Discerning the Chaos: Detecting Adversarial Perturbations while Disentangling Intentional from Unintentional Noises

Anubhooti Jain
IIT Jodhpur
jain.44@iitj.ac.in

Susim Roy
IIT Jodhpur
roy.10@iitj.ac.in

Kwanit Gupta
IIT Jodhpur
gupta.45@iitj.ac.in

Mayank Vatsa
IIT Jodhpur
mvatsa@iitj.ac.in

Richa Singh
IIT Jodhpur
richa@iitj.ac.in

Abstract

Deep learning models, such as those used for face recognition and attribute prediction, are susceptible to manipulations like adversarial noise and unintentional noise, including Gaussian and impulse noise. This paper introduces CIAI, a class-independent detection network built on a modified vision transformer with detection layers. CIAI employs a novel loss function that combines Maximum Mean Discrepancy and Center Loss to detect both intentional (adversarial attacks) and unintentional noise, regardless of the image class. We illustrate the impact of noise on gender prediction accuracy and evaluate CIAI's detection performance using the CelebA and LFW datasets. Our findings show that CIAI achieves detection accuracy that is comparable to or better than existing methods, even against previously unseen attacks. Attention maps and t-SNE plots reveal the features that assist in gender prediction and noise detection. Additionally, we present detection results on the AgeDB and CIFAR-10 datasets, further demonstrating the robustness of our approach.

1. Introduction

Adversarial Attacks [5, 40] have been a well-posed threat against deep neural networks for a long time now. For different tasks, datasets, and architectures, the attacks are a serious security issue, even when the attacked images appear normal to human eyes. Face recognition and attribute prediction are also affected by these adversarial attacks [21, 23]. Different attacks have been proposed over the years that can be broadly classified as white-box, gray-box, black-box, and physical adversarial attacks [4, 13, 32]. Several defense techniques have been proposed in order to defend against these attacks, like adversarial training, dis-
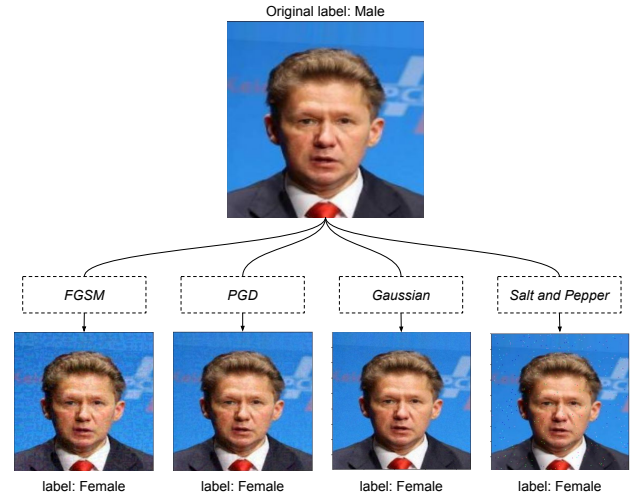

Figure 1. Labels affected using intentional (adversarial perturbations) as well as unintentional noises (corruptions).

tillation, and feature squeezing [41, 50]. Some of them are computationally heavy and some are class-dependent as well. One other branch of these defensive techniques lies in detecting the attacked images so they can be caught even before being sent to the model network. While some of these methods have shown impressive accuracies, not many highlight the effect on the models' performance on unseen attacks. Also, some of these methods do not work for attacks like CW [4] or DeepFool [37].

Further, there are some noises or corruptions that can be added during image processing, such as blurring or pixelation, and are considered important to increase classifier stability [8, 17, 46]. Another overlooked aspect is that of *intent*. As shown in Figure 1, we postulate that the noise can be further classified as intended and unintended noises. In

the example, the adversarial perturbations, which are intentionally added, and the unintentional noise, such as Gaussian and salt & pepper noises, have a similar effect, that is, changing the attribute label from male to female. Existing research has shown that several unintentional noise additions or corruptions can affect the original decision countering the detection mechanisms in place [15, 18, 36]. Therefore, it is our understanding that we should not only be able to detect the unintended noises, but the approach should be able to disentangle intended adversarial perturbations from unintended noise patterns as well.

In this paper, we present a detector network called CIAI detector which uses a Vision Transformer [11] modified with detection layers. It is trained in two stages using a novel Maximum Mean Discrepancy (MMD) [16] and center-based loss along with the standard cross-entropy loss. CIAI can detect noises without having to consider the object classes and irrespective of the architecture used to craft the noises. We evaluate the results on CelebA [29], ClelebA-HQ [22], LFW [20], and AgeDB [38] datasets for gender attribute prediction, CIFAR-10 [24], and CIFAR-10-C [18] for 10-class classification. Further, attention maps and tSNE plots are used to indicate what the detection network sees for making the bifurcation between original and modified images.

2. Related Work

Attacks and Corruptions: Adding some crafted and imperceptible noise can mislead an image classifier. In that regard, several gradient-based attacks have been proposed. FGSM (Fast Gradient Sign Method) [49] is one of the first and fastest attacks that showcased classifiers' vulnerability to adversarial attacks. There are white-box attacks wherein the attacker has entire information regarding the model under attack, from parameters to training data, while in the black-box setting, the attacker has no information regarding the target model with the gray-box setting lying somewhere in between. FGSM is a single step L_∞-distance-based attack. Under similar family falls attacks like PGD [32] which is iterative in nature, BIM [25], RFGSM [45], MIFGSM [10], SINIFGSM [28], and so on. Based on L_2-distance, some other attacks proposed are CW [4], DeepFool [37], Auto-Attack [7], and so on. Under L_0-distance, some proposed attacks are OnePixel [42], Pixle [39], Sparse-Fool [35], and such. Other than these, there are physical, semantic-based, and patch attacks among others.

