Using Depth for Pixel-Wise Detection of Adversarial Attacks in Crowd Counting

Weizhe Liu  Mathieu Salzmann  Pascal Fua

Computer Vision Laboratory, École Polytechnique Fédérale de Lausanne (EPFL)
{weizhe.liu, mathieu.salzmann, pascal.fua}@epfl.ch

Abstract. State-of-the-art methods for counting people in crowded scenes rely on deep networks to estimate crowd density. While effective, deep learning approaches are vulnerable to adversarial attacks, which, in a crowd-counting context, can lead to serious security issues. However, attack and defense mechanisms have been virtually unexplored in regression tasks, let alone for crowd density estimation.

In this paper, we investigate the effectiveness of existing attack strategies on crowd-counting networks, and introduce a simple yet effective pixel-wise detection mechanism. It builds on the intuition that, when attacking a multitask network, in our case estimating crowd density and scene depth, both outputs will be perturbed, and thus the second one can be used for detection purposes. We will demonstrate that this significantly outperforms heuristic and uncertainty-based strategies.

Keywords: Crowd Counting, Pixel-Wise Adversarial Detection, Security

1 Introduction

State-of-the-art crowd counting algorithms [69, 71, 48, 62, 53, 49, 26, 47, 52, 33, 19, 46, 4] rely on Deep Networks to regress a crowd density, which is then integrated to estimate the number of people in the image. Their application can have important societal consequences, for example when they are used to assess how many people attended a demonstration or a political event.

In the “Fake News” era, it is therefore to be feared that hackers might launch adversarial attacks to bias the output of these models for political gain. While such attacks have been well studied for classification networks [15, 22, 40, 41, 42, 6], they remain largely unexplored territory for people counting and even for regression at large. The only related approach we know of [45] is very recent and specific to attacking optical flow networks, leaving the pixel-wise detection of attacks untouched.

In this paper, our goal is to blaze a trail in that direction. Our main insight is that if a two-stream network is trained to regress both the people density and the scene depth, it becomes very difficult to affect one without affecting the other. In other words, pixels that have been modified to alter the density estimate will also produce incorrect depths, which can be detected by estimating
Fig. 1: Density and Depth (DaD) model. A two-stream network is trained to regress both the people density and the scene depth. The pixels that have been attacked to alter the density estimate will also produce incorrect depths and can thus be detected.

depth using unrelated means. As we will show, this can be done using a depth sensor, simple knowledge about the scene geometry, or even an unrelated deep model pre-trained to estimate depth from images. In other words, our approach works best for a static camera for which the depth of the scene it surveys can be accurately computed but remains applicable to mobile cameras that make such computation more difficult. This assumes that such reference depth maps can be kept safe from the attacker. When this cannot be done, we will show that using statistics from depth maps acquired earlier suffices to detect tampering at a later date.

Fig. 1 depicts our approach, which we will refer to as the Density and Depth (DaD) model. We will show that it is robust to both unexposed attacks, in which the adversary does not know the existence of our adversarial detector, and exposed attacks, in which the adversary can access not only the density regression network but also our adversarial detector. In other words, even when our approach is exposed, a hacker cannot mount an effective attack while avoiding detection. This is largely because, depth measurements in a crowded scene, are affected by the appearance and disappearance of people but the perturbations remain small whereas those of RGB pixels can be much larger due to illumination and appearance changes. Therefore, even if the attacker has access to the depth maps, it remains difficult to guarantee that they will be altered in a consistent and undetectable way.
Our contribution therefore is an effective approach to foiling adversarial attacks on people counting algorithms that rely on deep learning. Its principle is very generic and could be naturally extended to other image regression tasks. Our experiments on several benchmark datasets demonstrate that it outperforms heuristic-based and uncertainty-based attack detection strategies.

2 Related Work

Crowd Counting. Early crowd counting methods [60,59,28] tended to rely on counting-by-detection, that is, explicitly detecting individual heads or bodies and then counting them. Unfortunately, in very crowded scenes, occlusions make detection difficult and these approaches have been largely displaced by counting-by-density-estimation ones [69,62,33,53,64,43,49,71,48,4,58,34,38,50,36,35]. They rely on training a regressor to estimate people density in various parts of the image and then integrating. This trend began in [52,12], using either Gaussian Process or Random Forests regressors. Even though approaches relying on low-level features [9,18] can yield good results, they have now mostly been superseded by CNN-based methods [20,72,70,57,27,30,65,39,32,63,64,51,54,10,56,67,68], a survey of which can be found in [53]. In this paper, we therefore focus on attacks against these.

Defense against Adversarial Attacks. Deep networks trained to solve classification problems are vulnerable to adversarial attacks [15,22,40,41,42,6]. Existing attack strategies can be roughly categorized as optimization-based or gradient-based. The former [41,42,6] involve terms related to the class probabilities, which makes the latter [15,22,40] better candidates for attacks against deep regression models. The very recent work of [45] is the only one we know of that examines adversarial attacks against a regressor, specifically one that estimates optical flow. However, it does not propose defense mechanisms, which are the focus of this paper.

In the context of classification, one popular defense is adversarial training [55], which augments the training data with adversarial examples and has been shown to outperform many competing methods [10,14,17,25,11]. However, it needs access to adversarial examples, which are often not available ahead of time and must be generated during training. As a consequence, several alternative approaches have been proposed. This includes training auxiliary classifiers, ranging from simple linear regressors to complex neural networks, to predict whether a sample is adversarial or not. However, as shown in [5], such detection mechanisms can easily be defeated by an adversary targeting them directly.

In any event, none of these methods are designed to detect attacks at the pixel-level. Even the few researchers who have studied adversarial attacks for semantic segmentation [61,29], which is a pixel-level prediction task, do not go beyond detection at the global image level.

