PatchDEMUX: A Certifiably Robust Framework for Multi-label Classifiers Against Adversarial Patches

Dennis Jacob
UC Berkeley
Berkeley, CA, USA
djacob18@berkeley.edu

Chong Xiang
Princeton University
Princeton, NJ, USA
cxiang@princeton.edu

Prateek Mittal
Princeton University
Princeton, NJ, USA
pmittal@princeton.edu

Abstract

Deep learning techniques have enabled vast improvements in computer vision technologies. Nevertheless, these models are vulnerable to adversarial patch attacks which catastrophically impair performance. The physically realizable nature of these attacks calls for certifiable defenses, which feature provable guarantees on robustness. While certifiable defenses have been successfully applied to single-label classification, limited work has been done for multi-label classification. In this work, we present PatchDEMUX, a certifiably robust framework for multi-label classifiers against adversarial patches. Our approach is a generalizable method which can extend any existing certifiable defense for single-label classification; this is done by considering the multi-label classification task as a series of isolated binary classification problems to provably guarantee robustness. Furthermore, in the scenario where an attacker is limited to a single patch we propose an additional certification procedure that can provide tighter robustness bounds. Using the current state-of-the-art (SOTA) single-label certifiable defense PatchCleanser as a backbone, we find that PatchDEMUX can achieve non-trivial robustness on the MS-COCO and PASCAL VOC datasets while maintaining high clean performance[1].

1. Introduction

Deep learning-based computer vision systems have helped transform modern society, contributing to the development of technologies such as self-driving cars, facial recognition, and more [18]. Unfortunately, these performance boosts have come at a security cost; attackers can use *adversarial patches* to perturb patch-shaped regions in images and fool deep learning systems [4, 30]. The patch threat model presents a unique challenge for the security community due to its physically-realizable nature; for instance, even a single well-designed patch that is printed out can induce failure in the wild [4, 12, 26].

The importance of adversarial patches has made the design of effective defenses a key research goal. Defense strategies typically fall into one of two categories: *empirical defenses* and *certifiable defenses*. The former leverages clever observations and heuristics to prevent attacks, but can be vulnerable to *adaptive attacks* which bypass the defense through fundamental weaknesses in design [5, 14, 25]. As a result, certifiable defenses against patch attacks have become popular for computer vision tasks such as single-label classification and object detection [6, 17, 24, 29, 31–35, 37]; these methods feature provable guarantees on robustness under any arbitrary patch attack.

Despite these successes, progress on certifiable defenses against patch attacks has been limited for multi-label classification. Multi-label classifiers provide important capabilities for simultaneously tracking multiple objects while maintaining scalability. Many safety-critical systems depend on the visual sensing capabilities of multi-label classifiers, such as traffic pattern recognition in autonomous vehicles [16], video surveillance [10], and product identification for retail checkout [13]. Some of these applications have become mainstream in industry (i.e., Waymo robotaxis, Just Walk Out checkout, etc.).

To address this challenge we propose PatchDEMUX, a certifiably robust framework against patch attacks for the multi-label classification domain. Our design objective is to extend any existing certifiable defense for single-label classification to the multi-label classification domain. To do so, we leverage the key insight that any multi-label classifier can be separated into individual binary classification tasks. This approach allows us to bootstrap notions of certified robustness based on precision and recall; these are lower bounds on performance which are guaranteed *across all attack strategies in the patch threat model*. We also consider the scenario where an attacker is restricted to a single patch and propose a novel certification procedure that achieves stronger robustness bounds by using constraints in vulnerable patch locations.

We find that PatchDEMUX achieves non-trivial robustness on the MS-COCO and PASCAL VOC datasets while maintaining high performance on clean data. Specifically, when using the current SOTA single-label certifiable defense PatchCleanser as a backbone, PatchDEMUX attains 85.276% average precision on clean MS-COCO images and 44.902% certified robust average precision. On the PASCAL VOC dataset PatchDEMUX achieves 92.593% clean average precision and 56.030%

[1]Our source code is available at https://github.com/inspire-group/PatchDEMUX

certified robust average precision. For reference, an undefended model achieves 91.146% average precision on clean MSCOCO images and 96.140% average precision on clean PASCAL VOC images. Overall, the key contributions of our work can be summarized as follows:

- We address the challenge of patch attacks in the multi-label domain via a general framework that can interface with any existing/future single-label defense. To the best of our knowledge, our approach is the first of its kind.
- Our framework provably guarantees lower bounds on performance irrespective of the chosen patch attack (i.e., the patch can contain an optimized attack, random noise, etc.).
- We instantiate a version of our defense framework with the current SOTA single-label defense and achieve strong robust performance on popular benchmarks.

We hope that future work will integrate with the PatchDEMUX framework and further strengthen the robustness of multi-label classifiers to adversarial patches.

2. Problem Formulation

In this section, we provide a primer on the multi-label classification task along with standard metrics for evaluation. We next outline the adversarial patch threat model and its relevance in the multi-label setting. Finally, we discuss the concept of certifiable defenses and how they have been used so far to protect single-label classifiers against the patch attack.

2.1. Multi-label classification

Multi-label classification is a computer vision task where images $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{w \times h \times \gamma}$ with width w, height h, and number of channels γ contain multiple objects simultaneously, with each object belonging to one of c classes [38]. A classifier is then tasked with recovering each of objects present in an image. Note that this contrasts single-label classification, where exactly one object is recovered from an image.

More rigorously, each input datapoint is a pair $(\mathbf{x},\mathbf{y})$ where $\mathbf{x} \in \mathcal{X}$ corresponds to an image and $\mathbf{y} \in \mathcal{Y}$ is the associated image label. Each label $\mathbf{y} \in \mathcal{Y} \subseteq \{0,1\}^c$ is a bitstring where $\mathbf{y}[i]=1$ means class i is present and $\mathbf{y}[i]=0$ means class i is absent; this implies that the set of labels $\mathcal{Y}$ is 2^c in size, i.e., exponential. A *multi-label classifier* $\mathbb{F}:\mathcal{X}\to\mathcal{Y}$ is then trained with a loss function such that the predicted label $\hat{\mathbf{y}} := \mathbb{F}(\mathbf{x})$ is equivalent to $\mathbf{y}$. One popular loss function used for training is asymmetric loss (ASL) [3].

To evaluate the performance of a multi-label classifier, it is common to compute the number of *true positives* (i.e., classes i where $\mathbf{y}[i] = \hat{\mathbf{y}}[i] = 1$), the number of *false positives* (i.e., classes i where $\mathbf{y}[i]=0$ and $\hat{\mathbf{y}}[i]=1$), and the number of *false negatives* (i.e., classes i where $\mathbf{y}[i]=1$ and $\hat{\mathbf{y}}[i]=0$). These can be summarized by the *precision* and *recall* metrics [38]:

$$precision = \frac{TP}{TP+FP} \qquad recall = \frac{TP}{TP+FN} \tag{1}$$

The values TP, FP, and FN represent the number of true positives, false positives, and false negatives respectively.

