2019 IEEE Winter Conference on Applications of Computer Vision

ADA: Adversarial Data Augmentation for Object Detection

Sima Behpour
Univ. of Illinois at Chicago
sbehpo2@uic.edu

Kris M. Kitani
Carnegie Mellon University
kkitani@cs.cmu.edu

Brian D. Ziebart
Univ. of Illinois at Chicago
bziebart@uic.edu

Abstract

The use of random perturbations of ground truth data, such as random translation or scaling of bounding boxes, is a common heuristic used for data augmentation that has been shown to prevent overfitting and improve generalization. Since the design of data augmentation is largely guided by reported best practices, it is difficult to understand if those design choices are optimal. To provide a more principled perspective, we develop a game-theoretic interpretation of data augmentation in the context of object detection. We aim to find an optimal adversarial perturbations of the ground truth data (i.e., the worst case perturbations) that forces the object bounding box predictor to learn from the hardest distribution of perturbed examples for better test-time performance. We establish that the game-theoretic solution (Nash equilibrium) provides both an optimal predictor and optimal data augmentation distribution. We show that our adversarial method of training a predictor can significantly improve test-time performance for the task of object detection. On the ImageNet, Pascal VOC and MS-COCO object detection tasks, our adversarial approach improves performance by about 16%, 5%, and 2% respectively compared to the best performing data augmentation methods.

1. Introduction

There is no guarantee that human-labeled 'ground-truth' annotations of an image dataset are error free. Consider the bounding box annotations of three annotators of the image in Figure 1. Do all boxes contain the object? Are all three bounding boxes equally correct? Is there one bounding box which is most helpful for learning a detection model? These questions highlight the ambiguity in the annotation task. In response, many helpful heuristics have been utilized in the literature to obtain more consistent annotations. To deal with inter-annotator disagreement [1, 2, 3], previous work has relied primarily on reasonable heuristics for augmenting the ground truth through consensus [4, 5, 6, 7]. Despite these efforts, it is not clear if there is a principled approach for identifying the optimal ground truth distribution in the context of supervised learning.

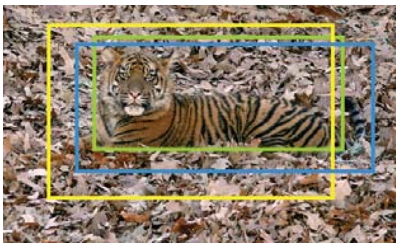

Figure 1. Tiger localization example with three different bounding box annotations illustrates ambiguity in 'ground truth' labels.

To partially address the uncertainty of the 'ground truth' annotations, dataset augmentation methods can be used to synthesize new annotations of images by perturbing annotations. In fact, heuristic data augmentation preprocessing such as random translation, flipping or scaling, has been shown to be essential for many modern visual learning tasks using deep networks. However, manually choosing perturbations to improve performance can be an error-prone process. While increasing the modes of data perturbations may effectively increase the amount of training data, it can also cause the learned predictor to drift. In this way, current techniques for data augmentation are more of an art than a science.

Towards a more principled approach to data augmentation, we propose to integrate annotation perturbations directly into the learning process. We do this by introducing an adversarial function that generates maximally perturbed version of the ground truth, which makes it as hard as possible for the predictor to learn. The adversary, however, is not completely free to perturb the data. It must retain certain feature statistics (*e.g.*, the features of the new bounding box distribution should still be close to the features of the original bounding box). Formally, we pose the data augmentation problem as a zero-sum game between a player (the predictor model) seeking to maximize performance and a constrained adversary (augmented data distribution) seeking to minimize expected performance [8, 9, 10]. This adversarial approach has been applied widely in different research studies and applications like Multi-label prediction [11] and object tracking [12]. To the best of our knowledge, this is the

978-1-7281-1975-5/19/$31.00 ©2019 IEEE
DOI 10.1109/WACV.2019.00137

first work to provide a theoretic basis for data augmentation in terms of an adversarial two player zero-sum game. As a consequence of our game-theoretic formulation, we develop a novel adversarial loss function that identifies the optimal data augmentation strategy which leads to the most robust predictor possible (*i.e.*, trained for the worst case perturbation of data). In our experiments, we focus on the task of object detection and show that our proposed adversarial data augmentation technique consistently improves performance over various competing loss functions, data augmentation levels, and deep network architectures.

2. Related Work

It is common to assume that the ground truth is singular and error-free. However, disagreement between annotators is a widely-known problem for many computer vision tasks [1], as well as a major concern when constructing an annotated computer vision corpora [2]. In large part, the difficulty arises because the set of possible "ground truth" annotations is typically extremely large for vision tasks. It is a powerset of possible descriptions (*e.g.*, words, noun phrases) in annotation tasks, multi-partitions of the pixels (exponential in the number of pixels) in segmentation tasks, and the possible bounding boxes (quadratic in the number of pixels) for localization tasks.

Methods to form a "consensus" annotation and to improve the annotation process through crowd-sourcing have been developed by averaging or combining together different independent annotations [4], verifying annotations with other independent annotators [5], and other strategies [6, 7]. For example, the ILSVRC2012 image dataset employs boundary box drawing, quality verification, and coverage verification as three separate subtasks [13] in a crowdsourcing pipeline. In the construction of that dataset, proposed bounding boxes are rejected 37.8% of the time [13], illustrating the inherent disagreement between annotators and the uncertainty of the task. Despite the added safeguards of the verification process, recent evaluations have also been performed by removing a substantial fraction of the training examples that are considered to have poor quality bounding boxes [3, 14, 15, 16, 17].

Many state-of-the-art methods for object detection [18] are based on CNN, and incorporate other improvements such as the use of very large scale datasets, more efficient GPU computation, and data augmentation [19]. Recently, most of the literature on data augmentation studies effective data augmentation methods for CNN features that increase the performance of different tasks (*e.g.*, classification, object recognition) [20, 21, 22, 23]. Chatfield [19] applies the data augmentation techniques commonly applied to CNN-based methods to shallow methods and shows an analogous performance boost [19]. Paulin et al. [21] claim that given a large set of possible transformations, all transformations are not equally informative and adding uninformative transformations increases training time with no gain in accuracy. They propose Image Transformation Pursuit (ITP) algorithm for the automatic selection of a compact set of transformations.

Complementary to our work, data augmentation can also be used to guard against adversarial attacks [24, 25, 26, 27, 28]. Total variance minimization and image quilting are presented as very effective defenses against adversarial-example attacks on image-classification systems [26]. The strength of these data augmentations lies in their non-differentiable nature and their inherent randomness resulting in difficult defenses for an adversary. Our work is different in that we seek to optimize the data augmentation process as part of a supervised learning problem.

