

Reconstructed Student-Teacher and Discriminative Networks for Anomaly Detection

Shinji Yamada¹, Satoshi Kamiya² and Kazuhiro Hotta³

摘要

异常检测是计算机视觉中的一个重要问题; 然而, 异常样本的稀缺使得这项任务很困难。因此, 最近的异常检测方法只使用没有异常区域的“正常图像”进行训练。提出了一种基于学生-教师特征金字塔匹配 (student-teacher feature pyramid matching, S TPM) 的异常检测方法, 该方法由学生和教师网络组成。生成模型是异常检测的另一种方法。它们从输入中重建正常图像, 并计算预测的正常图像与输入之间的差异。不幸的是, S TPM 没有生成正常图像的能力。为了提高 S TPM 的准确性, 本文使用学生网络, 就像在生成模型中一样, 来重建正常的特征。然而, 由于 S TPM 没有使用异常图像进行训练, 正常图像的异常图不干净, 降低了图像级异常检测的准确性。为了进一步提高准确率, 该方法使用了一个由异常图中的伪异常训练的判别网络, 该网络由两对学生-教师网络和一个判别网络组成。该方法在 MVTec 异常检测数据集上表现出较高的准确率。

1 引言

异常检测是对数据中偏离正常模式的样本进行识别。近年来, 卷积神经网络 (convolutional neural networks, CNN) 被应用于异常检测, 用于工业品的视觉检查, 定位入侵者的监视, 以及医学图像的病理诊断。以往研究了将每张图像归类为异常的图像级异常检测方法, 但图像级异常检测方法并不适用于需要在像素级进行异常检测的产品检验。著名的图像级异常检测方法依赖于生成模型准确地重建正常图像。基于生成对抗网络 (generative adversarial network, GAN) 和自动编码器的生成模型仅使用正常图像进行训练, 虽然由于模型仅使用正常图像进行训练, 无法重建异常区域, 但可以使用正常区域进行重建。使用生成模型的异常检测通过可视化重构能力差的区域来识别异常区域。然而, 这些方法难以识别异常区域, 除非生成的正常区域能够高精度重建。

使用预训练模型作为一种新的异常检测方法被提出, 如 S TPM 使用学生-教师网络。在 S TPM 中, 教师网络是预训练的 ResNet18, 学生网络是未训练的 ResNet18。S TPM 只使用正常图像进行训练。学生网络学习, 从而制作类似于教师网络中的特征图。由于训练只在正常图像上进行, 因此学生网络只能输出正常区域的特征。相比之下, 教师网络是在 ImageNet 上预训练的模型, 因此可以很好地表示异常区域的特征。学生网络和教师网络在特征表示上的差异在于异常区域。在 student ResNet18 中, S TPM 使用三种不同分辨率的异常图来检测异常区域。然而, stpm 有几个限制。S TPM 在学生和教师网络中使用三种不同分辨率的特征图之间的差异。在验证阶段, 通过将三个图相乘计算出最终的异常图。当 3 张异常图都检测到同一个异常区域时, 就可以进行高精度的异常检测。但是, 如果 3 个特征图中有一个未能检测到异常区域, 则由于 3 个特征图相乘, 无法检测到异常区域。

因此, 三个异常图都必须进行改进。观察到 S TPM 不具有像生成模型那样重建正常图像的能力,

而生成模型在异常检测中是有效的，我们引入了一个新的学生网络，该网络具有重建正常特征的能力。传统的 STPM 使用 ResNet18 作为教师网络。如果同一个教师网络用于同一个学生。有可能得到类似的异常图。为获取不同视角下的异常图，将预训练好的 ResNet50 作为教师网络用于学生网络的重建。然而，由于学生网络的架构是 ResNet18，对于不同结构的学生-教师对的知识蒸馏比在具有相同结构的 STPM 中学习学生-教师对更具挑战性。因此，在提出的方法中，利用注意力机制将教师网络中的一些特征传播到学生网络中进行学习。由于所提方法只使用正常图像进行训练，因此注意力机制以更高的精度重建正常区域。通过注意力机制传递提示来确保知识蒸馏。

使用师生对进行重构虽然提高了异常检测的准确率，但加入了判别网络，进一步提高了异常检测的准确率。由于知识蒸馏的异常检测仅使用正常图像进行训练，因此无法区分产品是否为异常。因此，有些情况下异常图中正常区域的值较大，如图 1 所示。异常图可以检测异常区域，但也可以将正常区域检测为异常。因此，将 STPM 通过两对学生-教师网络得到的异常图输入到判别网络中，在将伪异常图像输入 STPM 时重新考虑异常图。因此，判别网络学会了产生更准确的异常图。通过将判别网络得到的异常图与带有两对网络的 STPM 得到的异常图相乘，提高了异常检测的准确率。

2 相关工作

2.1 异常检测

异常检测有两种类型：图像级和像素级。图像级异常检测的目标是对异常样本进行正确分类。图像级异常检测有三种方法：生成模型、特征空间等数据分布和分类。生成模型基于重建损失程度检测异常。基于分布的方法将偏离正态分布的样本视为异常。当只生成一个正常产品的概率分布时，异常产品的概率密度低，允许对异常数据进行分类。基于分类的异常检测方法是一种将几何变换和分类相结合的异常检测方法。基于对未知异常数据分类准确率较低的思想，对异常样本进行检测。然而，这些方法不适用于像素级别的异常检测。

像素级异常检测是一种用于检测每个像素上的异常的方法。由于检测目标数量的增加，像素级异常检测的难度要高于图像级异常检测的难度。在像素级别检测异常的主流方法是基于生成模型^{[1][2][3][4]}，如 GAN 和自编码器。近年来，结合 GAN 和自编码器的方法被研究^{[5][6][7]}。生成模型必须以较高的精度重建正常图像，否则会降低异常检测的准确率。

提出了一种新的像素级异常检测方法 SPADE。SPADE 基于预训练模型比较正常和异常图像的特征，采用 k-means 聚类检测异常。除了预训练模型外，还提出了基于知识蒸馏的无信息学生模型。学生仅使用正常数据训练学生网络。然后从两个角度计算异常图：(i) 学生网络和教师网络输出的差异；(ii) 多个学生网络的不确定性。STPM 是一种同时培养有知识和无知识学生的方法。在 STPM 中，因为只在正常数据上进行训练，所以学生网络只表示正常区域的特征。相比之下，教师网络是在 ImageNet 上进行预训练的，因此可以表示异常区域的特征。学生网络和教师网络在特征表示上的差异在于异常区域。STPM 对三种不同分辨率的特征图使用知识蒸馏，并输出三种异常图。将 3 个异常图相乘得到最终的异常图。但是，如果三个异常图中有一个无法检测到异常区域，则最终的异常图无法检测到异常区域，如图 2 所示。为了解决这个问题，使用新的学生网络对每个异常图进行改进。

2.2 注意力机制

各种注意力机制被提出用于图像识别。残差注意力网络使用类似于残差块的结构来解决梯度消失问题。挤压激励网络 (SENet) 引入了一种强调特征图中重要通道的注意力机制。提出了一种仅使用注意力机制的 transformer 语言翻译模型。也提出了几种使用自注意力 (self-attention) 的图像识别方法。注意力分支网络通过聚合多个特征图提出用于分类的注意力图。注意力图可以用来可视化决策的基础。

在该方法中，注意机制的目的是将教师网络的特征泄露给学生网络，从而有效地重构正常数据的特征。因为异常必须在像素级检测到，所以一个能够强调和抑制像素的注意机制比强调像 SENet 这样的通道的注意机制更适合。如果教师网络中的几乎所有特性都泄漏到学生网络中，那么学生网络和教师网络将没有区别。因此，通过将教师网络中的特征聚合到一个通道中，可以在不提供全部信息的情况下对像素进行强调和抑制。因此，我们使用从教师网络生成的注意图。由于对学生网络的训练只使用常态数据，因此利用注意机制重构常态区域。

3 本文方法

3.1 本文方法概述

本文是对 STPM 网络进行重构，使用两对 student-teacher 进行综合判别异常区域，如图 2 所示，是整个网络的结构，其中 student1-teacher1 为原来的 STPM 的网络结构，而 student2-teacher2 文章提出的新的体系结构，总共两对组成 student-teacher 网络。在网络架构方面，student1 和 teacher1 使用预训练的 ResNet18 进行多尺度特征提取，而 teacher2 使用的是预训练的 ResNet50 进行特征提取，student2 是类似于 ResNet18 的重构子网络，对 teacher1 输出的高层特征进行特征的重构。

student-teacher 进行异常检测的原理可以概述为，在训练过程中，仅仅使用正常图片，两对 student-teacher 网络进行知识蒸馏，让 student 网络进行学习正常图像的特征，而在测试的时候则使用异常图片，因为无论 teacher1 网络还是 teacher2 网络，都是经过预训练的，可以认为他们是认识异常的，但是 student 在训练过程中没有见过异常，因而在测试过程中，student 和 teacher 在异常的表示上就会产生差异，而正常区域因为 student 在训练过程中已经见到过了，所以能被很好地进行重构，最终通过在 student 与 teacher 网络上各自对应层产生的多尺度特征图进行像素级别的对比，产生差异比较大的部分就认为是异常区域。其中知识蒸馏过程是通过两对 student-teacher 网络层与层提取出来的特征进行像素对像素的匹配完成的。

使用第二对 student2-teacher2 的原因是，原来的 STPM，也就是图中的 student1-teacher1 网络中，仅仅使用的是正常图像进行训练，并没有见过异常图片，因而会产生对于异常区域可能会与 teacher 混淆以及对正常区域判别失败的问题，从而导致模型精确度降低，如图 1。提出的重构的 student2-teacher2 从不同的角度进行看待问题，对 student1-teacher1 进行补充，最终提高精度。除此以外，为了进一步提高精确度，使用类似于 U-Net 的判别网络对异常进行学习，降低对正常区域的误判，从而提高精确度。

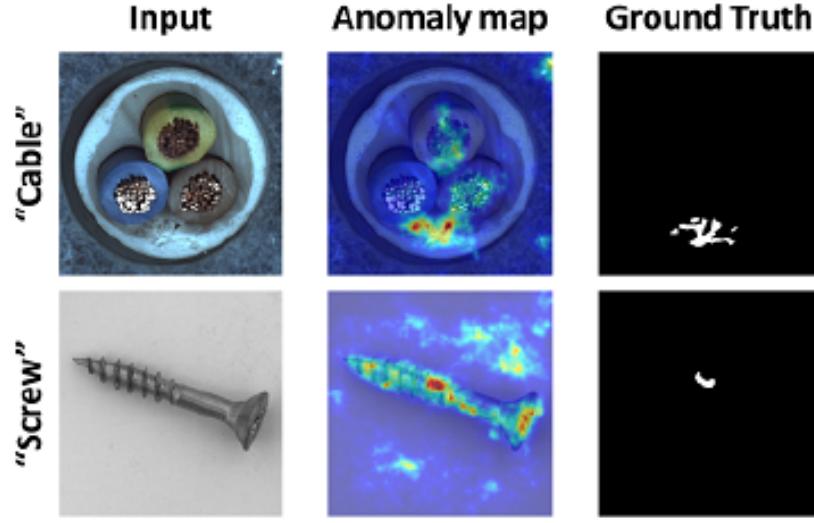


Fig. 1. The problem of STPM-based methods. The anomaly map has large values for both abnormal and normal regions.

