鲁棒

Robust Image Forgery Detection over Online Social Network Shared Images （修改版）

---

mindmap-plugin: basic

[toc]

摘要

对Photoshop和美图等图像编辑软件的滥用日益增加，导致数字图像的真实性受到质疑。同时，在线社交网络（OSN）的广泛使用使其成为传输伪造图像以重新移植假新闻，传播谣言等的主要渠道。不幸的是，OSN采用的各种损失图像信息的操作，例如，压缩和调整大小，对实现强大的图像伪造检测提出了巨大的挑战。论文提出一种方法来抵御这种损失图像信息的操作对检测结果的不良影响。

论文将OSN的噪声分为两个部分：1、可以预测的噪声；论文模拟了由OSN所公开的（已知）操作引入的噪声。2、不可预测的噪声（不可知）。论文将两个噪声分开建模。

然后，将建模的噪声集成到一个健壮的训练框架中，显着提高了图像伪造检测器的鲁棒性。

论文提出的方法相较于之前的方法更稳定，效果更好，最后，为了促进图像伪造检测的未来发展，论文基于现有的四个数据集和三个最流行的OSN构建了一个公共伪造数据集。

贡献

1、论文提出了一种新颖的训练方案，用于针对通过OSN传输的图像的伪造检测。训练方案不仅对OSN引入的可预测噪声进行建模，而且还结合了看不见的噪声，以进一步提高鲁棒性。

2、与几种最先进的方法相比，该模型实现了更好的检测性能，特别是在图像通过OSN传输的情况下。

3、论文基于四个已经存在的数据集和三个平台（脸书，微博和微信）构建了一个公共伪造数据集。

动机

以往的方法大都为特定的篡改设计，例如拼接，复制移动和内涂，而另一些则旨在识别更复杂或更复杂的伪造品。然而，很少有论文提出在OSN损失图像信息的情况下的检测模型。这种应对OSN损失的检测模型十分重要，以往OSN对图像执行的损失操作会严重降低普通检测器的性能。

为了应对上述挑战，论文的目标是设计一种强大的图像伪造检测方法，以击败OSN中的有损操作。具体来说，为了处理OSN退化，论文提出了一个噪声建模方案，并将模拟噪声集成到鲁棒的训练框架中。

How?

通过对脸书的图片传输实验，可以知道几种已知的常用对图片操作：

调整大小、增强滤镜和 JPEG 压缩。

论文的中心就是如何对噪声建模，如何模拟OSN对图像的损失操作，下图是主要的流程图。

Stage1和Stage2阶段致力于通过可微分的网络模拟可预测的噪声。Stage3通过对抗性噪声生成策略处理对看不见的噪声建模，Stage4对图像伪造检测器f_θ进行训练。

