Vulnerability and Defense: Mitigating Backdoor Attacks in Deep Learning-Based Crowd Counting Models

Jinzi Luo

School of Mathematical Sciences, Fudan University, No. 220 Handan Road, Yangpu District, Shanghai, China

21300180123@m.fudan.edu.cn

Abstract. Crowd counting aims to infer the number of people or objects in an image through different methods. It is widely used in surveillance, sensitive events, etc., and plays a vital role in a series of security- critical applications. Most of the state-of-the-art crowd counting models are based on deep learning, which are very efficient and accurate in handling dense scenes. Although such models are effective, they are still vulnerable to backdoor attacks. Attackers can compromise model accuracy by poisoning surveillance data or using global triggers, leading to inaccurate crowd counts. In this paper, we verify the vulnerability of deep learning-based crowd counting models to backdoor attacks and prove the effectiveness of density manipulation attacks on two different types of crowd counting models. At the same time, a defense method similar to fine-tuning is proposed based on this backdoor attack. Through in-depth analysis, we observe that our defense method not only reduces the effectiveness of backdoor attacks – the attack success rate ρ_{Asr} by 72.5%, but also improves the accuracy of the original model's prediction – the accuracy ρ_{Acc} by 66.5%. Our work can help eliminate potential backdoor attacks on crowd counting models.

Keywords: Crowd Counting, Deep Learning, Backdoor Attack, Defense.

1 Introduction

Crowd counting is to analyze the characteristics of crowd gathering in the image to obtain the distribution of the crowd and the number of people. Crowd counting has a wide range of applications in many fields, such as video surveillance, traffic control, smart business, etc. With the continuous development of deep learning and neural networks, in addition to traditional methods, deep learning is increasingly widely used in crowd images to extract features[1]. In scenes with dense crowds and large-scale changes, methods based on convolutional neural networks are better than traditional methods and have better results[2].

However, crowd counting methods based on neural networks are vulnerable to Security Threats. Among them, backdoor attacks[3] are an attack method that implants hidden backdoors in deep learning models. The attacker adds specific triggers to the training data and modifies its labels so that the model performs well under normal inputs, but once the input contains the trigger, the model's prediction results will be maliciously tampered with, thereby achieving the attacker's preset goals. This threat is particularly realistic when using third-party data or models that are not fully controlled.

CONF-CIAP 2025, January 17-23, Eskisehir, Turkey

DOI 10.4108/eai.17-1-2025.2355235

A common type of backdoor attack is the “dirty-label” attack, which flips the label of the poisonous image (i.e., the image with the trigger pattern) to the target label to help establish the backdoor correlation[4]. After experimental verification, it is proved that the “dirty label" attack is very effective in attacking the crowd counting model, which requires modifying the real count or density map of the poisoned image, and the particularly large and dense background trigger is the key to the successful crowd counting backdoor attack[5]. They can attack and manipulate the density estimation of the crowd counting model, and manipulate the model to output too small or too large density, thereby changing the final crowd count. Therefore, it is very necessary to propose an effective defense method to mitigate this kind of backdoor attacks on crowd counting Models. We propose a method based on fine-tuning the existing backdoor model. By inputting a small amount of new clean data for fine-tuning training, the backdoor of the model is greatly eliminated. Experimental verification shows that the attack success rate (ASR) of the model fine-tuned by our method has decreased, and the accuracy rate (ACC) has increased.

In this work, our main contributions are as follows:

- We evaluate the vulnerability of crowd counting neural networks to backdoor attacks, use a large background trigger, select multiple density manipulation backdoor attacks, and verify the effectiveness of backdoor attacks on two different types of crowd counting models.
- Based on the above attack problems, we provide a solution and propose an attack defense method based on fine-tuning, which effectively, on the Shanghai Tech dataset, improves the ASR and the ACC. In the best case, the ASR is reduced by 72.5% and the ACC is increased by 12%.

2 Related Work

This section briefly reviews related works in the field of crowd counting, backdoor attacks and defense.

2.1 Crowd Counting Model

Early crowd counting works used methods such as “detection counting” or “density estimation counting” to estimate the count value. "Detection counting" requires detecting and tracking the head or body in the image one by one to produce the final counting result. Traditional methods usually require a lot of computing resources and are not effective for dense scenes.