图 1: 异常检测

对判别网络的训练使用到了伪异常，并且判别网络是在两对 student-teacher 网络训练完成后再进行训练的，通过特定方法进行伪异常生成后，输入到训练好的两对 student-teacher 网络，两对 student-teacher 网络总共产生 6 个异常图，然后将 6 个异常图中，student1-teacher1 和 student2-teacher2 中对应层的特征图进行相加作为该层提取的结果，最终得到 3 个最终整合的特征图。对于判别网络的输入，使用的是产生的三个特征图在通道上的拼接，然后输入到判别网络之中不断进行训练。

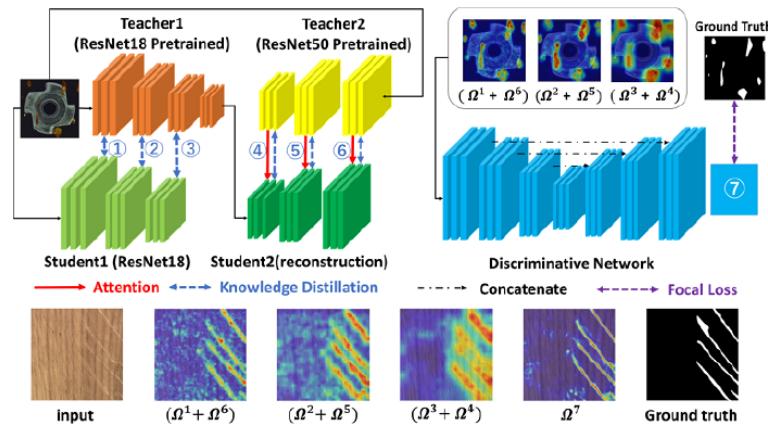


Fig. 6. Overview of the proposed network. The discriminant network uses pseudo-anomalies to reconsider the anomaly map obtained from STPM with two pairs of student-teacher networks. In the test phase, the anomaly map from the STPM is multiplied by that from the discriminant network.

图 2: 整体结构图

3.2 重构子网络

如图 3，是重构子网络的网络架构，类似于 ResNet18，通过 3×3 的卷积核以及残差层逐步对特征进行重构。并在相应位置对输入进行知识蒸馏以及注意力机制的添加。

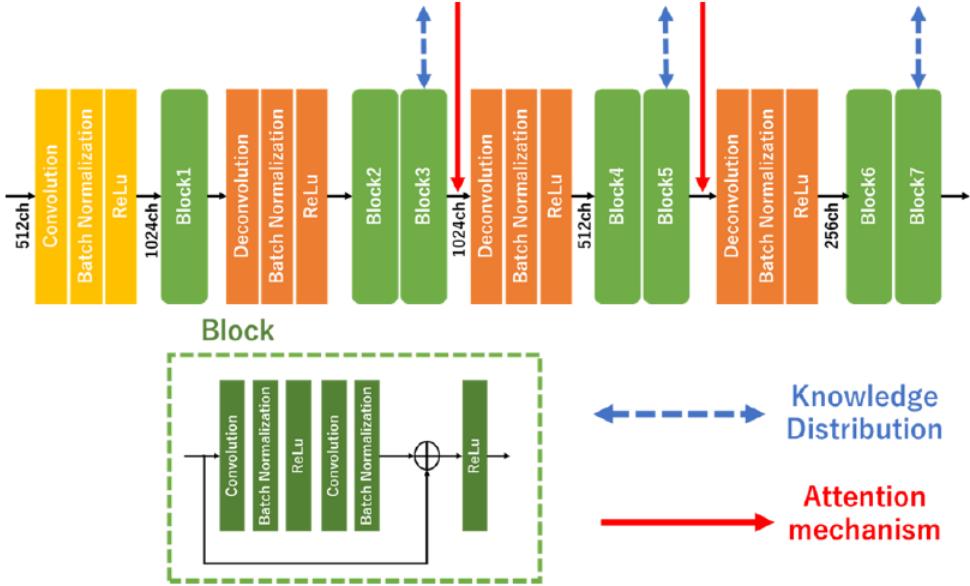


Figure 3: Structure of a student2 network

图 3: student2 的网络结构

3.3 注意力机制

如图 4 文章所使用注意力机制，注意力机制的作用在于，teacher2 与 student2 之间结构不同，可能会产生特征学习的差异，并且在训练过程中，网络仅使用正常图片，因而使用注意力机制让网络更多关注于正常纹理，而不是异常区域，进而突出 student2 与 teacher2 在异常之间的差异，提高异常检测的精确度，因为关注于像素级别的异常，而不是关注于通道的重要性，它输入的是 teacher2 对应层的多通道具体特征，但最终注意力机制会将输出单通道的注意力图。通过如图注意力机制的结构中 3×3 的卷积核网络实现特征的聚集，然后再通过 1×1 的卷积核对通道进行聚集，最终得到单通道注意力图。



Fig. 4. Attention mechanism in the proposed method. The student network can emphasize the important pixels for reconstruction.

图 4: 注意力机制

3.4 异常图生成

两对 student-teacher 网络通过层与层之间的特征图匹配，最终得到产生 6 个异常图，如图 5 所示，其中 Ω^1 对应产生 ① 的异常图，其他以此类推。

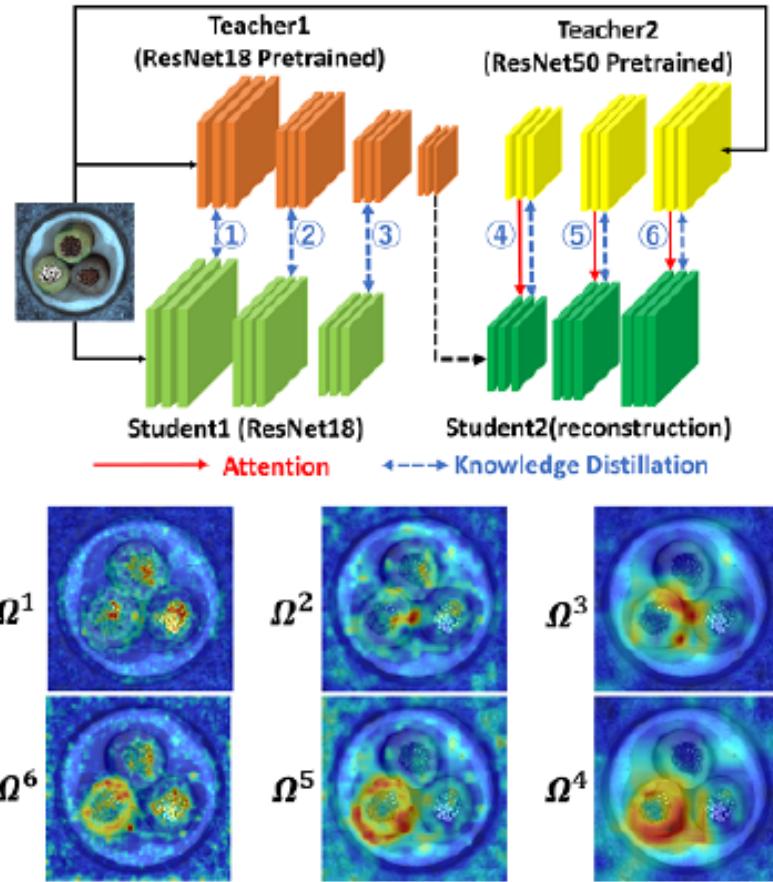


Fig. 3. Overview of STPM with two pairs of student-teacher networks.

图 5: 六层特征图

对于产生的 6 个特征图，最终会将两个网络中对应层的特征图进行相加，进行信息的互补，突出异常，进而获取不同层的特征图，如图 6。

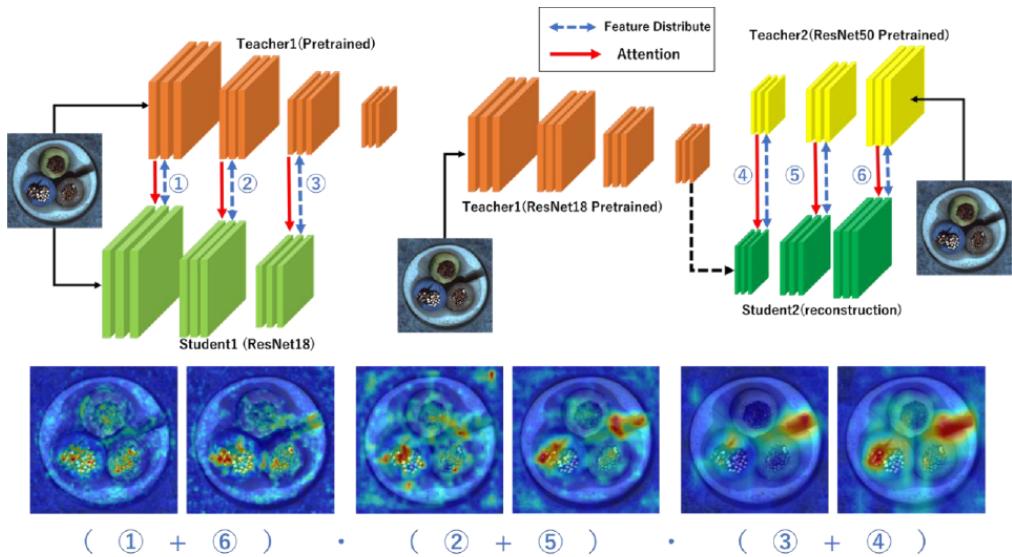


Figure 1: Overview of the proposed method

图 6: 特征图处理

上述过程中，将 6 个异常图对应层进行相加会得到 3 个不同维度的异常图，在训练以及测试判别

网络的时候会将这些异常图输入到判别网络当中，最终判别网络也会产生一个异常图，为 Ω^7 ，如图 2。最终的异常图生成公式如下。其中 $\Omega(J)$ 为最终输出的异常图， J 为输入的图像。

$$\Omega(J) = \{\Omega^1(J) + \Omega^6(J)\} \odot \{\Omega^2(J) + \Omega^5(J)\} \odot \{\Omega^3(J) + \Omega^4(J)\} \odot \Omega^7 \quad (1)$$