3. Problem Formulation

In order to understand the underlying theory of adversarial data augmentation proposed in this work, we must first understand the role of the *annotation distribution*, $p(y|\vec{x})$, which describes the distribution over labels y (*e.g.*, a bounding box annotation) given a feature vector $\vec{x}$ (*e.g.*, an RGB image). Note that a training dataset $\mathcal{D} = \{y_n, \vec{x}_n\}_{n=1}^{N}$, induces an annotation distribution $p(y|\vec{x})$. In other words, each label y_n in the training set can be interpreted to be a sample from the annotation distribution which is conditioned on a feature vector $\vec{x} \in \mathbb{R}^D$. When there is absolute certainty in the ground truth annotation, the annotation distribution $p(y|\vec{x})$ is an indicator function where it is one for the true label y^* and zero otherwise.

3.1. Data Augmentation

The process of data augmentation is a method of altering the annotation distribution. A typical method for data augmentation generates new examples $\tilde{\mathcal{D}}$ by perturbing the training data $\mathcal{D}$. For example, if the label is a structured output like a bounding box (*i.e.*, a vector of four values), we can generate a new structured label $\tilde{y}$ for the same image by slightly perturbing the original 'ground truth' bounding box y^*. This data augmentation process creates a new underlying annotation distribution $\tilde{p}(y|\vec{x})$. Since data augmentation can be used to generate multiple new labels for the same feature vector $\vec{x}$, the annotation probability $p(y|\vec{x})$ becomes a soft distribution over labels.

Now if we are given a loss function $\ell(\hat{y}, y)$ describing the distance between an estimated label $\hat{y}$ and annotation label y, we can compute the expected loss of the estimated label under the annotation distribution as: $\sum_{y \in \mathcal{Y}} P(y|\vec{x})\ell(\hat{y}, y)$. Notice that the expected loss is the smallest when the estimated label matches the annotation distribution. Conversely, the expected loss grows larger when the estimated label is far from the annotation distribution. It is important to note here that this marginalization over the annotation

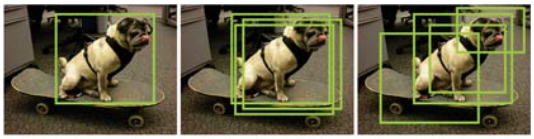

Figure 2. Types of annotation distributions. Adversarial augmentation selects bounding boxes that are maximally different from the ground truth but still contain important object features.

distribution is rarely made explicit in the loss function in most modern supervised learning objective functions because the distribution is assumed to be an indicator function at the 'ground truth' label.

Now consider the probabilistic predictor $f(y|\vec{x})$ which maps a feature vector $\vec{x}$ to a distribution over labels y. The expected loss over the entire dataset $\mathcal{D}$ under the predictor distribution and annotation distribution is defined as:

$$\min_{f\in\Gamma} \sum_{x\in\mathcal{D}} \overbrace{\sum_{y'} f(y'|\vec{x}) \sum_{y} P(y|\vec{x})\ell(y',y)}^{\text{expected loss for input } \vec{x}}. \quad (1)$$

The goal of supervised learning is to find the optimal predictor f (from some set of predictors Γ), that minimizes the above expected loss over the labeled training data. Understanding this verbose form of the supervised learning objective function is critical for the formulation that follows.

3.2. Adversarial Data Augmentation

If we adopt a pessimistic view of the annotated data and assume uncertainty in the 'ground truth' annotations, we can use data augmentation to perturb the 'ground truth' annotations to reflect this uncertainty. We go further and assume the worst case: that the quality of the annotation distribution is *maximally* perturbed. In other words, we make a strong pessimistic assumption that the annotation distribution was generated by an adversary. By making this worst case assumption, we hypothesize that we can train a more robust predictor that is resilient to large perturbations it might encounter at test time. Figure 2 illustrates three possible choices of annotation distributions for a single image.

More formally, instead of the common Empirical Risk Minimization (ERM) objective of Eq. (1), we aim to learn a predictor f that optimizes the following adversarial objective function:

$$\min_{f\in\Gamma} \sum_{x\in\mathcal{D}} \sum_{y'} f(y'|\vec{x}) \max_{P(y|x)} \sum_{y} P(y|\vec{x})\ell(y',y). \quad (2)$$

Notice that the maximization sub-problem has been inserted into the objective function which reflects our assumption that the annotation distribution is adversarial (*i.e.*, the worst case distribution). One might quickly notice that this is an unreasonable objective function without some additional constraints because the adversarial annotation distribution can be arbitrarily bad. In the next section we will incorporate constraints that limit the adversary from deviating very far from the original ground truth annotations.

3.3. Game Formulation

Our claim is that data augmentation should be included in the learning problem instead of being an independent data pre-processing step. By incorporating data augmentation into the predictor learning problem, we obtain a saddle point optimization problem where we pit a predictor trying to minimize the loss, against an adversarial annotation distribution that is trying to maximize the loss. In this form, the optimization can be seen as a *minimax* problem over a zero-sum two-player game.

In the language of game theory, the player (predictor) selects a label from a mixed strategy $y' \sim f(y'|\vec{x})$ to minimize the loss, while the opponent (annotation distribution) selects an annotation from the adversarial distribution $y \sim P(y|\vec{x})$ to maximize the loss. The equilibrium point of the game yields both the optimal predictor and an optimal data annotation distribution. The game is zero-sum because the negative loss of the player (predictor) is exactly the gain of the adversary (annotation distribution).

The value or payoff of the game for a particular feature vector $\vec{x}$ is the expected loss of the predictor distribution against the adversary's annotation distribution:

$$\mathbb{E}_{\substack{y'|\vec{x}\sim f \\ y|\vec{x}\sim P}}\left[\ell(y',y)\right] = \sum_{y',y} f(y'|\vec{x})\ell(y',y)P(y|\vec{x}) = \mathbf{f}^\top \mathbf{G}\mathbf{p}. \quad (3)$$

The expected loss of the game can also be written in matrix form, where $\mathbf{f}$ is the vector of probabilities obtained from the predictor over all labels, $\mathbf{G}$ is the game matrix where each element contains the loss between two labels, and $\mathbf{p}$ is the annotation distribution vector.