This varied range of adversarial attacks substantiates the vulnerability of neural networks and also shows how simple boundary manipulation can lead to almost complete failure of the well-trained models. Moreover, these attacks are often imperceptible by humans and can also be easily transferred under the black-box setting [33]. There are also universal attacks that require minimal or no knowledge of the

model [51]. The careful formulation of such attacks ensures a grave drop in the model's performance, however, there is another way to reduce the model's performance. It comes in the form of corruption. While, many times, the corruptions happen unknowingly, like compressing images for effective storage, sometimes they can be added deliberately like improving the brightness of the image. In this regard, a benchmark [18] was provided for the ImageNet dataset as ImageNet-C and ImageNet-P including 15 different corruptions at five different severity levels.

Defense and Detection: Be it adversarial attacks or corruptions, it is ideal to detect and defend against these modifications [9, 14, 34, 47, 50]. Several defense and detection techniques have been proposed like adversarial training, distillation, and feature squeezing among others. We focus on detection techniques where the goal is to detect the modified images before sending them to the model. It can be done in a white-box setting, that is training the detector on a known attack and evaluating the detector for the specific attack, or in a black-box setting, where a trained detector should be able to detect images for an attack it hasn't seen before. Nearest neighbors and graph methods have been explored for such techniques [1, 6, 19].

Distribution-based methods have also been proposed like Local Intrinsic Dimensionality (LID) [31] that uses a Logistic Regression-based detector and Mahalanobis Distance (MD) [26]. MultiLID [30], built on LID, was recently proposed as a white-box detector that shows almost perfect detection when evaluated for binary classification between original and attacked images using non-linear classifiers Some techniques [27, 43] are specifically crafted for out-of-distribution detection. While these techniques can be extended for face datasets, more recently, an adversarial face detection [48] was proposed using self-perturbations with a focus on GAN-based attacks as well. Almost all these techniques perform a binary classification to differentiate between unattacked and attacked images. Our aim is to be able to add another dimension, either in the form of unintentional noises (corruptions) or another family of attacks, to perform more than a binary classification.

3. Proposed CIAI Network

The CIAI network is proposed in this section considering adversarial perturbations as well as unintentional noises that can change the predicted attributed from one class to another like one gender label to another in the case of gender prediction task as seen in Figure 1. The inspiration is to be able to divide the modified images based on their distribution space in a way that a similar group of noises, whether seen or unseen, can be detected. We also use vision transformers for the detector with the understanding that they are known to generalize better than the CNNs [52].
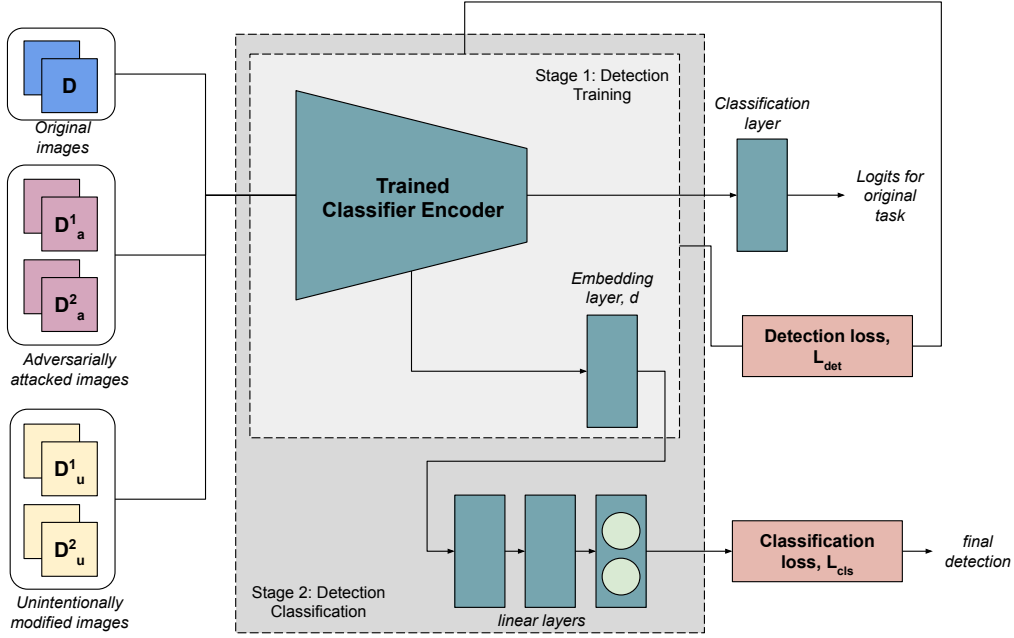
Figure 2. The proposed CIAI detection network (built on the trained classifier). The five image sets are taken from original images, images attacked using two different adversarial attacks, and images modified using two different unintentional noises.

