMS, and 0.92 milliseconds by NMS.

## 6. Related Work

Neural network ensembles are known to provide better generalization performance [10, 24]. Most of the existing attempts have been made to create DNN ensembles for image classifiers [17, 33]. In comparison, the DNN ensemble for object detection has received much less attention in both benign scenarios and under recent evasion attacks. Clearly, the consensus with majority voting popularly used for classification ensembles is not applicable. It fails miserably when dealing with detection inconsistency because different detectors may detect different sets of objects in terms of existence, the bounding box size and location of detected objects, and their classification prediction and confidence. NMS [19] and SoftNMS [2] are popularly used to merge disagreeable bounding boxes in training a DNN object detector. Hence, they are used as the baselines for comparison with ODEN. NMW [40] and FUSE [3] are recent enhancements for combining detection results from multiple detectors. Both use a set of hand-picked models pre-trained using different NN backbones to compose an ensemble, where FUSE uses SoftNMS and NMW uses soft-weighting to recompute the confidence for each detection. They do not consider the factor of effective teaming to achieve better performance, which can lead to the potential reduction in computation cost. ODEN is a significant enhancement of FUSE with two novel features: focal diversity-based ensemble selection and a three-tier inconsistency solver for robust detection combination.

## 7. Conclusions

We have presented ODEN, a principled approach to designing and deploying object detection ensembles. ODEN consists of two synergistic functional components: a focal detection diversity-based ensemble selection algorithm and a systematic ensemble detection calibration framework to combine object detection results from multiple detectors. ODEN can effectively identify ensembles with strong synergies and deliver ensemble mAP higher than any individual member detector in the team and outperforms existing representative approaches with higher adversarial robustness.

WACV #1772

## References

[1] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. In *IJCAI*, 2021.

[2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, 2017. 6, 8

[3] Ka-Ho Chow and Ling Liu. Robust object detection fusion against deception. In *ACM SIGKDD*, 2021. 8

[4] Ka-Ho Chow and Ling Liu. Boosting object detection ensembles with error diversity,. In *International Conference on Data Mining*. IEEE, 2022. 3

[5] Ka-Ho Chow, Ling Liu, Mehmet Emre Gursoy, Stacey Truex, Wenqi Wei, and Yanzhao Wu. Understanding object detection through an adversarial lens. In *Springer ESORICS*, 2020. 2, 11

[6] Ka-Ho Chow, Ling Liu, Margaret Loper, Juhyun Bae, Mehmet Emre Gursoy, Stacey Truex, Wenqi Wei, and Yanzhao Wu. Adversarial objectness gradient attacks in real-time object detection systems. In *IEEE TPS-ISA*, 2020. 2, 7, 11, 12

[7] Coral. Coral. https://coral.ai/. [Online; Accessed 2022/02/10].

[8] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015. 2, 6, 11

[9] Di Feng, Ali Harakeh, Steven L Waslander, and Klaus Dietmayer. A review and comparative study on probabilistic object detection in autonomous driving. *IEEE TITS*, 2021. 1, 11

[10] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992. 8

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 11

[12] Intel. Intel neural compute stick 2. https://ark.intel.com/content/www/us/en/ark/products/140109/intel-neural-compute-stick-2.html. [Online; Accessed 2022/02/10]. 8

[13] Paul F Jaeger, Simon AA Kohl, Sebastian Bickelhaupt, Fabian Isensee, Tristan Anselm Kuder, Heinz-Peter Schlemmer, and Klaus H Maier-Hein. Retina u-net: Embarrassingly simple exploitation of segmentation supervision for medical object detection. In *PMLR ML4H Workshop*, 2020. 11

[14] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *IJCV*, 2020. 2, 6

[15] Yuezun Li, Daniel Tian, Xiao Bian, Siwei Lyu, et al. Robust adversarial perturbation on deep proposal-based models. In *BMVC*, 2018. 2, 7, 11, 12

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 6

[17] Ling Liu, Wenqi Wei, Ka-Ho Chow, Margaret Loper, Emre Gursoy, Stacey Truex, and Yanzhao Wu. Deep neural network ensembles against deception: Ensemble diversity, accuracy and robustness. In *IEEE MASS*, 2019. 8

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

[19] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *IEEE ICPR*, 2006. 6, 8

[20] NVIDIA. Autonomous machines: The future of ai. https://www.nvidia.com/en-us/autonomous-machines/. [Online; Accessed 2022/02/10].

[21] Derek Partridge and Wojtek Krzanowski. Software diversity: practical statistics for its measurement and exploitation. *Elsevier IST*, 1997. 3

[22] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 11

[24] Amanda JC Sharkey. *Combining artificial neural nets: ensemble and modular multi-net systems*. Springer Science & Business Media, 2012. 8

[25] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *WOOT*, 2018. 2, 11

[26] Alexander Michael Staff, Jin Zhang, Jingyue Li, Jing Xie, Elizabeth Ann Traiger, Jon Arne Glomsrud, and Kristian Bertheussen Karolius. An empirical study on cross-data transferability of adversarial attacks on object detectors. In *AI-Cybersec@ SGAI*, pages 38–52, 2021.

[27] Pedro Teixidó, Juan Antonio Gómez-Galán, Rafael Caballero, Francisco J Pérez-Grau, José M Hinojo-Montero, Fernando Muñoz-Chavero, and Juan Aponte. Secured perimeter with electromagnetic detection and tracking with drone embedded and static cameras. *Sensors*, 21(21):7379, 2021. 1, 11

[28] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *CVPRW*, 2019. 2, 11

[29] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[30] Yue Wang, Alireza Fathi, Abhijit Kundu, David A Ross, Caroline Pantofaru, Tom Funkhouser, and Justin Solomon. Pillar-based object detection for autonomous driving. In *European Conference on Computer Vision*, pages 18–34. Springer, 2020. 11

[31] Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Mehmet Emre Gursoy, and Yanzhao Wu. Adversarial examples in deep learning: Characterization and divergence. *arXiv preprint arXiv:1807.00051*, 2018. 7

[32] Xingxing Wei, Siyuan Liang, Ning Chen, and Xiaochun Cao. Transferable adversarial attacks for image and video object detection. In *IJCAI*, 2019. 2, 7, 11, 12

[33] Yanzhao Wu and Ling Liu. Boosting deep ensemble performance with hierarchical pruning. In *IEEE ICDM*, 2021. 8

[34] Yanzhao Wu, Ling Liu, and Ramana Kompella. Parallel detection for efficient video analytics at the edge. In *IEEE CogMI*, 2021. 8

[35] Yanzhao Wu, Ling Liu, Zhongwei Xie, Ka-Ho Chow, and Wenqi Wei. Boosting ensemble accuracy by revisiting ensemble diversity metrics. In *CVPR*, 2021. 1, 3

[36] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *ICCV*, 2017. 2, 11, 12

[37] Yiqun Xie, Shashi Shekhar, Richard Feiock, and Joseph Knight. Revolutionizing tree management via intelligent spatial techniques. In *ACM SIGSPATIAL*, 2019. 11

[38] Haichao Zhang and Jianyu Wang. Towards adversarially robust object detection. In *ICCV*, 2019. 7, 8

[39] Xingwei Zhang, Xiaolong Zheng, and Wenji Mao. Adversarial perturbation defense on deep neural networks. *ACM Computing Surveys (CSUR)*, 54(8):1–36, 2021.

[40] Huajun Zhou, Zechao Li, Chengcheng Ning, and Jinhui Tang. Cad: Scale invariant framework for real-time object detection. In *ICCV Workshops*, 2017. 6, 8