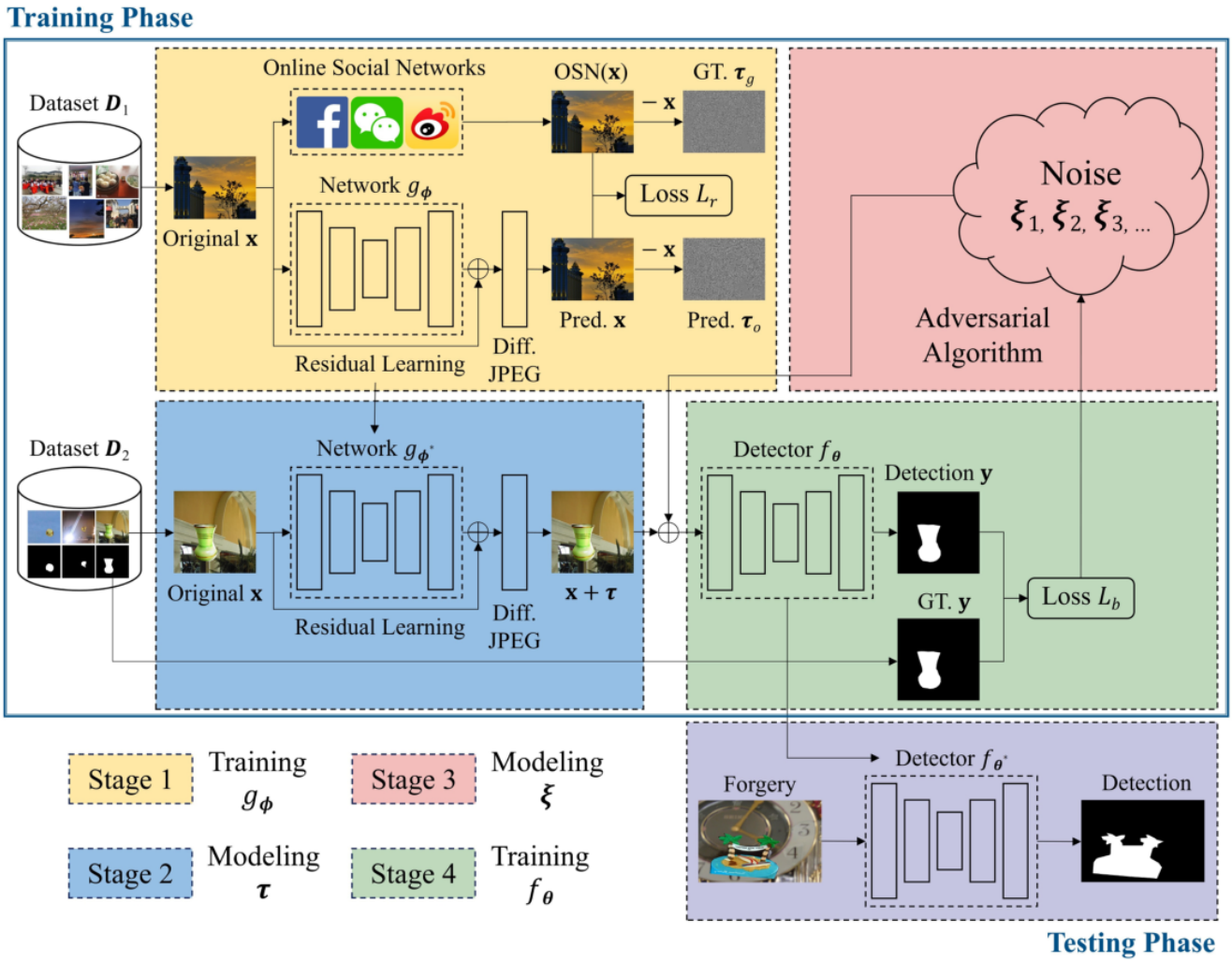


Figure 2. The overview of our proposed training scheme and the corresponding testing phase.

stage1 :训练g_ϕ

文章通过对社交网络图像退化操作的探讨得到了可预知噪声τ的主要来源是JPEG压缩的结论，其他后处理操作和可能出现的下采样都只对τ有部分影响。对于一个社交网络，产生的图像噪声可以表示为：

$$\tau_i = OSN(x_i) - x_i$$

可以从上式看出噪声是与信号有关的，τ是受到x影响的。

stage1有一张原图，对比输入OSN的结果OSN(x)和输入g_ϕ结果Pred.x，g_ϕ的模拟OSN网络的输出，采用残差优化模型，确保学习的噪声是OSN操作而不是图像本身带来的。训练g_ϕ的目标函数如下：

$$\min_{\phi}\left\{\mathcal{L}_r\left(x_i+g_\phi\left(\mathbf{x}_i\right),\mathrm{OSN}\left(\mathbf{x}_i\right)\right)\right\}$$

$\mathcal{L}_r(\mathbf{x},\mathbf{y})=\|\mathbf{x}-\mathbf{y}\|_2$.

因为要网络g_ϕ模拟OSN，需要一层J模拟压缩，在最后的输出结果上再嵌入可微分的代替压缩操作的J_q，将目标函数改成：

$$\min_{\phi}\left\{\mathcal{L}_r\left(J_q(x_i+g_\phi\left(\mathbf{x}_i\right)),\mathrm{OSN}\left(\mathbf{x}_i\right)\right)\right\}$$

这里的q代表JPEG压缩中的参数质量因子QF。

用于模拟噪声的模型g_ϕ采取U-net架构。

在Database D_1选取两张RGB三通道的原图（p1,p2）和一个binary mask y，取图p2的部分进行伪造（取决于binary mask y），其余部分取图p1的部分

$$\mathbf{x}=\mathbf{p}_1\odot(1-\mathbf{y})+\mathbf{p}_2\odot\mathbf{y}$$

这样就合成了伪造面孔x，检测器的目标是在给伪造面孔加入了噪声（两部分：OSN给图像所加的resize,压缩等已知噪声，还有未知的噪声）后还能被检测出。

τ代表可以预知的噪声，ξ是未知的噪声。最终添加的噪声是两者加和。

$$\boldsymbol{\delta}=\boldsymbol{\tau}+\boldsymbol{\xi}$$

目标函数如下：

$$\arg\min_{\boldsymbol{\theta}}\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{P(\boldsymbol{\tau})}\left\{\mathbb{E}_{P(\boldsymbol{\xi}|\boldsymbol{\tau})}\left\{\mathcal{L}_b\left(f_{\boldsymbol{\theta}}\left(\mathbf{x}_i+\boldsymbol{\tau}+\boldsymbol{\xi}\right),\mathbf{y}_i\right)\right\}\right\}$$

P_τ是可预期的噪声的分布，$P(\xi|\tau)$是条件分布。

stage2：得到 $P(\tau)$分布

在**OSN**不允许过量的上传和下载的情况下怎么获取噪声数据?

采用深度网络模拟，训练DNN模型g_ϕ，这个模型嵌入一个可微分的层来模拟压缩。

参见F2 stage1，第一步训练出一个好的模拟噪声τ的模型g_ϕ该模型被应用于第二部生成τ噪声的分布。

得到噪声τ，

$$\boldsymbol{\tau}_i(q) = \mathcal{J}_q\left(\mathbf{x}_i + g_{\phi^*}\left(\mathbf{x}_i\right)\right) - \mathbf{x}_i$$

，这里的的ϕ^*是通过目标函数的优化产生的。q 是与 JPEG 压缩关联的质量因子QF。对于给定的输入x_i，q是可以随着OSN的平台变化的。可以将可能的输出τ_i定义为：

$$\Omega_{\boldsymbol{\tau}_i} = \{\boldsymbol{\tau}_i\left(q_1\right), \boldsymbol{\tau}_i\left(q_2\right), \cdots\}$$