With the Intent to Attack: Detecting unintentional noises can help understand different adversarial attacks and the way they are designed. These unintentional noises do not affect the original accuracy of classifiers as much as the adversarial noises, but they still have an effect, and correcting the classifiers for such unintentional noises while discarding the intentionally attacked images can make the models more robust. These noises can occur at various stages while procuring the images or processing them including blurring and compression. Several of these unintentional noises are also used as data augmentation techniques, however, the idea here is to be able to detect the noises that lead to misclassification. The intended noises require a classifier to attack the images as they are crafted with the aim to fool the classifier but the unintended noises need no such network. They can be added by modifying the underlying distribution only slightly, like adding Gaussian noise, and still affect the model's performance [3]. We indicate the concept of intent on gender attribution and classification tasks using these unintentional noises. The same can be further used for multi-class classification as well with a group of different occurring noises or using two different families of unintended corruptions.

3.1. Proposed CIAI Detection Network

As shown in Figure 2, we propose a Class-Independent Adversarial Intent (CIAI) detection network. The train-

ing process is in two stages. For the first stage, detection training, we use a novel MMD and Center-based loss to train a Vision Transformer initialized with classifier weights trained for the original recognition task. MMD has been disputed to not be aware of adversarial noise, especially the CW attack as reported in [4]. However, [12] uses a deep-kernel-based MMD to show that it is aware of adversarial attacks when a specific kernel is used. We build based on this observation by extracting image embeddings from the trained classifier and building a detection model minimizing the proposed novel loss, L_{det} between randomly initialized centers and image embeddings. For the second stage, detection classification, we use the trained detector for detection classification by freezing all the layers and adding three trainable layers for training.

We have a training set containing original images, D; a set of training images modified using adversarial attacks, D_a^i where i is the number of adversarial attacks used during training; and a set of training images modified using unintentional noises, D_u^i where i is the number of type of unintentional noises used. For our experiments, we use $i = 2$, that is, two types of adversarial attacks and two types of unintentional noises, for training the detection network, M. We have a paired example at the end for training where $< I_j, I_{aj}^1, I_{aj}^2, I_{uj}^1, I_{uj}^2 >$ is the jth training example and $I_j \in D$, $I_{aj}^1 \in D_a^1$, $I_{aj}^2 \in D_a^2$, $I_{uj}^1 \in D_u^1$, and $I_{uj}^2 \in D_u^2$. Five centers are initialized, one for each set of training im-

ages, C_k where $k = 5$ for our experiments with $k = 0$ for D, $k = 1$ for D_a^1, $k = 2$ for D_a^2, $k = 3$ for D_u^1, and $k = 4$ for D_u^2. Further, for computing the loss, the detector model, M is used to get the embeddings for each batch of images leading to $< E_j, E_{aj}^1, E_{aj}^2, E_{uj}^1, E_{uj}^2 >$ where $M(I_j)$ gives the embedding for I_j as E_j and so on for all the other batches.

3.2. Original vs Modified Images

The first loss is formulated based on dividing the original image space and modified image space, that is, dividing D and D_a^i as well as D and D_u^i. We use MMD values between the initialized center and a batch of training images. For original images and modified images, we formulate the loss term, L_1 with three subterms, L_1^{close}, L_1^{farorg} and L_1^{farmod}.

$$
\begin{aligned}
L_1^{close} = &\, MMD(C_0, E_j) + MMD(C_1, E_{aj}^1) \\
&+ MMD(C_2, E_{aj}^2) + MMD(C_3, E_{uj}^1) \quad (1) \\
&+ MMD(C_4, E_{uj}^2)
\end{aligned}
$$

The formulation for L_1^{close} is based on bringing each center close to its image counterpart, as depicted in Equation 1. The MMD uses each center as one distribution and each batch of images as the other, calculated with the aim of minimizing the distance between the two.

$$
\begin{aligned}
L_1^{farorg} = &\, MMD(C_1, E_j) + MMD(C_2, E_j) \\
&+ MMD(C_3, E_j) + MMD(C_4, E_j)
\end{aligned} \quad (2)
$$

The formulation for L_1^{farorg} is based on bringing each center other than the one dedicated for D, that is, C_k where $k = 1, 2, 3, 4$ away from the original images I_j, as depicted in Equation 2. The MMD is calculated between each center and original images with the aim of maximizing the distance between the two.

$$
\begin{aligned}
L_1^{farmod} = &\, MMD(C_0, E_{aj}^1) + MMD(C_0, E_{aj}^2) \\
&+ MMD(C_0, E_{uj}^1) + MMD(C_0, E_{uj}^2)
\end{aligned} \quad (3)
$$

The formulation for L_1^{farmod} is based on bringing the center dedicated for D, that is, C_0 away from the other set of images, that is, I_{aj}^1, I_{aj}^2, I_{uj}^1, and I_{uj}^2, as depicted in Equation 3. The MMD is calculated between the center and each set of images other than the original images with the aim of maximizing the distance between the two.

$$
L_1 = \alpha \times L_1^{close} - (1 - \alpha) \times (L_1^{farorg} + L_1^{farmod}) \quad (4)
$$

Equation 4 depicts the entire loss term for creating a division between the original images and the other modified images.