A seemingly natural approach to performing pixel-level attack detection would be to rely on prediction uncertainty. In [11], the authors argue that Bayesian uncertainty [13] is a reliable way to detect the adversarial examples
Fig. 2: Density and Depth model. An input RGB image is first encoded to deep features by an encoder network. Then, these features are decoded to a crowd density map and a depth map by two different decoder networks. At inference time, we detect adversarial attacks in a pixel-wise manner by observing the depth estimation errors.

because the perturbed pixels generally have much higher uncertainty values. Uncertainty can be computed using dropout, as in [13], learned from data [21], or estimated using the negative log-likelihood of each prediction [23]. In our experiments, we will extend this strategy to pixel-wise adversarial attack detection and show that our approach significantly outperforms it.

3 Density and Depth Model

As discussed in Section 2, most state-of-the-art crowd counting algorithms rely on a deep network regressor \( F_d \), that takes an image \( I \) as input and returns \( D_{I^{est}} = F_d(I, \Theta) \), an estimated density map, which should be as close as possible to a ground-truth one \( D_{I^{gt}} \) in \( L^2 \) norm terms. Here, \( \Theta \) stands for the network’s weights, that have been optimized for this purpose.

An adversarial attack then involves generating a perturbation \( P = F_p(I, D_{I^{gt}}) \), where \( F_p \) maps the input image \( I \) and the density \( D_{I^{gt}} \) associated with the clean image to \( P \) in such a way that \( A = I + P \) is visually indistinguishable from \( I \) while yielding a crowd density estimate \( F_d(A, \Theta) \) that is as different as possible from the prediction obtained by the clean image \( I \).

We will review the best known ways to generate such attacks in Section 4.1. Here, our concern is to define \( F_d \) so as to defeat them by ensuring that they are easily detected. To this end, we leverage an auxiliary task, depth estimation, as discussed below.

Network Architecture. Instead of training a single regressor that predicts only people density, we train a two-stream network where one stream predicts people density and the other depth at each pixel. We write

\[
D_{I^{est}} = F_d(I, \Theta), \\
Z_{I^{est}} = F_z(I, \Theta),
\]

where \( D_{I^{est}} \) and \( Z_{I^{est}} \) are the estimated densities and depths while \( F_d \) and \( F_z \) are two regressors parameterized by the weights \( \Theta \). The network that implements
\( \mathcal{F}_d \) and \( \mathcal{F}_z \) comprises a single encoder and two decoders, one that produces the densities and the other the depths. It is depicted by Fig. 2, and we provide details of its architecture in the supplementary material. Note that some of the weights in \( \Theta \) are specific to the first decoder and others to the second.

As the two decoders use the same set of features as input, it is difficult to tamper with the results of one without affecting that of the other, as we will see in Section 4.4. More specifically, if a pixel is perturbed to change the local density estimate, the local depth estimate is likely to be affected as well. We therefore take the relative error in depth estimation with respect to a reference depth map \( Z^{ref} \)

\[
\frac{|Z^{est}(j) - Z^{ref}(j)|}{Z^{ref}(j)}
\]

(2)
to be an indicator of a potential disturbance. As will be shown in our experiments, the reference depth map can be either the ground-truth one, a rough estimate obtained using knowledge of the scene geometry, or the output of another monocular depth estimation network. In practice, we label a pixel as potentially tampered with if this difference is larger than 5% of the largest difference in the training dataset and we will justify this choice in Section 4.7. In test sequences for which the reference depth map can also be tampered with by the attacker, we can use the statistics of the training depth maps to also detect such tampering, as will be shown in Section 4.6.

Network Training. Given a set of \( N \) training images \( \{I_i\}_{1 \leq i \leq N} \) with corresponding ground-truth density maps \( \{D^{gt}_i\}_{1 \leq i \leq N} \) and ground-truth depth maps \( \{Z^{gt}_i\}_{1 \leq i \leq N} \), we learn the weights \( \Theta \) of the two regressors by minimizing the loss

\[
L(\Theta) = L_d + \lambda \cdot L_z ,
\]

(3)

\[
L_d = \frac{1}{2B} \sum_{i=1}^{B} \|D^{gt}_i - D^{est}_i\|^2 ,
\]

\[
L_z = \frac{1}{2B} \sum_{i=1}^{B} \|Z^{gt}_i - Z^{est}_i\|^2 .
\]

where \( B \) is the batch size and \( \lambda \) is a hyper-parameter that balances the contributions of the two losses. We found empirically that \( \lambda = 0.01 \) yields the best overall performance, as will be shown in Section 4.4.

To obtain the ground-truth density maps \( D^{gt}_i \), we rely on the same strategy as previous work \cite{26,48,71,47,31}. In each training image \( I_i \), we annotate a set of \( c_i \) 2D points \( O^{gt}_i = \{O^{gt}_j\}_{1 \leq j \leq c_i} \) that denote the position of each human head in the scene. The corresponding ground-truth density map \( D^{gt}_i \) is obtained by convolving an image containing ones at these locations and zeroes elsewhere with a Gaussian kernel of mean \( \mu \) and variance \( \sigma \).

4 Experiments

In this section, we first introduce the existing adversarial attack methods that can be used against a deep regressor and describe the evaluation metric and
the benchmark datasets we used to assess their performance. We then use them to attack our approach to demonstrate its robustness, and conclude with an ablation study that demonstrates that our approach is robust to the hyper-parameter setting and works well when used in conjunction with several recent crowd density regressors \cite{71,26,36}.

4.1 Attacking a Deep Regressor

While there are many adversarial attackers \cite{15,22,40,41,42,6}, their effectiveness has been proven mostly against classifiers but far more rarely against regressors \cite{45}. As discussed in Section 2, the gradient-based methods \cite{15,22,40} are the most suitable ones to attack regressors, and we focus on the so-called Fast Gradient Sign Methods (FGSMs), which are the most successful and widely used ones. We will distinguish between unexposed attacks in which the attacker does not know that we use depth for verification purposes and exposed attacks in which they do.

**Unexposed Attacks.** If the attacker is unaware that we use the depth map for verification purposes, they will only try to affect the density map. They might then use one of the following variants of FGSM.