2.2. The patch threat model

Theoretical formulation. In the patch threat model, attackers possess the ability to arbitrarily adjust pixel values within a restricted region located anywhere on a target image $\mathbf{x} \in \mathcal{X}$; the size of this region can be tuned to alter the strength of the attack [4]. As discussed in Sec. 1, defending against this threat model is critical due to its physically realizable nature [4, 12, 26]. In this paper, we primarily focus on defending against a single adversarial patch as it is a popular setting in prior work [6, 17, 24, 31–33, 35]. However, our baseline certification methods can also handle multiple patches, provided the underlying single-label defense strategy already has this capability [33].

We can formally specify patch attacks for an image $\mathbf{x} \in \mathcal{X}$ as follows. Define $\mathcal{R} \subseteq \{0,1\}^{w \times h}$ as the set of binary matrices which represent restricted regions, where elements inside the region are 0 and those outside the region are 1 [33]. Then, the associated patch attacks are:

$$S_{\mathbf{x},\mathcal{R}} := \{\mathbf{r} \circ \mathbf{x} + (\mathbf{1}-\mathbf{r}) \circ \mathbf{x}' | \mathbf{x}' \in \mathcal{X}, \mathbf{r} \in \mathcal{R}\} \tag{2}$$

The $\circ$ operator refers to element-wise multiplication with broadcasting to ensure shape compatibility. Note that this formulation demonstrates how the patch attack can be considered a special case of the ℓ_0-norm threat model [17].

Adversarial patches in the multi-label setting. Patch attacks in multi-label classification aim to induce class mismatches between a ground-truth label $\mathbf{y} \in \mathcal{Y}$ and prediction $\hat{\mathbf{y}} \in \mathcal{Y}$. Unlike single-label classification, different types of mismatches are possible in this setting; for instance, patches can increase the number of false negatives and/or the number of false positives predicted by the classifier $\mathbb{F}$. In general, adversarial patches are generated by representing the desired objective as an optimization problem and then applying an iterative technique such as projected gradient descent (PGD) over $S_{\mathbf{x},\mathcal{R}}$ [21].

2.3. Certifiable defenses against patch attacks

At a high-level, certifiable defenses against patch attacks (CDPA) provide provable guarantees on performance for deep learning-based computer vision systems $\mathbb{F}:\mathcal{X}\to\mathcal{Y}$ against all possible attacks in the patch threat model [6, 17, 24, 29, 31–35, 37]. This ensures that defense robustness will not be compromised by future adaptive attacks.

We formulate a CDPA as having an inference procedure and a certification procedure; additional security parameters, denoted by σ, manage the trade-off between robust performance and inference time [33]. The inference procedure $INFER_{[\mathbb{F},\sigma]} : \mathcal{X} \to \mathcal{Y}$ takes an image $\mathbf{x} \in \mathcal{X}$ as input and outputs a prediction $\hat{\mathbf{y}} \in \mathcal{Y}$. The quality of prediction $\hat{\mathbf{y}}$ with respect to the ground-truth label $\mathbf{y}$ can be evaluated using a performance metric (e.g., precision, recall), which we denote by $\rho:\mathcal{Y}\times\mathcal{Y}\to\mathbb{R}$. In addition to the inference procedure, the certification procedure $CERT_{[\mathbb{F},\sigma]} : \mathcal{X} \times \mathcal{Y} \times \mathbb{P}(\mathcal{R}) \to \mathbb{R}$ ($\mathbb{P}()$ denotes power set) takes image $\mathbf{x}$, ground-truth label $\mathbf{y}$, and the threat model represented by the set of allowable patch regions $\mathcal{R}$ to determine the worst possible performance of $INFER$ on image $\mathbf{x}$. The certification procedure is only used for evaluation.

Formally, for a performance metric ρ and a patch threat model $S_{\mathbf{x},\mathcal{R}}$ we will have

$$\rho(INFER_{[\mathbb{F},\sigma]}(\mathbf{x}'),\mathbf{y}) \geq \tau, \forall \mathbf{x}' \in S_{\mathbf{x},\mathcal{R}} \tag{3}$$

Here, $\tau := CERT_{[\mathbb{F},\sigma]}(\mathbf{x},\mathbf{y},\mathcal{R})$ is the lower bound of model prediction quality against an adversary who can use any patch region $\mathbf{r} \in \mathcal{R}$ and introduce arbitrary patch content. Datapoints with a non-trivial lower bound are considered *certifiable*.

We can summarize these concepts as follows.

Definition 1 (CDPA). *A* certifiable defense against patch attacks (CDPA) *for model $\mathbb{F}: \mathcal{X} \to \mathcal{Y}$ is a tuple of procedures $DEF := (INFER_{[\mathbb{F},\sigma]} : \mathcal{X} \to \mathcal{Y}, CERT_{[\mathbb{F},\sigma]} : \mathcal{X} \times \mathcal{Y} \times \mathbb{P}(\mathcal{R}) \to \mathbb{R})$ where the former is the inference procedure, the latter is the certification procedure, and $\sigma \subseteq \{0,1\}^*$ are security parameters. Certifiable datapoints satisfy Eq. (3) for a performance metric $\rho: \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.*

We note that we have different ρ for different tasks. For instance, CDPAs for single-label classifiers ensure that the output label is preserved for certifiable datapoints[2].

Definition 2 (Single-label CDPA). *A* single-label CDPA *is a CDPA for single-label classifiers $\mathbb{F}_s : \mathcal{X} \to \{1,2,...,c\}$. The performance metric is $\rho(y_1,y_2) := [y_1 = y_2]$. The certification procedure $CERT$ evaluates to 1 for certifiable datapoints and 0 otherwise.*

For multi-label classification, we consider the interpretation where the performance metric is $\rho(\mathbf{y_1},\mathbf{y_2}) := \Sigma_{i=1}^{c}[\mathbf{y_1}[i] = 1 \cap \mathbf{y_2}[i] = 1]$ and $CERT$ lower bounds the number of true positives. This helps bootstrap robust metrics such as certified precision and recall (see Sec. 3.3).

2.4. Certifiable defenses for single-label classifiers against patch attacks

A variety of CDPA have been developed for single-label classifiers [6, 17, 24, 29, 31–33, 35]. Current techniques roughly fall into one of two categories: *small receptive field* defenses and *masking* defenses. With regards to the former, the general principle involves limiting the set of image features exposed to the undefended model and then robustly accumulating results across several evaluation calls. Some examples of this approach include De-randomized Smoothing [17], BagCert [24], and PatchGuard [31, 32]. On the other hand, masking defenses curate a set of masks to provably occlude an adversarial patch regardless of location. PatchCleanser, the current SOTA certifiable defense, uses such a method [33]. Our proposed framework PatchDEMUX is theoretically compatible with any of these techniques.