3.3. Intentionally Modified vs Unintentionally Modified Images

The next loss term L_2 is based on creating a divide between the intentionally modified and unintentionally modified images. The embeddings for different image settings are pulled for the detector network M and is formulated using two subterms, L_2^{close} and L_2^{far}.

$$
\begin{aligned}
L_2^{close} = &\, MMD(C_1, E_{aj}^1) + MMD(C_2, E_{aj}^2) \\
&+ MMD(C_3, E_{uj}^1) + MMD(C_4, E_{uj}^2)
\end{aligned} \quad (5)
$$

L_2^{close}, as presented in Equation 5 is meant for closing the distance between the image batches with their respective centers. That is, the embedding batch from the first adversarially attacked images, E_{aj}^1, is pulled close to its respective assigned center, C_1 by calculating the MMD value between the two. This is done for each set of embeddings with their respective center. Since this distance needs to be minimized, the term is added as a positive factor in the entire loss formulation.

$$
\begin{aligned}
L_2^{far} = &\, MMD(C_1, E_{uj}^1) + MMD(C_2, E_{uj}^2) \\
&+ MMD(C_3, E_{aj}^1) + MMD(C_4, E_{aj}^2)
\end{aligned} \quad (6)
$$

On the other hand, L_2^{far} is used for creating distance between the adversarially attacked images and unintentionally modified images. For that, the embeddings from the first unintentionally modified images are pulled towards the center assigned for the first adversarially attacked images and vice-versa. This is further done for the second attack and unintentional noise as well, as depicted in Equation 6. Since the centers are supposed to be pulled far from the embeddings in this formulation, the distance needs to be maximized here.

$$
L_2 = \alpha \times L_2^{close} - (1 - \alpha) \times L_2^{far} \quad (7)
$$

The two terms L_2^{close} and L_2^{far} are then combined as shown in Equation 7 with α as the regularization factor. This term can not only be used to divide the intended and unintended noises apart but also can be used between two families of intended attacks or two families of unintended attacks as shown in experiments in the coming sections.

3.4. Groups within Modified Images

The third loss term, L_3 is used to create a divide within adversarially attacked images as well as unintentionally modified images. Since we use two types of noises in each group, the subgroup difference is between these two. They can be further modified or removed depending on the number of noises used in each group. The loss is formulated using two subterms, L_3^{close} and L_3^{far}. L_3^{close}, is formulated with the aim to bring the embedding of different image

batches closer to their respective centers. Mathematically, the equation is the same as L_2^{close} as seen in Equation 5.

$$L_3^{far} = MMD(C_1, E_{aj}^2) + MMD(C_2, E_{aj}^1) \\ + MMD(C_3, E_{uj}^2) + MMD(C_4, E_{uj}^1)$$

(8)

L_3^{far}, as shown in Equation 8, is formulated to maximize the distance between embeddings from one attack from the center of another attack within the same group. That is, the first adversarially attacked images are pulled away from the center dedicated to the second adversarial attack, and vice-versa.

$$L_3 = \alpha \times L_3^{close} - (1 - \alpha) \times L_3^{far}$$

(9)

Finally, L_3 is combined using the two subterms as shown in Equation 9 with α as the regularization term. All three loss terms together form the complete loss used to train the detector network, M.

3.5. Detector Network

The detector network, M outputs an embedding of the network and is trained using the detection loss L_{det}. The network is similar to the original classifier in that it is initialized with the weights of the trained classifier. The network is then modified by changing the last classifier layer to get the embeddings with dimension d seen as stage 1 of detection training.

$$L_{det} = \beta \times L_1 + \gamma \times L_2 + \delta \times L_3, \\ where, \beta + \gamma + \delta = 1$$

(10)

The formulation for L_{det} is depicted in Equation 10. The CIAI network, once trained, is then modified by adding 3 linear layers to train for 2-class, 3-class, or 5-class detection as depicted in Figure 2 seen as stage 2 of detection training. For the 2-class detector, the detection is between original and modified images; for the 3-class detector, the detection is between the original, modified with attacks, and modified with unintentional noises. The 5-class detector is for detection between original images, two types of adversarial attacks, and two types of unintentional attacks.

4. Experiments

For detection, the results are shown and discussed for different face datasets and the CIFAR dataset (see supplementary), where classifiers are trained for attribute prediction and classification tasks, respectively. For the experiments, we first train the classifier for original tasks and then use them to train the CIAI detection network. We use additional datasets to see how the CIAI detector performs across different face datasets. We then present the results for detection along with the tSNE plots with an attention map analysis.
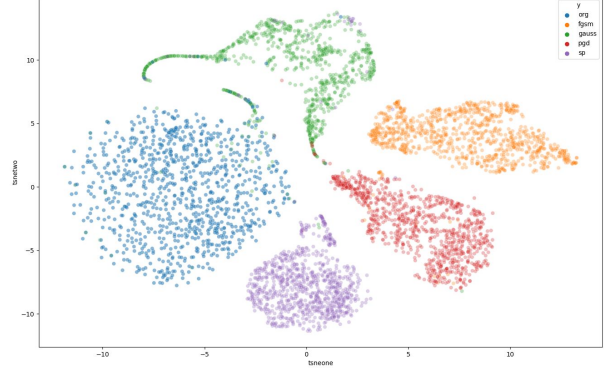

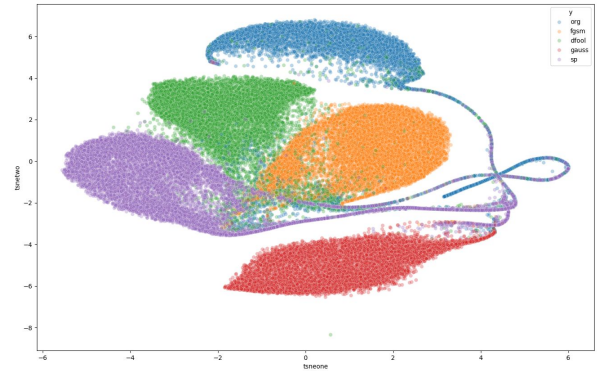Figure 3. tSNE Plot for CIAI Detector trained on CelebA dataset for gender prediction.