1. **Untargeted FGSM (FGSM-U(n))** \cite{15,22}. It generates adversarial examples designed to increase the network loss as much as possible from that of the prediction obtained from the clean image, thereby preventing the network from predicting it. Given an input image $I$, the density predicted from the clean image $D$, and the regressor $F_d$ of Eq. 1 parametrized by $\Theta$, the attack is performed by iterating

$$ I_{adv}^n = I, $$

$$ I_{adv}^{i+1} = \text{clip}(I_{adv}^i - \alpha \cdot \text{sign}(\nabla I L_d(F_d(I_{adv}^i; \Theta), D)), \epsilon). $$

$n$ times. The adversarial example is then taken to be $I_{adv}^n$, and we will refer to this as FGSM-U(n). It is a single-step or multiple-step attack without target and $\text{clip}$ guarantees that the resulting perturbation is bounded by $\epsilon$. For consistency with earlier work \cite{15,22}, when $n = 1$, we reformulate this attack as

$$ I_{adv} = I + \epsilon \cdot \text{sign}(\nabla I L_d(F_d(I; \Theta), D)). $$

Unless otherwise specified we use $\epsilon = 15$, $\alpha = 1$, and $n = 19$, as recommended in earlier work \cite{22}. These numbers are chosen to substantially increase the crowd counting error while keeping the perturbation almost imperceptible to the human eye. We will analyze the sensitivity of our approach to these values in Section 4.7. An example of this attack is shown in Fig. 3. By comparing Fig. 3(d) and (e), we can see that the attack made some people “disappear”.

2. **Targeted FGSM (FGSM-T(n))** \cite{22}. Instead of simply preventing the network from finding the right answer $D$, we can target a specific wrong answer $D_t$. This is achieved using the slightly modified iterative scheme

$$ I_{adv}^0 = I, $$

$$ I_{adv}^{i+1} = \text{clip}(I_{adv}^i - \alpha \cdot \text{sign}(\nabla I L_d(F_d(I_{adv}^i; \Theta), D_t)), \epsilon). $$
Again, we take the adversarial example to be $I_{adv}^n$ and use the same values as before for $\epsilon$ and $\alpha$. We will refer to this as FGSM-T(n). In our experiments, we take the targets to be the original density values plus one, which creates an obvious error while yielding tampered images that are undistinguishable from the original ones.

**Exposed Attacks.** If the attacker knows that we are using the depth maps and has access to both $F_d$ and $F_z$, the two regressors of Eq. 1, their natural reaction will be to try to modify the density maps while leaving the depth maps as unchanged as possible. To this end, we propose the following exposed variations of the untargeted and targeted FGSM attacks described above.

(1) Untargeted Exposed FGSM (FGSM-UE(n)). The iterative scheme becomes

$$I_{adv}^0 = I,$$

$$L_{i+1}^{all} = L_d(F_d(I_{adv}^i; \Theta), D) - \lambda \cdot L_z(F_z(I_{adv}^i; \Theta), z),$$

$$I_{adv}^{i+1} = \text{clip}(I_{adv}^i + \alpha \cdot \text{sign}(%(\nabla I_{adv}^{i} L_{i+1}^{all})), \epsilon).$$

where $z$ is the depth map associated with clean image. When $n = 1$, we reformulate the final line of Eq. 7 as in Eq. 5 for consistency with earlier work. The additional term in the loss function aims to preserve the predicting power of $F_z$ while compromising that of $F_d$ as much as possible. We again use the same values as before for $\epsilon$ and $\alpha$, and $\lambda = 0.01$ is the same balancing factor as in Eq. 4.

(2) Targeted Exposed FGSM (FGSM-TE(n)). Similarly, the targeted attack iterative scheme becomes

$$I_{adv}^0 = I,$$

$$L_{i+1}^{all} = L_d(F_d(I_{adv}^i; \Theta), D_t) + \lambda \cdot L_z(F_z(I_{adv}^i; \Theta), z),$$

$$I_{adv}^{i+1} = \text{clip}(I_{adv}^i - \alpha \cdot \text{sign}(\nabla I_{adv}^{i} L_{i+1}^{all}), \epsilon).$$

When $n = 1$, we again reformulate the final line of Eq. 8 as in Eq. 5.

### 4.2 Evaluation Datasets

We use three different datasets to evaluate our approach. The first two are RGB-D datasets with ground-truth depth values obtained from sensors. Since depth...
sensors may not always be available, we also evaluate our model on a third dataset that contains RGB images with accurate perspective maps, that is, maps inferred from the scene geometry without using depth sensors. Such maps only represent the scene to the exclusion of the people in it. This will let us show that our approach not only works for RGB-D datasets but also achieves remarkable performance in RGB images if scene geometry is available. Furthermore, we will show that our approach also applies when the reference depth is obtained using a separate monocular depth estimation network.

ShanghaiTechRGBD [27]. This is a large-scale RGB-D dataset with 2,193 images and 144,512 annotated heads. The valid depth ranges from 0 to 20 meters due to the limitation in depth sensors. The lighting condition ranges from very bright to very dark in different scenarios. We use the same setting as in [27], with 1,193 images as training set and the remaining ones as test set, and normalize the depth values from $[0,20]$ to $[0,1]$ for both training and evaluation.

MICC [2]. This dataset was acquired by a fixed indoor surveillance camera. It is divided into three video sequences, named FLOW, QUEUE and GROUPS. The crowd motion varies from sequence to sequence. In the FLOW sequence, people walk from point to point. In the QUEUE sequence, people walk in a line. In the GROUPS sequence, people move inside a controlled area. There are 1,260 frames in the FLOW sequence with 3,542 heads. The QUEUE sequence contains 5,031 heads in 918 frames, and the GROUPS sequence encompasses 1,180 frames with 9,057 heads. We follow the same setting as in [27], taking 20% of the images of each scene as training set and using the remaining ones as test set.

Venice [36]. The above two RGB-D datasets contain depth information acquired by sensors. Such information is hard to obtain in outdoor environments, particularly if the scene is far from the camera. Therefore, we also evaluate our approach on the Venice dataset. This dataset contains RGB images and an accurate perspective map of each scene. It was inferred from the grid-like ground pattern that is provided in the supplementary material and without using a depth sensor. The dataset contains 4 different sequences for a total of 167 annotated frames with fixed $1,280 \times 720$ resolution. Our experimental setting follows that of [36,35], with 80 images from a single long sequence as training data, and the images from the remaining 3 sequences for testing purposes.

