# DefectNet: Towards Fast and Efficient Defects Detection

Feng Li  Feng Li

**Abstract. pass**

## 1. Introduction

（背景:）近几年来，深度卷积神经网络在目标检测领域取得了巨大的成功，涌现出了一大批的优秀的检测框架，例如，基于区域推荐的检测算法（Faster R-CNN），单次分类和回归的单阶段的算法如 SSD、YOLO 等，他们大都集中于建立一个统一通用的目标检测框架，并且有着较高的准确率和效率，虽然他们能满足大多数目标检测领域的需要，（抛出问题:）但是，在小部分目标检测领域，尤其是在工厂瑕疵检测中，例如，酒瓶质量监控和纺织品瑕疵检测，我们不仅仅只关注于一张图像中某个对象的识别和定位，同时我们还需要对该图像是否包含瑕疵对象进行判决，即图像分类问题，甚至有时候对瑕疵进行判决比对物体检测更加重要，因为在大多数瑕疵检测领域，不包含瑕疵的图像往往占据了绝大多数，而包含瑕疵的图像只有一小部分，同时更高的对瑕疵的判决能力意味着更小的计算成本和更低的错误率，这能降低大量的经济成本。虽然我们也能够利用（state-of-the-art）业界前沿的检测框架解决这些问题，但是他们并不完美，他们对瑕疵的判决能力仍然取决于某个阈值，而这往往会容易受到主观方面的影响，这种情况下他们产生错误的概率仍然很高，所以我们还有可以改进和提升的空间。从另一方面讲，当然，我们可以先训练一个图像分类网络，先对图像是否包含瑕疵进行判决，然后再训练一个目标检测网络，再检测出图像中的瑕疵对象，但是这样太繁琐了，会带来更多的时间和计算成本，因此（hence），我们希望有一个框架能一次性解决上述问题。



(a)


(b)

(c)

| 背景是否参与训练 | IoU:0.50:0.95 | Iou:0.50 | F1-score | aIT | aIT of normal image | aIT of defect images |
|---|---|---|---|---|---|---|
| Yes | | | | | | |
| No | | | | | | |

在这篇文章中，我们建议了一个新颖的瑕疵检测框架，名字为 DefectNet，它将一个 Defect Finding Network 插入到目前（state-of-the-art）的目标检测框架中，以便一次性的实现对是否包含瑕疵进行判决和对瑕疵对象进行物体检测。同时，它成功的充分的利用了不包含瑕疵的图像，这在以往的目标检测算法中是做不到的。首先，它先使用 DFN 判断图像是否包含瑕疵，然后根据判断的结果再决定是否再对图像进行物体检测，如果不含有瑕疵，则直接返回无瑕疵信号，否则需要进一步的对瑕疵进行检测。图？展示了三种不同类型的进行瑕疵检测方法。

实验结果表明我们的 DefectNet 对瑕疵的检验能力（acc）比 One-model Method 高出了 10%，速度提升了 10%，其中，对正常图像的检出速度更是提升了一倍，同时 mAP 仅仅只是降低了 1-2%，提升效果十分明显。对于 Two-model Method，我们的检验能力（acc）比 One-model Method 高出了 10%，速度提升了 10%，其中，对正常图像的检出速度更是提升了一倍。表？展示了我们的实验结果。

我们的方法的目的不是为了取代和超越现今的物体检测框架，而是为了弥补它们在某些细分的物体检测领域方面的不足。我们的贡献也许很微小，但是它促进了深度卷积网络在目标检测领域更加系统和完善。我们总结我们的贡献如下：

1. 在绝大多数 One-model 检测算法中，不包含任何瑕疵的图像是很难用上的，我们建议的方法成功充分有效的利用了这些图像数据，避免了数据资源的浪费；
2. 在绝大多数瑕疵检测领域，瑕疵图像只占一小部分，而绝大多数都是不包含瑕疵的图像，我们建议的 DefectNet 它对瑕疵的检验能力超过了 One-model Method 和 Two-model Method，而且推理速度更快，因此它能够减少许多不必要的计算量，同时降低成本，并且弥补了他们在瑕疵检测领域方面的不完美；
3. 我们在 Fabric Detects Dataset 对我们的模型进行了实验和分析，并且和 One-model Method 和 Two-model Method 做了比较。实验结果表明我们的框架具有较强的实用性。