3. PatchDEMUX Design

In this section we propose *PatchDEMUX*, a certifiably robust framework for multi-label classifiers against patch attacks. We first outline the key property that any multi-label classification problem can be separated into constituent binary classification tasks. Next, we use this observation to construct a generalizable framework which can theoretically integrate any existing single-label CDPA. We then describe the inference and certification procedures in more detail along with robust evaluation metrics. Finally, we propose a novel location-aware certification method which provides tighter robustness bounds.

3.1. An overview of the defense framework

Isolating binary classifiers in multi-label classification. As discussed in Sec. 2.1, labels $\mathbf{y} \in \{0,1\}^c$ in multi-label classification are bitstrings where $\mathbf{y}[i] \in \{0,1\}$ corresponds to the presence/absence of class $i \in \{1,2,...,c\}$. Note that predictions for each class $\mathbf{y}[i]$ are independent of each other; therefore, the multi-label classification task can be represented as a series of isolated binary classification problems corresponding to each class. This motivates a defense formulation for multi-label classifiers in terms of "isolated" binary classifiers, where each class is individually protected by a single-label CDPA. Given a multi-label classifier[3] $\mathbb{F}: \mathcal{X} \to \mathcal{Y}$, we use the notation $\mathbb{F}[i]: \mathcal{X} \to \{0,1\}$ to refer to the isolated classifier for class i.

In practice, defining the isolated classifier is complicated as some single-label CDPA designs require architectural restrictions [24, 31, 32]. Nevertheless, a workaround is possible; specifically, we can initialize the multi-label classifier $\mathbb{F}: \mathcal{X} \to \mathcal{Y}$ as an ensemble of c binary classifiers which each satisfy the required architecture. Then, for each class $i \in \{1,2,...,c\}$ we can define the isolated classifier $\mathbb{F}[i]$ as the associated ensemble model. Other defenses are architecture-agnostic [33]. In these cases we can use any off-the-shelf multi-label classifier $\mathbb{F}: \mathcal{X} \to \mathcal{Y}$ and for each class $i \in \{1,2,...,c\}$ define the isolated classifier $\mathbb{F}[i]$ as having outputs $\mathbb{F}[i](\mathbf{x}) := \mathbb{F}(\mathbf{x})[i]$ for all $\mathbf{x} \in \mathcal{X}$.

Our framework. At a high-level, the PatchDEMUX defense framework takes advantage of the isolation principle to extend any existing single-label CDPA to the multi-label classification task. The *PatchDEMUX inference procedure* consists of three stages (see Fig. 1). In the input stage, it preprocesses the input image $\mathbf{x} \in \mathcal{X}$. In the demultiplexing stage it isolates binary classifiers for each class $i \in \{1,2,...,c\}$ and applies the underlying single-label CDPA inference procedure. Finally, in the aggregation stage we return the final prediction vector by pooling results from the individual classes. The *PatchDEMUX certification procedure* works similarly. It separately applies the underlying single-label CDPA certification procedure to each isolated classifier and then creates a lower bound for true positives by accumulating the results.

3.2. PatchDEMUX inference procedure

The PatchDEMUX inference procedure is described in Algorithm 1. We first take the inference procedure $SL\text{-}INFER$ from a single-label CDPA and prepare it with security parameters σ. On line 2, we initialize a $preds \in \{0,1\}^c$ array to keep track of individual class predictions. Finally, on line 4 we run $SL\text{-}INFER$ with the isolated binary classifier $\mathbb{F}[i]$ on input

[2] We use Iverson bracket notation for convenience

[3] From here on, $\mathcal{Y}$ will denote a multi-label label set with c classes

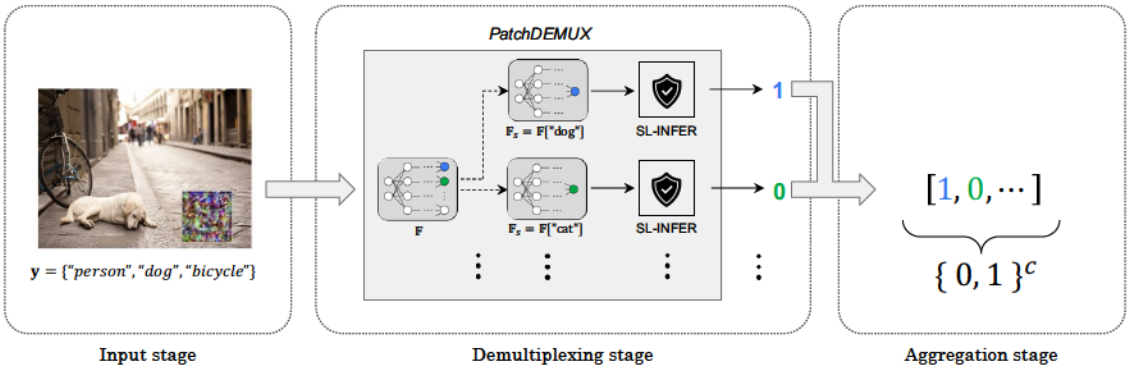


Figure 1. *A diagram which illustrates the defense framework from PatchDEMUX. In the input stage, the (potentially attacked) image is preprocessed. In the demultiplexing stage, the $SL\text{-}INFER$ inference procedure from a single-label CDPA is applied to each individual class in the multi-classification task. This is done by considering the multi-label classifier $\mathbb{F}$ as a series of isolated binary classifiers $\mathbb{F}[i]$ for $i \in \{1,2,...,c\}$. Finally, in the aggregation stage the individual outputs are returned as a single label.*

Algorithm 1 *The inference procedure associated with PatchDEMUX*

Input: Image $\mathbf{x} \in \mathcal{X}$, multi-label classifier $\mathbb{F} : \mathcal{X} \to \mathcal{Y}$, inference procedure $SL\text{-}INFER$ and security parameters σ from a single-label CDPA, number of classes c
Output: Prediction $preds \in \{0,1\}^c$

procedure DEMUXINFER($\mathbf{x}$,$\mathbb{F}$,$SL\text{-}INFER$,σ,c)
  $preds \leftarrow \{0\}^c$ ▷ Set predictions to zero vector
  for $i \leftarrow 1$ to c **do** ▷ Consider classes individually
    $preds[i] \leftarrow$ SL-INFER$_{[\mathbb{F}[i],\sigma]}(\mathbf{x})$
  end for
  return $preds$
end procedure

image $\mathbf{x}$ and update $preds$ for class i.

Remark. If the time complexity for $SL\text{-}INFER$ is $\mathcal{O}(f(n))$, the time complexity for Algorithm 1 will be $\mathcal{O}(c \cdot f(n))$. However, in practice it is possible to take advantage of relatively negligible defense post-processing and effectively reduce the time complexity to $\mathcal{O}(f(n))$. See *Supplementary Material*, Appendix G.