4.3 Metrics and Baselines

In all our experiments, we partition the images into four parts and tamper with one while leaving the other three untouched. We then measure:

- How well we can detect the pixels that have been tampered with. We measure this in terms of the mean Intersection over Union

$$mIoU = \frac{1}{N} \sum_{i=1}^{N} \frac{|v_i \cap \hat{v}_i|}{|v_i \cup \hat{v}_i|}$$

(9)
where $N$ is the number of images, $\hat{v}_i$ is 1 for the pixels in image $I_i$ predicted to have been tampered with according to Eq. 2 and $v_i$ is the ground-truth perturbation mask.

- How well the modifications of the depth map correlate with those of the predicted density. As in many previous works [71,69,43,48,62,53,36,35], we quantify these modifications in terms of the mean absolute error for densities and depths along with the root mean squared error for density. They are defined as

$$DMAE = \frac{1}{N} \sum_{i=1}^{N} |d_i - \hat{d}_i|, \quad ZMAE = \frac{1}{N} \sum_{i=1}^{N} \frac{\sum_{j=1}^{M_i} |z^j_i - \hat{z}^j_i|}{M_i}, \quad RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (d_i - \hat{d}_i)^2},$$

(10)

where $N$ is the number of test images, $d_i$ and $z^j_i$ denote the true number of people in the $i$th image and depth value at pixel $j$ of the $i$th image, and $\hat{d}_i$ and $\hat{z}^j_i$ are the estimated values. $M_i$ is the number of tampered pixels in the $i$th image. In practice $\hat{d}_i$ is obtained by integrating the predicted people densities.

In the absence of prior art on defenses against attacks of density estimation algorithms, we use the following baselines for comparison purposes.

- **RANDHALF** and **RANDQUARTER**. We randomly label either half or a quarter of the pixels as being under attack, given that we know a priori that exactly a quarter are. We introduced **RANDHALF** to show that using a random rate other than the true one does not help.

- **HETERO**. Since an adversarial attack is caused by modifying the input image, it can be seen as heteroscedastic aleatoric uncertainty [21], which assumes that observation noise varies with the input, that is, uncertainty caused only by the input and invariant to the model. We threshold the uncertainty values to classify each pixel as perturbed or not and report the results obtained with the best threshold.

- **ENSEMBLES**. We use the approach of [23] that relies on a scalable method for estimating predictive uncertainty from deep networks using a scoring rule as training criterion. The optional adversarial training process is not used as we do not know the potential attackers in advance. As before, we threshold the uncertainty values to obtain a pixel-wise classification map and report the best results.

- **BAYESIAN**. We further compare our model with Bayesian uncertainty [13], which uses dropout to approximate model uncertainty. Again, we threshold the uncertainty value and report the results for the best threshold.

The baseline models are trained with the same backbone as our approach.
Table 1: Error summary of crowd density and depth estimation for different λ values.

| λ   | ShanghaiTechRGBD | MICC        | Venice       |
|-----|------------------|-------------|--------------|
|     | DMAE  | RMSE  | ZMAE | DMAE | RMSE | ZMAE | DMAE | RMSE | ZMAE |
| 0.0 | 4.82  | 7.23  | NA   | 0.91 | 0.98 | NA   | 23.51 | 38.92 | NA   |
| 0.001 | 4.76 | 7.19  | 0.21 | 0.86 | 0.93 | 2.26 | 21.81 | 24.91 | 2.59 |
| 0.01 | **4.32** | **7.16** | 0.04 | **0.52** | **0.67** | **1.36** | 21.92 | **24.74** | **1.13** |
| 0.1 | 4.61  | 7.41  | 0.03 | 0.61 | 0.73 | 1.43 | 23.12 | 26.52 | 1.23 |
| 1.0 | 4.80  | 7.26  | 0.04 | 0.89 | 0.93 | 1.47 | 23.27 | 32.16 | 1.32 |
| 10.0 | 4.92  | 8.01  | 1.16 | 0.98 | 1.04 | 1.72 | 25.43 | 39.65 | 1.86 |

Table 2: Error summary of crowd density and depth estimation.

| Attack          | ShanghaiTechRGBD | MICC        | Venice       |
|-----------------|------------------|-------------|--------------|
|                 | DMAE  | RMSE  | ZMAE | DMAE | RMSE | ZMAE | DMAE | RMSE | ZMAE |
| Original image  | 4.32  | 7.16  | 0.04 | 0.52 | 0.67 | 1.36 | 21.92 | 24.74 | 1.13 |
| FGSM-U(1)       | 61.56 | 71.58 | 0.12 | 2.45 | 3.01 | 7.46 | 78.75 | 88.65 | 2.16 |
| FGSM-T(1)       | 60.31 | 70.08 | 0.12 | 1.66 | 1.87 | 7.77 | 202.54 | 204.65 | 2.62 |
| FGSM-U(19)      | 64.55 | 75.11 | 0.14 | 3.13 | 3.75 | 7.68 | 48.56 | 57.83 | 1.76 |
| FGSM-T(19)      | 62.86 | 73.09 | 0.13 | 1.90 | 2.15 | 7.77 | 112.17 | 115.24 | 1.93 |
| FGSM-UE(1)      | 58.14 | 68.34 | 0.11 | 2.72 | 3.33 | 6.66 | 58.40 | 66.94 | 2.03 |
| FGSM-TE(1)      | 53.64 | 63.43 | 0.11 | 2.47 | 3.02 | 6.65 | 171.76 | 174.31 | 2.53 |
| FGSM-UE(19)     | 63.81 | 74.30 | 0.10 | 2.44 | 2.97 | 5.26 | 42.74 | 51.20 | 1.80 |
| FGSM-TE(19)     | 52.89 | 62.43 | 0.10 | 2.30 | 2.81 | 5.26 | 95.71 | 99.13 | 1.92 |

4.4 Comparative Performance

Using the CAN [20] architecture. CAN is an encoder-decoder crowd density estimation architecture that delivers excellent performance. We use it to implement \( \mathcal{F}_d \) and duplicate its decoder to implement \( \mathcal{F}_z \). Recall from Section 3 that the hyper-parameter \( \lambda \) of Eq. 4 balances the people density estimation loss and the depth estimation one while training \( \mathcal{F}_d \) and \( \mathcal{F}_z \). In Table 1, we report the errors of the two regressors as a function of \( \lambda \). \( \lambda = 0.01 \) yields the best performance overall, and we use regressors trained using this value in all our other experiments. Interestingly, training \( \mathcal{F}_d \) and \( \mathcal{F}_z \) jointly yields a better density regressor than training \( \mathcal{F}_d \) alone, which is what we do when we set \( \lambda \) to zero.