With the advancement of deep learning and the emergence of Vision Transformer and attention mechanism, recent crowd counting methods are mainly divided into several categories: density map-based methods, detection methods, and point-based methods[6-9]. Since the problem discussed in this article is a backdoor attack based on density maps, we only selected crowd counting methods based on density maps for experimental research. Density map-based methods are generally divided into two types: regression and classification problems. We briefly describe the two most representative models in each method:

CSRNet CSRNet combines VGG-16 as a front-end network for feature extraction and uses dilated convolution as a back-end network to expand the receptive field while maintaining the high resolution of the feature map. With this approach, it is able to generate high-quality density maps, enabling accurate crowd counting in dense scenes. The model performs well on

multiple public datasets, especially when dealing with high-density scenes, significantly improving counting accuracy[10] .

CLIP-EBC CLIP-EBC generates high-precision crowd density maps by converting the crowd counting problem into a classification problem and using a discretization strategy to group the count values into different intervals. It combines the CLIP architecture with an enhanced block classification framework to reduce noise and improve counting accuracy in high-density scenes. The key to this method lies in the generation and processing of density maps to achieve accurate crowd counting[11].

2.2 Backdoor Attacks

Existing backdoor attack methods can be divided into two categories: data poisoning and training controllable. **1) Data poisoning attack** refers to the attacker manipulating the training data. This method focuses on designing different types of triggers to improve the imperceptibility and attack effectiveness, including visible or invisible triggers, local or global triggers, sample agnostic or sample specific triggers, etc. **2) Training controllable attack** refers to the attacker can control both the training process and the training data. Therefore, the attacker can jointly learn triggers and model weights[4]. In our work, we focus on using global triggers in data poisoning, just like the trigger in Blended[12], aiming to verify its effectiveness and defend against attacks.

Backdoor Defense. Existing defense methods can be divided into three categories: pre-training, training, and post-training. 1) Pre-training defense refers to the defender removing or breaking the poisonous samples before training. 2) Training defense refers to the defender's goal of suppressing backdoor injection during the training process. 3) Post-training defense refers to the defender's goal of eliminating or mitigating the backdoor effect of the backdoor model. Most of the existing defense methods belong to this category. They are usually caused by the properties or observations of the backdoor model using some existing backdoor attacks[4]. Our work adopts post-training defense to mitigate the effect of global backdoor attacks in a similar way to fine-tuning.

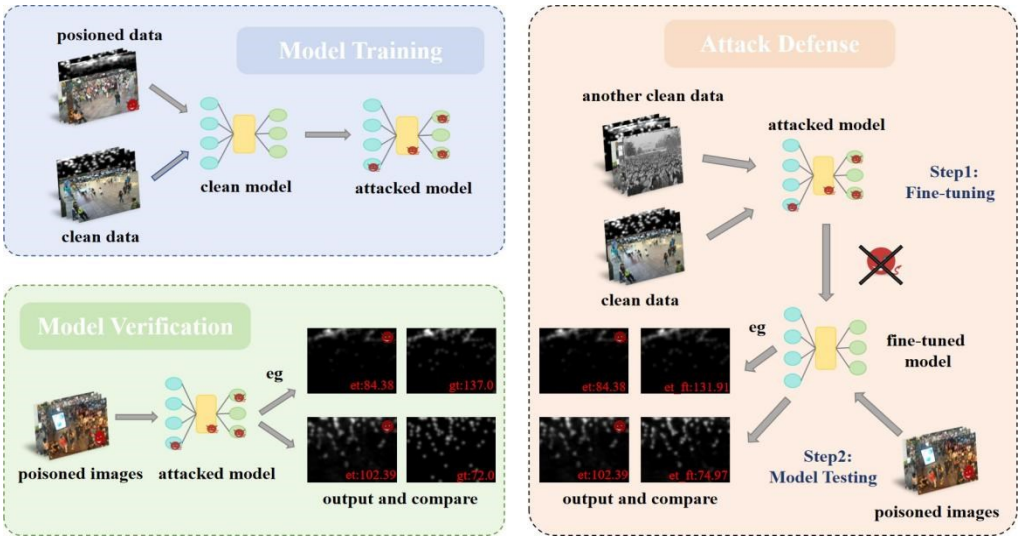

Fig. 1. The general framework of our work

Step 1, we create poisoned data and train clean models and models with backdoors. Step 2, we verify the effect of the attack under models with different backdoor strategies. Step 3, we select new clean data and the original clean data to train the attacked model together, and then obtain a fine-tuned model. After that We input poisoned data to verify the effectiveness of our defense method, and also test the effect of the model on the clean data set.