3.3. PatchDEMUX certification procedure

The PatchDEMUX certification procedure is outlined in Algorithm 2. We first initialize the certification procedure $SL\text{-}CERT$ from a single-label CDPA with security parameters σ. On line 2, we create the $\boldsymbol{\kappa}$ array to store certifiable classes. On line 5, we run $SL\text{-}CERT$ with the isolated binary classifier $\mathbb{F}[i]$ on datapoint $(\mathbf{x},\mathbf{y}[i])$ and place the result in $\boldsymbol{\kappa}[i]$; recall from Definition 2 that $SL\text{-}CERT$ returns 1 for protected datapoints and 0 otherwise. Finally, on lines $7-10$ we count a successful true positive for classes with $\mathbf{y}[i]=1$ and $\boldsymbol{\kappa}[i]=1$. Otherwise, we assign a false negative or false positive as we cannot guarantee the accuracy of these classes. We now establish the correctness of these bounds.

Algorithm 2 *The certification procedure associated with PatchDEMUX*

Input: Image $\mathbf{x} \in \mathcal{X}$, ground-truth $\mathbf{y} \in \mathcal{Y}$, multi-label classifier $\mathbb{F} : \mathcal{X} \to \mathcal{Y}$, certification procedure $SL\text{-}CERT$ and security parameters σ from a single-label CDPA, patch locations $\mathcal{R}$
Output: Certified number of true positives TP_{lower}, false positives upper bound FP_{upper}, false negatives upper bound FN_{upper}, class certification list $\boldsymbol{\kappa}$

1: **procedure** DEMUXCERT($\mathbf{x}$,$\mathbf{y}$,$\mathbb{F}$,$SL\text{-}CERT$,σ,$\mathcal{R}$)
2: $c \leftarrow$ len($\mathbf{y}$)
3: $\boldsymbol{\kappa} \leftarrow [0]^c$
4: **for** $i \leftarrow 1$ to c **do** ▷ Certify each class separately
5: $\boldsymbol{\kappa}[i] \leftarrow$ SL-CERT$_{[\mathbb{F}[i],\sigma]}(\mathbf{x},\mathbf{y}[i],\mathcal{R})$
6: **end for**
7: ▷ Compute robust metrics
8: $TP_{lower}, FP_{upper}, FN_{upper} \leftarrow 0,0,0$
9: $TP_{lower} \leftarrow \Sigma_{i=1}^{c}[\boldsymbol{\kappa}[i]=1 \cap \mathbf{y}[i]=1]$
10: $FP_{upper} \leftarrow \Sigma_{i=1}^{c}[\boldsymbol{\kappa}[i]=0 \cap \mathbf{y}[i]=0]$
11: $FN_{upper} \leftarrow \Sigma_{i=1}^{c}[\boldsymbol{\kappa}[i]=0 \cap \mathbf{y}[i]=1]$
12: **return** $TP_{lower}, FP_{upper}, FN_{upper}, \boldsymbol{\kappa}$
13: **end procedure**

Theorem 1 (Algorithm 2 Correctness). *Suppose we have an image data point $(\mathbf{x},\mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$, a single-label CDPA $SL\text{-}DEF$, and a multi-label classification model $\mathbb{F} : \mathcal{X} \to \mathcal{Y}$. Then, under the patch threat model $S_{\mathbf{x},\mathcal{R}}$ the bounds returned by Algorithm 2 are correct.*

Proof. See *Supplementary Material*, Appendix A. □

Thus, using Algorithm 2 we have established the lower bound on true positives (TP_{lower}) and the upper bound on both false positives (FP_{upper}) and false negatives (FN_{upper}) when using Algorithm 1. This allows us to bootstrap notions of *certified precision* and *certified recall* by referencing Eq. (1):

$$certifiedprecision = \frac{TP_{lower}}{TP_{lower} + FP_{upper}} \quad (4)$$

$$certifiedrecall = \frac{TP_{lower}}{TP_{lower} + FN_{upper}} \quad (5)$$

Note by construction that both metrics provide lower bounds for precision and recall on a datapoint (x,y) *irrespective of any attempted patch attack*; the real-world performance of our defense will always be higher. Therefore, an empirical evaluation of existing multi-label attack vectors is not necessary [1, 2, 22, 23]. Furthermore, micro-averaging these metrics across datapoints provides lower bounds on precision and recall for an entire dataset [38].

3.4. Location-aware certification

We now discuss an improved method called *location-aware certification* which extends Algorithm 2. This method works in the scenario where an attacker is restricted to a single patch. The general intuition is that if we track vulnerable patch locations for each class, we can use the constraint that an adversarial patch can only be placed at one location to extract stronger robustness guarantees. For instance, suppose we have an image with a dog, a bicycle, and people (see Fig. 2). If we directly apply Algorithm 2, it is possible that each of these classes would individually fail to be certified. However, this method does not account for the fact that different classes may be vulnerable at different locations; for example, the "dog" and "bicycle" classes might be at risk in the bottom left corner of the image, while the "people" class is at risk near the top. Because the patch cannot exist in two places simultaneously, at least one class must be robust and the actual certified recall will be $1/3$.

3.4.1. Tracking vulnerable patch locations

We now give a formal treatment of our core idea. Suppose we have a single-label CDPA $SL\text{-}DEF$. For many existing single-label defenses, it is possible to relate the certification procedure $SL\text{-}CERT$ to the complete list of patch locations $\mathcal{R}$ from Eq. (2) [6, 17, 24, 29, 31–33, 35]. In these cases, we extend Definition 2 and allow $SL\text{-}CERT$ to return a *vulnerability status array*, which we denote by $\boldsymbol{\lambda} \in \{0,1\}^{|\mathcal{R}|}$. A value of 1 implies the image $\mathrm{x} \in \mathcal{X}$ is protected from attacks located in $\mathrm{r} \in \mathcal{R}$, while 0 means it is not.

This provides a convenient formulation with which to express our improved method. Consider a multi-label classifier $\mathbb{F}: \mathcal{X} \to \mathcal{Y}$. We first obtain vulnerability status arrays $\boldsymbol{\lambda}$ for each class in Algorithm 2 that could not be certified; this is done by isolating the associated binary classifiers. We then note that given k classes of a common failure mode (i.e., FN or FP), the sum of the inverted arrays $\mathbf{1} - \boldsymbol{\lambda}$ will represent the frequency of the failure type at each patch location. The key insight is that the maximum value, v_{opt}, from the combined array will represent the patch location $\mathrm{r_{opt}} \in \mathcal{R}$ of the image most vulnerable to a patch attack; an attacker must place an adversarial patch at this location to maximize malicious effects. Note however that it is possible $v_{opt} < k$. Then, as per the construction of each $\boldsymbol{\lambda}$ these $k - v_{opt} > 0$ classes will be *guaranteed robustness under the optimal patch location.*

3.4.2. Proposing our novel algorithm

Algorithm 3 *Location-aware certification for FN*

Input: Image $\mathrm{x} \in \mathcal{X}$, ground-truth $\mathrm{y} \in \mathcal{Y}$, multi-label classifier $\mathbb{F}: \mathcal{X} \to \mathcal{Y}$, certification procedure $SL\text{-}CERT$ and security parameters σ from a single-label CDPA, patch locations $\mathcal{R}$
Output: Certified number of true positives TP_{new}, false negatives upper bound FN_{new}