3.5 伪异常生成

在训练判别网络的时候，会使用到伪异常，如图 7，是生成为为异常的整体过程，其中使用的是柏林噪声，因为柏林噪声生成的噪声带有随机性质的同时产生连续的块，常用于生成游戏地图的生成。

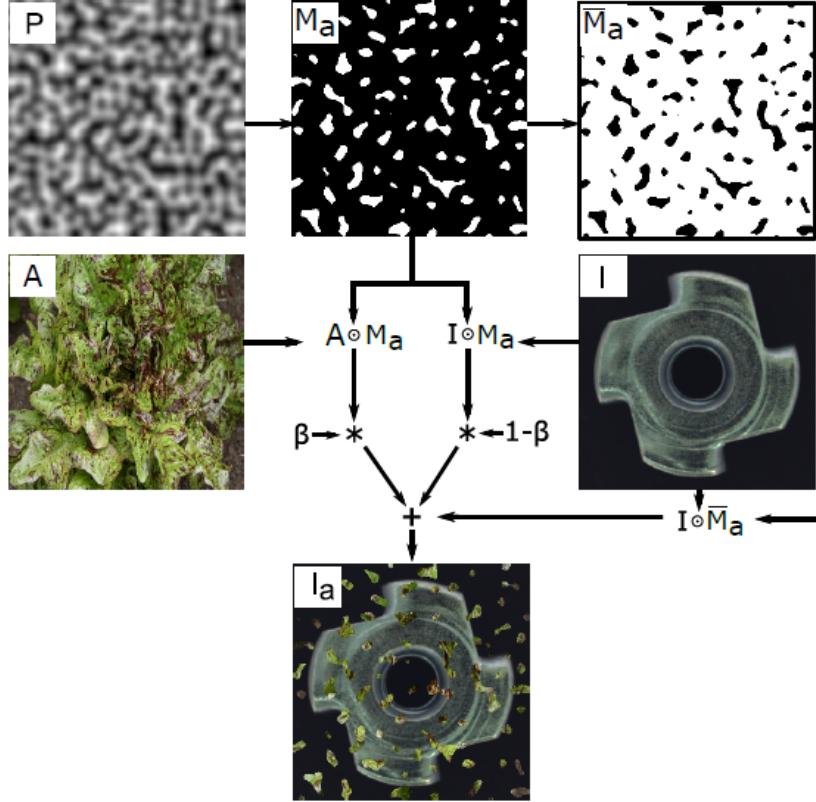


Figure 4. Simulated anomaly generation process. The binary anomaly mask M_a is generated from Perlin noise P . The anomalous regions are sampled from A according to M_a and placed on the anomaly free image I to generate the anomalous image I_a .

图 7: 伪异常生成

通过柏林噪声发生器进行生成随机噪声 P ，然后再通过均值二值化生成异常图 M_a ，通过对异常图 M_a 取反得到 \bar{M}_a ，整体生成过程可以表达为如下公式，其中 β 表示的是透明度，在透明度参数的影响下，让异常生成更显得真实。

$$I_a = \bar{M}_a \odot I + (1 - \beta)(M_a \odot I) + \beta(M_a \odot A) \quad (2)$$

其中异常源图片为 DTD 数据库，而无异常图片来源于 MVTec 数据库，并且对于异常源数据库会进行三种随机的数据增强策略，其中数据增强的总集包括分隔，锐度，日光化，均匀化，亮度变化，颜色变化，自动对比度，从其中抽选三种。如图 8为增强后的效果。

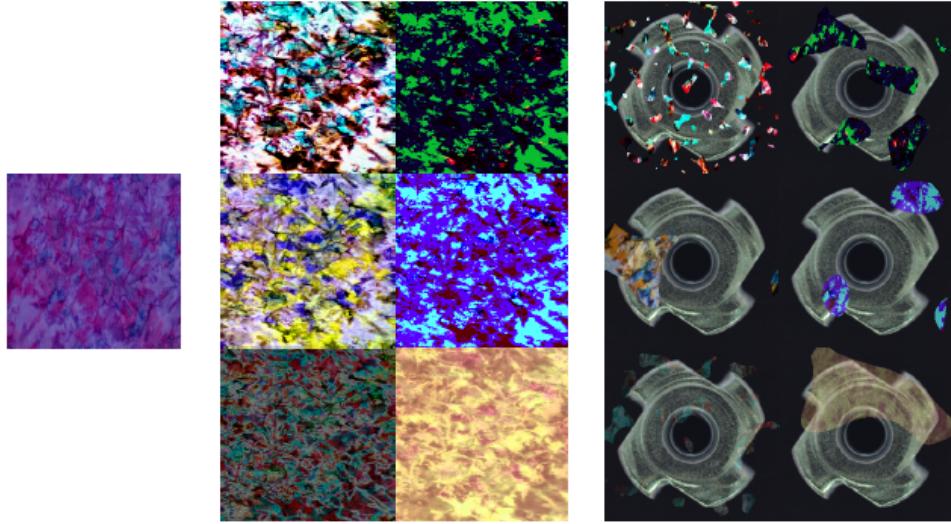


Figure 5. The original anomaly source image (left) can be augmented several times (center) to generate a wide variety of simulated anomalous regions (right).

图 8: 异常源数据增强

3.6 损失函数定义

损失函数包含两对 student-teacher 之间多尺度特征图的匹配损失以及判别网络的重构损失。对于 student-teacher 中损失函数首先需要对特征进行归一化。

$$F_t^l(\hat{I}_k)_{ij} = \frac{F_t^l(I_k)_{ij}}{\|F_t^l(I_k)_{ij}\|_{l_2}^2}, F_t^l(\hat{I}_k)_{ij} = \frac{F_t^l(I_k)_{ij}}{\|F_t^l(I_k)_{ij}\|_{l_2}^2} \quad (3)$$

上述公式(3)中参数 l 为第 l 层的特征， i 和 j 分别为对应的像素的位置，最终完成对输出的某一层的特征图进行 l_2 范数的归一化。在经过 l_2 范数归一化以后，在进行计算特征图之间的 l_2 距离，如下公式(4)

$$l^2(I_k)_{ij} = \frac{1}{2} \|\hat{F}_t^l(I_k)_{ij} - \hat{F}_t^l(I_k)_{ij}\|_{l_2}^2 \quad (4)$$

经过 l_2 范数归一化后，特征图上每个像素点的像素值范围是 $[0, 1]$ 。通过上述操作以后，得到第 l 层损失如公式(5)。

$$l^l(I_k) = \frac{1}{w_l h_l} \sum_{i=1}^{w_l} \sum_{j=1}^{h_l} l^2(I_k)_{ij} \quad (5)$$

全部层的总损失如公式(6)，其中 L 为总层数。

$$l^l(I_k) = \sum_{l=1}^L l^l(I_k) \quad (6)$$

4 复现细节与改进

4.1 与已有开源代码对比

复现论文没有现成的代码，复现过程中主要是依据作者的逻辑以及文章参考文献中模型相应部分所涉及的参考文献进行复现，参考论文包括^{[8][9][10][11]}。

4.2 参数配置说明

所有的图片都统一转换成分辨率 为 256×256 , 两对 student-teacher 模型训练 epochs 为 100, 使用 SGD 优化器, momentum 为 0.9, learning rate 为 0.4, batch size 为 32, weight decay 为 0.0001, 判别网络训练 epochs 为 300, 使用 Adam 优化器, momentum 为 0.9, learning rate 为 0.0001, batch size 为 32, weight decay 为 0.0001。

4.3 对比实验与论文结果

如图 9 是论文中进行测试的结果, 其中 PdDim、DREAM、STPM 是在与复现代码在同一硬件环境中进行测试而来, 而 Ours(S-T)、Ours(Dis)、Ours 是论文中给出的分别关于两对 student-teacher、仅判别网络、整体网络的结果。