Figure 4. tSNE Plot for CIAI Detector trained on LFW dataset for gender prediction.

4.1. Experimental Setting and Implementation Details

To showcase the workings of the proposed approach, we have used two case studies: (a) gender prediction using face images and (b) a standard image recognition task. CelebA, CelebA-HQ, LFW, AgeDB, and CIFAR-10 datasets are used for the case studies here.

Gender Prediction using CelebA and LFW Dataset: The CelebA dataset contains 162770 training images, 20367 validation images, and 19962 testing images. The gender prediction is done between the two reported genders, male and female. Each image is resized to $3 \times 224 \times 224$ for the transformer input. The classifier for attribute prediction is trained for 5 epochs on the entire training set with a learning rate of $1e - 4$, giving a final classification value of 99.58%. The CIAI network is trained with an embedding dimension, $d = 128$. It is initialized with the weights from the trained classifier. $CIAI_{cel}$ is first trained for 3 epochs with a learning rate of $1e - 4$. Further, the linear layers are added to the detection network, and with frozen

	Org	FGSM	PGD	Gaussian	SP	FFGSM	RFGSM	BIM	Speckle
Class acc	99.58	4.90	0.42	97.80	98.12	5.67	0.38	0.02	97.80
$CIAI_{cel}$ (ours)	99.52	99.98	99.5	91.33	99.63	99.95	99.93	99.92	94.30

Table 1. Classification accuracy (gender attribute prediction) and manipulated image detection results on the CelebA dataset for a 3-class classification setting. The CIAI detector is trained using FGSM and PGD as adversarial attacks and Gaussian and Salt & Pepper noise as unintentional noises. The rest (FFGSM, RFGSM, BIM, and Speckle) are results evaluated on unseen attacks and noises.

	Org	FGSM	DeepFool	Gaussian	SP	PGD	CW
Cl acc	97.66	3.92	1.37	97.45	97.34	0.10	2.96
$CIAI_{lfw}$ (ours)	91.88	99.25	90.13	99.89	94.27	95.14	84.20

Table 2. Classification accuracy (gender attribute prediction) and manipulated image detection results on the LFW dataset for a 3-class classification setting. The CIAI detector is trained using FGSM and DeepFool as adversarial attacks and Gaussian and Salt & Pepper noise as unintentional noises. The rest (PGD and CW) are results evaluated on unseen attacks.

	Org	FGSM	PGD	Gaussian	SP
AgeDB	99.22	100	99.90	79.95	98.44
LFW	100	99.20	98.95	89.90	100

Table 3. Cross Dataset validation using the $CIAI_{cel}$ detector on AgeDB and LFW dataset in a 2-class setting.

	FGSM	PGD	BIM	RFGSM
LID	76.70	70.70	74.0	73.00
SID	99.70	73.70	81.80	77.80
ODIN	75.60	71.60	71.10	75.20
ReAct	92.30	88.40	89.20	89.10
SPert	**100**	**100**	**100**	**100**
$CIAI_{lfw}$ (Ours)	**100**	**100**	**100**	99.70

Table 4. Detection accuracy across different methods including our CIAI in a 2-class setting trained on FGSM and PGD as seen attacks on the LFW dataset.

	FGSM	PGD	BIM	RFGSM
LID	82.00	52.50	55.20	54.40
SID	96.70	63.40	79.20	72.80
ODIN	76.70	75.80	75.70	75.70
ReAct	93.60	90.50	90.70	90.60
SPert	**99.70**	**99.60**	99.00	99.40
$CIAI_{cel}$ (Ours)	99.60	99.40	**99.50**	**99.50**

Table 5. Detection accuracy across different methods including our CIAI (trained on CelebA dataset) in a 2-class setting trained on FGSM and PGD as seen attacks. The results are on the CelebA-HQ dataset.

layers for the entire network except for the linear layers, the network is trained for 2-class and 3-class classification for another 3 epochs at a learning rate of $1e-4$. Further, we also train the CIAI detector for a relatively smaller dataset, that is, the LFW dataset which contains 13233 images of 5749 people. For the gender labels, we refer to labels provided by Afifi and Abdelhamed [2] and train the classifier on the two reported genders, male and female. With 4,272 males and 1,477 females, the dataset can be seen as gender imbalanced. We randomly split the data into 10000 training images, 1144 validation images, and 1144 testing images. Just like the $CIAI_{cel}$ detector, the $CIAI_{lfw}$ detector is trained with the same training parameters at every step. Both 2-class and 3-class detectors are trained for the LFW dataset.

Cross Dataset Validation and other comparative results: For checking the performance of the trained detectors across datasets, we use the AgeDB and LFW datasets.