补充——q的取值：在噪声建模方案中，在训练期间对QF进行动态采样，从而模拟通用网络以模仿OSN平台的一般行为，或者，我们可以为每个单独的q训练一个特定的网络g_ϕ，以便更准确地模仿可预测的噪声。这个想法类似于一些现有的去噪网络，它们为每个可能的噪声水平训练一个网络。然而，我们通过实验发现，这种替代方案未能带来明显的改善; 同时，它大大增加了培训成本，并使其在实践中使用起来很麻烦。最终采用第一种动态采样的方式，对蒙特卡罗（MC）采样方案进行修改，以生成大量噪声样本，以模拟分布$P(\tau)$。

stage3：得到ξ的条件分布$P(\xi|\tau)$

在未知噪声的情况如何进行建模?

鉴于噪声是未知的，所以采用对τ的建模方法是不实际的，为了解决这一挑战，我们将位置从噪声方面转移到检测器f_θ，在看不见的噪声ξ

中，我们实际上只需要注意那些降低检测性能的噪声，而忽略了那些对检测影响不大的噪声。

从本质上讲，对抗性噪声通常对人类感官来说是不可察觉的，同时能够导致严重的模型输出错误。同时，我们关注的看不见的噪声ξ是能够愚弄探测器的噪声ξ，并且通常很小（高度失真的图像会偏离伪造的目的）。对抗噪声与该噪声具有很高的相似性，所以采取对抗噪声进行ξ建模。

引入对比噪声，从对比的角度看，可以定义噪声ξ为，如果对正常样本加入噪声可以对分类产生影响（即使得样本越过决策边界），那么就可以将该噪声定义为未知的第二类噪声。具体如下图：（其中虚线表示对抗噪声的几个可能方向）

注意到噪声ξ通常具有小振幅的事实，论文建议沿损失函数的输入的梯度来设置ξ，以最小化噪声能量（参见下图中的红虚线）。

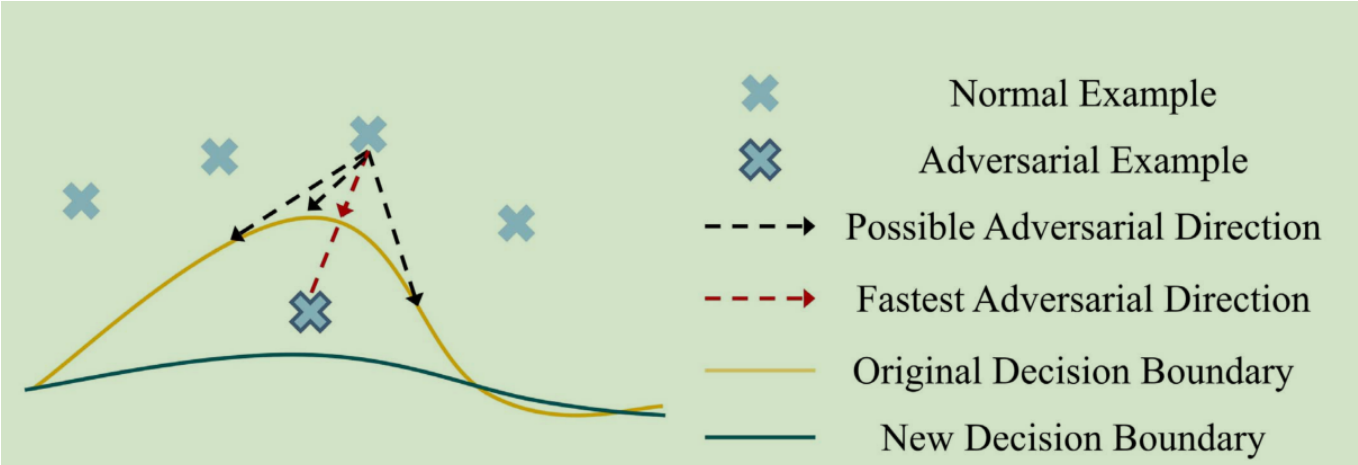


Fig. 5. An illustrative example of the learned decision boundary with or without the adversarial noise.

噪声ξ的扰动都很小，可以将其定义为检测器损失函数沿着x_i的梯度（也就是对x_i的偏导）。

$$\boldsymbol{\xi}_i = \mathcal{S}\left(\nabla_{\mathbf{x}_i}\mathcal{L}_b\left(f_{\boldsymbol{\theta}}\left(\mathbf{x}_i + \boldsymbol{\tau}_i\right), \mathbf{y}_i\right)\right)$$

$$\nabla_{\mathbf{x}_i}\mathcal{L}_b\left(f_{\boldsymbol{\theta}}\left(\mathbf{x}_i+\tau_i\right),\mathbf{y}_i\right)=\frac{\partial\mathcal{L}_b\left(f_{\boldsymbol{\theta}}\left(\mathbf{x}_i+\boldsymbol{\tau}_i\right),\mathbf{y}_i\right)}{\partial\mathbf{x}_i}$$

ξ其实就是损失函数相对于输入的偏导。S是梯度下降之意。通过在训练期间添加这种对抗性噪声，预计这将使学习的模型不仅能够抵抗特定的对抗性噪声，还可以抵抗更一般的看不见的噪声。