	PaDim	DREAM	STPM	Ours(S-T)	Ours(Dis)	Ours
bottle	0.998 / 0.982	0.992 / 0.987	1.0 / 0.987	1.0 / 0.99	1.0 / 0.993	1.0 / 0.993
cable	0.922 / 0.968	0.918 / 0.955	0.94 / 0.955	0.967 / 0.973	0.989 / 0.979	0.996 / 0.983
capsule	0.915 / 0.986	0.985 / 0.984	0.841 / 0.984	0.873 / 0.985	0.945 / 0.972	0.93 / 0.985
carpet	0.999 / 0.99	0.97 / 0.991	0.986 / 0.991	0.981 / 0.988	0.964 / 0.985	0.987 / 0.992
grid	0.957 / 0.965	0.999 / 0.992	0.999 / 0.992	0.984 / 0.994	1.0 / 0.996	1.0 / 0.996
hazelnut	0.934 / 0.979	1.0 / 0.987	1.0 / 0.987	1.0 / 0.991	0.987 / 0.995	0.998 / 0.995
leather	1.0 / 0.99	1.0 / 0.995	1.0 / 0.995	0.998 / 0.984	1.0 / 0.996	1.0 / 0.996
metal_nut	0.992 / 0.971	0.987 / 0.972	0.999 / 0.972	1.0 / 0.982	0.999 / 0.989	1.0 / 0.989
pill	0.944 / 0.961	0.989 / 0.979	0.966 / 0.979	0.967 / 0.972	0.98 / 0.99	0.981 / 0.987
screw	0.844 / 0.983	0.939 / 0.988	0.92 / 0.988	0.948 / 0.993	0.941 / 0.989	0.968 / 0.993
tile	0.974 / 0.939	0.996 / 0.967	0.969 / 0.967	0.953 / 0.97	0.986 / 0.987	0.999 / 0.988
toothbrush	0.972 / 0.987	1.0 / 0.987	0.85 / 0.987	0.9 / 0.989	0.987 / 0.994	0.979 / 0.993
transistor	0.978 / 0.975	0.931 / 0.82	0.927 / 0.82	0.975 / 0.898	0.978 / 0.872	0.983 / 0.907
wood	0.988 / 0.941	0.991 / 0.956	0.992 / 0.956	0.992 / 0.97	0.966 / 0.978	0.993 / 0.981
zipper	0.909 / 0.984	1.0 / 0.984	0.942 / 0.984	0.898 / 0.985	0.998 / 0.988	0.993 / 0.992
mean	0.955 / 0.973	0.98 / 0.97	0.955 / 0.97	0.962 / 0.977	0.981 / 0.962	0.987 / 0.977

图 9: 对比实验与论文结果 (Image -Level AUC / Pixel-level AUC)

4.4 损失函数

对于两对 student-teacher 的损失函数, 文章给出了如下的定义, 它包括了 6 个层中不同尺度特征图之间的特征的损失, 但如图如图 5 所示, teacher1 是预训练的 ResNet18, 以及 teacher2 是预训练的 ResNet50 模型, 论文给出的总损失是六个特征提取层中的匹配损失之和。

但是在按照逻辑而言, student2 的损失函数事实上跟 student1 是不完全关联的, 因为 student2 的输入是 teacher1 的提取结果以及是与 teacher2 之间的特征匹配都是一些已经确定的条件, 那么处于这方面的思考, 考虑是否使用论文的总损失函数以及统一的优化器还是使用各自的损失以及各自的优化器展开实验, 如图 10 是二者的实验情况对比。其中 All-loss(S-T) 是两对 student-teacher 网络使用总共的损失, Split-loss(S-T) 是使用各自的损失以及各自的优化器。

	All-loss(S-T)	Split-loss(S-T)
bottle	1.0 / 0.988	1.0 / 0.988
cable	0.995 / 0.979	0.978 / 0.976
capsule	0.931 / 0.988	0.85 / 0.985
carpet	0.992 / 0.992	0.99 / 0.992
grid	0.987 / 0.994	0.987 / 0.983
hazelnut	1.0 / 0.99	1.0 / 0.99
leather	0.67 / 0.918	1.0 / 0.995
metal_nut	1.0 / 0.98	1.0 / 0.982
pill	0.961 / 0.973	0.916 / 0.959
screw	0.963 / 0.992	0.947 / 0.994
tile	0.965 / 0.961	0.942 / 0.952
toothbrush	0.908 / 0.99	0.894 / 0.988
transistor	0.988 / 0.867	0.977 / 0.885
wood	0.994 / 0.954	0.995 / 0.954
zipper	0.881 / 0.989	0.897 / 0.99
mean	0.949 / 0.97	0.958 / 0.974

图 10: 损失函数实验 (Image -Level AUC / Pixel-level AUC)

通过实验可以看到使用总损失以及共同的优化器在除了 leather 类以外的其他类取得了相对于分开做损失以及优化器的情况下要更好的情况，其中在 pill、capsule 的结果比较明显，其中一个原因是，使用分开的损失计算中，对 pill 以及 capsule 中的不明显的形变的异常并不敏感，检测能力比较弱。对于 screw 以及 toothbush 等类也有较好的结果，说明在这种情况下，使用总损失函数，模型具有更强的检测轻微的形变结构异常以及细小异常的能力，但是在 leather 上效果非常差，根据它的特点而言，发现 leather 训练样本也非常少，仅仅是 38 张，不到其他类的 1/5 的训练样本数，因而使用总损失做优化，会使得整体模型对样本的数量比较敏感，整体参数收敛相对较慢，而原来 student1-teacher1 两对网络中，收敛速度比较快并且存在在较少的样本情况下也具有较好的检测性能，因而说明重构 student 可能存在重构能力相对较差或者是二者参数更新不对称的问题。

4.5 重构子网络复现

重构子网络的实现结构如下，该结构原文中并没有给出，相关参考本文同作者 2021 的文章^[11]，实现结构为类似一个 ResNet18 的重构网络，通过 5 中可以看到，重构子网络是跟预训练的 ResNet50 进行知识蒸馏，以及重构的特征为 teacher1 的最后一层输出的高维度特征信息，那么结合图 3 可以进行推测其中的卷积层的信息。如图 11 是 ResNet18 和 ResNet50 的网络结构，图 3 中类似于 ResNet18 的结构，输入的图片为 256×256，那么 teacher1 在第四层最终输出的分辨率为 8×8，按照 ResNet18 中第一个块的卷积核大小为 7×7，其相对于原来的输入是 64×64，不符合我们的结构，因为首个 block 考虑使用 3×3 的卷积，并且不改变分辨率以通道数，因为按照 ResNet18 的结构而言是一个整体通道数以及分辨率线性变化的。在重构 student 与 teacher2 进行第一次知识蒸馏，teacher2 提取出来的多尺度特征中，其分辨率为 16×16，根据图 3 的特点，反卷积核设置为 3×3 的卷积并且提升分辨率为原来的 4 倍，也就是宽高各自两倍，并且负责通道数的改变，中间 Block 只负责特征的提取，那么最终设计出具体的网络层。

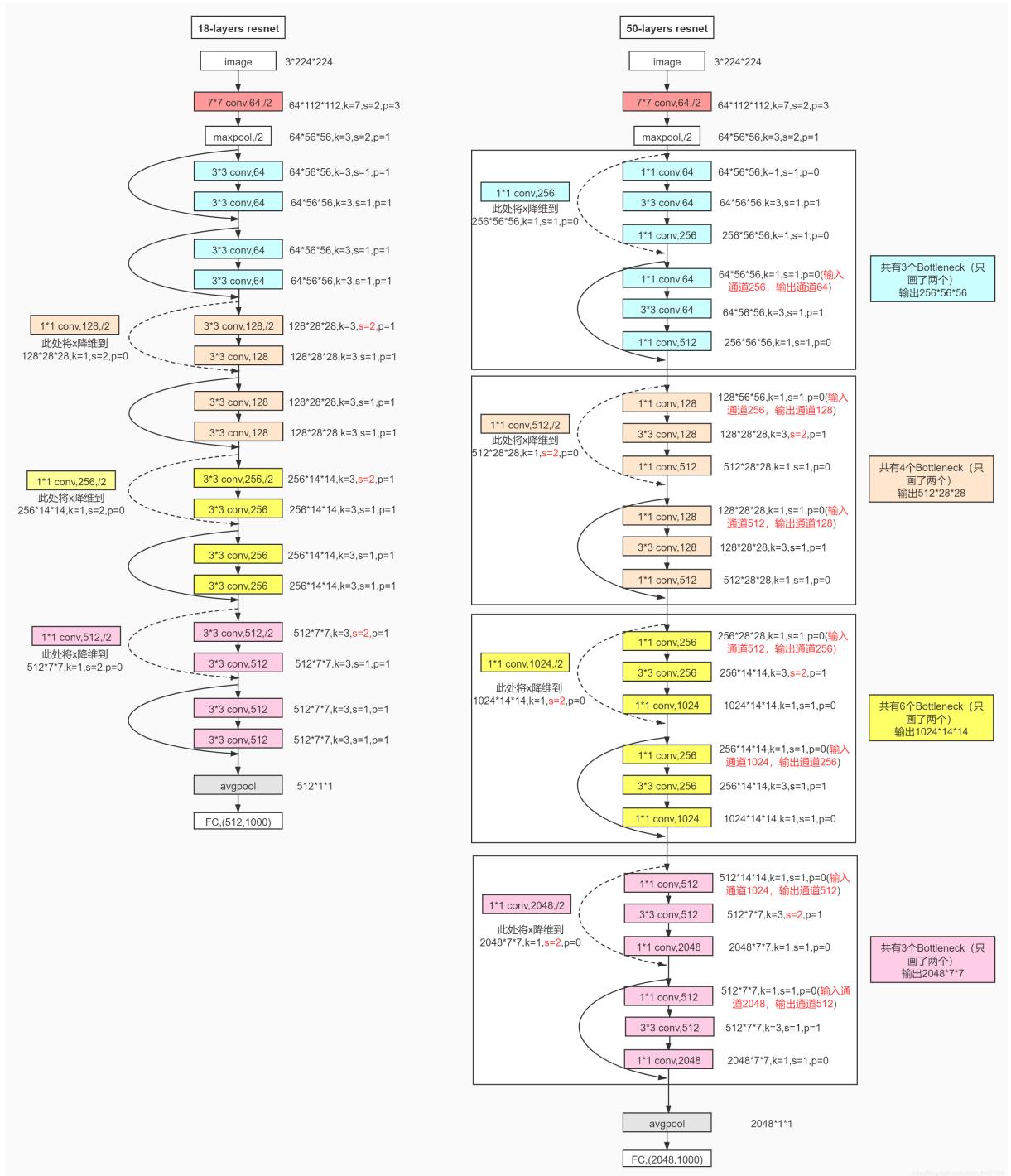


图 11: ResNet18 和 ResNet50 网络结构

如图 12 是对两对 student-teacher 的整体测试，包含了不同的 epoch，因为测试过程发现通过一些 early stop 策略发现，实际上整体的能力并没有达到一个最好的结果，因而展开了对网络的不同 epoch 之间的测试，其中 S-T(100)、S-T(200)、S-T(300) 表示两对 student-teacher 在 100、200 以及 300 个 epoch 下的结果，可以看到在 epoch=200 的情况下模型达到比较好的结果，并且性价比较高。

并且随着迭代次数的提升，在如 capsule、zipper、toothbrush 等类上有着更好的性能，对比与 STPM 相关实验，可以知道，随着网络训练，重构子网络部分性能不断提升促进整体网络性能的提升。