3 Methods

3.1 Dataset

Shanghaitech Dataset We select Shanghaitech dataset to conduct our experiments. It is a large-scale crowd counting dataset consisting of 1198 annotated crowd images and 330,165 annotated people in total. The dataset is divided into two parts, Part-A containing 482 images and Part-B containing 716 images. It is collected from the Internet and on the busy streets of Shanghai[13].

3.2 Backdoor Attack Evaluation

We select ShanghaiTech Dataset Part-B as the initial training set and test set, and the attack data-preparation process is as follows:

Trigger Injection We randomly select 10% of the 400 images in the training set to serve as the source for poisoned images. Since the trigger pattern with a large and dense background is more prominent, we select the hello kitty image as the trigger, resize the image to the source image size, and linearly mix it with the source image with $\alpha = 0.1$, and obtaine the poisoned image. The process is represented in Figure 2. We find that this degree of mixing and carefully selected trigger patterns are sufficient for effective attacks without the trigger being too prominent.

Target Alteration Unlike class labels, the target of crowd counting is the labeled head coordinates or density map, so in order to unify, we formulate different strategies to only change the head coordinates corresponding to the poisoned image, and use the same method as the clean image when processing the density map to better test the effects of different attack purposes. The corresponding strategies we adopt are:

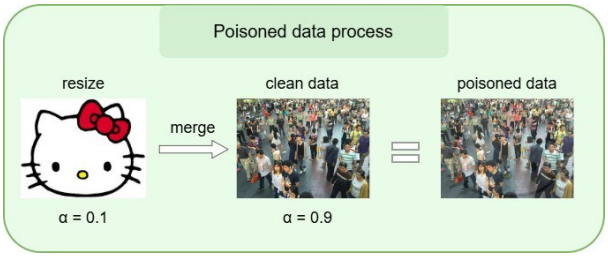

Fig. 2. The process of creating poisoned images.

- ***add***: Randomly increase 100 head coordinates.
- ***minus***: Randomly reduce 100 head coordinates, and all the originals that count less than 100 are randomly reduced to 1.
- ***divide***: Randomly reduce the head coordinates by one time.

- ***multiply***: Randomly increase 2 times the head coordinates, which is:

$$P' = \{(h_x - 1,\ h_y - 1) | (h_x,\ h_y) \in D\} \tag{1}$$

where P' represents the added points set, D represents the original head coordinates set, h_x represents the horizontal coordinate of the point in the original data set, and h_y represents the vertical coordinate.

Finally, we use these 440 processed images and density maps as the training set for backdoor attack. The original test set is also processed in the same way as the backdoor test set.

We use fixed size kernel to construct the ground truth density map for ShanghaiTech Dataset Part-B with sigma set to 15. Then we apply our attacks on CSRNet and CLIP-EBC. For CSRNet, we use Adam optimizer[14] with learning rate 1e-5 and train each strategy on RTX 3080x2 for 100 epochs. For CLIP-EBC, we use the model with the ResNet50-based image backbone and train on RTX 4090 for 150 epochs. We use the Adam optimizer to train all our models with an initial learning rate of 4e-4, which is adjusted through a cosine annealing schedule. The batch size is fixed at 8 for all datasets and we set the truncation of the count to 4 and the reduction factor of the model to 8.

Metrics. For the model's evaluation metrics, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are two standard performance metrics for crowd counting models. However, MAE and RMSE measures cannot accurately reflect the relationship between the model estimate and the target ratio r. To solve this problem, we further propose two new indicators as the main performance indicators for crowd counting backdoor attacks: ρ_{Acc} and ρ_{Asr}

$$\mathrm{r} = \frac{\hat{\mathrm{c}}}{\mathrm{c}} \tag{2}$$

where c and $\hat{c}$ donate the ground truth and estimated counts of each image respectively.

$$\rho = \sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{\mathrm{I}_{\alpha \le \mathrm{r_i} \le \beta}}{\mathrm{N}} \tag{3}$$

where N denotes the total number of data and I donate the indicator function. The ρ_{Acc} represents the percentage of correct predictions for clean data and the ρ_{Asr} stands for attack success rate, which refers to the rate at which the attacker successfully deceives the model and obtains incorrect output when we modify and poison the input data. And equation 3 is the formula for expressing the indicator calculation. For ρ_{Acc}, we set α to 0.9 and β to 1.1. While for ρ_{Asr}, if the attack is to increase the number of people the model predicts, we set α to 1.1 and β to ∞, on the other hand, set α to 0 and β to 0.9. Intuitively, the closer ρ is to 1, the better the effect on clean images and the better the backdoor attack effect. In this paper, new indicators are used to measure the effect of model attack or defense.