The adversarial objective function, Eq. (2) in its current form is problematic because the adversarial annotation distribution is free to perturb the ground truth annotations in arbitrary ways that have no similarity to the original annotations. This can be prevented by constraining the adversarial annotation distribution to choose label distributions in a way that retains feature statistics of the original ground truth annotation. For example, we may want the mean of a set of augmented bounding box annotations to be the same as the mean of the original bounding box annotation. Formally, we can define the first-order statistic of the ground truth data as: $\mathbb{E}_{y,x\sim\mathcal{D}}\left[\phi(y,x)\right] = \frac{1}{N}\sum_{n=1}^{N}\phi(y_n,\vec{x}_n)$, where $(y_n,\vec{x}_n)$ is the n^{th} training example in $\mathcal{D}$. We are now ready to define the constrained adversarial optimization problem.

Definition 1. *The* **Primal Adversarial Data Augmentation** *(ADA-P) game is:*

$$\min_{f}\max_{P} \mathbb{E}_{\substack{\vec{x}\sim\mathcal{D}, \\ y'|\vec{x}\sim f, \\ y|\vec{x}\sim P}}\left[\ell(y',y)\right] \textit{ such that:} \quad (4)$$

$$\mathbb{E}_{\substack{\vec{x}\sim\mathcal{D}, \\ y|\vec{x}\sim P}}\left[\phi(y,\vec{x})\right] = \mathbb{E}_{y,x\sim\mathcal{D}}\left[\phi(y,\vec{x})\right]$$

where $f(y'|\vec{x})$ and $P(y|\vec{x})$ are distributions over all potential predicted labels for each feature vector $\vec{x}$.

Due to strong Lagrangian duality [29], a dual problem with an equivalent solution can be formulated by including the constraint in the objective function using a vector of Lagrangian multipliers, θ. This resulting Lagrangian potential $\theta^\top \phi(\cdot,\cdot)$ links together a set of otherwise independent zero-sum games.

Definition 2. *The* **Dual Adversarial Data Augmentation** *(ADA-D) game is:*

$$\min_{\theta} \mathbb{E}_{\mathbf{x},y^*\sim\mathcal{D}}\Big[\min_{f}\max_{P}\mathbb{E}_{\substack{y'\sim f,\\ y\sim P}}\Big[\ell(y',y) + \theta^\top\{\phi(y,\vec{x})-\phi(y^*,\vec{x})\}\Big]. \tag{5}$$

We make two important observations based on this dual optimization perspective. First, since P is adversarially chosen, there is no need to restrict or parameterize f to avoid overfitting to P as is typically done in supervised learning. Instead, the feature potential based on θ is learned to provide constraints on the adversary that make prediction easier. Further, since P is chosen *after* f in Eq. (5), the predictor is incentivized to randomize.

4. Adversarial Object Detection

Up to this point, we have described our proposed adversarial data augmentation learning approach in general terms, as it can apply to many structured output tasks. Now, we shift our focus to the concrete problem of object detection. This explicit focus will help us to describe our approach in concrete terms.

Label Space. Each structured output label y is represented by the four coordinates of a bounding box. The domain of a label is denoted $\mathcal{Y}$. The set of all possible bounding boxes $\mathcal{Y}$ is very large for an image of modest size and therefore it is rarely practical to evaluate all possible bounding boxes. This means that the sums over labels used in the formulation above (Section 3) are not tractable and that some form of distribution approximation is needed. To discretize the label space $\mathcal{Y}$, we use a bounding box proposal algorithm, Edgebox [30] or a Region Proposal Network [31] to generate a set of k bounding boxes to define the label space $\mathcal{Y}$.

Feature Statistics. To represent the feature statistics $\phi(y,\vec{x})$ of a bounding box y over an image $\vec{x}$, we use the FC7 features of the VGG16 [32] network. Concretely, it is a 4096 dimensional vector over a sub-image defined by the bounding box y. The feature statistic constraint $|\phi(y',\vec{x})-\phi(y^*,\vec{x})|$ described in the ADA-D definition represents the difference between the FC7 features of an arbitrary bounding box y' and the FC7 features of the ground truth bounding box y^*. Also known as the *perceptual loss*, this quantity ensures that the adversarial bounding box label remains perceptually similar to the ground truth bounding box label. **Loss Function.** The loss function used for object detection is based on the classical intersection over union (IoU) score, $IoU(y,y') = \text{area}(y\cap y')/\text{area}(y\cup y')$. Here, y and y' are two bounding boxes. In this work, we focus on losses defined in terms of the amount of non-overlap, $\ell(y,y') = 1 - \text{IoU}(y,y')$, which equals to one when y and y' are disjoint, zero when they are identical, and smoothly transitions in between those extremes. Another loss function we use is the overlap loss with a threshold: $\ell_{t\alpha}(y,y') = 1$ if $\text{IoU}(y,y') < \alpha$ and 0 otherwise, which assigns binary loss to bounding boxes depending on the overlap threshold.

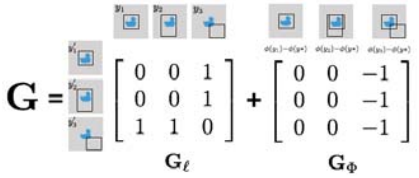

Figure 3. Example Game Matrix for a duck image with three bounding boxes. Each black bounding box is a potential label for the same duck image.

4.1. Game Matrix

As noted in Eq. (3), the expected loss of the adversarial game can be written in matrix form, $\mathbf{f}^\top\mathbf{G}\mathbf{p}$. The game (or payoff) matrix $\mathbf{G}$ for ADA-D is constructed from Eq. (5) as an $|\mathcal{Y}|\times|\mathcal{Y}|$ matrix, where each element is defined as:

$$\mathbf{G}(y',y) = \ell(y',y) + \theta^\top|\phi(y,\vec{x})-\phi(y^*,\vec{x})|, \tag{6}$$

where the first term $\ell(\cdot,\cdot)$ is the IoU based loss and the second term is the weighted difference between FC7 features of the annotation distribution label y and the ground truth y^* label. To better understand the structure of the game matrix, we can decompose it as $\boldsymbol{G} = \boldsymbol{G}_\ell + \boldsymbol{G}_\Phi$. The elements of each matrix are illustrated for a toy example in Figure 3. The first matrix $\boldsymbol{G}_\ell$ contains the pairwise loss between the label of the predictor and the label of the adversary, $\ell(y',y)$. The second matrix $\boldsymbol{G}_\Phi$ contains the difference in feature statistics between the adversary and the ground truth. Since the constraint matrix $\boldsymbol{G}_\Phi$ does not depend on the predictor label y', each row is identical. The elements of the last column are all -1 in this example because the feature statistic of the bounding box y_3 are very different from the ground truth bounding box y^*, whereas the first two columns are zero because the content of their bounding boxes are similar.