	S-T(100)	S-T(200)	S-T(300)
bottle	1.0 / 0.988	1.0 / 0.987	1.0 / 0.987
cable	0.978 / 0.976	0.99 / 0.976	0.997 / 0.979
capsule	0.85 / 0.985	0.918 / 0.988	0.947 / 0.988
carpet	0.99 / 0.992	0.991 / 0.984	0.989 / 0.992
grid	0.987 / 0.983	1.0 / 0.992	1.0 / 0.993
hazelnut	1.0 / 0.99	1.0 / 0.991	1.0 / 0.991
leather	1.0 / 0.995	1.0 / 0.987	0.926 / 0.948
metal_nut	1.0 / 0.982	0.989 / 0.986	0.996 / 0.983
pill	0.916 / 0.959	0.932 / 0.969	0.941 / 0.976
screw	0.947 / 0.994	0.957 / 0.994	0.952 / 0.994
tile	0.942 / 0.952	0.985 / 0.952	0.966 / 0.936
toothbrush	0.894 / 0.988	0.906 / 0.989	0.919 / 0.99
transistor	0.977 / 0.885	0.978 / 0.916	0.978 / 0.923
wood	0.995 / 0.954	0.992 / 0.953	0.996 / 0.946
zipper	0.897 / 0.99	0.905 / 0.989	0.941 / 0.987
mean	0.958 / 0.974	0.969 / 0.977	0.97 / 0.974

图 12: 不同 epoch 的测试 (Image -Level AUC / Pixel-level AUC)

4.6 网络直连实验

这里会产生一个一个疑问，即为什么是直接从 teacher1 提取特征进行重构，用于重构 student 的重构输入，论文中给出的解释为，多尺度特征提取过程中，最后一层包含更多的并且更为全局的信息，可用于更全面的重构，但是实际上论文中给出的损失函数是六个层相加的结果，那么是否可以考虑重构 student 的输入是直接通过 student1 的输出而得到，基于该疑问展开实验，从前边的实验实验中可以推论出，重构网络可能在重构过程中重构能力比较差，整体组成的网络可能会产生收敛比较慢的问题，在直连二者进行测试时，这里适当增加了迭代的 epoch，并展开对比实验，如图 13 是实验对比的结果，Direct-early(S-T) 表示的是使用基于损失的 early stop 策略对模型进行测试，根据损失在一定时间内的变化情况进行提前结束模型的训练，Direct-200(S-T)、Direct-300(S-T) 表示的是直连网络分辨在 epoch 为 200 以及 300 的情况下进行测试，而 origin(S-T) 是复现的并且在 epoch=100 的结果。

	Direct-early(S-T)	Direct-200(S-T)	Direct-300(S-T)	Origin(S-T)
bottle	1.0 / 0.988	1.0 / 0.988	1.0 / 0.988	1.0 / 0.988
cable	0.995 / 0.979	0.988 / 0.977	0.94 / 0.968	0.978 / 0.976
capsule	0.931 / 0.988	0.956 / 0.988	0.965 / 0.988	0.85 / 0.985
carpet	0.992 / 0.992	0.987 / 0.992	0.996 / 0.993	0.99 / 0.992
grid	0.987 / 0.994	0.993 / 0.994	0.848 / 0.953	0.987 / 0.983
hazelnut	1.0 / 0.99	1.0 / 0.992	0.992 / 0.979	1.0 / 0.99
leather	0.67 / 0.918	0.519 / 0.904	0.987 / 0.978	1.0 / 0.995
metal_nut	1.0 / 0.98	1.0 / 0.982	0.969 / 0.976	1.0 / 0.982
pill	0.961 / 0.973	0.981 / 0.973	0.97 / 0.967	0.916 / 0.959
screw	0.963 / 0.992	0.958 / 0.995	0.963 / 0.994	0.947 / 0.994
tile	0.965 / 0.961	0.98 / 0.918	0.97 / 0.956	0.942 / 0.952
toothbrush	0.908 / 0.99	0.917 / 0.99	0.917 / 0.987	0.894 / 0.988
transistor	0.988 / 0.867	0.976 / 0.879	0.972 / 0.855	0.977 / 0.885
wood	0.994 / 0.954	0.993 / 0.959	0.989 / 0.962	0.995 / 0.954
zipper	0.881 / 0.989	0.967 / 0.986	0.972 / 0.981	0.897 / 0.99
mean	0.949 / 0.97	0.948 / 0.968	0.963 / 0.968	0.958 / 0.974

图 13: 直连网络测试 (Image -Level AUC / Pixel-level AUC)

从图 13 可以看到在 epoch=300 的情况下，整体模型效果达到了较好的结果，对比 Origin(S-T) 而言整体结果相近，但是实际上训练时间更长，耗费资源更多，但是直连网络情况下对 capsule、pill、zipper 具有更强的能力，这里原因更上述使用全部损失进行优化器的结果是一样的，在检测微小的结构形变的能力上具有更好的性能，但是在 grid 类上效果下降十分明显，通过分析可能存在的其中一个原因是，grid 中产生形变的样品中，其样品区域和背景相似程度比较高，并且产生微小形变的结构异常，例如仅仅是原来的轻微弯曲的效果，会对模型引入偏差导致较差的性能，如图 14。

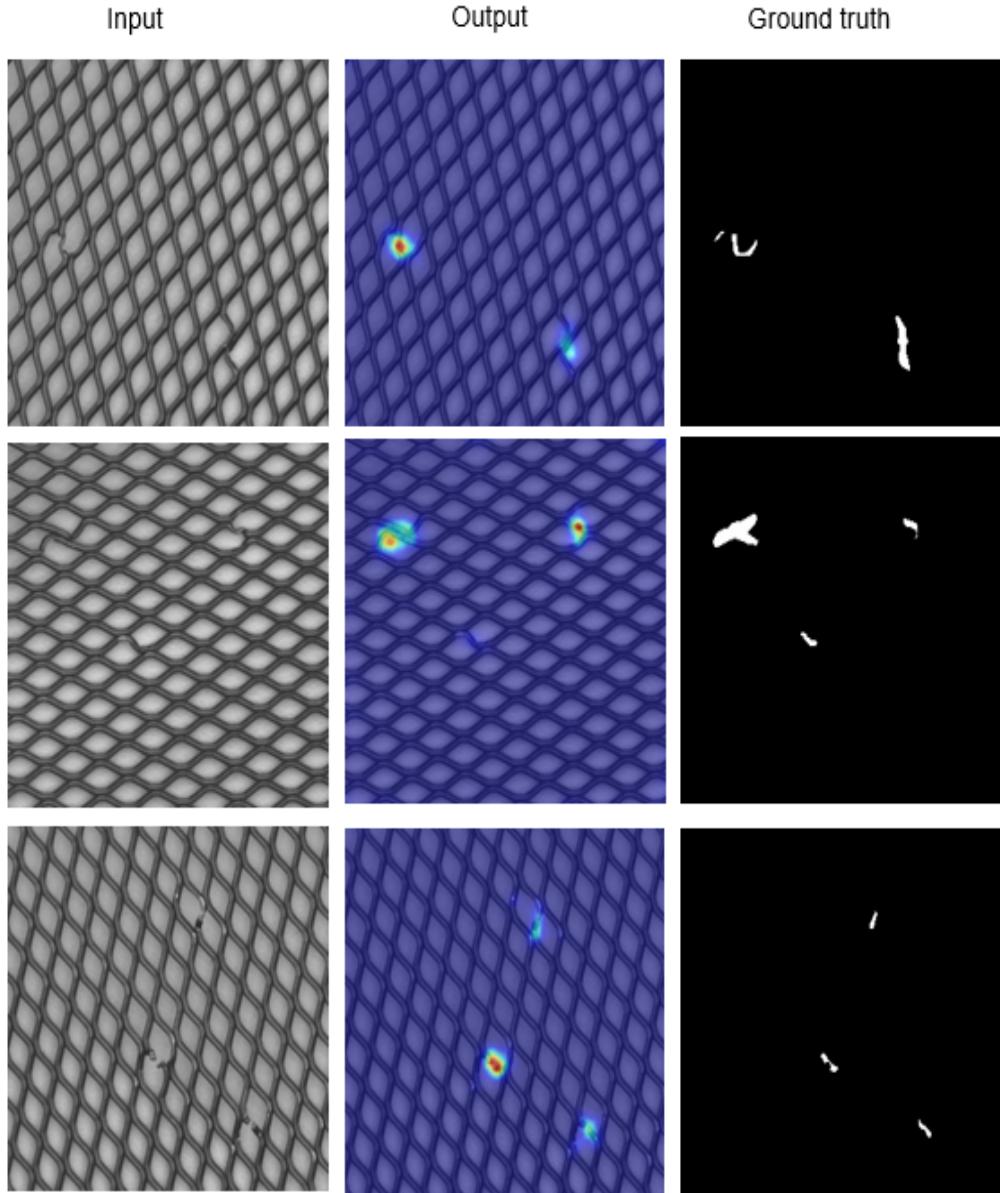


图 14: 直连网络 grid 测试结果

4.7 重构 student 进一步改进

参考^[12]中给出了一个类似的思想对如图 3 的重构 student2 进行改造，如图 15 是改造后的结果，仅仅使用三个特征提取层进行特征提取，去掉了前缀的卷积层，尽量减少参数量，在保持性能的同时加快训练的速度，让模型更快收敛，同时保留多尺度提取特征的各层，用于最终的多尺度特征的提取，最终网络组成仅是三个特征提取层。