然而，由（9）计算的噪声取决于特定的输入x_i，而不适用于一般输入。为了全面增强探测器的泛化能力，建议将对抗噪声的方向调整为全局梯度方向。如果采用全局梯度，出现的另一个关键问题是如何以有效的方式准确计算全局梯度。为此，我们采用类似于随机梯度下降（SGD）的策略，也就是说对于在第t+1轮输入的图像x_{t+1}，ξ_{t+1}（以τ为条件）可以设置为从第一个 t 输入开始计算的平均梯度，即：

$$\xi_{t+1}=\frac{1}{t}\sum_{i=0}^{t}\mathcal{S}\left(\nabla_{\mathbf{x}_i}\mathcal{L}_b\left(f_{\boldsymbol{\theta}}\left(\mathbf{x}_i+\boldsymbol{\tau}_i+\boldsymbol{\xi}_i\right),\mathbf{y}_i\right)\right)$$

其中ξ_0初始化为0。虽然（10）式可用于估计平均梯度，但它仅反映特定已知数据（训练数据）的梯度，从而失去了普遍性。为了缓解上述问题并进一步提高鲁棒性，我们建议在小范围内扰动ξ_t。在这里，使用参数模型来表示平均梯度会更理想。

我们首先采用数据驱动的方法，分析从训练过程中随机选择的1000个ξ样本的统计数据。

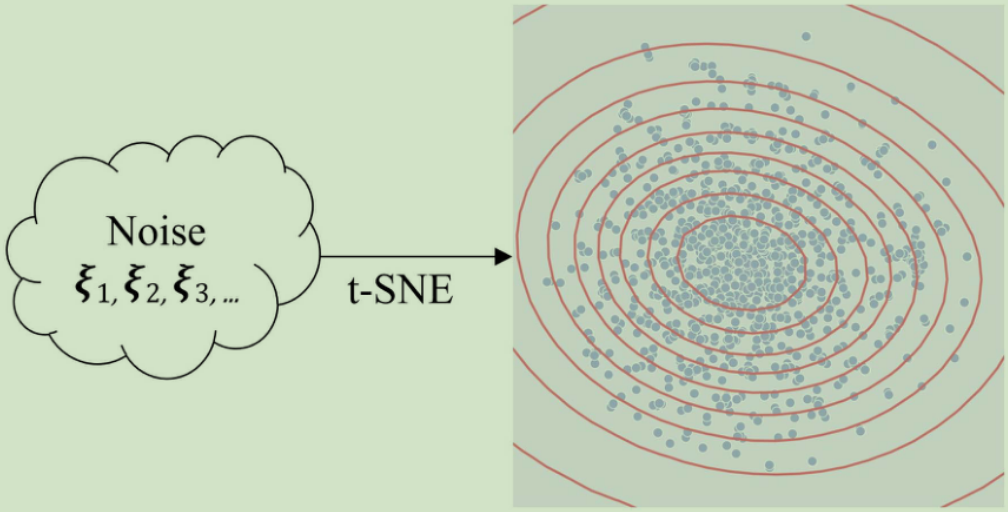


Fig. 6. Visualization of 1000 noise samples of $\boldsymbol{\xi}$ by using t-SNE [32].

对于1000个样本计算ξ时可以看出样本点都在某个特定中心点附近，如果样本离中心远，那么梯度就会消失。该现象说明可以采用高斯分布模拟平均梯度分布。

$$\boldsymbol{\xi}_{t+1} \mid \boldsymbol{\tau} \sim \mathcal{N}\left(\boldsymbol{u}_{t+1}, \sigma^2 \mathbf{I}\right)$$

中心（均值是）u，u如下所示：

$$\boldsymbol{u}_{t+1} = \epsilon \cdot \frac{1}{t} \sum_{i=0}^{t} \mathcal{S}\left(\nabla_{\mathbf{x}_i} \mathcal{L}_b\left(f_{\boldsymbol{\theta}}\left(\mathbf{x}_i + \boldsymbol{\tau}_i + \boldsymbol{\xi}_i\right), \mathbf{y}_i\right)\right)$$

ϵ是用于约束扰动大小以避免不必要的模型退化的参数。

鲁棒训练细节

模型最后的目标函数是：

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \sum_{j=1}^{m} \sum_{k=1}^{h} \mathcal{L}_b\left(f_{\boldsymbol{\theta}}\left(\mathbf{x}_i + \boldsymbol{\tau}_j + \boldsymbol{\xi}_k\right), \mathbf{y}_i\right)$$

Algorithm 1: The training algorithm

Input: Training datasets $\mathcal{D}_1$ and $\mathcal{D}_2$; training epochs N_1 and N_2; learning rates l_ϕ and l_θ.
Output: Trained detector f_{θ^*}