4.2. Nash Equilibria

The solution of the game, the Nash equilibrium pair $(\mathbf{f},\mathbf{p})$, is defined as the optimal strategy for each player such that:

$$\max_{\mathbf{p}'}\mathbf{f}^\top\boldsymbol{G}\mathbf{p}' \le \mathbf{f}^\top\boldsymbol{G}\mathbf{p} \le \min_{\mathbf{f}'}\mathbf{f}'^\top\boldsymbol{G}\mathbf{p}. \tag{7}$$

For the example in Figure 3, the lower potential for the third bounding box in $\boldsymbol{G}_\Phi$ offsets the larger loss that might be produced in $\boldsymbol{G}_\ell$ by having $p(y_3) > 0$. Due to the symmetries in $\boldsymbol{G}$, the equilibrium solution is then simply: $P(y_1|\mathbf{x}) = P(y_2|\mathbf{x}) = f(y_1'|\mathbf{x}) = f(y_2'|\mathbf{x}) = 0.5$ and $P(y_3|\mathbf{x}) = f(y_3|\mathbf{x}) = 0$.

In general, this equilibrium solution pair can be obtained efficiently using a pair of linear programs:

$$\min_{v,\mathbf{f}\geq\mathbf{0}} v \text{ such that: } \mathbf{f}^\top\boldsymbol{G} \leq v\mathbf{1} \text{ and } \mathbf{f}^\top\mathbf{1} = 1; \text{ and} \quad (8)$$
$$\max_{v,\mathbf{p}\geq 0} v \text{ such that: } \boldsymbol{G}\mathbf{p} \geq v\mathbf{1}^\top \text{ and } \mathbf{p}^\top\mathbf{1} = 1,$$

where v is the value of the game (*i.e.*, the expected loss). This first linear program finds $\mathbf{f}$ that produces the maximum value against the worst choice of $\mathbf{p}'$ using the left-hand side of Eq. (7) via constraints for each deterministic choice of $\mathbf{p}'$ (i.e., the $\mathbf{1}$ vector). The second linear program is constructed in a likewise manner to obtain $\mathbf{p}$.

4.3. Constraint Generation for Large Games

In practice, forming and solving a zero-sum adversarial game over a very large set of labels (*e.g.*, the set of all possible bounding boxes in an image) for each image is computationally expensive. To obtain the same result more efficiently, we employ a constraint-generation method [33, 10] to solve ADA-D without explicitly constructing the entire payoff matrix $\boldsymbol{G}$. It is based on the key insight that the equilibrium distributions, $\mathbf{f}$ and $\mathbf{p}$, both assign zero probabilities to many bounding boxes and those bounding boxes can then be effectively removed from the game matrix without changing the solution. Constraint-generation uses a set of the most violated constraints to grow a game matrix that supports the equilibrium distribution that is much smaller than the full game matrix.

The approach works by iteratively obtaining a Nash equilibrium for a game defined over a subset of the possible labels (not all of them), finding a player's best response strategy (either the predictor or the annotation distribution) to that equilibrium distribution. Then the best response to the set of opponent strategies defining the game is added as a new strategy. When additional best responses no longer improve either player's game value, the subgame equilibrium is guaranteed to be an equilibrium to the larger game [33].

4.4. Algorithm Details

Algorithm 1 details the ADA equilibrium computation structure. The preprocessing step extracts image box proposals (*e.g.*, EdgeBox or RPN) and their CNN FC features in Lines 1-4. S_p and S_f are set of annotations (box proposals) for $\mathbf{p}$ and $\mathbf{f}$ game players. In the main portion of the algorithm in lines 5-12, `solveGame` obtains a Nash equilibrium using linear programming (Gurobi LP solver

Algorithm 1 ADA Equilibrium Computation

Input: Image $\vec{x}$; Parameters θ; Ground Truth y^*
Output: Nash equilibrium, $(\mathbf{f}, \mathbf{p})$
1: $\mathcal{Y} \leftarrow \text{EdgeBox}(\vec{x})$
2: $\Phi = \text{CNN}(\mathcal{Y}, \vec{x})$
3: $\boldsymbol{\psi} \leftarrow \theta^\top(\Phi - \Phi(y^*))$
4: $\mathcal{S}_p \leftarrow \mathcal{S}_f \leftarrow \text{argmax}_y\, \psi(y)$
5: **repeat**
6: $\quad(\mathbf{f}, \mathbf{p}, v_p) \leftarrow \texttt{solveGame}(\psi(\mathcal{S}_p), \text{loss}(\mathcal{S}_f, \mathcal{S}_p))$
7: $\quad(y_{\text{new}}, v_{\text{max}}) \leftarrow \max_y \mathbb{E}_{y'\sim f}[\text{loss}(y, y') + \psi(y)]$
8: $\quad$**if** $(v_p \neq v_{\text{max}})$ **then** $\mathcal{S}_p \leftarrow \mathcal{S}_p \cup y_{\text{new}}$
9: $\quad(\mathbf{f}, \mathbf{p}, v_f) \leftarrow \texttt{solveGame}(\psi(\mathcal{S}_p), \text{loss}(\mathcal{S}_f, \mathcal{S}_p))$
10: $\quad(y'_{\text{new}}, v_{\text{min}}) \leftarrow \min_{\hat{y}} \mathbb{E}_{y\sim p}[\text{loss}(y, y')]$
11: $\quad$**if** $(v_f \neq v_{\text{max}})$ **then** $\mathcal{S}_f \leftarrow \mathcal{S}_f \cup y'_{\text{new}}$
12: **until** $v_p = v_{\text{max}} = v_f = v_{\text{min}}$
13: **return** $(\mathbf{f}, \mathbf{p})$

[34]) and constraint generation using $\max$ and $\min$ operations is performed in lines 7 and 10. After reaching the loop termination condition (line 12), the $\mathbf{f}$ and $\mathbf{p}$ distributions are returned. In order to obtain the optimal model parameters θ, we perform stochastic gradient descent over data mini-batches $\mathcal{B}$, and use ADA (Algorithm 1) for the inner optimization problem using θ and y^* to compute the (sub)gradient of the minimax objective function in Eq. (5). It can be shown that feature residual is the (sub)gradient of the objective function, as it is the only part of the objective function that depends on θ, yielding updates: $\theta \leftarrow \theta - \lambda(\mathbb{E}_{\substack{\vec{x}\sim\mathcal{B},\\ y|\vec{x}\sim P}}[\phi(y,\vec{x})] - \mathbb{E}_{y,x\sim\mathcal{B}}[\phi(y,\vec{x})])$. We use AdaGrad [35] for the parameter updates to effectively converge near the point where the data augmentation features match the training data features.