1: **procedure** LocCert(x,y,$\mathbb{F}$,$SL\text{-}CERT$,σ,$\mathcal{R}$)
2: ▷ Pass all args to DemuxCert(...)
3: TP,FP,FN,$\boldsymbol{\kappa} \leftarrow$ DemuxCert(...)
4: ▷ Initialize array with list of FN indices
5: $c \leftarrow \mathrm{len}(\mathrm{y})$
6: $fnIdx \leftarrow \mathrm{list}(\{1 \le i \le c : \boldsymbol{\kappa}[i] = 0 \cap \mathrm{y}[i] = 1\})$
7: $fnCertFails \leftarrow [0]^{FN \times |\mathcal{R}|}$
8: **for** $k \leftarrow 1$ to FN **do** ▷ Isolate each FN classifier
9: $\mathbb{F}_s \leftarrow \mathbb{F}[fnIdx[k]]$
10: $\boldsymbol{\lambda} \leftarrow$ SL-CERT$_{[\mathbb{F}_s,\sigma]}$(x,y$[fnIdx[k]]$,$\mathcal{R}$)
11: $fnCertFails[k] = \mathbf{1} - \boldsymbol{\lambda}$
12: **end for**
13: $fnTotal \leftarrow \mathrm{sum}(fnCertFails, \mathrm{dim} = 0)$
14: $FN_{new} = \max(fnTotal)$ ▷ Pick worst location
15: $TP_{new} = TP + (FN - FN_{new})$
16: **return** TP_{new}, FN_{new}
17: **end procedure**

These insights are encapsulated by Algorithm 3, the location-aware certification method for false negatives.[4] It works by first computing robustness bounds for data point (x,y) via Algorithm 2. On line 5 we determine the false negative classes that failed certification in Algorithm 2. During the *for* loop on lines 8–12, we extract the vulnerability status array $\boldsymbol{\lambda}$ for each false negative by isolating the associated binary classification task. Finally, we sum the inverted arrays $\mathbf{1} - \boldsymbol{\lambda}$ on line 13 and pick the patch location with the largest value; this is the max number of false negatives an attacker can induce at test time. We then alter the lower bound for true positives on line 16.

We now demonstrate that Algorithm 3 provides superior bounds to Algorithm 2.

Theorem 2 (Algorithm 3 Correctness). *Suppose we have an image data point $(\mathrm{x},\mathrm{y}) \in \mathcal{X} \times \mathcal{Y}$, a single-label CDPA $SL\text{-}DEF$, and a multi-label classification model $\mathbb{F}: \mathcal{X} \to \mathcal{Y}$. If $SL\text{-}CERT$ returns the vulnerability status array $\boldsymbol{\lambda}$ associated with each $\mathrm{r} \in \mathcal{R}$, then under the patch threat model $S_{\mathrm{x},\mathcal{R}}$ the bounds from Algorithm 3 are correct and stronger than Algorithm 2.*

Proof. See *Supplementary Material*, Appendix A. □

An analogue to Theorem 2 also exists for FP bounds, and can be proved using a modified version of Algorithm 3 that tracks FP indices.

[4] Obtaining FP_{new} is similar, with line 5 changed to track FP indices

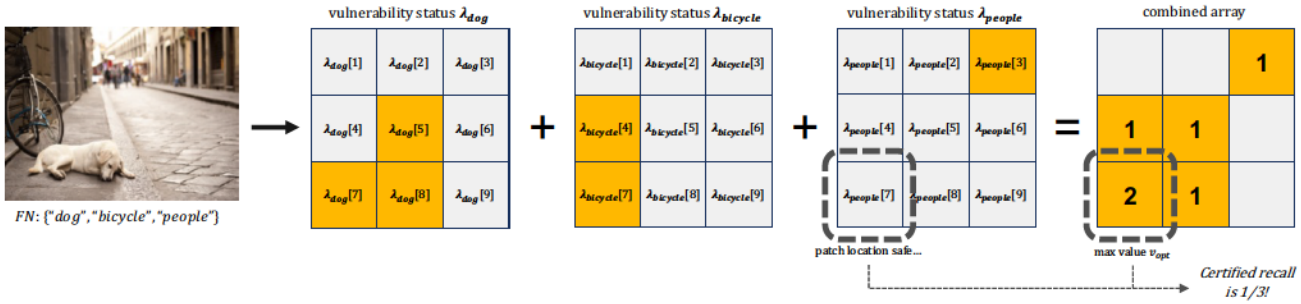


Figure 2. *A diagram which illustrates the key intuition for the location-aware approach. In the sample image we assume all three objects (i.e., "dog", "bicycle", "people") are false negatives. Thus, for each FN we extract the vulnerability status over all patch locations (orange means vulnerable) and accumulate them to find the most vulnerable patch location; this happens to be in the bottom left corner of the image. However, the "people" class by itself is not vulnerable to this location; thus, we can claim stronger robustness bounds than initially suggested by Algorithm 2.*

4. Main Results

4.1. Setup

In this section, we discuss our evaluation setup. The associated source code is available at https://github.com/inspire-group/PatchDEMUX.

Backbone initialization and parameters. Recall from Sec. 3.1 that PatchDEMUX requires an underlying single-label CDPA to operate. For our experiments we choose PatchCleanser, as it is the current SOTA single-label CDPA and is architecture-agnostic (i.e., it is compatible with any off-the-shelf multi-label classifier) [33]. PatchCleanser works by using a novel double-masking algorithm along with a specially generated certification mask set to provably remove adversarial patches [33]. The mask generation process has two security parameters. The first is the number of masks for each image dimension $k_1 \times k_2$; using more masks leads to longer inference time but results in stronger robustness, effectively serving as a "computational budget" [33]. The second is the estimated size of the patch p in pixels. Our experiments with PatchDEMUX use 6×6 masks and assume the patch is $\sim$2% of the overall image size, which are the default settings in Xiang et al. [33]; we vary these parameters in Sec. 5. For more details on how PatchCleanser fits into the PatchDEMUX framework see *Supplementary Material*, Appendix B.

We note that PatchCleanser can also provide protection against multiple patches [33]. Because our baseline certification method provably extends single-label guarantees to multi-label setting, it will also feature resistance against multiple patches. In our experiments, we focus on the single patch setting for simplicity.

Dataset and model architectures. We evaluate our defense on two datasets: MS-COCO [19] and PASCAL VOC [11]. The former is a challenging collection of images that feature "common objects in context" [19], while the latter focuses on "realistic scenes" [11]. For our experiments we test on the MS-COCO 2014 validation split, which contains $\sim$41,000 images and 80 classes, and the PASCAL VOC 2007 test split, which has $\sim$5,000 images and 20 classes. Both of these splits are commonly used in the multi-label classification community [3, 20, 27, 36].