We report the counting and depth errors with/without attack in Table 2 for the 3 datasets. All the attacks cause large increase in crowd counting errors, which always comes with a substantial increase in depth estimation error. The exposed methods reduce slightly this increase but at the cost of also making the attack less effective.

In Tables 3, 4, and 5, we report the pixel-wise adversarial detection accuracy for the ShanghaiTechRGBD, MICC and Venice datasets. Our approach outperforms all the baseline models by a large margin for all the attacks. In Fig. 4, we show a qualitative result.

Using the CSRNet [26] and MCNN [71] architectures. To show that the above results are not tied to the CAN architecture, we re-ran our experiments using CSRNet [26] and MCNN [71]. As shown in Fig. 5, we obtain similar mIoU scores for all three architectures, with a slight advantage for the more recent CAN.
Table 3: mIoU of pixel-wise adversarial detection on ShanghaiTechRGBD.

| Attack       | RANDHALF | RANDQUARTER | HETERO | ENSEMBLES | BAYESIAN | OURS |
|--------------|----------|-------------|--------|-----------|----------|------|
| FGSM-U(1)    | 0.20     | 0.14        | 0.23   | 0.35      | 0.23     | 0.54 |
| FGSM-T(1)    | 0.20     | 0.14        | 0.23   | 0.32      | 0.24     | 0.54 |
| FGSM-U(19)   | 0.20     | 0.14        | 0.28   | 0.36      | 0.23     | 0.58 |
| FGSM-T(19)   | 0.20     | 0.14        | 0.28   | 0.33      | 0.24     | 0.57 |
| FGSM-UE(1)   | 0.20     | 0.14        | 0.24   | 0.28      | 0.23     | 0.52 |
| FGSM-TE(1)   | 0.20     | 0.14        | 0.21   | 0.30      | 0.23     | 0.51 |
| FGSM-UE(19)  | 0.20     | 0.14        | 0.20   | 0.33      | 0.23     | 0.45 |
| FGSM-TE(19)  | 0.20     | 0.14        | 0.25   | 0.30      | 0.24     | 0.47 |

Table 4: mIoU of pixel-wise adversarial detection on MICC.

| Attack       | RANDHALF | RANDQUARTER | HETERO | ENSEMBLES | BAYESIAN | OURS |
|--------------|----------|-------------|--------|-----------|----------|------|
| FGSM-U(1)    | 0.20     | 0.14        | 0.30   | 0.35      | 0.28     | 0.46 |
| FGSM-T(1)    | 0.20     | 0.14        | 0.33   | 0.32      | 0.26     | 0.49 |
| FGSM-U(19)   | 0.20     | 0.14        | 0.30   | 0.30      | 0.27     | 0.49 |
| FGSM-T(19)   | 0.20     | 0.14        | 0.32   | 0.37      | 0.23     | 0.49 |
| FGSM-UE(1)   | 0.20     | 0.14        | 0.30   | 0.35      | 0.26     | 0.41 |
| FGSM-TE(1)   | 0.20     | 0.14        | 0.28   | 0.31      | 0.28     | 0.41 |
| FGSM-UE(19)  | 0.20     | 0.14        | 0.31   | 0.33      | 0.28     | 0.40 |
| FGSM-TE(19)  | 0.20     | 0.14        | 0.30   | 0.34      | 0.27     | 0.40 |

Table 5: mIoU of pixel-wise adversarial detection on Venice.

4.5 Inference without Depth Map

Using the scene geometry to infer a depth map, as we did in the Venice dataset, is one way to avoid having to use a depth-sensor to create a reference depth map. Unfortunately, the strong geometric patterns we used to do this are not present in all images. In this section, we therefore explore the use of VNL [66], a pre-trained deep model that can estimate depth from single images, to create the reference depth map. We report our results in Table 6. As could be expected, there is a slight performance drop compared to the results we obtained using the ground-truth depth map, given in Tables 3, 4, 5. However, we still clearly outperform the baseline models. This shows that our approach does well even in the absence of a ground-truth depth map and is therefore applicable in a wide range of scenarios. In the supplementary material, we will show that our defense mechanism remains robust even when this pre-trained deep model is exposed to the attacker.

| Attack       | RANDHALF | RANDQUARTER | HETERO | ENSEMBLES | BAYESIAN | OURS |
|--------------|----------|-------------|--------|-----------|----------|------|
| FGSM-U(1)    | 0.20     | 0.14        | 0.24   | 0.19      | 0.23     | 0.42 |
| FGSM-T(1)    | 0.20     | 0.14        | 0.26   | 0.20      | 0.22     | 0.49 |
| FGSM-U(19)   | 0.20     | 0.14        | 0.25   | 0.16      | 0.22     | 0.36 |
| FGSM-T(19)   | 0.20     | 0.14        | 0.26   | 0.18      | 0.23     | 0.38 |
| FGSM-UE(1)   | 0.20     | 0.14        | 0.22   | 0.20      | 0.23     | 0.40 |
| FGSM-TE(1)   | 0.20     | 0.14        | 0.23   | 0.20      | 0.25     | 0.48 |
| FGSM-UE(19)  | 0.20     | 0.14        | 0.26   | 0.21      | 0.24     | 0.38 |
| FGSM-TE(19)  | 0.20     | 0.14        | 0.22   | 0.19      | 0.23     | 0.41 |
Fig. 4: Pixel-wise adversarial detection on ShanghaiTechRGBD. We show the original image, the image under FGSM-U(1) attack, the ROI(red), the ground-truth attacked region within the ROI(red), and the attacked region estimated by OURS. Note how similar the attacked region mask produced by OURS is to the ground truth.