## 2. DefectNet

~~现今基于 CNN 的主流的目标检测框架主要分为两种，一种是基于区域推荐的，即先用传统的图像算法或者训练一个 CNN 生成候选建议框，然后再用 CNN 进行目标的分类和候选框的回归，另一种是使用单个 CNN 直接实现目标的分类和锚点的回归。图 2(a)和(b)分别展示了这两种结构。然而，在实际的瑕疵检测中，不含瑕疵的图像应该比含有瑕疵的图像要多得多，如果采用(a)和(b)这两种结构，它们的计算量应该会比先进行瑕疵判决再进行瑕疵检测的计算量要多，因此本文提出了 defectNet~~，DefectNet 由 backbone 网络、Defect Finding Network、head、Box 回归、Box 分类组成。图 2（c）展示了

这种结构。

## 2.1. Backbone

The backbone network is mainly used to extract image features for object classification and location in the object detection frameworks, there are a number of excellent networks for image classification so far, such as VGGNet, GoogleNet, ResNet, DenseNet and so on. In this paper, we select ResNet-50 as our backbone network for the efficiency and accuracy. Figure 3 illustrates the ResNet-50 network that consists of an input head, four sequential stages and the final output layer. The input head includes a 7×7 convolution kernel with an output channel of 64 and a stride of 2, a max pooling layer with a stride of 2. After an image passes through the input head, its width and height decreases by 4 times and its channels size increases to 64.

Each stage of ResNet-50 starts with a downsampling block and followed by several residual blocks. In the downsampling block, there are two paths, namely path A and path B. Path A is the main path, where includes three convolutions, we call them conv1, conv2 and conv3, their kernels size is 1×1, 3×3, 1×1 respectively. Instead, path B has only a 1×1 convolution, which is called conv4. Path B is also called shortcut connection. The conv1 and conv2 have the same output channels, but the output channels of conv3 are four times as many as those of conv1 and conv2. In order to sum outputs of both paths, conv4 has the same output channels with conv3. A residual block is similar to the downsampling block except without conv4 in path B and only using convolutions with a stride of 1. In stage 1, the strides of all convolutions are 1, which make the input width and height remain unchanged. Starting from stage 2, the strides of conv2 and conv4 are 2 to halve the input width and height. The downsampling block together with residual block are named as bottleneck, stage 1, stage 2, stage 3, stage4 of ResNet-50 have 3, 4, 6, 3 bottlenecks respectively. The final output layer is an average pooling layer and a full connection layer. There are many different residual network models by setting the different number of residual blocks in each stage, such as ResNet-101, ResNet-152, where the number represents the number of layers in the network.

## 2.2. Defect Finding Network

在几乎所有的目标检测框架中，往往只需要提取 backbone 中前面几层的特征图信息，而最后的输出层往往丢弃。在本文中，我们重新提取最后的输出层信息，因为它能够帮助我们对图像是否包含瑕疵进行判决。输入图像经过 backbone 的卷积操作后，最后只剩下高级的语义信息，我们对最后一层特征图进行平均池化操作，然后展开成一维向量，在经过两个全连接操作，最后经过 Soft Max 层，从而输出是否包含有瑕疵的判决信息。我们引入 binary cross entropy loss 给图像瑕疵分类，定义如下：

$$L_{judge}(\mathbf{X}, y) = \begin{cases} -y\log(p) \\ -(1-y)\log(1-p) \end{cases}$$

$$L_{judge}(\mathbf{X}, y) = -[y\log(p) + (1-y)\log(1-p)]$$

where X 是输入的图像，y（0，1）是图像的标签，0 表示没有瑕疵，1 表示有瑕疵，p 表示包含瑕疵的概率。

## 2.3. Features Pyramid Network

For the classification problems, we usually need deeper semantic information, while for location, the semantic information in the shallow layers is more important. The feature pyramid network adopts a lateral connection of feature maps to have both of these characteristics. Hence, we use FPN as our head. Figure 1(a) describes the construction of the feature pyramid networks. The structure of feature pyramid networks involves a bottom-up pathway, a top-down pathway and corresponding lateral connections. The bottom-up pathway is the feed-forward computation of