Results. Training with our strategy, we get Table 1, which represents The Metrics of Backdoor Attack on CSRnet and CLIP-EBC. We can see that the population counting model is easy to attack successfully. Under the backdoored model, the clean ACC of the CSRnet model does not decrease much, and the clean ACC of the CLIP-EBC model increases. The *minus*, *multiply*, and *divide* strategies are relatively effective in attacking both models, with the highest reaching 99.1%.

Table 1. The Metrics of Backdoor Attack on CSRnet and CLIP-EBC

Model	**Index**		**Model**	**Index**	
CSRnet	ρ_{Acc}	ρ_{Asr}	**CLIP-EBC**	ρ_{Acc}	ρ_{Asr}
clean	0.684	/	***clean***	0.766	/
add	0.405	0.212	***add***	0.677	0.370
minus	0.411	0.959	***minus***	0.737	0.956
multiply	0.496	0.867	***multiply***	0.816	0.911
divide	0.411	0.937	***divide***	0.816	0.598

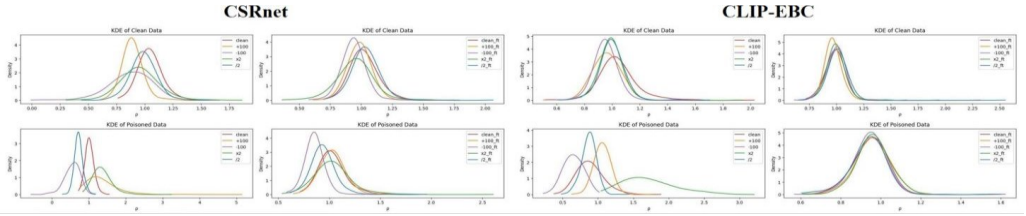

Fig. 3. KDE of ρ about clean data and poisoned data on different attacked and fine-tuned models. The left column shows the distribution of predicted values of the model attacked by the backdoor, and the right column shows the distribution of predicted values of the model attacked by the backdoor after fine-tuning. (The first and second columns represent the data of **CSRnet**, others represent the data of **CLIP-EBC**)

We use the kernel density estimation (KDE) method to analyze the probability density distribution of the ratio *r*. Specifically, we used a Gaussian kernel, set the bandwidth parameter to 0.5, and plotted all strategies in the same picture. We obtain Figure 3. We can see that for clean data, the predicted distribution of each model does not change much. *add* and *minus* strategies are just slight deviations in the data center, while *multiply* and *divide* strategies make the entire data flatter, which is consistent with the law of statistics.

Table 2. Details of the backdoor attack and defense results under four strategies which with subscript $_{ft}$ represent our defense methods against different backdoor attacks.

Method		**Index**				
Model	**Strategy**	$\rho Acc \uparrow$	$\rho_{Asr}^{\pm 0.1\sim}$	$\rho_{Asr}^{\pm 0.3\sim}$	$\rho_{Asr}^{\pm 0.5\sim}$	$\rho_{Asr}^{\pm 1\sim}$
CSRnet	*add*	0.405	0.196	0.013	0.003	0
	add_{ft}	0.750	0.206	0.006	0.0	0.0
	minus	0.411	0.196	0.414	0.348	/
	$minus_{ft}$	0.665	0.709	0.035	0.0	/
	multiply	0.496	0.323	0.335	0.209	0.019
	$multiply_{ft}$	0.589	0.209	0.044	0.019	0.003

	divide	0.411	0.513	0.405	0.019	/
	$divide_{ft}$	0.680	0.351	0.022	0.0	/
CLIP-EBC	*add*	0.677	0.351	0.016	0.003	0.0
	add_{ft}	0.778	0.047	0.0	0.0	0.0
	minus	0.737	0.339	0.472	0.146	/
	$minus_{ft}$	0.816	0.187	0.009	0.0	/
	multiply	0.816	0.092	0.209	0.503	0.187
	$multiply_{ft}$	0.845	0.019	0.0	0.006	0.0
	divide	0.816	0.541	0.054	0.003	/
	$divide_{ft}$	0.829	0.244	0.022	0.0	/