1 Randomly initialize ϕ and θ
2 **for** *epoch = 1 to* N_1 **do**
3 **for** *minibatch* $(\mathbf{x}_i, \mathbf{y}_i) \subset \mathcal{D}_2$ **do**
4 $\mathbf{g}_\phi = \nabla_\phi[\mathcal{L}_r(\mathcal{J}_q(\mathbf{x}_i + g_\phi(\mathbf{x}_i)), \mathbf{y}_i)]$ ▷ Eq. (9)
5 $\phi = \phi - l_\phi \cdot \mathbf{g}_\phi$ ▷ Update g_ϕ
6 **end**
7 **end**
8 Temporary output $g_{\phi^*} = g_\phi$
9 Initialize $\boldsymbol{u}_0 = \mathbf{0}$
10 **for** *epoch = 1 to* N_2 **do**
11 **for** *minibatch* $(\mathbf{x}_i, \mathbf{y}_i) \subset \mathcal{D}_2$ **do**
12 Initialize $\mathbf{L}_0 = \mathbf{0}$
13 **for** *j = 1 to m* **do**
14 $q_j \sim \text{Uniform}(71, 95)$ ▷ Sample QF
15 $\boldsymbol{\tau}_j = \mathcal{J}_{q_j}(\mathbf{x}_i + g_{\phi^*}(\mathbf{x}_i)) - \mathbf{x}_i$ ▷ Model $\boldsymbol{\tau}$
16 $\{\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_h\} \sim \mathcal{N}(\boldsymbol{u}_{i-1}, \sigma^2\mathbf{I})$ ▷ Model $\boldsymbol{\xi}$
17 $\mathbf{L}_j = \mathbf{L}_{j-1} + \sum_{k=1}^{h} \mathcal{L}_b(f_{\boldsymbol{\theta}}(\mathbf{x}_i + \boldsymbol{\tau}_j + \boldsymbol{\xi}_k), \mathbf{y}_i)$ ▷ Eq. (15)
18 **end**
19 $\mathbf{g}_{\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} \mathbf{L}_m, \mathbf{g}_{\mathbf{x}_i} = \nabla_{\mathbf{x}_i} \mathbf{L}_m$
20 $\boldsymbol{\theta} = \boldsymbol{\theta} - l_{\boldsymbol{\theta}} \cdot \mathbf{g}_{\boldsymbol{\theta}}$ ▷ Update $f_{\boldsymbol{\theta}}$
21 $\boldsymbol{u}_i = \boldsymbol{u}_{i-1} + \epsilon \cdot \mathcal{S}(\mathbf{g}_{\mathbf{x}_i})$ ▷ Eq. (14)
22 **end**
23 **end**
24 Final output $f_{\theta^*} = f_{\boldsymbol{\theta}}$

g_ϕ是模拟OSN输出结果的模型，梯度下降得到参数 ϕ

模拟未知噪声 ξ 的高斯分布的中心初始为0

$\mathcal{L}$是损失函数，三重循环模拟损失函数加和

每一轮计算完一组图像的损失函数，就更新检测器参数 θ 和高斯分布的中心u

在算法 1 中，第 2~7 行专门用于训练网络g_ϕ^*以估计用于第 15 行的可预测噪声 τ。第 16 行利用提出的对比噪声的方法对 τ 上看不见的噪声 ξ 条件进行建模。然后，在第 18 行中，计算最终目标函数，并在第 20~22 行中更新参数。最终，我们在第26行中生成经过训练的探测器 f_θ^*。

注：鲁棒的训练方案也可以看作是一种数据增强技术，其中注入的噪声数据根据OSN平台和探测器本身的特性进行了精心设计。众所周知，在许多情况下，数据增强带来的鲁棒性改进通常会导致原始探测器的性能下降。应该注意的是，当探测器的训练和测试基于相同的数据分布时，就会发生这种性能下降。然而，情况往往并非如此，因为测试数据可能来自截然不同的分布，从这个角度来看，考虑探测器的泛化是非常关键的，在这种情况下，数据增强通常具有积极作用。正如预期的那样，并会通过实验验证，我们提出的方案可以有效地提高对各种OSN平台传输的鲁棒性。

SGD梯度下降（随机梯度下降）

普通梯度下降算法有一些缺点。我们需要仔细研究我们为算法的每次迭代所做的计算量。

假设我们有 10000 个数据点和 10 个特征。我们需要计算目标函数相对于每个特征的导数，所以实际上我们每次迭代将做10000 *10* = *100000次计算，梯度下降通常需要 1000 次迭代，实际上我们有 100000* 1000 = 100000000 次计算来完成算法。这开销过大，因此在大量数据上梯度下降很慢。