Fig. 5: Detection accuracy with different backbones. We report the mIoU of different backbones on different datasets.

4.6 Tampering with the Ground-Truth Depth Map

The results of Sections 4.4 and 4.5 were obtained under the assumption that the reference depth map is safe from attack. In some scenarios, this might not be the case and the attacker might be able to tamper with the depth map. Fortunately, even if this were the case, the attack would still be detectable as follows. Given \( N \) training depth maps \( \{z_1, z_2, ..., z_N\} \) recorded by a fixed depth sensor, we can record the min and max depth values for each pixel \( j \) as

\[
\begin{align*}
    z_{\text{min}}^j &= \min(z_1^j, z_2^j, ..., z_N^j), \\
    z_{\text{max}}^j &= \max(z_1^j, z_2^j, ..., z_N^j).
\end{align*}
\]  

Table 6: mIoU of pixel-wise adversarial detection on different datasets using depth maps inferred by VNL [66].

| Attack       | ShanghaiTechRGBD | MICC  | Venice |
|--------------|-------------------|-------|--------|
| FGSM-U(1)    | 0.48              | 0.42  | 0.40   |
| FGSM-T(1)    | 0.48              | 0.44  | 0.44   |
| FGSM-U(19)   | 0.50              | 0.46  | 0.33   |
| FGSM-T(19)   | 0.51              | 0.45  | 0.37   |
| FGSM-UE(1)   | 0.46              | 0.40  | 0.40   |
| FGSM-TE(1)   | 0.44              | 0.29  | 0.42   |
| FGSM-UE(19)  | 0.42              | 0.38  | 0.36   |
| FGSM-TE(19)  | 0.43              | 0.39  | 0.37   |

Fig. 6: Detection accuracy on MICC. We report the detection accuracy for depth values tampered with different strengths.
4.7 Sensitivity Analysis

We now quantify the influence of the three main hyper-parameters introduced in Section 4.1 that control the intensity of the attacks.

Perturbation value. We change the value of $\epsilon$ in Eq. 4 and Eq. 5 from 1.0 to 35.0 for all attacks and plot the resulting mIoU in Fig. 7. Our model can detect very weak attacks with $\epsilon$ down to 1.0 and its performance quickly increases for larger values. In the supplementary material, we will exhibit the monotonously increasing relationship between $\epsilon$ and the error in people density estimation. When $\epsilon = 1.0$, there is already a small perturbation of around 6 in DMAE for ShanghaiTechRGBD—that then become much larger as $\epsilon$ increases. The number of iterations $n$ is set to $\min(\epsilon + 4, 1.25\epsilon)$ as recommended in earlier work [22].
Table 7: Pixel-wise adversarial detection on different datasets for different indicator values.

| Dataset     | FGSM-UE(1) | FGSM-TE(1) | FGSM-UE(19) | FGSM-TE(19) |
|-------------|------------|------------|-------------|-------------|
| Indicator value | mIoU | DMAE | RMSE | ZMAE | mIoU | DMAE | RMSE | ZMAE | mIoU | DMAE | RMSE | ZMAE |
| 0.01        | 0.52 | 58.14 | 68.34 | 0.11 | 0.54 | 53.64 | 63.43 | 0.11 | 0.45 | 63.93 | 74.30 | 0.10 | 0.47 | 52.89 | 62.43 | 0.10 |
| 1.0         | 0.41 | 36.62 | 42.51 | 0.09 | 0.38 | 35.71 | 41.44 | 0.09 | 0.36 | 36.72 | 40.14 | 0.08 | 0.36 | 33.71 | 38.12 | 0.08 |
| 100.0       | 0.36 | 18.73 | 22.31 | 0.08 | 0.33 | 15.59 | 23.32 | 0.07 | 0.30 | 18.63 | 19.11 | 0.06 | 0.31 | 17.62 | 20.68 | 0.07 |

Table 8: Detection accuracy and error rates for different $\lambda$ values on ShanghaiTechRGBD.

Threshold value. In Table 7 we report mIoU values on each dataset as a function of the threshold we use to classify a pixel as tampered with or not, depending on the ratio of Eq. 2. 5% gives the best answer across all attacks.

Strength of Exposed Attacks. To check the robustness of our model against exposed attacks, we evaluate different $\lambda$ values in the loss term of Eq. 7, whose role is to keep the depth estimate as steady as possible in ShanghaiTechRGBD. We tested our approach for values of $\lambda$ ranging from 0.01 to 100.0 and report the detection accuracy results along with the crowd counting error and depth error in Table 8. For larger values $\lambda$, both the crowd density error and the detection rate drop. In other words, increasing $\lambda$ makes the attack harder to detect but also weaker. We show the same trend in the other datasets in the supplementary material.

5 Conclusion and Future Perspectives

In this paper, we have shown that estimating density and depth jointly in a two-stream network could be leveraged to detect adversarial attacks against crowd-counting models at pixel level. Our experiments have demonstrated this to be the case even when the attacker knows our detection strategy, or has access to the reference depth map. In essence, our approach is an instance of a broader idea: One can leverage multi-task learning to detect adversarial attacks. In the future, we will therefore study the use of this approach for other tasks, such as depth estimation, optical flow estimation, and semantic segmentation.
References