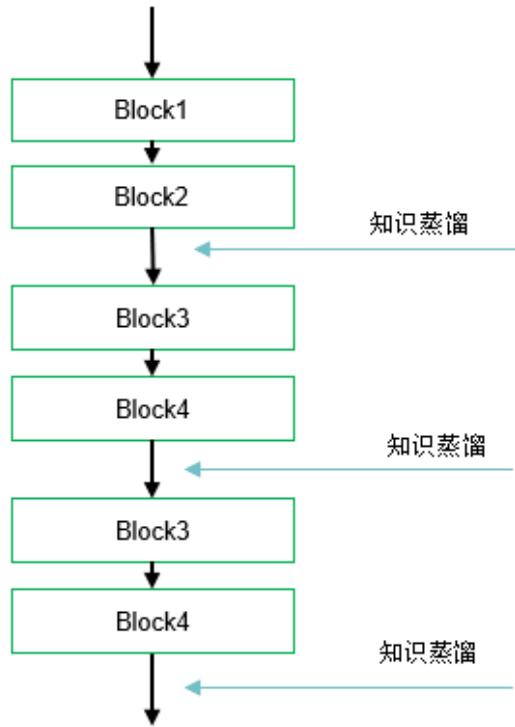


图 15: 重构网络

如图 16 是如图 15 的重构网络对应的块，使用 2×2 的反卷积对输入的特征表示进行通道数的提升以及分辨率的提升，并且跳跃连接同样使用 2×2 的反卷积块形成一个残差的跳跃连接。

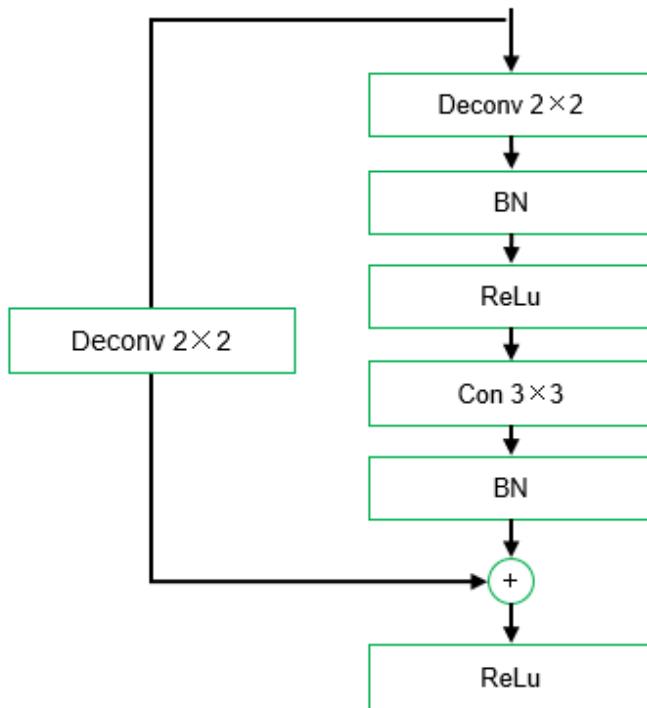


图 16: 重构网络的 block

如图 17，其中 RSTPM-100、RSTPM-1-200 表示的是原实现的重构子网络的在 epoch=100、epoch=200 的结果，RSTPM-2-100 是改造后在 epoch=100 的结果，可以看到结果是接近原来网络的 epoch=200 的效果，并且改造后模型参数是原来复现的重构 student 的参数量的一半，训练速度也大大加快。

	RSTPM-1-100	RSTPM-1-200	RSTPM-2-100
bottle	1.0 / 0.988	1.0 / 0.987	1.0 / 0.988
cable	0.978 / 0.976	0.99 / 0.976	0.956 / 0.973
capsule	0.85 / 0.985	0.918 / 0.988	0.915 / 0.987
carpet	0.99 / 0.992	0.991 / 0.984	0.994 / 0.992
grid	0.987 / 0.983	1.0 / 0.992	1.0 / 0.994
hazelnut	1.0 / 0.99	1.0 / 0.991	1.0 / 0.99
leather	1.0 / 0.995	1.0 / 0.987	1.0 / 0.995
metal_nut	1.0 / 0.982	0.989 / 0.986	0.997 / 0.98
pill	0.916 / 0.959	0.932 / 0.969	0.963 / 0.966
screw	0.947 / 0.994	0.957 / 0.994	0.947 / 0.992
tile	0.942 / 0.952	0.985 / 0.952	0.971 / 0.959
toothbrush	0.894 / 0.988	0.906 / 0.989	0.9 / 0.99
transistor	0.977 / 0.885	0.978 / 0.916	0.975 / 0.901
wood	0.995 / 0.954	0.992 / 0.953	0.992 / 0.964
zipper	0.897 / 0.99	0.905 / 0.989	0.869 / 0.988
mean	0.958 / 0.974	0.969 / 0.977	0.965 / 0.977

图 17: 重构结果 (Image -Level AUC / Pixel-level AUC)

4.8 注意力机制复现

对于注意力机制而言，如图 4给出了具体的注意力机制的结构，论文中提到，因为是对像素级别的异常检测，因而更多应该关注于像素层面上，通过注意力机制后，最终通道数会被压缩到一个通道，而不是关注多个通道的重要性，那么按此逻辑最终 1×1 的卷积应该是将整体通道压缩，最终得到一个通道的注意力图，而 3×3 的卷积可以作为特征具体的作用。从 teacher2 中提取出来的对应层次的特征，通过注意力机制后加载 student2 上，参考 [4] 中，注意力机制的条件可以存在如下两种形式，其中 $g'(x)$ 最终添加注意机制后的输出， $M(x_t)$ 是注意力机制输出的注意力图，而 x_t 是从 teacher2 中接收到的多尺度特征信息， x 是 student2 上层传到下层的特征信息。

$$g'(x) = M(x_t) \cdot x \quad (7)$$

$$g'(x) = (1 + M(x_t)) \cdot x \quad (8)$$

对于公式 (7) 起到对特征图像素点加权的作用，但是可能会造成局部像素点的值为 0 的情况，而公式 (8) 可以防止局部出现 0 的情况，并且可以突出峰值的特征图，为了验证两种方式的使用，对两种添加方式都进行了实验并输出对应的注意力图进行观察其效果，如图 18，因为对网络中需要添加的注意力机制的地方有两处，所以尝试不同的添加方式，其中 S-T(++) 表示两个都是使用公式 (8) 的形式，S-T(**) 表示两处都使用公式 (2) 的形式，S-T(*+) 表示第一处使用公式 (7) 的形式而第二处使用公式 (8) 的形式，S-T(None) 表示不使用注意力机制。如图 18，在 S-T(++) 以及 S-T(None) 中性能是最差的，通过分析结果图可知，重构 student 的重构能力在 epoch=100 的情况下并没有很好的性能，而在特征提取过程中并没有很好的发挥作用，而通过公式 (7) 的形式凑巧减少了上层的影响，而不会将上层的坏影响更多传递到下层，从而 S-T(*+) 有着更好的结果。

	S-T(++)	S-T(**)	S-T(*+)	S-T(None)
bottle	1.0 / 0.988	1.0 / 0.987	1.0 / 0.988	1.0 / 0.988
cable	0.985 / 0.978	0.975 / 0.974	0.978 / 0.976	0.992 / 0.976
capsule	0.871 / 0.987	0.849 / 0.987	0.85 / 0.985	0.861 / 0.986
carpet	0.994 / 0.993	0.994 / 0.993	0.99 / 0.992	0.994 / 0.988
grid	0.948 / 0.985	0.982 / 0.986	0.987 / 0.983	0.987 / 0.985
hazelnut	1.0 / 0.989	0.999 / 0.989	1.0 / 0.99	0.999 / 0.989
leather	0.894 / 0.911	0.994 / 0.993	1.0 / 0.995	0.877 / 0.908
metal_nut	0.994 / 0.977	0.997 / 0.972	1.0 / 0.982	0.994 / 0.98
pill	0.932 / 0.943	0.932 / 0.967	0.916 / 0.959	0.928 / 0.959
screw	0.946 / 0.994	0.94 / 0.995	0.947 / 0.994	0.932 / 0.994
tile	0.964 / 0.955	0.985 / 0.953	0.942 / 0.952	0.965 / 0.952
toothbrush	0.861 / 0.987	0.778 / 0.986	0.894 / 0.988	0.858 / 0.987
transistor	0.977 / 0.904	0.967 / 0.917	0.977 / 0.885	0.98 / 0.906
wood	0.996 / 0.962	0.994 / 0.947	0.995 / 0.954	0.992 / 0.953
zipper	0.869 / 0.986	0.893 / 0.981	0.897 / 0.99	0.888 / 0.985
mean	0.949 / 0.969	0.952 / 0.975	0.958 / 0.974	0.95 / 0.969

图 18: 注意力机制测试对比 (Image -Level AUC / Pixel-level AUC)

通过具体的实验并输出注意力图，其中比较理想的注意力图是论文中给出的，如图 4，在有异常的部分是没有明显的反映，而在正常区域是有较大的反映的，这是因为注意力机制在训练的时候是仅仅使用正常样本去训练的，但是实际上复现过程中输出的注意图，如图 19，并没有出现预期的结果。

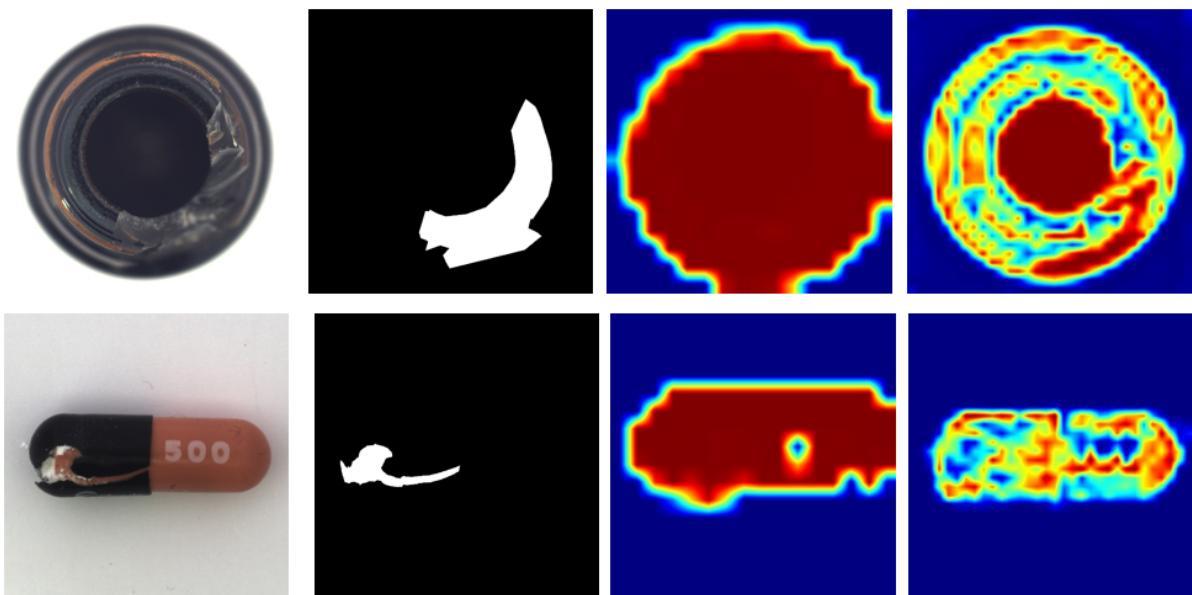


图 19: 注意力机制输出

实验过程还发现一个现象，就是某次实验中，发现输出的特征图中都是全 0 的情况，这也就是说每一次经过注意力图的加权后，特征图都变成了 0，将上层提取到的特征全部清楚了，但是达到了相同的结果，考虑到之前猜测重构 student 的能力可能会更差，因而尝试将每次注意力机制都进行置 0，结果如图 20，其中 Error(S-T) 表示的是注意力机制仅做清 0 操作，而 Correct(S-T) 是论文中提到的实现方式，可以看到进行置 0 操作实现了更好的结果，下层特征输出不依赖于上层，但是达到了较好的效果。