the backbone, where feature maps are selected from the output of the last layer of each stage in the backbone network. The selected feature maps have the different number of channels and their width and height are halved in sequence. Through upsampling spatially coarser, the top-down pathway generates higher resolution feature maps by two times, whose semantic information is stronger but localized information is weaker. The lateral connections between the top-down pathway and the bottom-up pathway settle the matter, which enhance the localized information of feature maps in the top-down pathway by merging feature maps of the same spatial size from the bottom-up pathway and the top-down pathway and make more accurate predictions.

### 2.4. Bounding Box Regression and Classification

A bounding box 这里介绍候选框回归息。

### 2.5. Loss Function

我们的损失函数由三部分组成：瑕疵图像分类的损失，物体检测对象的置信度损失和位置损失。

Defect Image Classification Loss

参数 alpha 意味着瑕疵图像分类损失所占的权重。

## 3. Experiments

### 3.1. Bottled Liquor Dataset for Quality Control

瓶装白酒数据集来自于数智重庆.全球产业赋能创新大赛（初赛），均拍摄于企业真实生产的产品，该数据集总共有 4516 张图像，其中，正常图像有 1146 张，含有瑕疵的图像有 3370 张。瓶装酒的瑕疵可分为五个大类，但是该数据集只有三个大类，总共十个小类，特别地，还包括一个特殊的类别——背景，不包含在瑕疵类别里面，它的作用只是为了区分正常的图像和包含瑕疵的图像，当一张图像不含有任何一类瑕疵类别时，该图像才为正常的图像。我们将该数据集按照 8：2 的比例划分为训练集和测试集，其中，训练集总共有 3612 张图像，包含 921 正常图像和 2691 张瑕疵图像；测试集总共有 904 张图像，包含 225 正常图像和 679 张瑕疵图像。

### 3.2. Implementation Details

我们采用 mmdetection 工具箱作为我们的检测工具，如果没有做出特别的说明，本文中所有的实验都是使用单个的 GTX 1080Ti GPU 进行训练的，并且 batch size 为 2，学习率为 0.00125，epoch 数量为 12，当然，如果采用不同大小的学习率，可能得到的结果会和本文的结果不一样，甚至 mAP 的结果会更好，但这不是我们关注的重点。

### 3.3. Evaluation metrics

我们使用 mAP、f1-score、average infer time per image 来评价我们的模型。mAP 采用标准的 COCO 数据集的评价尺度，包括 Iou：0.50：0.95 和 IoU：0.50 两种评价尺度，用于对各个不同类别的瑕疵的检出能力作出评价。F1-score 采用 macro average f1-score，用于评价模型对瑕疵的发现检出能力，即判断是否为非瑕疵图像，f1-score 由 acc 和 AR 决定，f1-score 得分越高，就代表对瑕疵的敏感程度越高。aIT 用于评价模型的推理效率，在这里我们每次仅对一张图像进行推理，同时仅统计模型对输入图像的处理时间的平均值，不包含图像读取、转换为张量、图像增强等在输入模型之前的处理时间，因为不同大小的图片每次的预处理时间可能都不一样。

### 3.4. One-model Method

我们采用 Cascade R-CNN 检测算法作为我们的 One-model method，同时我们不做任何的改动。考虑到数据集中有一个特殊的背景类别，把背景类别加入到网络中训练与不加入到网络中训练应该有不同的结果，所以我们分为两种情况，一类背景参与训练，另一类背景不参与训练。每种情况我们花费了 3 个小时完成了训练过程。表？展示了我们的

实验结果。从表？中我们可以看出，背景类别是否参与训练不影响评价结果，所以我们在训练过程中去除了背景类别，因为它浪费了训练时间和计算量。

因为 One-model method 对瑕疵的判决能力和阈值相关，于是我们在不同的阈值条件下来观察评价结果和阈值的关系。图？展示了我们的实验结果。从图？中我们可以看出，不同的阈值条件严重地影响了评价结果，因此它受主观因素的影响较大。