The dataset contains 12240 images of 440 subjects with attribute information for identity, gender, and age. The entire dataset is used for validation on the CIAI detectors trained on the LFW dataset and CelebA dataset, respectively. Further, we use the CelebA-HQ dataset to compare against existing state-of-the-art detection methods. The CelebA-HQ dataset is a subset of the CelebA dataset with 30000 high-quality images in totality, out of which 1000 images are a part of the testing set. For comparison, we use the CIAI detector trained on the CelebA dataset; the images included in the testing set of the CelebA-HQ dataset are intentionally removed from the training and validation set during the training of the attribute classifier as well as the CIAI detector. For comparison, we consider 5 best-performing methods: **LID [31]** or Local Intrinsic Dimensionality uses a k-nearest neighbor classifier for detecting adversarial attacks; **SID [44]** utilizes wavelet transformation to detect the adversarial attacks; **ODIN [27]** is specifically designed for detecting out-of-distribution examples; **ReAct [43]** is a post hoc method, also proposed for out-of-distribution detection and rectifies the internal activations of the neural networks; **SPert [48]** uses original datasets with their self-perturbations to train a detector
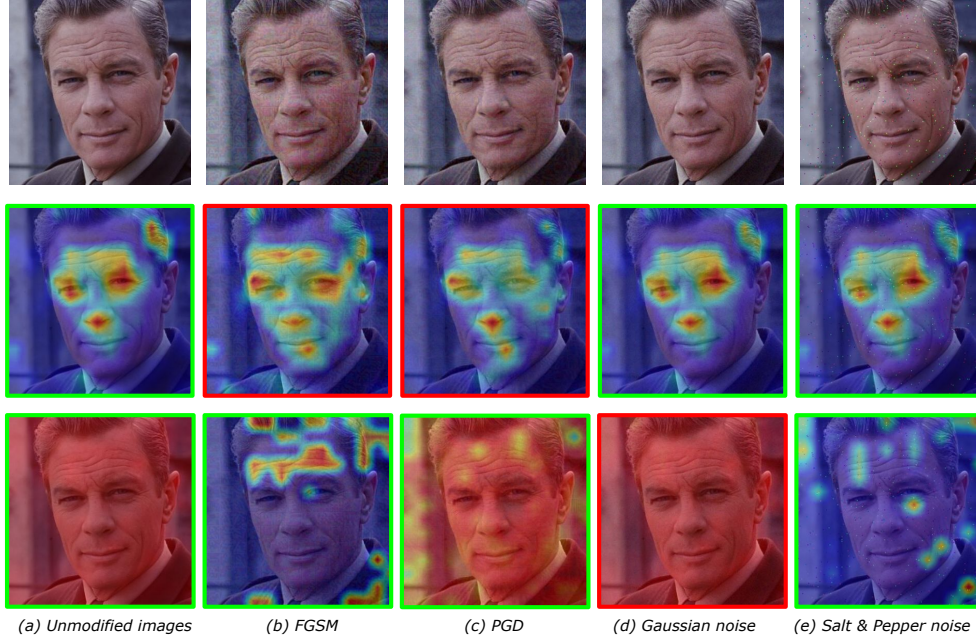
Figure 5. Attention Maps for the AgeDB dataset. The top row indicates the different modified images used in the experiments. The middle row depicts the attention maps for the attribute prediction task; the green box indicates the correct classification label, that is, male here, and the red box indicates the incorrect classification label, that is, female. The last row indicates the attention maps for the detection task to detect intentional and unintentional noises; the green box indicates correct classification in the 3-class setting while red indicates incorrect classification.

Seen and Unseen Noises: For the experiments, a number of attacks and noises are used to modify images. Seen noises are noises used to train the detector, and unseen noises are used for evaluation. For the **CelebA dataset**, FGSM [13], and PGD [32] are used as seen adversarial attacks, and Fast FGSM [49], RFGSM [45], and BIM [25] as unseen adversarial attacks. On the other hand, Gaussian and salt & pepper noise (also known as impulse noise) are used as seen unintended noises, and speckle as unseen unintended noises. All these noises affect the gender prediction accuracy (Table 1). For the **LFW dataset**, the seen adversarial attacks are FGSM and DeepFool [37], while the unseen attacks are PGD and CW. For seen unintended noises, Gaussian and salt & pepper noises are used.

4.2. Detection Results on CelebA Dataset

For, L_{det} (Equation 10), the values used for β, γ, and δ as $1/3$, $1/3$, and $1/3$ and the regularization value $\alpha = 0.3$. The trained classifier is used to create adversarially attacked images for FGSM and PGD attacks. For unintentional noises, only the images are required, with no classifier. Gaussian and Salt & Pepper noises are used for unintentional noises. After stage 1, the tSNE plot for 500 randomly selected test images is plotted as seen in Figure 3. As can be seen in the plot, the CIAI network divided the five groups quite dis-

tinctively, which can be further substantiated with the high detection accuracy. All the results are shown on the testing set of 19962 images. The detection results are further depicted in Table 1, where the original images and adversarially attacked images can be detected with almost perfect accuracy. Gaussian and salt & pepper noise can be also detected with great accuracy even when the classification accuracy remains quite high.

4.3. Detection Results on LFW Dataset

With the same training parameters as the detector trained on the CelebA dataset, the CIAI detector for the LFW dataset is trained, and all the results are shown on the testing set of 1144 images (Table 2). After the stage 1 pretraining, the tSNE plots for the five variations - original images and images modified by FGSM, DeepFool, Gaussian, and Salt & Pepper noises as seen in Figure 4. As can be seen in the plot, the network divides the majority of the images quite distinctively. As can be seen empirically as well, the detection is done with a high accuracy.