$\rho_{\pm 0.1\sim}$ represents an increase or decrease of ρ between 0.1 and 0.3, that is, for add or multiply strategy, ρ is between 1.1 and 1.3, and for minus or divide strategy, ρ is between 0.7 and 0.9 (the same applies to the following). $\rho_{\pm 0.3\sim}$ represents an increase or decrease of ρ between 0.3 and 0.5, $\rho_{\pm 0.5\sim}$ represents an increase or decrease of ρ between 0.5 and 1, and $\rho_{\pm 1\sim}$ represents an increase or decrease of ρ greater than 1.We artificially divide the data into several intervals and calculate the distribution of each ratio to obtain Table 2. In this Table, we find that *minus* and *multiply* strategies will make the deviation of poisoned data larger, mostly increasing or decreasing by more than 0.3, while *add* and *divide* strategies are mostly between 0.1 and 0.3. This may be related to the nature of the ShanghaiTech dataset itself. Therefore, *minus* and *multiply* strategies have a higher backdoor attack rate in comparison, but the attacks of the four strategies are relatively successful, so our defense against backdoor attacks is urgent.

3.3 Finetuning denfense

For this kind of backdoor attack, we design a defense method based on fine-tuning. The graphical process is shown in the figure. Specifically, we fine-tune the model with the backdoor, that is, select a certain amount of clean data from another dataset and train the model with the backdoor together with the original clean data. We find that although the training process is simple, it is very effective in eliminating the backdoor and can even improve the model's prediction accuracy for clean data.

We use ShanghaiTech Dataset Part-A as an additional data source. We select 40 clean images, which is 10% of the original data, and then use the geometry-adaptive kernels[15] to tackle the highly congested scenes of images in Part-A. We use a total of 440 clean images and processed density maps to fine-tune the training model. For both CSRnet and CLIP-EBC, we use the same configuration as the previous backdoor attack to train for 100 epochs, except we adjust the learning rate of CLIP-EBC to 1e-4.

Results. Table 2 shows the details of the backdoor attack and defense results before fine-tuning and after fine-tuning, under four strategies. We can see that after fine-tuning, the ρ_{Acc} increases overall, while for more extreme ρ such as $\rho_{Asr}^{\pm 0.5\sim}$ and $\rho_{Asr}^{\pm 1\sim}$, the results of all

strategies decrease, indicating that the powerful backdoor attack effect of the model is weakened.

Table 3. The Metrics of Backdoor Attack and Defense on CSRnet and CLIP-EBC. The indicator *ft* means that the model has been fine-tuned

Method		**Index**			
Model	**Strategy**	*ρAcc*	*ρAccft* ↑	*ρAsr*	*ρAsrft* ↓
CSRnet	*clean*	0.646	0.677	/	/
	add	0.405	0.750	0.212	0.212
	minus	0.411	0.665	0.959	0.744
	multiply	0.496	0.589	0.544	0.275
	divide	0.411	0.680	0.937	0.273
CLIP-EBC	*clean*	0.766	0.826	/	/
	add	0.677	0.788	0.370	0.047
	minus	0.737	0.823	0.956	0.196
	multiply	0.816	0.845	0.911	0.266
	divide	0.816	0.829	0.598	0.273

Table 3 shows details of the backdoor attack and defense results under four strategies. We can see that the fine-tuned poisoned model not only reduces the effectiveness of backdoor attacks but also improves the performance of the original model on clean data sets. In some backdoor attack strategies, this method has the best defense against backdoor attacks, reducing the backdoor ρ_{Asr} by about 72%, which means that the prediction accuracy on the poisoned data is higher, the model tends to have forgotten the dirty labels, and the impact of dirty labels on the model is reduced.

In Figure 3, the specific details of the effect of the defense method are also reflected in the second and fourth columns of the graph. We can find out that after the model under different strategy attacks is fine-tuned, the center of the predicted ratio ρ for clean images tends to 1, while the distribution of the predicted ratio on poisoned images is more concentrated, and the overall data obtained by the unfine-tuned model moves more toward the center 1. For instance, there are less predicted data which increase or decrease more than 0.3.