Test-time Inference. At testing time, we can compute $\mathbf{f}$ by solving the minimax problem in Eq. (5) using ADA in Algorithm 1. However, there is one important change. Since the parameters θ are fixed at test time, the final term of Eq. (5) is a constant $\theta^\top\phi(y^*, \vec{x})$ and is excluded from the optimization (we do not use the ground truth at test time!). ADA yields two distributions $\mathbf{p}$ and $\mathbf{f}$. Our final prediction is the K (for example, top 5 box proposals for Pascal VOC) most probable predictor bounding box(es) under the equilibrium distribution, $\hat{y} = \text{argmax}_y\, f(y|\vec{x})$.

Extension to Multi-Class Multi-Instance Detection. For ease of exposition, we have limited our above description of ADA to single class detection. We can extend our method for multi-class detection by allowing an additional label option of *no instance detected* and learning parameter vectors for each class $\theta_1, \ldots, \theta_N$. As before, we perform stochastic gradient descent over data mini-batches, and use ADA (Algorithm 1) in the inner optimization problem. However, since we are dealing with multi-class detection, we must perform ADA N times (compute a prediction for every class) for each image against the ground truth y_n^*, where

n is the index of the true class. This produces a big list of (class probability, bounding box) pairs, where the box with the highest probability is treated as the prediction result. If the highest class probability matches the ground truth class, we only update θ_n. If there is a mismatch (wrong class was predicted), then both θ_n and θ_m are updated, where m is the index of the false prediction.

5. Experiments

In the following experiments, we show that our proposed adversarial optimization for data augmentation provides meaningful improvements in test time prediction performance. We first compare our adversarial data augmentation (ADA) objective function against baselines models with no data augmentation on the task of localization in Section 5.1. Second, we compare our approach to baseline models with varying levels of data augmentation in Section 5.2. Third, we evaluate our approach on the task of detection (joint recognition and localization) in Section 5.3. Finally, we use our adversarial optimization over various deep features to show consistent improvements across networks in Section 5.4.

Baselines. We benchmark the performance of our adversarial approach (ADA) for object detection against two classical objective functions.

(1) SSVM: The structured output support vector machine (SSVM) [36] is a large margin classifier with a variable margin depending on a structured loss function ℓ. The objective function is defined as:

$$\hat{\theta} = \underset{\theta}{\text{argmin}} \quad \lambda||\theta||_2 + \sum_n \xi_n \tag{9}$$
$$\text{s.t.} \quad \theta^\top(\phi(y_n^*, \vec{x}_n) - \phi(y, \vec{x}_n)) \geq \ell(y_n^*, y) - \xi_n \quad \forall\, y,$$

where θ is the weight vector, ϕ is the feature function (image feature statistic), ℓ is the loss function and ξ is the slack variable. To solve the SSVM objective function, we employ an iterative constraint generation strategy to accelerate the learning process by adding a few constraints per iteration (instead of the entire constraint set defined by each label $y \in \mathcal{Y}$). At test time, we generate a set of bounding box proposals (EdgeBox) and take the bounding box with the highest potential using the learned weight vector, $\hat{y} = \text{argmax}_y\, \theta^\top \phi(y, \vec{x})$ to identify the predicted structured output.

(2) Softmax: The soft maximum (logistic regression) objective function is a probabilistic predictor. For the softmax objective function, we estimate a distribution over all proposed bounding boxes y that maximizes the conditional likelihood of proposed bounding boxes with an IoU above a given threshold.

$$\hat{\theta} = \underset{\theta}{\text{argmax}} \prod_n P(y_n|\vec{x}_n;\theta) = \underset{\theta}{\text{argmax}} \prod_n \frac{e^{\theta^\top \phi(y_n, \vec{x}_n)}}{\sum_y e^{\theta^\top \phi(y, \vec{x}_n)}},$$

Table 1. No augmentation baseline comparison (IoU>0.5)

Model	ImageNet Object Categories										
	Plane	Bird	Bus	Car	Cat	Cow	Dog	Hors	Moni	Sofa	mAP
ADA+VGG (Ours)	**92.0**	**93.5**	**92.0**	**100.0**	**89.1**	**100.0**	**93.0**	**96.4**	**96.0**	**90.0**	**94.2**
Softmax+VGG	84.0	86.5	84.0	87.0	70.9	77.5	62.0	72.7	72.0	80.0	77.7
SSVM+VGG	90.0	82.5	82.0	82.0	40.0	87.5	72.0	72.7	90.0	78.0	77.7

Table 2. No augmentation baseline comparison (IoU>0.7)

Model	ImageNet Object Categories										
	Plane	Bird	Bus	Car	Cat	Cow	Dog	Hors	Moni	Sofa	mAP
ADA+VGG (Ours)	**58.0**	**61.5**	**64.0**	**91.0**	**30.9**	**77.4**	**58.0**	**58.2**	**61.8**	**61.9**	**62.3**
Softmax+VGG	47.6	45.7	40.0	62.8	20.0	42.5	25.1	25.4	31.4	44.2	38.5
SSVM+VGG	51.8	55.5	44.0	61.7	21.8	54.7	31.6	43.6	56.0	57.3	47.8

where θ is the weight vector, ϕ is the potential function (FC7 feature) and ℓ is the loss function. At test time, we compute the Bayesian optimal decision to identify the most likely bounding box from a set of proposed bounding boxes according to: $\hat{y} = \text{argmin}_y \sum_{y' \in \mathcal{Y}} P(y'|\mathbf{x};\theta)\ell(y, y')$, where $P(y|\mathbf{x};\theta)$ is the learned conditional distribution parameterized by θ and ℓ is the loss function.

5.1. Baseline Comparisons with No Augmentation

We begin with the simplest evaluation, where we compare our proposed adversarial data augmentation approach with two baseline method that use only the ground truth annotation, without augmenting the training data, to learn a predictor. We compare our method **ADA+VGG** against **SSVM+VGG** and **Softmax+VGG**. The suffix **+VGG** for each objective function specifies the deep network from which the features are used, in this case VGG16 [32]. We compute the mean Average Precision (mAP) score for several classes in ImageNet dataset for each of the competing methods. We train a bounding box predictor for each object category, and consider an object to be correctly detected when the IoU is greater than a threshold. We emphasize here that we are decoupling the recognition task from the localization task by learning class specific bounding box regressor and testing only on images that contain the target class. Later experiments will evaluate on both recognition and localization. For this experiment, we give results for two thresholds, 50% IoU and 70% IoU for each object category. Since our method explicitly augments the dataset as part of the optimization process whereas the two baselines have no data augmentation, we expect our approach will outperform the two baselines.