For the multi-label classifier architecture, we evaluate two options. The first is a ResNet-based architecture from Ben-Baruch et al. [3] that uses convolution kernels and has an input size of 448×448. The second is a vision transformer-based (ViT) architecture from Liu et al. [20] that uses the self-attention mechanism and has an input size of 384×384 [9, 20, 36]. These models are chosen as they perform well on the multi-label classification task and have publicly available checkpoints. We resize images to fit on each model and apply different defense fine-tuning methods (i.e., Random Cutout [8], Greedy Cutout [28]) to achieve stronger robustness guarantees.

Evaluation settings and metrics. Our results feature several evaluation settings.

1. *Undefended clean:* This setting represents evaluation on clean data without the PatchDEMUX defense.
2. *Defended clean:* This setting refers to evaluation on clean data with the PatchDEMUX defense activated.
3. *Certified robust:* This setting represents lower bounds on performance determined using Algorithm 2.
4. *Location-aware robust:* This setting represents the tighter certification bounds from Algorithm 3. We report performance corresponding to the worst-case attacker (see *Supplementary Material*, Appendix E).

The first two are *clean settings*, where precision and recall metrics are empirically computed for each datapoint. The latter two are *certified robust settings*, where certified precision and certified recall metrics are computed using Algorithm 2 and Algorithm 3. In all four evaluation settings we micro-average metrics over the entire dataset [38]. In addition, we sweep model outputs across a range of threshold values to create *precision-recall plots*. The associated area-under-curve values aggregate performance and are used to approximate *average precision* (AP); more details are in *Supplementary Material*, Appendix C.

4.2. PatchDEMUX overall performance

In this section we report our main results for PatchDEMUX on the MS-COCO 2014 validation dataset. We summarize the precision values associated with key recall levels in Tab. 1. Fig. 3

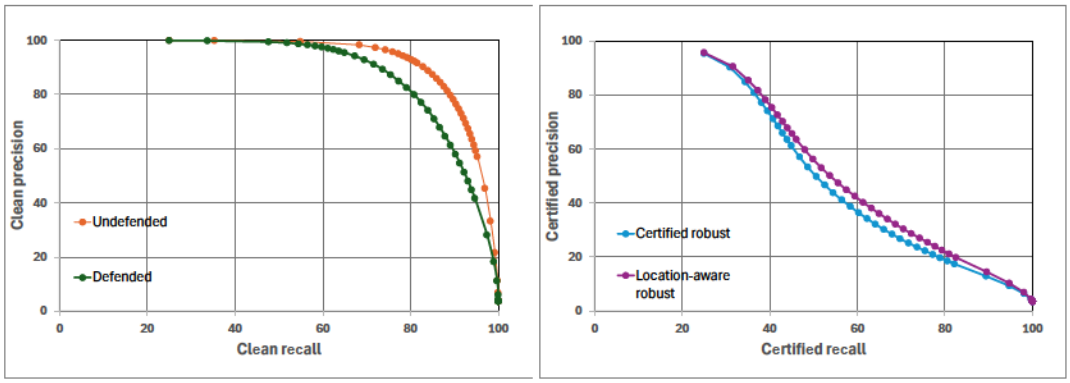
Table 1. *PatchDEMUX performance with ViT architecture on the MS-COCO 2014 validation dataset. Precision values are evaluated at key recall levels along with the approximated average precision. We assume the patch attack is at most 2% of the image area and use a computational budget of 6×6 masks*

(a) *Clean setting precision values*

Architecture	ViT			
Clean recall	25%	50%	75%	*AP*
Undefended	99.930	99.704	96.141	91.146
Defended	99.894	99.223	87.764	85.276

(b) *Certified robust setting precision values*

Architecture	ViT			
Certified recall	25%	50%	75%	*AP*
Certified robust	95.369	50.950	22.662	41.763
Location-aware	95.670	56.038	26.375	44.902



(a) *Clean setting precision-recall curves* (b) *Certified robust setting precision-recall curves*

Figure 3. *PatchDEMUX precision-recall curves with ViT architecture over the MS-COCO 2014 validation dataset. We consider the clean and certified robust evaluation settings. We assume the patch attack is at most 2% of the image area and use a computational budget of 6×6 masks.*

features precision-recall plots, while AP values are present in Tab. 1. Because the ViT architecture outperforms the Resnet architecture (see *Supplementary Material*, Appendix D) we focus on the ViT model here. Performance of the ViT architecture on the PASCAL VOC 2007 test dataset is in *Supplementary Material*, Appendix H.

High clean performance. As shown in Tab. 1a and Fig. 3a, the PatchDEMUX inference procedure features excellent performance on clean data. Specifically, the defended clean setting achieves ∼94% of the undefended model's AP. These results demonstrate that PatchDEMUX can be deployed at test time with minimal loss in performance utility.

Non-trivial robustness. Tab. 1b and Fig. 3b also show that PatchDEMUX attains non-trivial certifiable robustness on the MS-COCO 2014 validation dataset. For instance, when fixed at 50% certified recall PatchDEMUX achieves 56.038% certified precision. This performance remains stable across a variety of thresholds, as evidenced by the 44.902% certified AP value. Location-aware certification is a key factor in these results, improving certified AP by almost 3 points compared to the certified robust setting. Improvements are most notable in the *mid recall-mid precision* region of the certified robust precision-recall plot (Fig. 3b).

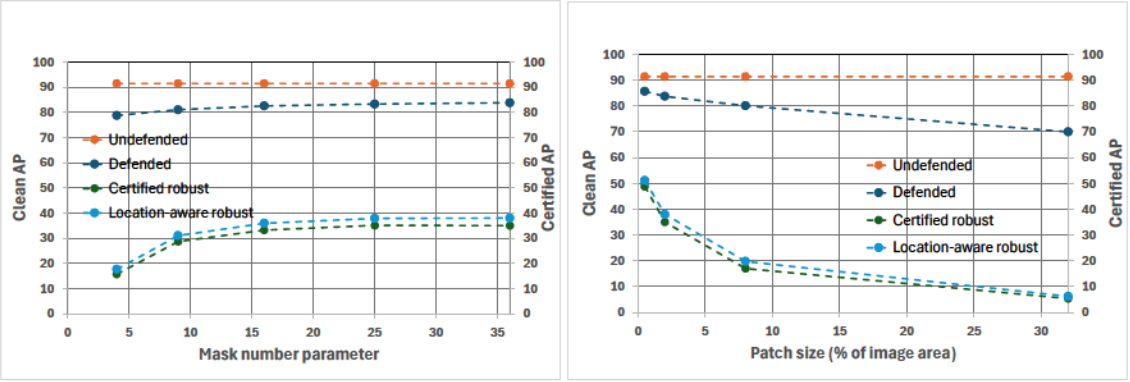
Interestingly, the defended clean precision-recall plot (Fig. 3a) is concave in shape while the certified robust plots (Fig. 3b) are slightly convex. This performance gap is likely due to the sensitivity of PatchCleanser's certification procedure to object occlusion from the generated mask set. This limitation is compounded by the fact that many MS-COCO images contain objects that are small relative to the overall image size [19, 33].