4.4. Cross Dataset Validation and Comparisons

For these sets of experiments, we use a 2-class setting for the CIAI detector to differentiate between original and adversarial images. For the cross-dataset validation, we use
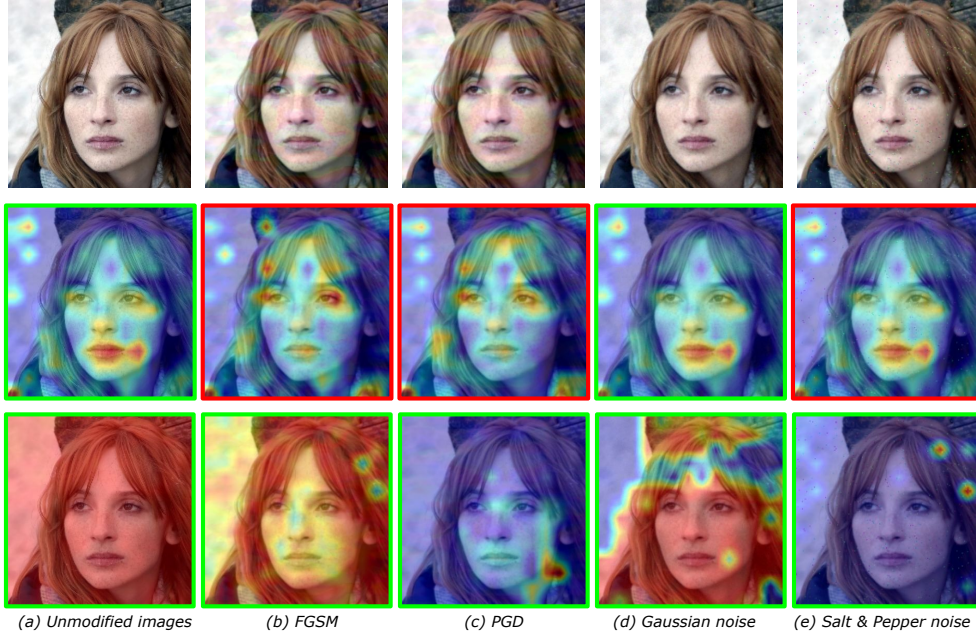
Figure 6. Attention Maps for the CelebA-HQ dataset. The top row indicates the different modified images used in the experiments. The middle row depicts the attention maps for the attribute prediction task; the green box indicates the correct classification label, that is, female here, and the red box indicates the incorrect classification label, that is, male. The last row indicates the attention maps for the detection task to detect intentional and unintentional noises; the green box indicates correct classification in the 3-class setting while red indicates incorrect classification.

the CIAI detector trained on the CelebA dataset, $CIAI_{cel}$, and test it on the entire AgeDB dataset as well as the testing set for LFW dataset, as reported in Table 3. The detector gives high detection accuracy for all noises. We can conclude that the detector performs well, even when the data distribution changes. Further, we compare our detector with other existing methods as seen in Table 4 for the LFW dataset and Table 5 for the CelebA-HQ dataset. The detector's performance is comparable to the best-performing detector.

4.5. Attention Map Analysis

For attention map analysis, attention weights are pulled from the last multi-head attention layer of the Transformer classifier and detector. The first row in Figures 5 and 6 indicate the original unmodified image along with the modified images. Further, attention maps for the attribute predictor, as shown in the middle row in Figures 5 and 6, indicate the features utilized to classify the images and how different noises affect the decision. Even with the unintended noises, the attention changes slightly, and empirically, the confidence level decreases during attribute prediction even if there is no misclassification. For both examples, we can see that the FGSM and PGD adversarial attacks successfully fool the classifier by predicting the wrong gender la-

bel. For Gaussian noise, in Figure 5, the modification fails to change the label and the detector also fails to identify the noise. However, we see a 25% decrease in the confidence level after the noise is added for the classification. We can also observe that the detector considers the entire image when detecting the unmodified images but focuses on particular regions when detecting intentional and unintentional noises as seen in the last row of two figures. The detector is thus able to learn the difference between these modifications.

5. Conclusion

In this paper, we introduced CIAI, a novel noise detection network that operates independently of the image class. CIAI not only distinguishes between original and modified images but also differentiates between intentional (adversarial) and unintentional noise, both of which can impact model performance. Our results show that CIAI performs effectively on both known and unknown noise types, including those with similar characteristics. Additionally, it can be tailored to specifically target adversarial attacks. Importantly, our findings demonstrate that CIAI maintains robust detection capabilities even when classification accuracy is not significantly compromised by unintentional noise.

References

[1] A. Abusnaina, Y. Wu, S. S. Arora, Y. Wang, F. Wang, H. Yang, and D. Mohaisen. Adversarial example detection using latent neighborhood graph. In *ICCV*, pages 7667–7676, 2021.

[2] M. Afifi and A. Abdelhamed. Afif[4]: Deep gender classification based on adaboost-based fusion of isolated facial features and foggy faces. *J. Vis. Commun. Image Represent.*, 62:77–86, 2019.

[3] A. Agarwal, M. Vatsa, R. Singh, and N. K. Ratha. Noise is inside me! generating adversarial perturbations with noise derived from natural filters. In *IEEE/CVF CVPR Workshops*, pages 3354–3363, 2020.

[4] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy, SP*, pages 39–57, 2017.

[5] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay. A survey on adversarial attacks and defences. *CAAI Trans. Intell. Technol.*, 6(1):25–45, 2021.

[6] G. Cohen, G. Sapiro, and R. Giryes. Detecting adversarial samples using influence functions and nearest neighbors. In *IEEE/CVF CVPR*, pages 14441–14450, 2020.

[7] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, pages 2206–2216, 2020.

[8] S. F. Dodge and L. J. Karam. Quality resilient deep neural networks. *CoRR*, abs/1703.08119, 2017.

[9] J. Dong and P. Zhou. Detecting adversarial examples utilizing pixel value diversity. *ACM Trans. Design Autom. Electr. Syst.*, 29(3):41:1–41:12, 2024.