The test time localization accuracy at 50% IoU on 10 object classes from ImageNet are given in Table 1. As expected, we observed significant test time improvement in bounding box regression accuracy for every object category. On average, our proposed adversarial data augmentation approach improved mean average precision by 17% percentage points.

We repeated the same experiment for a more strict loss, a 0.7 thresholded IOU loss function. The mean average precision of the predicted bounding boxes are given in Table 2. Since we are evaluating performance with a more strict loss

Table 3. Effect of Data Augmentation (IoU $> 70\%$)

Augmentation	AlexNet Object Category										
	Plane	Bird	Bus	Car	Cat	Cow	Dog	Hors	Moni	Sofa	Avg
$SSVM_{t50}$+VGG	53.8	57.9	49.7	64.0	22.6	59.9	37.5	45.5	56.7	57.8	50.5
$SSVM_{t60}$+VGG	54.7	58.9	52.7	67.7	23.7	64.9	42.0	48.6	57.3	58.4	52.9
$SSVM_{t70}$+VGG	**56.4**	**61.6**	**56.8**	**70.8**	**25.4**	**67.3**	**49.1**	**51.9**	**58.6**	**58.8**	**55.7**
$SSVM_{t75}$+VGG	52.6	61.0	51.7	64.4	20.2	61.2	42.6	44.0	57.3	56.0	51.1
$SSVM_{t80}$+VGG	49.8	52.0	44.9	60.3	20.2	55.8	33.1	41.4	55.8	52.7	46.6
ADA+VGG (Ours)	**58.0**	**61.5**	**64.0**	**91.0**	**30.9**	**77.4**	**58.0**	**58.2**	**61.8**	**61.9**	**62.3**

function, the absolute mAP values decrease as expected. However, notice that our proposed approach still obtains a significant improvement over the baseline algorithms improving mAP by 15% percentage points over the strongest baseline **SSVM+VGG**.

5.2. Baseline Comparisons with Augmentation

We now compare the performance of our approach to the strongest baseline model, **SSVM+VGG**, trained with different levels of data augmentation. As mention earlier, data augmentation such as random translations of bounding boxes, is a common heuristic used to help supervised learning methods avoid overfitting. We prepare five levels of data augmentation to train the **SSVM+VGG** baseline. Instead of using random translations within a range of the ground truth bounding box annotation, which generate many similar bounding boxes, we use the EdgeBox proposal network to generate a diverse set of bounding boxes. We keep the top 250 EdgeBox proposals with the highest scores and filter them according to five thresholds with respect to the original ground truth bounding box: (1) IoU$>$50%; (2) IoU$>$60%; (3) IoU$>$70%; (4) IoU$>$75%; and (5) IoU$>$80%. We denote the experiment using the subscript `t50` to represent a model trained on a collection of bounding boxes with IoU$>$50%. We consider the bounding boxes that pass the threshold test, as new 'ground truth' and use them as the training set. We note here again that our proposed method automatically selects (gives weights to) the bounding box proposals during the learning process and does not require a separate augmentation step.

The results of this experiment are shown in Table 3. We note that augmenting SSVM in this manner improves test performance compared to the model without data augmentation (Table 2). We also observe that the performance gain is maximized around the 70% overlap threshold. However, we find that with the exception of the *Bird* class, the performance gains of data augmentation do not reach the performance of our ADA approach. Our proposed approach still outperforms the model with the best level data augmentation by 6.4% percentage points.

We also performed a second data augmentation experiments by only varying the number of bounding boxes with the top EdgeBox scores (instead of using IoU) for every image label to understand how the amount of augmented data affects test time performance. At test time, we use the 50% overlap criteria for successful localization. Table 4 shows

Table 4. Effect of Number of Augmented Data Annotations. ADA outperforms best configuration SSVM+VGG baseline by 12%.

SSVM+VGG	k=1	k=2	k=4	k=6	k=8	k=10	k=12	ADA+VGG
mAP	77.6	79.7	81.4	**83.8**	83.7	79.8	75.3	**94.2**

Table 5. Detection Performance Comparison (IoU $> 70\%$).

Model	Image Net Object Category										
	Plane	Bird	Bus	Car	Cat	Cow	Dog	Hors	Moni	Sofa	Avg
ADA+VGG (Ours)	**46.0**	**55.5**	**60.0**	**86.0**	**25.4**	**70.0**	**47.0**	**52.7**	**60.0**	**48.0**	**55.1**
SSVM+VGG	42.0	46.0	38.0	53.0	16.4	52.5	25.0	36.4	42.0	42.0	39.3
Softmax+VGG	40.0	42.5	42.0	55.0	16.4	32.5	16.0	29.1	22.0	34.0	33.0

the mAP performance over same 10 object categories in ImageNet as a function of the number of augmented data annotations per image. The augmented data annotations are selected from a rank list of EdgeBox proposals from each image. The performance of the **SSVM+VGG** tops out at 8 augmented data annotations and is still 12% points below our proposed approach (94.2% mAP).

5.3. Detection Performance Comparison

We now address the object detection task of jointly locating and recognizing the category of an unknown object, to evaluate the performance of our approach on a harder task. In order for a model to obtain a correct result, the predictor must output the correct category label and also generate a bounding box that overlaps with the ground truth by at least 70% IoU. For our baseline models, **SSVM+VGG** and **Softmax+VGG**, we use the best performing data augmentation scheme from Table 3 that includes EdgeBox proposals that have 70% IoU threshold with the original ground truth annotation.

Table 5 shows the object detection performance when evaluated at the 70% IoU threshold for correctness. We again find strong support for our adversarial approach to deal with uncertainty. Specifically, we find that ADA_{70} provides the best performance for all object classes. Though the relative performance advantage differs by object type, for classes like `Dog`, the improvement over the other approaches is nearly double. On average, ADA provides a significant performance improvement of 15.8% percentage points on this task over the strongest performing **SSVM+VGG** baseline.

5.4. Generalization Across Deep Architectures

To understand how our proposed adversarial loss function, ADA, generalizes across various deep architectures, we conduct two sets of experiments.

(1) Generalizing ADA across various deep features. We run this experiment over 20 classes of VOC 2007 dataset, 5000 training images and 4952 testing images. In this experiment setting, the same box proposals are used for different networks but the features are different because the architectures are different. We first extract the top 250 box proposals of every image using

Table 6. ADA Generalization Across Deep Features. VOC2007 mAP for IoU>0.5.