4.3. Ablation studies

We also perform a series of ablation studies for PatchDEMUX using the MS-COCO 2014 validation dataset. We first empirically compare different attackers in the location-aware robust setting and find that attacks targeting false positives are relatively "weaker" (see *Supplementary Material*, Appendix E). We then investigate the impact of different defense fine-tuning routines, and find that variants of cutout fine-tuning (i.e., Random Cutout [8], Greedy Cutout [28]) can boost model robustness (see *Supplementary Material*, Appendix F); the strongest results for the defended clean setting are featured in the previous section.

5. Security Parameter Experiments

As discussed in Sec. 4.1, the PatchCleanser backbone has two security parameters: the number of masks desired in each dimension $k_1 \times k_2$ (i.e., the "computational budget") and the estimated size of the patch p in pixels [33]. In this section, we study the impact of these parameters on PatchDEMUX performance. To isolate the effects of security parameter variation, we use ViT checkpoints without defense fine-tuning. Experiments are done on the MS-COCO 2014 validation dataset.



(a) *Clean AP and certified AP as a function of mask number.* (b) *Clean AP and certified AP as a function of patch size.*

Figure 4. *The impact of varying PatchCleanser security parameters on PatchDEMUX performance. Experiments performed on MS-COCO 2014 validation dataset. We compute clean AP for the clean setting evaluations, and certified AP for the certified robust setting evaluations.*

5.1. Impact of varying mask number

We present results when varying the mask number parameter in Fig. 4a (the associated table is in *Supplementary Material*, Appendix I). We assume the number of masks in each dimension is the same (i.e., $k := k_1 = k_2$) and evaluate with respect to k^2. We keep the patch size parameter its default value of $\sim$2%.

Limited tradeoff between computational budget and robustness. We find that PatchDEMUX provides consistent defended clean and certified robust performance even after greatly reducing the number of masks. For instance, decreasing the number of masks from 36 to 16 results in a maximum AP drop of 2 points across all evaluation settings. At the extreme of $k^2 = 4$ masks more substantial performance drops are noticeable. This is expected, as the mask generation method from PatchCleanser will create larger masks to compensate for reduced mask number; this leads to increased occlusion and fewer certification successes [33].

5.2. Impact of varying patch size

We present results when varying the patch size estimate in Fig. 4b (the associated table is in *Supplementary Material*, Appendix I). We keep the mask number at its default value of 6×6 masks.

Strong clean performance over different patch sizes. We find that the defended clean performance of PatchDEMUX is resilient to increasing patch size; indeed, clean AP only drops from 85.731 in the smallest patch setting to 69.952 in the largest. Thus, even in unlikely scenarios (i.e., a patch size of $\geq 32\%$ would be easily detectable by hand) PatchDEMUX maintains strong inference performance. For the certified robust settings, PatchDEMUX provides relatively strong robustness guarantees on smaller patches (i.e., $\leq 2\%$) and performance degrades for larger patches (i.e., $\geq 8\%$); certified AP drops close to 0% when a patch size of 32% is considered. These trends align with experiments done by Xiang et al. [33] in the single-label classification domain; the general intuition is that larger patch sizes require PatchCleanser to generate larger masks, making certification failures more likely [33].

5.3. Overall takeaways

Overall, we find that PatchDEMUX performance tradeoffs corroborate with findings from Xiang et al. [33]. This illustrates a key feature of our defense framework: PatchDEMUX successfully adapts the strengths of underlying single-label CDPAs to the multi-label classification setting.

6. Related Work

Certifiable defenses against patch attacks. CDPAs have been designed for various computer vision applications. In single-label classification, defense strategies include bound propagation methods [6], small receptive field methods [17, 24, 31, 32], and masking methods [33]. CDPAs have also been proposed for object detection [34] and semantic segmentation [37], although notions of certifiable robustness are more difficult to define in these domains.

Certifiable defenses in multi-label classification. Jia et al. [15] proposed MultiGuard, a certifiably robust defense for multi-label classifiers that generalizes randomized smoothing [7]. However, MultiGuard is designed to protect against ℓ_2-norm attacks and does not address adversarial patches.

7. Conclusion

The threat of adversarial patch attacks has compromised real-world computer vision systems, including those that depend on multi-label classifiers. To this end we introduced PatchDEMUX, a certifiably robust framework for multi-label classifiers against adversarial patches. PatchDEMUX can extend any existing single-label CDPA, including the current SOTA single-label CDPA PatchCleanser, and demonstrates strong performance on the MS-COCO and PASCAL VOC datasets. We hope that future work will take advantage of our modular framework to significantly mitigate the impact of adversarial patches.

8. Acknowledgements

We would like to thank the anonymous CVPR reviewers for their helpful feedback. This work was supported by National Science Foundation grants IIS-2229876 (the ACTION center) and CNS-2154873. Prateek Mittal acknowledges the support of NSF grant CNS-2131938, Princeton SEAS Innovation award, and OpenAI & FarAI superalignment grants.

References

[1] Abhishek Aich, Calvin-Khang Ta, Akash Gupta, Chengyu Song, Srikanth V. Krishnamurthy, M. Salman Asif, and Amit K. Roy-Chowdhury. GAMA: Generative Adversarial Multi-Object Scene Attacks. In *NeurIPS 2022*. arXiv, 2022. 5

[2] Abhishek Aich, Shasha Li, Chengyu Song, M. Salman Asif, Srikanth V. Krishnamurthy, and Amit K. Roy-Chowdhury. Leveraging Local Patch Differences in Multi-Object Scenes for Generative Adversarial Attacks. In *WACV 2023*, pages 1308–1318, Waikoloa, HI, USA, 2023. IEEE. 5

[3] Emanuel Ben-Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric Loss For Multi-Label Classification. In *ICCV 2021*. arXiv, 2021. arXiv:2009.14119 [cs]. 2, 6, 4, 8

[4] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial Patch. In *NeurIPS 2017 Workshops*. arXiv, 2018. arXiv:1712.09665 [cs]. 1, 2

[5] Nicholas Carlini and David Wagner. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. In *CCS 2017 Workshop on Artificial Intelligence and Security (AISec 2017)*, pages 3–14, Dallas Texas USA, 2017. ACM. 1

[6] Ping-Yeh Chiang, Renkun Ni, Ahmed Abdelkader, Chen Zhu, Christoph Studer, and Tom Goldstein. Certified Defenses for Adversarial Patches. In *ICLR 2020*. arXiv, 2020. arXiv:2003.06693 [cs, stat]. 1, 2, 3, 5, 8

[7] Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified Adversarial Robustness via Randomized Smoothing. In *ICML 2019*. arXiv, 2019. arXiv:1902.02918 [cs, stat]. 8

[8] Terrance DeVries and Graham W. Taylor. Improved Regularization of Convolutional Neural Networks with Cutout, 2017. arXiv:1708.04552 [cs]. 6, 7