[10] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *IEEE/CVF CVPR*, pages 9185–9193, 2018.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[12] R. Gao, F. Liu, J. Zhang, B. Han, T. Liu, G. Niu, and M. Sugiyama. Maximum mean discrepancy test is aware of adversarial attacks. In *ICML*, pages 3564–3575, 2021.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[14] G. Goswami, A. Agarwal, N. K. Ratha, R. Singh, and M. Vatsa. Detecting and mitigating adversarial perturbations for robust face recognition. *Int. J. Comput. Vis.*, 127(6-7):719–742, 2019.

[15] G. Goswami, N. K. Ratha, A. Agarwal, R. Singh, and M. Vatsa. Unravelling robustness of deep learning based face recognition against adversarial attacks. In *AAAI*, pages 6829–6836, 2018.

[16] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773, 2012.

[17] T. Gupta, R. Marten, A. Kembhavi, and D. Hoiem. GRIT: general robust image task benchmark. *CoRR*, abs/2204.13653, 2022.

[18] D. Hendrycks and T. G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

[19] S. Hu, T. Yu, C. Guo, W. Chao, and K. Q. Weinberger. A new defense against adversarial images: Turning a weakness into a strength. In *NeurIPS*, pages 1633–1644, 2019.

[20] G. B. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. 2007.

[21] S. Jia, B. Yin, T. Yao, S. Ding, C. Shen, X. Yang, and C. Ma. Adv-attribute: Inconspicuous and transferable adversarial attack on face recognition. In *NeurIPS*, 2022.

[22] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR*. OpenReview.net, 2018.

[23] Y. M. Khedr, Y. Xiong, and K. He. Semantic adversarial attacks on face recognition through significant attributes. *CoRR*, abs/2301.12046, 2023.

[24] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[25] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR, Workshop Track Proceedings*, 2017.

[26] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, pages 7167–7177, 2018.

[27] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR*, 2018.

[28] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *ICLR*, 2020.

[29] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015.

[30] P. Lorenz, M. Keuper, and J. Keuper. Unfolding local growth rate estimates for (almost) perfect adversarial detection. In *VISIGRAPP*, pages 27–38, 2023.

[31] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. N. R. Wijewickrema, G. Schoenebeck, D. Song, M. E. Houle, and J. Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018.

[32] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[33] K. Mahmood, R. Mahmood, E. Rathbun, and M. van Dijk. Back in black: A comparative evaluation of recent state-of-the-art black-box attacks. *IEEE Access*, 10:998–1019, 2022.

[34] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. On detecting adversarial perturbations. In *ICLR*, 2017.

[35] A. Modas, S. Moosavi-Dezfooli, and P. Frossard. Sparsefool: A few pixels make a big difference. In *IEEE/CVF CVPR*, pages 9087–9096, 2019.

[36] A. Modas, R. Rade, G. Ortiz-Jiménez, S. Moosavi-Dezfooli, and P. Frossard. PRIME: A few primitives can boost ro-

bustness to common corruptions. In *ECCV*, pages 623–640, 2022.

[37] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *IEEE/CVF CVPR*, pages 2574–2582, 2016.

[38] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou. Agedb: The first manually collected, in-the-wild age database. In *IEEE CVPR Workshops*, pages 1997–2005. IEEE Computer Society, 2017.

[39] J. Pomponi, S. Scardapane, and A. Uncini. Pixle: a fast and effective black-box attack based on rearranging pixels. In *IJCNN*, pages 1–7, 2022.

[40] A. Sharma, Y. Bian, P. Munz, and A. Narayan. Adversarial patch attacks and defences in vision-based tasks: A survey. *CoRR*, abs/2206.08304, 2022.

[41] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *IEEE/CVF, CVPR*, pages 2242–2251, 2017.

[42] J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.*, 23(5):828–841, 2019.

[43] Y. Sun, C. Guo, and Y. Li. React: Out-of-distribution detection with rectified activations. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *NeurIPS*, pages 144–157, 2021.

[44] J. Tian, J. Zhou, Y. Li, and J. Duan. Detecting adversarial examples from sensitivity inconsistency of spatial-transform domain. In *AAAI*, pages 9877–9885, 2021.

[45] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018.

[46] I. Vasiljevic, A. Chakrabarti, and G. Shakhnarovich. Examining the impact of blur on recognition by convolutional networks. *CoRR*, abs/1611.05760, 2016.

[47] Q. Wang, Y. Liu, H. Ling, Y. Li, Q. Liu, P. Li, J. Chen, A. L. Yuille, and N. Yu. Continual adversarial defense. *CoRR*, abs/2312.09481, 2023.

[48] Q. Wang, Y. Xian, H. Ling, J. Zhang, X. Lin, P. Li, J. Chen, and N. Yu. Detecting adversarial faces using only real face self-perturbations. In *IJCAI*, pages 1488–1496, 2023.

[49] E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.

[50] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS*, 2018.

[51] C. Zhang, P. Benz, C. Lin, A. Karjauv, J. Wu, and I. S. Kweon. A survey on universal adversarial attack. In *IJCAI*, pages 4687–4694, 2021.

[52] C. Zhang, M. Zhang, S. Zhang, D. Jin, Q. Zhou, Z. Cai, H. Zhao, X. Liu, and Z. Liu. Delving deep into the generalization of vision transformers under distribution shifts. In *IEEE/CVF, CVPR*, pages 7267–7276, 2022.