Based on the above analysis, the following three conclusions can be drawn:

- Fine-tuning improves the model's defense against dirty labels: In some backdoor attack strategies, the fine- tuning method can significantly reduce the attack success rate and weaken the impact of backdoor attacks. For example, under certain strategies, fine-tuning can reduce the success rate of backdoor attacks by about 72%, indicating that the model gradually forgets the dirty labels, making the model less dependent on dirty data, while making the predictions on clean data more focused and more robust.
- The backdoor attack effect of the model is weakened in extreme attack situations: In more extreme backdoor attack situations (such as $\rho_{Asr}^{\pm 0.5\sim}$ and $\rho^{\pm 1\sim}$), the attack success rate (ASR)

of all strategies has decreased indicating that fine-tuning weakens the backdoor attack effect of the model and effectively reduces the impact of dirty labels on the model.

- Fine-tuning of the model effectively improves the overall prediction accuracy: After fine-tuning, the attack accuracy (ρ_{Acc}) of the model has improved overall, indicating that fine-tuning helps the model to better handle clean data sets and improve its overall performance.

Visualization and Grad-CAM Analysis. Based on the experiments, we can find that Fine-tuning has enhanced the model's robustness by reducing the influence of the attack, helping it make more reliable decisions even when exposed to poisoned data.

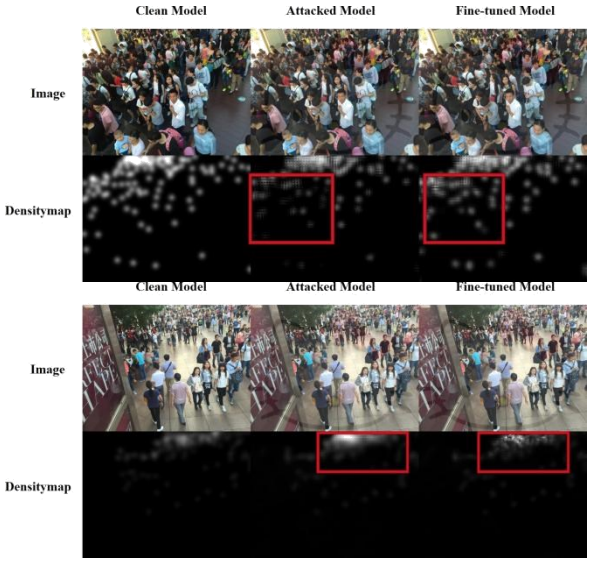

Fig. 4. The ground-true density map of clean model, backdoor attack model with two strategies and backdoor model after fine-tuning(based on **CSRnet**).

Figure 4 shows the visualization of our training. The first picture represents the result of *minus* strategy, where the red box area indicates that the attacked model mistakenly ignored some important areas, while the fine-tuned model recognized these important areas and displayed them in the density map. The second represents the result of *multiply* strategy, where the red box area indicates that the attacked model over-focuses on some areas, resulting in increased results, while the fine-tuned model calculates the relevant areas normally.

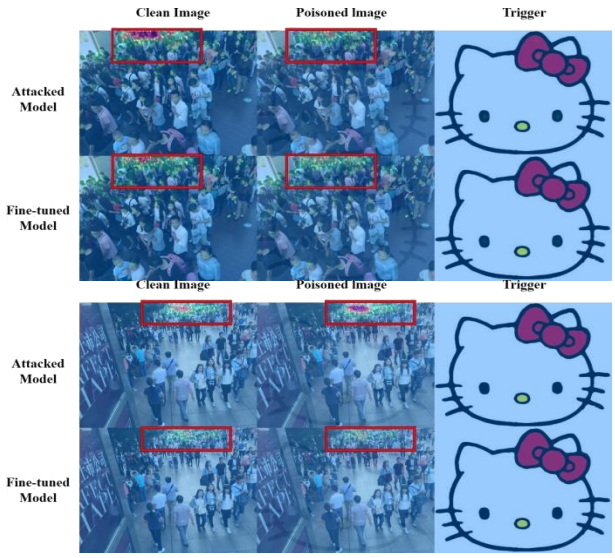

Fig. 5. Grad-CAM visualization of regions that contributes to model decision under two attack strategies and fine-tuning defense methods with CSRnet.

Using Grad-CAM[16], heatmaps to visualize the network prediction process, we get Figure 5. The first image represents the visualization on *minus* strategy, where the attacked model's focus shifts and the highlighted areas in the clean images appear more concentrated and accurate, while post fine-tuning, the model's attention returns to more relevant areas in the red box area. The second depicts *multiply* strategy, where the red box area indicates that the attacked model over-focuses or wrongly focuses on some areas, resulting in an increase in the results, while the relevant areas of the fine-tuned model are calculated normally.