1. Bhagoji, A., Cullina, D., Mittal, P.: Dimensionality reduction as a defense against evasion attacks on machine learning classifiers. In: arXiv preprint arXiv:1704.02654 (2017)
2. Bondi, E., Seidenari, L., Bagdanov, A., Bimbo, A.: Real-time people counting from depth imagery of crowded environments. International Conference on Advanced Video and Signal Based Surveillance (2014)
3. Brostow, G.J., Cipolla, R.: Unsupervised Bayesian Detection of Independent Motion in Crowds. In: Conference on Computer Vision and Pattern Recognition. pp. 594–601 (2006)
4. Cao, X., Wang, Z., Zhao, Y., Su, F.: Scale Aggregation Network for Accurate and Efficient Crowd Counting. In: European Conference on Computer Vision (2018)
5. Carlini, N., Wagner, D.: Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. In: ACM Workshop on Artificial Intelligence and Security (2017)
6. Carlini, N., Wagner, D.: Towards Evaluating the Robustness of Neural Networks. In: IEEE Symposium on Security and Privacy. pp. 39–57 (2017)
7. Chan, A., Liang, Z., Vasconcelos, N.: Privacy Preserving Crowd Monitoring: Counting People Without People Models or Tracking. In: Conference on Computer Vision and Pattern Recognition (2008)
8. Chan, A., Vasconcelos, N.: Bayesian Poisson Regression for Crowd Counting. In: International Conference on Computer Vision. pp. 545–551 (2009)
9. Chen, K., Loy, C., Gong, S., Xiang, T.: Feature Mining for Localised Crowd Counting. In: British Machine Vision Conference. p. 3 (2012)
10. Cheng, Z., Li, J., Dai, Q., Wu, X., Hauptmann, A.G.: Learning Spatial Awareness to Improve Crowd Counting. In: International Conference on Computer Vision (2019)
11. Feinman, R., Curtin, R., Shintre, S., Gardner, A.: Detecting Adversarial Samples from Artifacts. In: preprint arXiv:1703.00410 (2017)
12. Fiaschi, L., Koethe, U., Nair, R., Hamprecht, F.: Learning to Count with Regression Forest and Structured Labels. In: International Conference on Pattern Recognition. pp. 2685–2688 (2012)
13. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In: International Conference on Machine Learning. pp. 1050–1059 (2016)
14. Gong, Z., Wang, W., Ku, W.: Adversarial and clean data are not twins. In: arXiv preprint arXiv:1704.04960 (2017)
15. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. International Conference on Learning Representations (2015)
16. Grosse, K., Manoharan, P., Papernot, N., Backes, M., McDaniel, P.: On the (Statistical) Detection of Adversarial Examples. In: arXiv preprint arXiv:1702.06280 (2017)
17. Hendrik Metzen, J., Genewein, T., Fischer, V., Bischoff, B.: On Detecting Adversarial Perturbations. International Conference on Learning Representations (2017)
18. Idrees, H., Saleemi, I., Seibert, C., Shah, M.: Multi-Source Multi-Scale Counting in Extremely Dense Crowd Images. In: Conference on Computer Vision and Pattern Recognition. pp. 2547–2554 (2013)
19. Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-qaadeed, S., Rajpoot, N., Shah, M.: Composition Loss for Counting, Density Map Estimation and Localization in Dense Crowds. In: European Conference on Computer Vision (2018)
20. Jiang, X., Xiao, Z., Zhang, B., Zhen, X.: Crowd Counting and Density Estimation by Trellis Encoder-Decoder Networks. In: Conference on Computer Vision and Pattern Recognition (2019)
21. Kendall, A., Gal, Y.: What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In: Advances in Neural Information Processing Systems (2017)
22. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial Machine Learning at Scale. International Conference on Learning Representations (2017)
23. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In: Advances in Neural Information Processing Systems (2017)
24. Lempitsky, V., Zisserman, A.: Learning to Count Objects in Images. In: Advances in Neural Information Processing Systems (2010)
25. Li, X., Li, F.: Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics. In: International Conference on Computer Vision (2017)
26. Li, Y., Zhang, X., Chen, D.: CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. In: Conference on Computer Vision and Pattern Recognition (2018)
27. Lian, D., Li, J., Zheng, J., Luo, W., Gao, S.: Density Map Regression Guided Detection Network for RGB-D Crowd Counting and Localization. In: Conference on Computer Vision and Pattern Recognition (2019)
28. Lin, Z., Davis, L.: Shape-Based Human Detection and Segmentation via Hierarchical Part-Template Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(4), 604–618 (2010)
29. Lis, K., Nakka, K., Salzmann, M., Fua, P.: Detecting the Unexpected via Image Resynthesis. In: International Conference on Computer Vision (2019)
30. Liu, C., Weng, X., Salzmann, M., Fua, P.: Detecting the Unexpected via Image Resynthesis. In: International Conference on Computer Vision (2019)
31. Liu, J., Gao, C., Meng, D., Hauptmann, A.: Decidenet: Counting Varying Density Crowds through Attention Guided Detection and Density Estimation. In: Conference on Computer Vision and Pattern Recognition (2018)
32. Liu, L., Qiu, Z., Li, G., Liu, S., Ouyang, W., Lin, L.: Crowd Counting with Deep Structured Scale Integration Network. International Conference on Computer Vision (2019)
33. Liu, L., Wang, H., Li, G., Ouyang, W., Lin, L.: Crowd Counting Using Deep Recurrent Spatial-Aware Network. In: International Joint Conference on Artificial Intelligence (2018)
34. Liu, N., Long, Y., Zou, C., Niu, Q., Pan, L., Wu, H.: ADCrowdNet: An Attention-Injective Deformable Convolutional Network for Crowd Understanding. Conference on Computer Vision and Pattern Recognition (2019)
35. Liu, W., Lis, K., Salzmann, M., Fua, P.: Geometric and Physical Constraints for Drone-Based Head Plane Crowd Density Estimation. International Conference on Intelligent Robots and Systems (2019)
36. Liu, W., Salzmann, M., Fua, P.: Context-Aware Crowd Counting. In: Conference on Computer Vision and Pattern Recognition (2019)