Model	VOC 2007 Object Category																				
	Aero	Bike	Bird	Boat	Bott	Bus	Car	Cat	Chair	Cow	DinT	Dog	Hors	Mbik	Pers	Plnt	Shee	Sofa	Train	TV	mAP
ADA+VGG16	67.6	71.1	67.4	63.0	46.4	75.4	78.5	80.2	**50.7**	77.9	64.2	79.8	71.5	72.8	66.6	30.0	69.7	72.3	**80.2**	61.8	**67.4**
SSVM+VGG16	70.1	74.5	63.2	46.0	43.2	74.8	78.0	78.3	43.2	73.3	61.5	79.2	73.4	72.0	63.7	34.3	67.0	66.8	70.2	**71.1**	65.1
SVM+VGG16 [37]	73.4	**77.0**	63.4	45.4	44.6	75.1	78.1	79.8	40.5	73.7	62.2	79.4	**78.1**	73.1	64.2	35.6	66.8	67.2	70.4	**71.1**	66.0
ADA+AlexNet	61.9	68.8	62.5	62.4	44.9	72.5	74.4	79.5	43.7	81.6	64.2	**81.1**	70.4	68.1	71.2	**38.6**	64.7	69.8	79.0	58.2	**65.9**
SSVM+AlexNet	66.8	72.0	57.3	44.3	41.5	66.6	73.1	69.2	34.9	53.9	54.2	61.6	69.5	68.0	58.8	35.5	63.2	51.6	63.1	62.9	58.4
SVM+AlexNet [37]	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8	58.5
ADA+ResNet101	**74.1**	74.2	**69.5**	**63.1**	**47.9**	**77.1**	**79.8**	**84.2**	47.8	**82.2**	**64.5**	78.1	71.2	**73.0**	**71.4**	36.4	**70.3**	**72.6**	78.1	64.2	**69.0**
SSVM+ResNet101	67.3	68.6	69.3	56.2	47.1	75.9	79.1	83.9	46.0	81.5	63.0	77.9	69.2	72.6	64.6	35.9	68.0	68.6	75.2	64.3	63.8

Table 7. ADA Generalization Across Deep Architectures over VOC dataset (mAP for IoU>0.5)

Model	VOC 2007 Object Category																				
	Model	train	aero	bike	bird	boat	bott	bus	car	cat	chair	cow	table	dog	horse	mbik	prsn	plant	sheep	sofa	train	tv	mAP
ADA+Faster	vgg16	07	77.7	80.1	71.3	60.0	48.1	82.0	80.3	84.5	50.5	77.6	68.2	84.3	75.6	78.3	73.9	41.2	69.9	65.6	75.4	77.4	**71.1**
OHEM[38]	vgg16	07	71.2	78.3	69.2	57.9	46.5	81.8	79.1	83.2	47.9	76.2	68.9	83.2	80.8	75.8	72.7	39.9	67.5	66.2	75.6	75.9	69.9
Faster-RCNN	vgg16	07	73.8	78.5	70.0	57.3	50.2	79.8	78.2	85.1	48.5	74.3	65.7	83.5	76.9	75.7	72.4	40.2	67.6	65.2	70.7	74.1	69.4
ADA+Faster	resnet	07,12	80.1	82.1	75.2	71.4	56.5	86.2	85.5	90.5	88.0	89.0	71.5	88.5	91.5	78.7	79.8	44.2	76.4	80.5	83.4	74.3	**78.7**
Faster-RCNN	resnet	07,12	79.8	80.7	76.2	68.3	55.9	85.1	85.3	89.8	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	85.3	72.0	76.4

EdgeBox. These box proposals are then passed through VGG16 (Matconvnet pre-trained model [39]) to extract the box feature, which are activations of the last fully connected layer (FC7). The box proposals and their features are used to train ADA (called ADA+VGG16 in table 6) and SSVM (called SSVM+VGG16 in table 6). We repeat the same procedure with pre-trained models of Alexnet [16] and ResNet101 [14]. As reference, we also report the result of SVM+VGG16 and SVM+AlexNet FC7 presented in [40]. Table 6 shows that ADA consistently outperforms the baseline models across all deep network architectures. In particular, notice that ADA+ResNet101 outperforms SSVM+ResNet101 by 5% in mAP. This experiment shows that ADA can provide improved performance across different deep features.

(2) Generalizing ADA across state of the art architectures. In this experiment, we benchmark ADA against state of the art deep architectures (Faster-RCNN[31], Fast-RCNN[41], and OHEM[38]) for object detection on Pascal VOC and MS-COCO datasets. We use the pre-trained Faster-RCNN (VGG 16) [31, 42], extract 300 box proposals and their features (512-d for VGG). The width and the height of every box proposal are also appended to the features. We use these box proposals and their features as input for ADA(is called **ADA+Faster** in Tables 7, 8), train it on VOC 2007-training set and evaluate it on VOC2007-testing set. We calculate mAP for this dataset using the evaluation function provided in VOC2007 development kit [43]. The results for **Faster-RCNN** and **ADA+Faster-RCNN** are presented in Table 7. Table 7 shows that ADA+Faster-RCNN outperforms Faster-RCNN for a majority of the classes except *Bottle*, *Cat* and *Horse*. We repeat the same experiment using a pretrained Faster-RCNN with ResNet. We train the network on VOC2007 + VOC2012 training set and test it on VOC2007 testing set, following [31, 42]. **ADA+Faster-RCNN** improves the mAP performance by nearly 2% when compared to Faster-RCNN.

Table 8. ADA Generalization Across Deep Architectures over MS-COCO dataset.(mAP over varying IoU)

Method	proposals	mAP@0.5	mAP@0.75	mAP@[.5,.95]
Fast-RCNN[41]	SS,2000	35.9	19.9	19.7
OHEM[38]	RPN,300	42.5	22.2	22.6
Faster-RCNN	RPN,300	42.7	21.9	21.5
ADA+Faster	RPN,300	**44.3**	**23.8**	**23.6**

We also repeat this experiment on more challenging Microsoft COCO object detection dataset [44]. MS-COCO involves 80 object categories and we run experiment using 80k images for the training set and 20k images for the testing set, following [31, 42]. We evaluate the mAP averaged (using COCO's standard metric) for IOU $\in [0.5 : 0.05 : 0.95]$ denoted as mAP@[.5,.95] in Table 8) and mAP@0.5 (PASCAL VOC's metric [43]). Results in Table 8 show that ADA+Faster-RCNN provides 2% improvement in mAP in comparison with Faster-RCNN. The reported mAP at different thresholds confirm that **ADA+Faster-RCNN** consistently outperforms state of the art methods.