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR 2021*. arXiv, 2021. arXiv:2010.11929 [cs]. 6, 5

[10] Mohamed Elhoseiny, Amr Bakry, and Ahmed Elgammal. MultiClass Object Classification in Video Surveillance Systems - Experimental Study. In *CVPR 2013 Workshops (CVPRW 2013)*, pages 788–793, OR, USA, 2013. IEEE. 1

[11] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 6

[12] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust Physical-World Attacks on Deep Learning Visual Classification. In *CVPR 2018*, pages 1625–1634, Salt Lake City, UT, USA, 2018. IEEE. 1, 2

[13] Marian George and Christian Floerkemeier. Recognizing Products: A Per-exemplar Multi-label Image Classification Approach. In *ECCV 2014*, pages 440–455. Springer International Publishing, 2014. 1

[14] Jamie Hayes. On Visible Adversarial Perturbations & Digital Watermarking. In *CVPR 2018 Workshops (CVPRW 2018)*, pages 1678–16787, Salt Lake City, UT, USA, 2018. IEEE. 1

[15] Jinyuan Jia, Wenjie Qu, and Neil Zhenqiang Gong. MultiGuard: Provably Robust Multi-label Classification against Adversarial Examples. In *NeurIPS 2022*. arXiv, 2022. arXiv:2210.01111 [cs]. 8

[16] Chi-Hsi Kung, Shu-Wei Lu, Yi-Hsuan Tsai, and YiTing Chen. Action-Slot: Visual Action-Centric Representations for Multi-Label Atomic Activity Recognition in Traffic Scenes. In *CVPR 2024*, pages 18451–18461, Seattle, WA, USA, 2024. IEEE. 1

[17] Alexander Levine and Soheil Feizi. (De)Randomized Smoothing for Certifiable Defense against Patch Attacks. In *NeurIPS 2020*. arXiv, 2021. arXiv:2002.10733 [cs, stat]. 1, 2, 3, 5, 8

[18] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):6999–7019, 2022. 1

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. In *ECCV 2014*. arXiv, 2015. arXiv:1405.0312 [cs]. 6, 7

[20] Shilong Liu, Lei Zhang, Xiao Yang, Hang Su, and Jun Zhu. Query2Label: A Simple Transformer Way to Multi-Label Classification, 2021. arXiv:2107.10834 [cs]. 6

[21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR 2018*. arXiv, 2019. arXiv:1706.06083 [cs, stat]. 2

[22] Hassan Mahmood and Ehsan Elhamifar. Semantic-Aware Multi-Label Adversarial Attacks. In *CVPR 2024*, pages 24251–24262, Seattle, WA, USA, 2024. IEEE. 5

[23] Stefano Melacci, Gabriele Ciravegna, Angelo Sotgiu, Ambra Demontis, Battista Biggio, Marco Gori, and Fabio Roli. Domain Knowledge Alleviates Adversarial Attacks in Multi-Label Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9944–9959, 2022. 5

[24] Jan Hendrik Metzen and Maksym Yatsura. Efficient Certified Defenses Against Patch Attacks on Image Classifiers. In *ICLR 2021*. arXiv, 2021. arXiv:2102.04154 [cs, stat]. 1, 2, 3, 5, 8

[25] Muzammal Naseer, Salman H. Khan, and Fatih Porikli. Local Gradients Smoothing: Defense against localized adversarial attacks. In *WACV 2019*. arXiv, 2018. arXiv:1807.01216 [cs]. 1

[26] Federico Nesti, Giulio Rossolini, Saasha Nair, Alessandro Biondi, and Giorgio Buttazzo. Evaluating the Robustness of Semantic Segmentation for Autonomous Driving against Real-World Adversarial Patch Attacks. In *WACV 2022*. arXiv, 2021. arXiv:2108.06179 [cs]. 1, 2

[27] Tal Ridnik, Gilad Sharir, Avi Ben-Cohen, Emanuel Ben-Baruch, and Asaf Noy. ML-Decoder: Scalable and Versatile Classification Head. In *WACV 2023*, pages 32–41, Waikoloa, HI, USA, 2023. IEEE. 6

[28] Aniruddha Saha, Shuhua Yu, Mohammad Sadegh Norouzzadeh, Wan-Yi Lin, and Chaithanya Kumar Mummadi. Revisiting Image Classifier Training for Improved Certified Robust Defense

against Adversarial Patches. *Transactions on Machine Learning Research*, 2023. 6, 7

[29] Hadi Salman, Saachi Jain, Eric Wong, and Aleksander Madry. Certified Patch Robustness via Smoothed Vision Transformers. In *CVPR 2022*, pages 15116–15126, New Orleans, LA, USA, 2022. IEEE. 1, 2, 3, 5

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014. arXiv:1312.6199 [cs]. 1

[31] Chong Xiang and Prateek Mittal. PatchGuard++: Efficient Provable Attack Detection against Adversarial Patches. In *ICLR 2021 Workshop on Security and Safety in Machine Learning Systems*. arXiv, 2021. arXiv:2104.12609 [cs]. 1, 2, 3, 5, 8

[32] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking. In *USENIX Security 2021*. arXiv, 2021. arXiv:2005.10884 [cs, stat]. 3, 8

[33] Chong Xiang, Saeed Mahloujifar, and Prateek Mittal. PatchCleanser: Certifiably Robust Defense against Adversarial Patches for Any Image Classifier. In *USENIX Security 2022*. arXiv, 2022. arXiv:2108.09135 [cs]. 2, 3, 5, 6, 7, 8

[34] Chong Xiang, Alexander Valtchanov, Saeed Mahloujifar, and Prateek Mittal. ObjectSeeker: Certifiably Robust Object Detection against Patch Hiding Attacks via Patch-agnostic Masking. In *IEEE Symposium on Security and Privacy 2023*. arXiv, 2022. arXiv:2202.01811 [cs]. 8

[35] Chong Xiang, Tong Wu, Sihui Dai, Jonathan Petit, Suman Jana, and Prateek Mittal. PatchCURE: Improving Certifiable Robustness, Model Utility, and Computation Efficiency of Adversarial Patch Defenses. In *USENIX Security 2024*. arXiv, 2024. arXiv:2310.13076 [cs]. 1, 2, 3, 5

[36] Shichao Xu, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Zhu Qi. Open Vocabulary Multi-Label Classification with Dual-Modal Decoder on Aligned Visual-Textual Features, 2023. arXiv:2208.09562 [cs]. 6

[37] Maksym Yatsura, Kaspar Sakmann, N. Grace Hua, Matthias Hein, and Jan Hendrik Metzen. Certified Defences Against Adversarial Patch Attacks on Semantic Segmentation. In *ICLR 2023*. arXiv, 2023. arXiv:2209.05980 [cs]. 1, 2, 8

[38] Min-Ling Zhang and Zhi-Hua Zhou. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014. 2, 5, 6