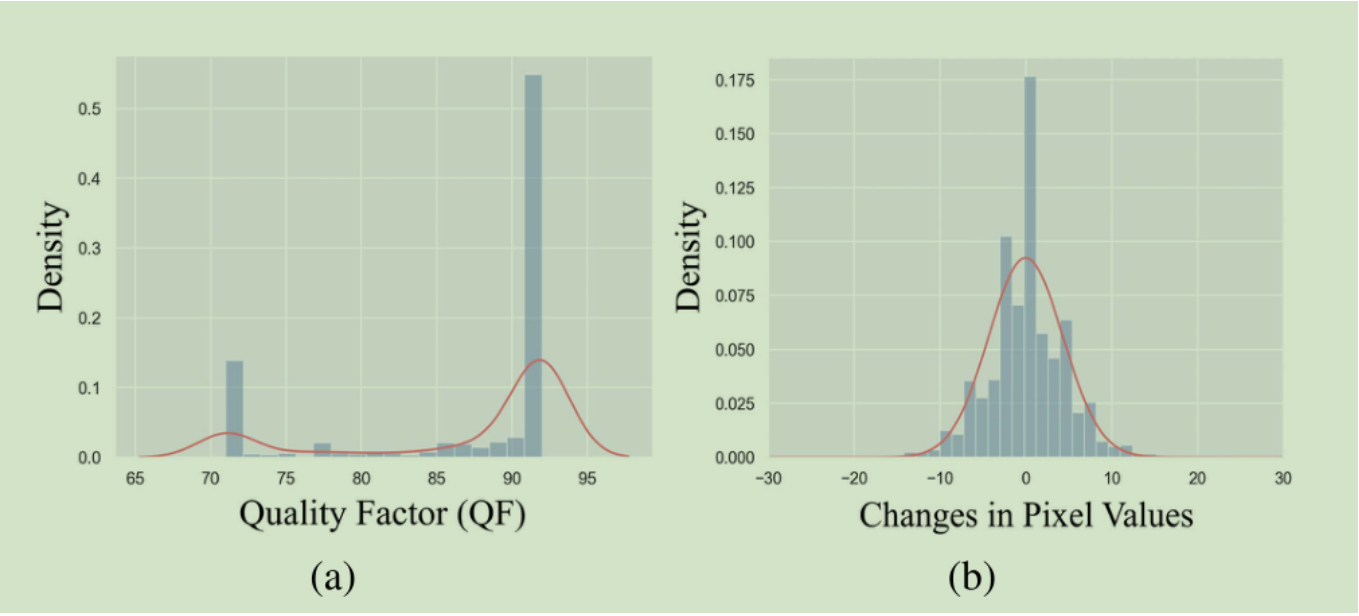
而SGD在每次迭代时从整个数据集中随机选择一个数据点进行计算，以大大减少计算量。梯度下降对少量数据点进行采样也很常见，每次迭代取一小部分数据点而非只取一个，这称为“小批量”梯度下降。小批量梯度下降在梯度下降的优良性和 SGD 的速度之间取得平衡。

论文细节补充

OSN是怎样处理图像的（以**facebook**为例）

一般有四步：格式转换、调整大小、增强过滤和 JPEG 压缩。具体来说，首先将上传的图像转换为像素域并且确保像素值在 [0, 255] 范围内。之后，如果图像的分辨率高于 2048 像素，则会调整图像大小，随后，图像中的一些选定块经过高度自适应和复杂的增强过滤，而由于是自适应的操作，增强过滤操作是难以预知的。最后，OSN对图像进行一轮JPEG压缩，并根据图像内容自适应地确定质量因子（QF）。