	Error(S-T)	Correct(S-T)
bottle	1.0 / 0.984	1.0 / 0.988
cable	0.987 / 0.981	0.978 / 0.976
capsule	0.823 / 0.981	0.85 / 0.985
carpet	0.992 / 0.992	0.99 / 0.992
grid	0.99 / 0.988	0.987 / 0.983
hazelnut	0.999 / 0.989	1.0 / 0.99
leather	0.995 / 0.99	1.0 / 0.995
metal_nut	0.997 / 0.979	1.0 / 0.982
pill	0.911 / 0.94	0.916 / 0.959
screw	0.908 / 0.99	0.947 / 0.994
tile	0.983 / 0.96	0.942 / 0.952
toothbrush	0.967 / 0.984	0.894 / 0.988
transistor	0.987 / 0.942	0.977 / 0.885
wood	0.998 / 0.927	0.995 / 0.954
zipper	0.895 / 0.987	0.897 / 0.99
mean	0.962 / 0.974	0.958 / 0.974

图 20: 注意力机制改造测试 (Image -Level AUC / Pixel-level AUC)

4.9 结果展示

如图 21 是分割的结果，其中 STPM、RSTPM 分别表示本地实现的 STPM 以及复现的 RSTPM 的方法，分别对如下样例进行测试得到如下结果。

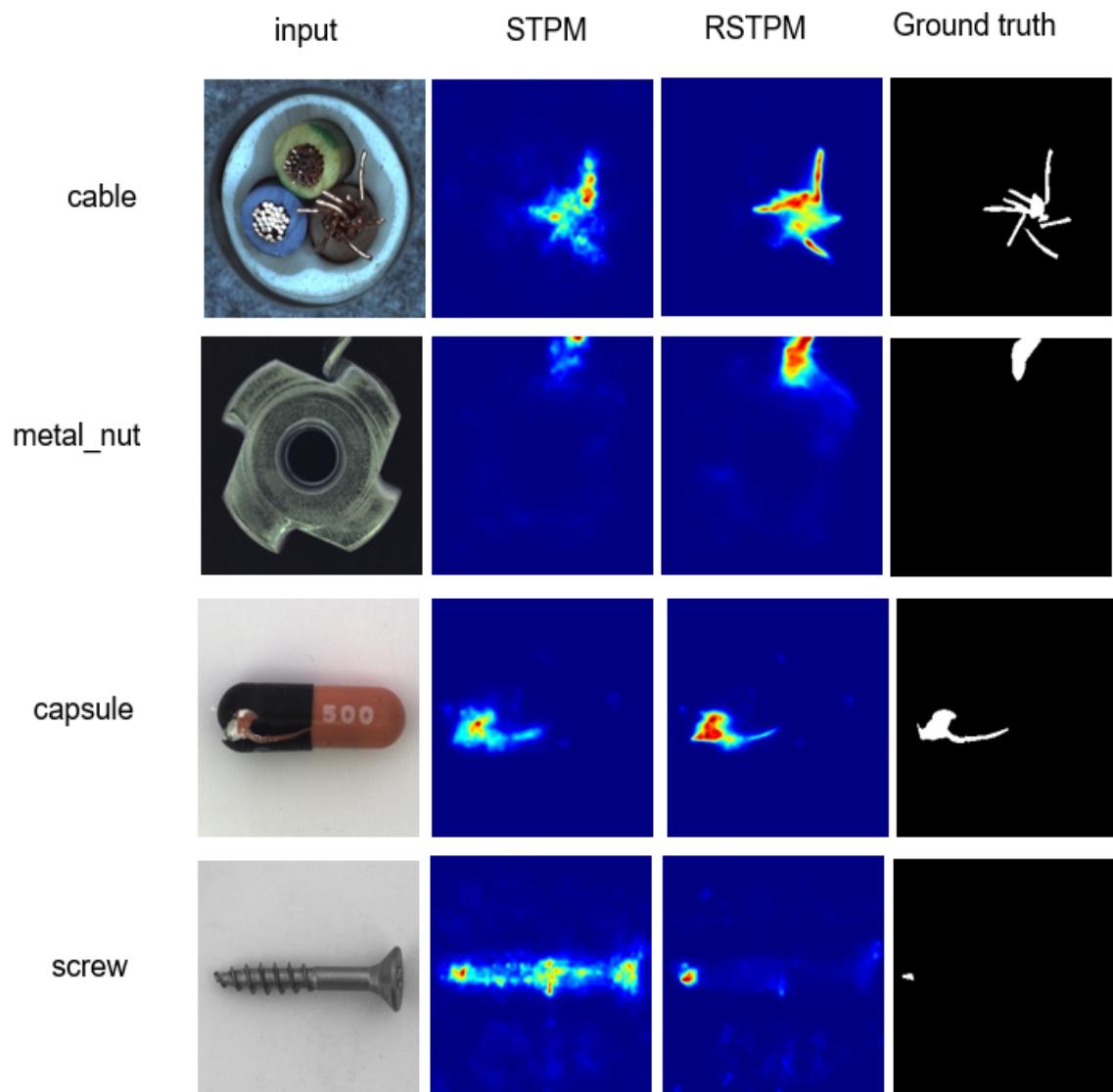


图 21: 分割结果

如图 22 是用于生成的伪异常的结果，其中① + ⑥、② + ⑤、③ + ④ 分别表示两对 student-teahcer 各个多尺度特征层提取的特征的和的结果，就是作为输入到判别网络的结果，augment 表示做数据增强后的图片，作为一个伪异常，从图 22 可以看到，模型具有相对较好的泛化能力。

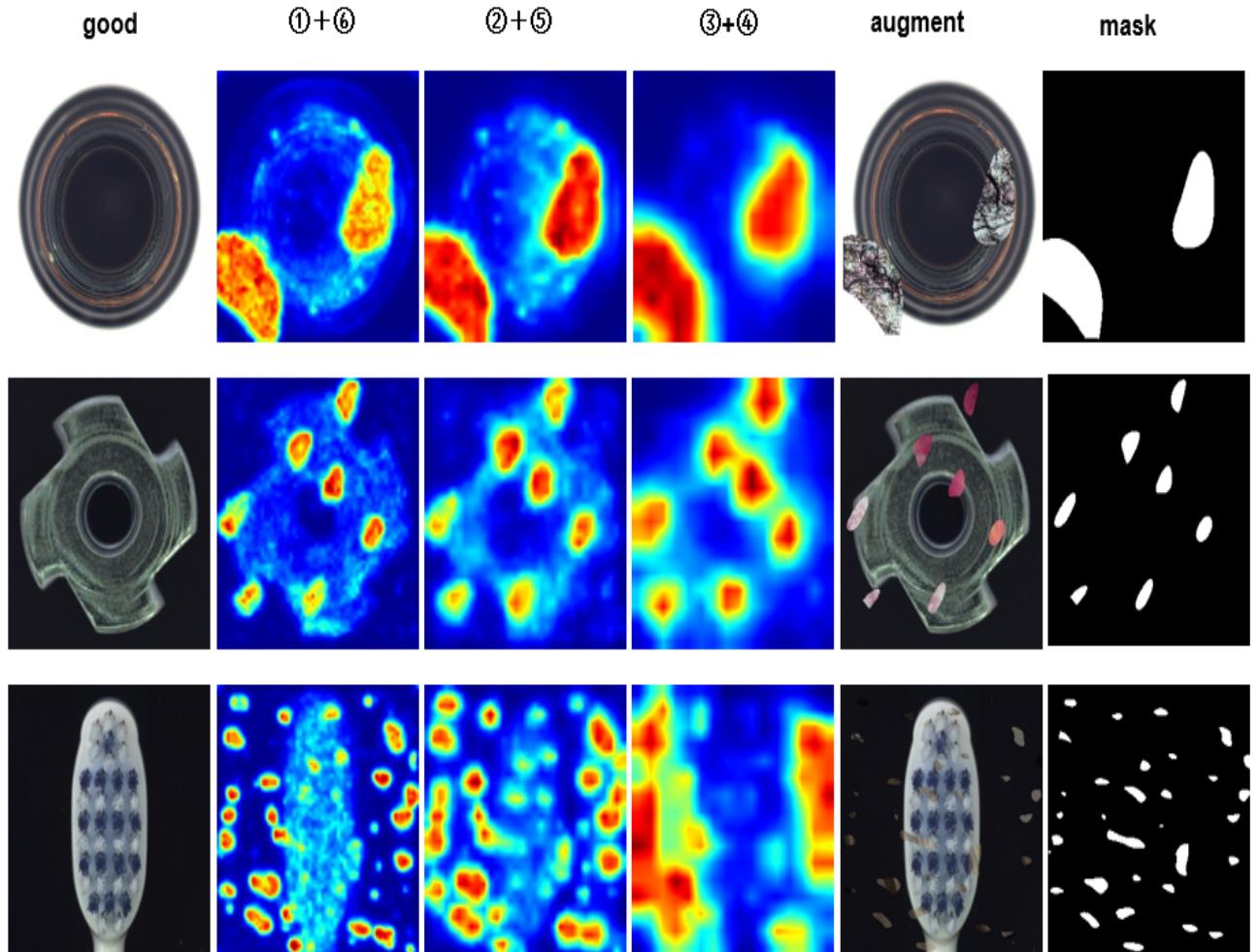


图 22: 伪异常生成与分割结果

如图 23 是一些失败的案例，观察这些案例中，主要失败的因素在于，类似于 capsule、zipper 中产生的形变从角而言，跟正常图像的差别并不大，如图 capsule 中的扁平情况以及 zipper 中拉链的变形，整体结构跟无异常的分布大致相同并且颜色也相同从而导致误差，而对于 toothbrush，属于比较细微的异常并且具有方向性，模型处理相关异常能力也比较差，对于 transostor，出现整体形变的情况，因为经过像素级别的特征提取，整体学习过程中并不能很好把握空间的关系，出现如图的结果是因为整体的样品的移动挡住了背景的颜色，让模型认为整个都是一个异常，所以会将整个模型都认为是一个异常，这种异常也是比较复杂，模型在此方面并没有很好的性能。