6. Conclusions

In this paper, we have developed a game-theoretic formulation for data augmentation that perturbs image annotations adversarially. This provides robustness in the learned predictor that is achieved by training from a number of augmentations that are adaptively selected to be difficult, while still approximating the ground truth annotation. We demonstrated the benefits for object localization and detection using experiments over ten different object classes for the ILSVRC2012 dataset, twenty different object classes for the VOC2007 dataset and MS-COCO dataset, showing significant improvements for our approach under 50% and 70% thresholded IoU evaluation measures.

Acknowledgments

This research was supported in part by NSF CAREER grant #1652530 and RI grant #1526379.

References

[1] Welinder, P., Branson, S., Perona, P., Belongie, S.J.: The multidimensional wisdom of crowds. In: Advances in neural information processing systems. (2010) 2424–2432

[2] Nowak, S., Rüger, S.: How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In: Proceedings of the international conference on Multimedia information retrieval, ACM (2010) 557–566

[3] Szegedy, C., Ioffe, S., Vanhoucke, V.: Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261 (2016)

[4] Upchurch, P., Sedra, D., Mullen, A., Hirsh, H., Bala, K.: Interactive consensus agreement games for labeling images. (2016)

[5] Saragih, J.M., Lucey, S., Cohn, J.F.: Deformable model fitting by regularized landmark mean-shift. International Journal of Computer Vision **91**(2) (2011) 200–215

[6] Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: CVPR, 2012 IEEE Conference on, IEEE (2012) 2879–2886

[7] Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: A semi-automatic methodology for facial landmark annotation. In: The IEEE Conference on CVPR Workshops. (June 2013)

[8] Grünwald, P.D., Dawid, A.P.: Game theory, maximum entropy, minimum discrepancy, and robust Bayesian decision theory. Annals of Statistics **32** (2004) 1367–1433

[9] Asif, K., Xing, W., Behpour, S., Ziebart, B.D.: Adversarial cost-sensitive classification. In: Proceedings of the Conference on UAI. (2015)

[10] Wang, H., Xing, W., Asif, K., Ziebart, B.: Adversarial prediction games for multivariate losses. In: Advances in Neural Information Processing Systems. (2015) 2710–2718

[11] Behpour, S., Xing, W., Ziebart, B.: Arc: Adversarial robust cuts for semi-supervised and multi-label classification. In: AAAI Conference on Artificial Intelligence. (2018)

[12] Fathony, R., Behpour, S., Zhang, X., Ziebart, B.: Efficient and consistent adversarial bipartite matching. In Dy, J., Krause, A., eds.: Proceedings of the 35th International Conference on Machine Learning. Volume 80 of Proceedings of Machine Learning Research., Stockholmsmssan, Stockholm Sweden, PMLR (10–15 Jul 2018) 1457–1466

[13] Su, H., Deng, J., Fei-Fei, L.: Crowdsourcing annotations for visual object detection. In: Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence. (2012)

[14] Gokberk Cinbis, R., Verbeek, J., Schmid, C.: Multi-fold mil training for weakly supervised object localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2014) 2409–2416

[15] Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 2691–2699

[16] Sukhbaatar, S., Fergus, R.: Learning from noisy labels with deep neural networks. arXiv preprint arXiv:1406.2080 **2**(3) (2014) 4

[17] Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., Rabinovich, A.: Training deep neural networks on noisy labels with bootstrapping. arXiv preprint arXiv:1412.6596 (2014)

[18] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105

[19] Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. arXiv preprint arXiv:1405.3531 (2014)

[20] Taylor, L., Nitschke, G.: Improving deep learning using generic data augmentation. arXiv preprint arXiv:1708.06020 (2017)

[21] Paulin, M., Revaud, J., Harchaoui, Z., Perronnin, F., Schmid, C.: Transformation pursuit for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 3646–3653

[22] Masi, I., Trn, A.T., Hassner, T., Leksut, J.T., Medioni, G.: Do we really need to collect millions of faces for effective face recognition? In: European Conference on Computer Vision, Springer (2016) 579–596

[23] He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence **37**(9) (2015) 1904–1916

[24] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. (2014) 2672–2680

[25] Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. arXiv preprint arXiv:1610.08401 (2016)

[26] Guo, C., Rana, M., Cisse, M., van der Maaten, L.: Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117 (2017)

[27] Joon Oh, S., Fritz, M., Schiele, B.: Adversarial image perturbation for privacy protection–a game theory perspective. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 1482–1491

[28] Tramèr, F., Kurakin, A., Papernot, N., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204 (2017)

[29] Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge university press (2004)

[30] Larry Zitnick, P.D.: Edge boxes: Locating object proposals from edges. In: ECCV, European Conference on Computer Vision (September 2014)

[31] Yang, J., Lu, J., Batra, D., Parikh, D.: A faster pytorch implementation of faster r-cnn. https://github.com/jwyang/faster-rcnn.pytorch (2017)

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

[33] McMahan, H.B., Gordon, G.J., Blum, A.: Planning in the presence of cost functions controlled by an adversary. In: Proceedings of the International Conference on Machine Learning. (2003) 536–543

[34] Gurobi Optimization, I.: Gurobi optimizer reference manual (2016)

[35] Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**(Jul) (2011) 2121–2159

[36] Vedaldi, A.: A MATLAB wrapper of $\mathrm{SVM}^{\mathrm{struct}}$. `http://www.vlfeat.org/~vedaldi/code/svm-struct-matlab.This` (2011)

[37] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation tr. In: arXiv:1311.2524. (2014)

[38] Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 761–769

[39] Vedaldi, A., Lenc, K.: Matconvnet: Convolutional neural networks for matlab. In: Proceedings of the 23rd ACM international conference on Multimedia, ACM (2015) 689–692

[40] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2014) 580–587

[41] Girshick, R.: Fast r-cnn. arXiv preprint arXiv:1504.08083 (2015)

[42] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS). (2015)

[43] Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge 2007 (voc 2007) results (2007) (2008)

[44] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, Springer (2014) 740–755

[45] Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. Journal of machine learning research **2**(Dec) (2001) 265–292

[46] Liu, Y.: Fisher consistency of multicategory support vector machines. In: International Conference on Artificial Intelligence and Statistics. (2007) 291–298

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

[48] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia, ACM (2014) 675–678