通过对facebook图像数据集的分析，Facebook使用的QF值范围为71到95，其中更详细的分布如下图(a)所示：

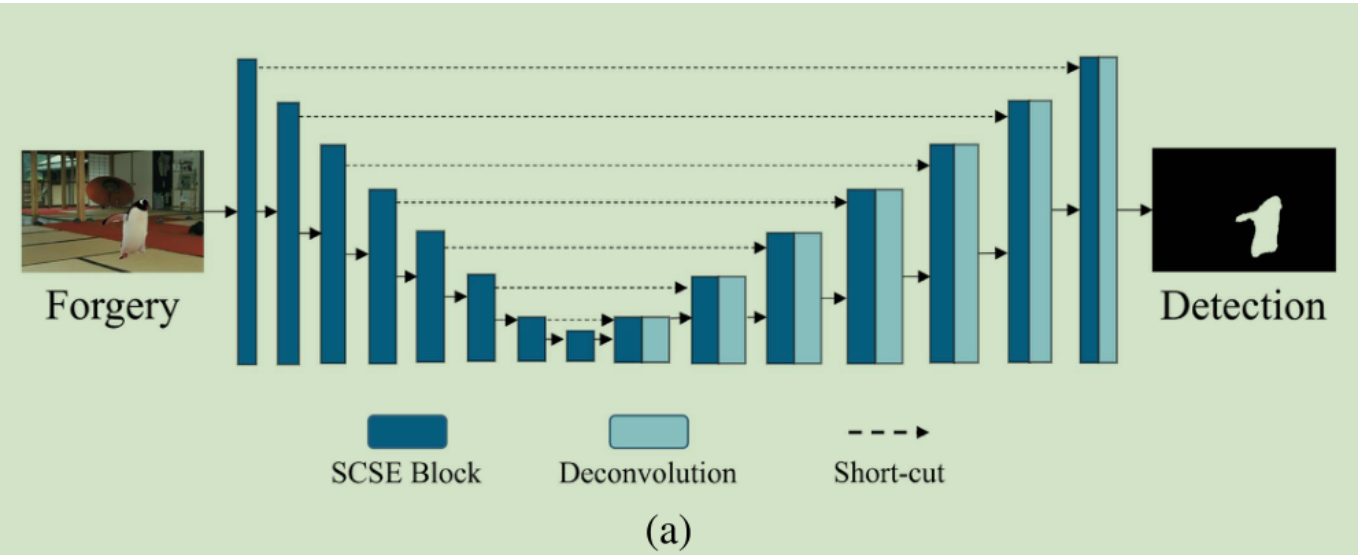


（b）显示了图像像素值的变化。

尽管不同 OSN 平台上的图像处理不同，但主流 OSN 进行的操作仍然有许多相似之处（例如，无处不在的 JPEG 压缩）。

基本图像伪造检测器

在讨论检测器的鲁棒性之前，这是整个方案的基础。检测网络旨在以像素级精度检测伪造区域。基本检测器的架构如下图(SE-U-Net)：



解释：检测器f_θ的输入是$H \times W \times 3$的图像，输出是检测结果的二值图。（0和1构成，代表是否伪造）。

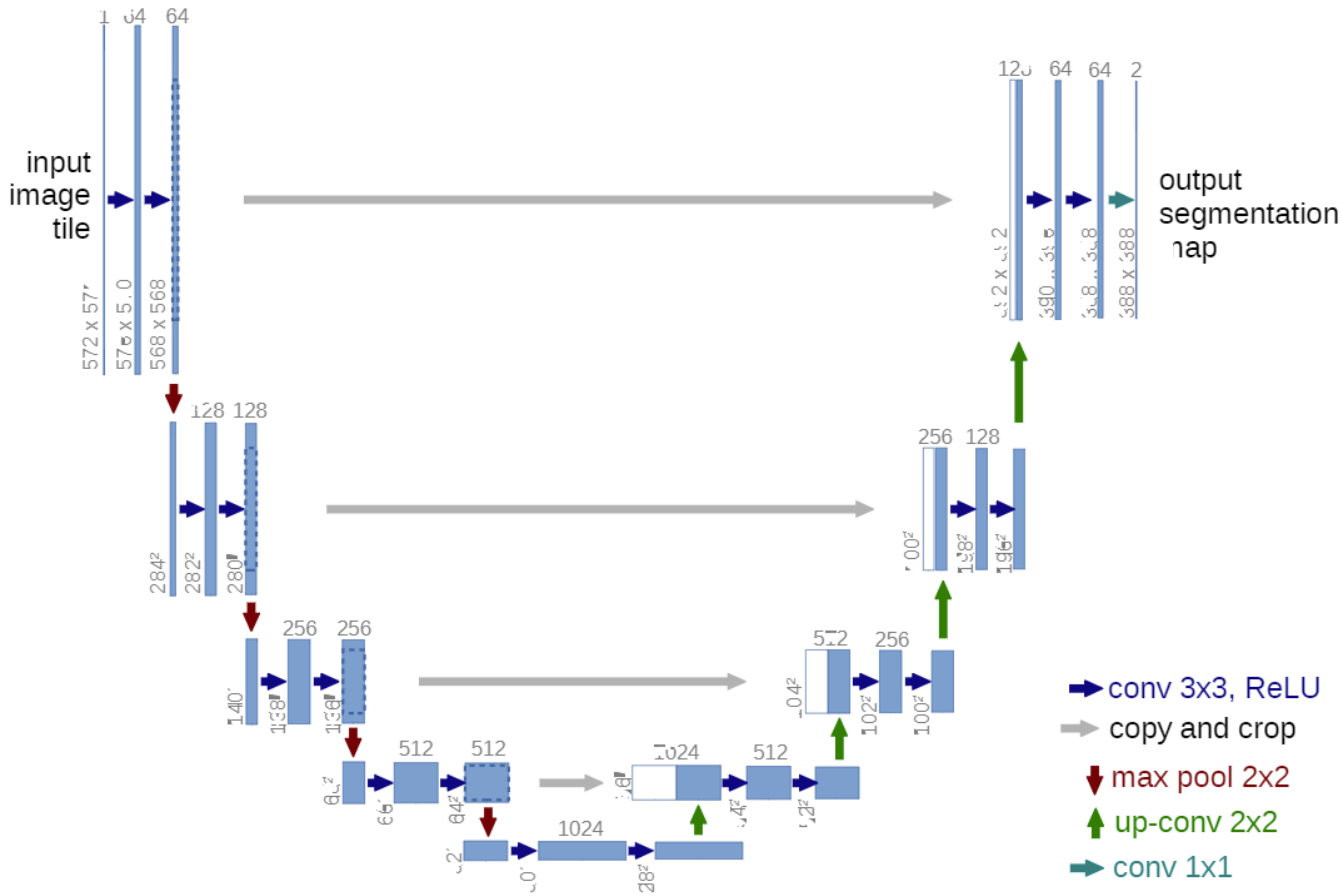
主要采用的架构：**U-NET**

U-net常用于图像的分类，与传统卷积神经网络不同，它是用于处理生物医学图像的网络，不仅输出标签（是否患病），还定位异常区域。U-net的整体结构如下图：

U-Net 由四个连续的编码器和四个对称的解码器组成，其中每个编码器包含重复的卷积层，ReLU 激活和最大池化操作。在编码阶段，不断缩小空间维度，提取更重要的特征信息，在解码阶段，通过重新调用从相应编码器学习到的特征作为额外的上下文信息，解码器可以更好地优化各种任务的结果。