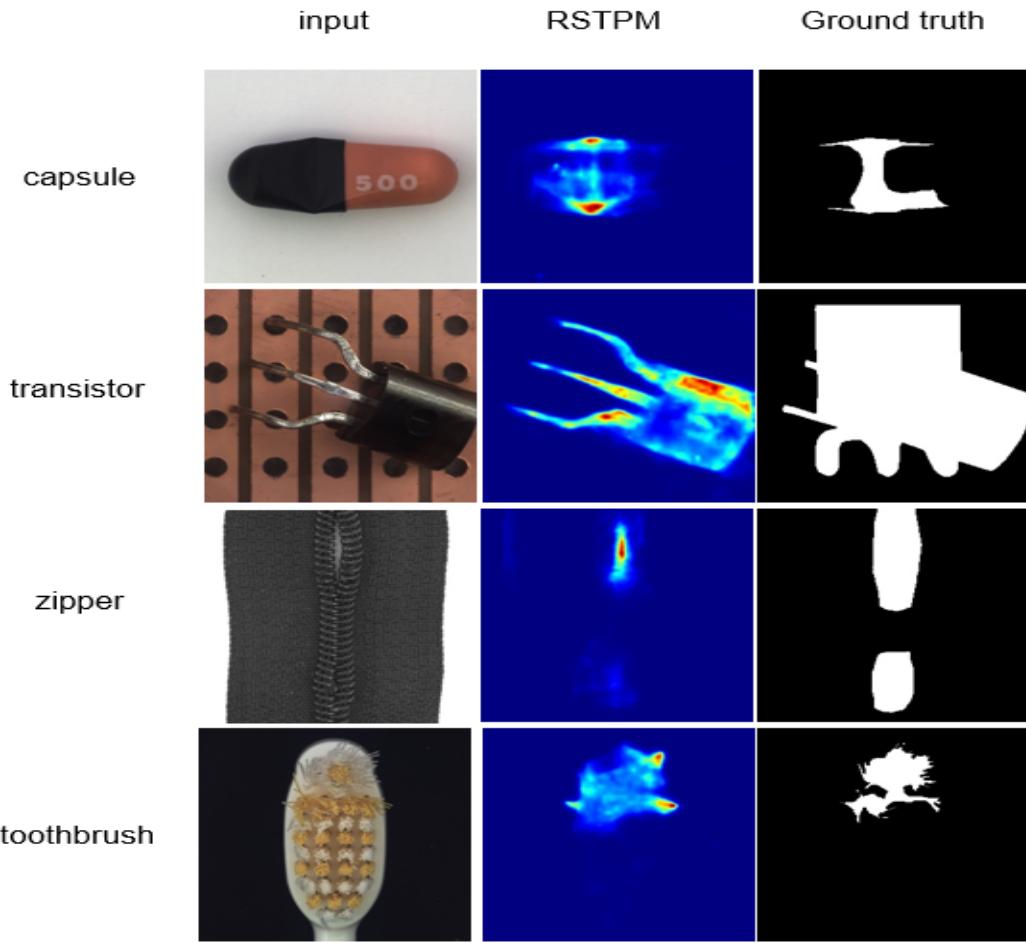


图 23: 一些失败的案例

如图 24 是整体实验结果比较，其中 Paper(S-T)、Paper(Dis)、Paper(total) 分别表示的是论文中关于两对 student-teacher、仅仅判别网络判定以及整体网络判定的结果，Repro(S-T)、Repro(Dis)、Repro(total) 分别表示的是复现的关于两对 student-teacher、仅仅判别网络判定以及整体网络判定的结果。对于两对 student-teacher 的结果差异并不大，但是在判别网络方面复现结果中效果好于论文中给出的结果，但是在总体结果上却出现了明显的差异，在判别网络的实现方面还是存在一定的差异。

	Paper(S-T)	Repro(S-T)	Paper(Dis)	Repro(Dis)	Paper(total)	Repro(total)
bottle	1.0 / 0.99	1.0 / 0.988	1.0 / 0.993	1.0 / 0.994	1.0 / 0.993	1.0 / 0.994
cable	0.967 / 0.973	0.978 / 0.976	0.989 / 0.979	0.996 / 0.973	0.996 / 0.983	0.998 / 0.984
capsule	0.873 / 0.985	0.85 / 0.985	0.945 / 0.972	0.929 / 0.991	0.93 / 0.985	0.876 / 0.99
carpet	0.981 / 0.988	0.99 / 0.992	0.964 / 0.985	1.0 / 0.961	0.987 / 0.992	0.997 / 0.99
grid	0.984 / 0.994	0.987 / 0.983	1.0 / 0.996	1.0 / 0.996	1.0 / 0.996	1.0 / 0.995
hazelnut	1.0 / 0.991	1.0 / 0.99	0.987 / 0.995	0.999 / 0.997	0.998 / 0.995	1.0 / 0.996
leather	0.998 / 0.984	1.0 / 0.995	1.0 / 0.996	1.0 / 0.974	1.0 / 0.996	1.0 / 0.995
metal_nut	1.0 / 0.982	1.0 / 0.982	0.999 / 0.989	1.0 / 0.975	1.0 / 0.989	1.0 / 0.987
pill	0.967 / 0.972	0.916 / 0.959	0.98 / 0.99	0.973 / 0.981	0.981 / 0.987	0.966 / 0.978
screw	0.948 / 0.993	0.947 / 0.994	0.941 / 0.989	0.952 / 0.991	0.968 / 0.993	0.943 / 0.994
tile	0.953 / 0.97	0.942 / 0.952	0.986 / 0.987	1.0 / 0.978	0.999 / 0.988	0.979 / 0.982
toothbrush	0.9 / 0.989	0.894 / 0.988	0.987 / 0.994	0.983 / 0.995	0.979 / 0.993	0.931 / 0.993
transistor	0.975 / 0.898	0.977 / 0.885	0.978 / 0.872	0.974 / 0.865	0.983 / 0.907	0.985 / 0.903
wood	0.992 / 0.97	0.995 / 0.954	0.966 / 0.978	0.996 / 0.961	0.993 / 0.981	0.995 / 0.971
zipper	0.898 / 0.985	0.897 / 0.99	0.998 / 0.988	1.0 / 0.99	0.993 / 0.992	0.969 / 0.993
mean	0.962 / 0.977	0.958 / 0.974	0.981 / 0.962	0.987 / 0.975	0.987 / 0.977	0.976 / 0.983

图 24: 复现对比 (Image -Level AUC / Pixel-level AUC)

5 总结与展望

该研究使用一种新的学生网络对 STPM 异常检测方法进行改进并重建。将教师网络修改为新的学生网络，并使用教师到学生的注意力机制来保证学习的成功。为进一步提高异常检测的准确率，使用判别网络对异常图进行重新考虑。通过将伪异常用于判别网络的训练，为正常区域生成更准确的异常图。所提方法同时使用 STPM 和两对师生网络以及一个判别网络，与传统方法相比取得了较高的异常检测准确率。

参考文献

- [1] BERGMANN P, LÖWE S, FAUSER M, et al. Improving unsupervised defect segmentation by applying structural similarity to autoencoders[J]. arXiv preprint arXiv:1807.02011, 2018.
- [2] SCHLEGL T, SEEBÖCK P, WALDSTEIN S M, et al. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery[C]//International conference on information processing in medical imaging. 2017: 146-157.
- [3] SCHLEGL T, SEEBÖCK P, WALDSTEIN S M, et al. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks[J]. Medical image analysis, 2019, 54: 30-44.
- [4] ZAVRTANIK V, KRISTAN M, SKOČAJ D. Reconstruction by inpainting for visual anomaly detection [J]. Pattern Recognition, 2021, 112: 107706.
- [5] AKCAY S, ATAPOUR-ABARGHOUEI A, BRECKON T P. Ganomaly: Semi-supervised anomaly detection via adversarial training[C]//Asian conference on computer vision. 2018: 622-637.
- [6] AKÇAY S, ATAPOUR-ABARGHOUEI A, BRECKON T P. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection[C]//2019 International Joint Conference on Neural Networks (IJCNN). 2019: 1-8.
- [7] TANG T W, KUO W H, LAN J H, et al. Anomaly detection neural network with dual auto-encoders GAN and its industrial inspection applications[J]. Sensors, 2020, 20(12): 3336.
- [8] FUKUI H, HIRAKAWA T, YAMASHITA T, et al. Attention branch network: Learning of attention mechanism for visual explanation[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 10705-10714.
- [9] WANG G, HAN S, DING E, et al. Student-teacher feature pyramid matching for unsupervised anomaly detection[J]. arXiv preprint arXiv:2103.04257, 2021.
- [10] ZAVRTANIK V, KRISTAN M, SKOČAJ D. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 8330-8339.
- [11] YAMADA S, HOTTA K. Reconstruction Student with Attention for Student-Teacher Pyramid Matching [J]. arXiv preprint arXiv:2111.15376, 2021.

- [12] DENG H, LI X. Anomaly Detection via Reverse Distillation from One-Class Embedding[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 9737-9746.