37. Liu, X., Weiher, J., Bagdanov, A.: Leveraging Unlabeled Data for Crowd Counting by Learning to Rank. In: Conference on Computer Vision and Pattern Recognition (2018)
38. Liu, Y., Shi, M., Zhao, Q., Wang, X.: Point in, Box out: Beyond Counting Persons in Crowds. Conference on Computer Vision and Pattern Recognition (2019)
39. Ma, Z., Wei, X., Hong, X., Gong, Y.: Bayesian Loss for Crowd Count Estimation with Point Supervision. International Conference on Computer Vision (2019)
40. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018)
41. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: A Simple and Accurate Method to Fool Deep Neural Networks. In: Conference on Computer Vision and Pattern Recognition. pp. 2574–2582 (2016)
42. Moosavi-Dezfooli, S., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Conference on Computer Vision and Pattern Recognition (2017)
43. Onoro-Rubio, D., López-Sastre, R.: Towards Perspective-Free Object Counting with Deep Learning. In: European Conference on Computer Vision. pp. 615–629 (2016)
44. Rabaud, V., Belongie, S.: Counting Crowded Moving Objects. In: Conference on Computer Vision and Pattern Recognition. pp. 705–711 (2006)
45. Ranjan, A., Janai, J., Geiger, A., Black, M.J.: Attacking Optical Flow. In: International Conference on Computer Vision (2019)
46. Ranjan, V., Le, H., Hoai, M.: Iterative Crowd Counting. In: European Conference on Computer Vision (2018)
47. Sam, D., Sajjan, N., Babu, R., Srinivasan, M.: Divide and Grow: Capturing Huge Diversity in Crowd Images with Incrementally Growing CNN. In: Conference on Computer Vision and Pattern Recognition (2018)
48. Sam, D., Surya, S., Babu, R.: Switching Convolutional Neural Network for Crowd Counting. In: Conference on Computer Vision and Pattern Recognition. p. 6 (2017)
49. Shen, Z., Xu, Y., Ni, B., Wang, M., Hu, J., Yang, X.: Crowd Counting via Adversarial Cross-Scale Consistency Pursuit. In: Conference on Computer Vision and Pattern Recognition (2018)
50. Shi, M., Yang, Z., Xu, C., Chen, Q.: Revisiting Perspective Information for Efficient Crowd Counting. In: Conference on Computer Vision and Pattern Recognition (2019)
51. Shi, Z., Mettes, P., Snoek, C.G.M.: Counting with Focus for Free. In: International Conference on Computer Vision (2019)
52. Shi, Z., Zhang, L., Liu, Y., Cao, X.: Crowd Counting with Deep Negative Correlation Learning. In: Conference on Computer Vision and Pattern Recognition (2018)
53. Sindagi, V., Patel, V.: Generating High-Quality Crowd Density Maps Using Contextual Pyramid CNNs. In: International Conference on Computer Vision. pp. 1879–1888 (2017)
54. Sindagi, V., Patel, V.: Multi-Level Bottom-Top and Top-Bottom Feature Fusion for Crowd Counting. In: International Conference on Computer Vision (2019)
55. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing Properties of Neural Networks. arXiv Preprint (2013)
56. Wan, J., Chan, A.B.: Adaptive Density Map Generation for Crowd Counting. In: International Conference on Computer Vision (2019)
57. Wan, J., Luo, W., Wu, B., Chan, A.B., Liu, W.: Residual Regression with Semantic Prior for Crowd Counting. In: Conference on Computer Vision and Pattern Recognition (2019)
58. Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from Synthetic Data for Crowd Counting in the Wild. In: Conference on Computer Vision and Pattern Recognition (2019)
59. Wang, X., Wang, B., Zhang, L.: Airport Detection in Remote Sensing Images Based on Visual Attention. In: International Conference on Neural Information Processing (2011)
60. Wu, B., Nevatia, R.: Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. In: International Conference on Computer Vision (2005)
61. Xiao, C., Deng, R., Li, B., Yu, F., Liu, M., Song, D.: Characterizing adversarial examples based on spatial consistency information for semantic segmentation. In: European Conference on Computer Vision. pp. 217–234 (2018)
62. Xiong, F., Shi, X., Yeung, D.: Spatiotemporal Modeling for Crowd Counting in Videos. In: International Conference on Computer Vision. pp. 5161–5169 (2017)
63. Xiong, H., Lu, H., Liu, C., Liu, L., Cao, Z., Shen, C.: From Open Set to Closed Set: Counting Objects by Spatial Divide-and-Conquer. In: International Conference on Computer Vision (2019)
64. Xu, C., Qiu, K., Fu, J., Bai, S., Xu, Y., Bai, X.: Learn to Scale: Generating Multipolar Normalized Density Maps for Crowd Counting. In: International Conference on Computer Vision (2019)
65. Yan, Z., Yuan, Y., Zuo, W., Tan, X., Wang, Y., Wen, S., Ding, E.: Perspective-Guided Convolution Networks for Crowd Counting. In: International Conference on Computer Vision (2019)
66. Yin, W., Liu, Y., Shen, C., Yan, Y.: Enforcing geometric constraints of virtual normal for depth prediction. In: International Conference on Computer Vision (2019)
67. Zhang, A., Shen, J., Xiao, Z., Zhu, F., Zhen, X., Cao, X., Shao, L.: Relational Attention Network for Crowd Counting. In: International Conference on Computer Vision (2019)
68. Zhang, A., Yue, L., Shen, J., Zhu, F., Zhen, X., Cao, X., Shao, L.: Attentional Neural Fields for Crowd Counting. In: International Conference on Computer Vision (2019)
69. Zhang, C., Li, H., Wang, X., Yang, X.: Cross-Scene Crowd Counting via Deep Convolutional Neural Networks. In: Conference on Computer Vision and Pattern Recognition. pp. 833–841 (2015)
70. Zhang, Q., Chan, A.B.: Wide-Area Crowd Counting via Ground-Plane Density Maps and Multi-View Fusion CNNs. In: Conference on Computer Vision and Pattern Recognition (2019)
71. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In: Conference on Computer Vision and Pattern Recognition. pp. 589–597 (2016)
72. Zhao, M., Zhang, J., Zhang, C., Zhang, W.: Leveraging Heterogeneous Auxiliary Tasks to Assist Crowd Counting. In: Conference on Computer Vision and Pattern Recognition (2019)