关于U-Net：（参考论文：U-Net: Convolutional Networks for Biomedical Image Segmentation）

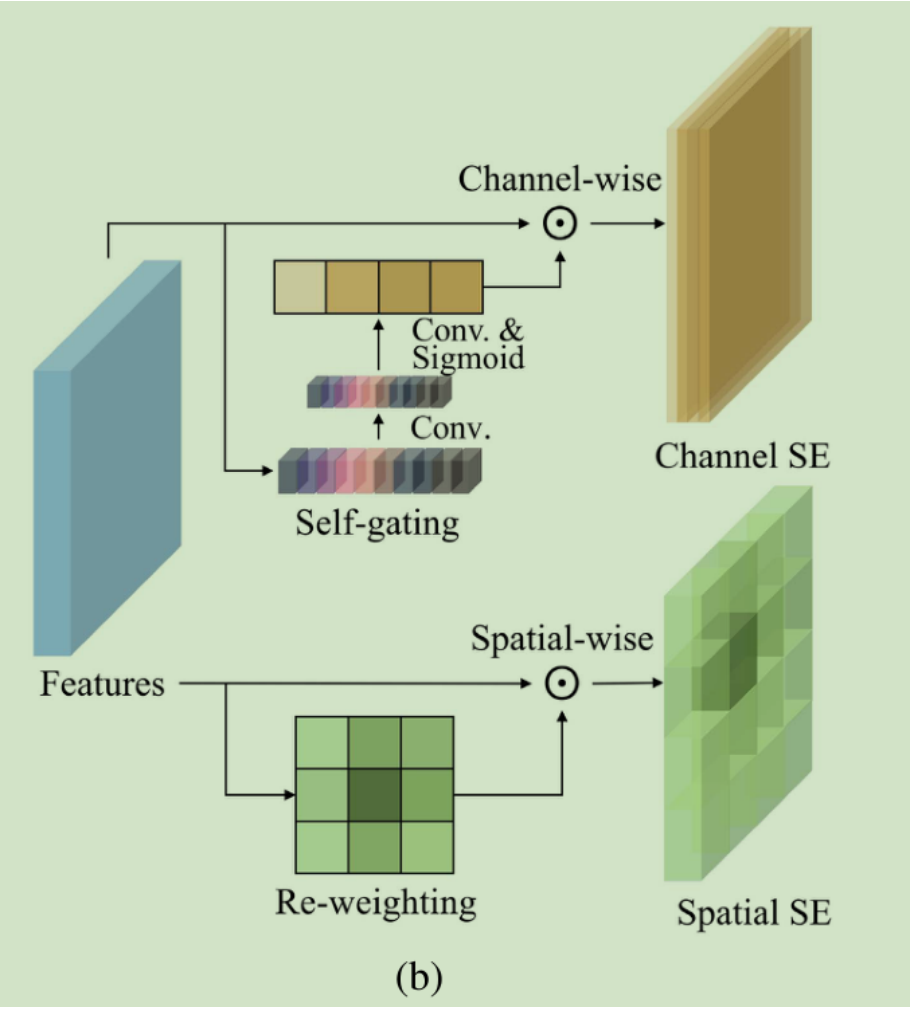
构造U-Net（可以进行伪造定位）



U-net 体系结构（例如，最低分辨率的 32$\times$32 像素）。每个蓝框对应一个多信道特征图。信道数在框顶部表示。左边进行一次卷积，通道数就翻一倍，右边正相反，进行一次转置卷积通道数就减半。

有人指出，标准卷积层通常学习的是表示输入图像内容的特征，而不是底层的伪造痕迹。为了提高提取伪造相关特征的能力，我们通过结合“空间信道挤压和激励”（SCSE）机制来进一步增强架构，而不是简单地使用传统的普通 U-Net。

下图的构造是SCSE模块：



改进过的SE-U-Net可以有选择地强调信息特征，同时抑制其余特征。具体而言，所使用的SCSE层由两个分支组成，每个分支分别在空间域和信道域中进行操作。对于一个初始的给定特征图

$$\mathbf{F} \in \mathbb{R}^{H \times W \times C}$$

，空间域操作实现生成一个更新加权的矩阵

$$\mathbf{S} \in \mathbb{R}^{H \times W}$$

通过操作：（⊗代表卷积操作，W_1代表卷积层参数）

$$\mathbf{S} = \text{Sigmoid}\left(\mathbf{W}_1 \otimes \mathbf{F}\right)$$

然后S和F相乘，实现自适应激励。并且生成的重新校准的空间要素用F_S表示:

$$\mathbf{F}_S = \text{Sigmoid}\left(\mathbf{W}_1 \otimes \mathbf{F}\right) \odot_s \mathbf{F}$$

$\odot_s$是空间上乘法操作（spatial-wise multiplication）。

另一支在信道域实现操作:

首先是通过self-gating操作生成中间矢量$$
\mathbf{v} \in \mathbb{R}^{1 \times 1 \times C}

，v代表了各个向量的权重信息， v是如何生成的？ #### SE模块： 该模块

\mathbf{v}^*=\operatorname{Sigmoid}\left(\mathbf{W}_2 \otimes \operatorname{ReLU}\left(\mathbf{W}_3 \otimes \mathbf{v}\right)\right)

包含信道特征的特征图$$F_C$$由下式获得：

\mathbf{F}_C=\mathbf{v}^* \odot_c \mathbf{F}

$\odot_c$代表信道乘法（channel-wise multiplication）。 还应该强调