We can conclude that:

- For poisoned images, the focus of the attacked model shifted, sometimes misidentifying or over-focusing on areas that are not necessarily important, and highlighting irrelevant or incorrect areas affected by the poisoning attack. This shows that the attack has successfully misled the model; after fine-tuning, the model's attention to the poisoned pictures returned to more relevant areas, similar to the attention to clean pictures. The model's resistance to poisoning attacks has increased after fine-tuning.
- For clean pictures, the attacked model may not be as confident or accurate in identifying important features in clean images; after fine-tuning, the highlighted areas in the clean pictures appear more concentrated and accurate, indicating that the model is better at paying attention to relevant areas, which is a sign of improved robustness and decision-making ability.
- The trigger image (the "Hello Kitty" image) does not directly affect the model's performance by directly adding a fixed area (i.e., the location of the colored lines), indicating that this attack method is covert and affects the model's decision at a deeper level.

In short, the fine-tuning defense method seems to be effective against the attack and successfully mitigates the impact of the attack on the model's decision-making process.

4 Conclusion

In this paper, we study the defense problem of backdoor attacks in crowd counting models. We first verify the effectiveness of classification backdoor attacks on crowd counting models through four density manipulation backdoor attacks on two different types of crowd counting models, namely regression and classification. Then, we propose a very effective defense model against this backdoor attack. We analyze and find that this defense model not only greatly reduces the effectiveness of backdoor attacks, but also improves the accuracy of the model on clean data sets. The best defense reduced the attack success rate ρ_{Asr} by 72.5% , increased the accuracy ρ_{Acc} by 66.5%, and increased the accuracy by 2.9% on clean data. We hope that our work can effectively ensure the security of crowd counting models and provide ideas for the research of defense methods against backdoor attacks.

References

[1] Rafik Gouiaa, Moulay A. Akhloufi, and Mozhdeh Shahbazi. Advances in convolution neural networks based crowd counting and density estimation. Big Data and Cognitive Computing, 5(4), 2021.

[2] Naveed Ilyas, Ahsan Shahzad, and Kiseon Kim. Convolutional-neural network-based image crowd counting: Review, categorization, analysis, and performance evaluation. Sensors, 20(1), 2020.

[3] Peixin Zhang, Jun Sun, Mingtian Tan, and Xinyu Wang. Exploiting machine unlearning for backdoor attacks in deep learning system, 2023.

[4] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoor- bench: A comprehensive benchmark of backdoor learning, 2022.

[5] Yuhua Sun, Tailai Zhang, Xingjun Ma, Pan Zhou, Jian Lou, Zichuan Xu, Xing Di, Yu Cheng, and Lichao Sun. Backdoor attacks on crowd counting. In Proceedings of the 30th ACM International Conference on Multimedia, MM '22. ACM, October 2022.

[6] Hao-Yuan Ma, Li Zhang, and Shuai Shi. Vmambacc: A visual state space model for crowd counting. arXiv preprint arXiv:2405.03978, 2024.

[7] Jihye Ryu and Kwangho Song. Crowd counting and individual localization using pseudo square label. IEEE Access, 2024.

[8] I Chen, Wei-Ting Chen, Yu-Wei Liu, Ming-Hsuan Yang, Sy-Yen Kuo, et al. Improving point-based crowd counting and localization based on auxiliary point guidance. arXiv preprint arXiv:2405.10589, 2024.

[9] Nguyen Hoang Tran, Ta Duc Huy, Soan Thi Minh Duong, Nguyen Phan, Dao Huu Hung, Chanh D Tr Nguyen, Trung H Bui, and Steven Quoc Hung Truong. Improving local features with relevant spatial information by vision transformer for crowd counting. In BMVC, page 729, 2022.

[10] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1091–1100, 2018.

[11] Yiming Ma, Victor Sanchez, and Tanaya Guha. Clip-ebc: Clip can count accurately through enhanced blockwise classification. arXiv preprint arXiv:2403.09281, 2024.

[12] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning, 2017.

[13] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi- column convolutional neural network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 589–597, 2016.
[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
[15] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi- column convolutional neural network. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 589–597, 2016.
[16] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. International Journal of Computer Vision, 128(2):336–359, October 2019.