表？

| 背景是否参与训练 | IoU:0.50:0.95 | Iou:0.50 | F1-score | aIT | aIT of normal image | aIT of defect images |
|---|---|---|---|---|---|---|
| Yes | | | | | | |
| No | | | | | | |

表？

| 背景是否参与训练 | IoU:0.50:0.95 | Iou:0.50 | F1-score | aIT | aIT of normal image | aIT of defect images |
|---|---|---|---|---|---|---|
| Yes | | | | | | |
| No | | | | | | |

### 3.5. Two-model Method

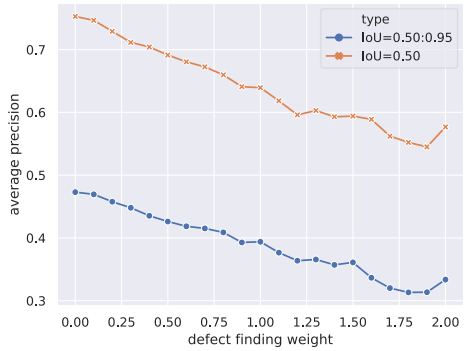
对于 Two-model Method，为了和其他的实验所用的模型保持一致，我们采用 ResNet50 作为我们的第一个模型，采用 Cascade R-CNN 检测算法作为我们的 second model，我们只统计 first model 和 second model 的推理时间用来评价，不包含任何的数据处理时间。其中，first model 不做任何的图像增强措施，second Model 只接受 first model 判断为瑕疵图像的图像作为输入。First model 用了 2 个小时完成了训练过程，second model 用了 3 个小时完成了训练过程。表？展示了我们的训练结果。从表？中我们可以看出，two-model method 耗时较多，极大的影响了整个瑕疵检测的推理时间。

### 3.6. Defect Network

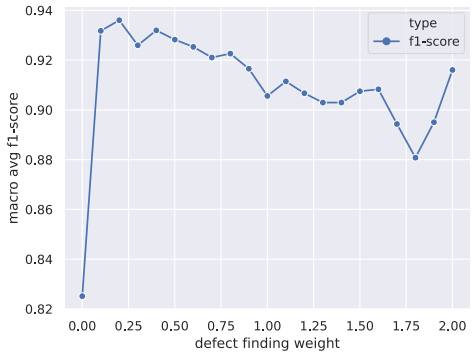
为了公平的比较，我们仍然使用 Cascade R-CNN 作为我们的检测框架，我们在 backbone 网络后面添加 DFN 网络就形成了所谓的 Defect Network。就模型数量来说，我们的方法属于单模型方法。每一次的模型训练过程花费了约 3 个小时。刚开始时，我们没有添加任何的改动，我们仅仅是将 DFN 加入到我们的目标检测模型中，得到的结果和我们想象的一样，速度比 One-model Method 提升了 10%，比 Two-model Method 提升了 10%，但是 mAP 却下降了 10%多，对应着图？中 defect finding weight 为 1，这困扰了我们一阵子。于是我们猜想是不是 DFN 网络过早的收敛同时 loss 在 DFN 网络中集中过多，于是我们引入了 defect finding weight 来平衡 DFN 网络和其他网络的损失。实验结果表明了我们的猜想是对的。

**Different defect finding weight.** 我们从 0-2 每隔 0.1 的间隔设置一个权重，总共 21 个权重。特别地，权重 0.0 实际上就是 One-model Method。图？展示了我们的实验结果。从图？(a)中我们可以看出，随着 defect finding weight 权重的增大，我们的 AP 结果接近线性减少，的确说明了 DFN 的权重影响模型的 AP，特别地在越接近 0 的地方 AP 值越高。从图？(b)中我们可以看出，在权重不为 0 的地方的 f1-score 比权重为 0 的 f1-score 要高的多，约提高了为 10%，说明 DFN 网络对瑕疵的检测能力要强很多，也间接说明了 DefectNet 的实用性。从图？(c)中我们可以看出，在非 0 权重的地方 all images,

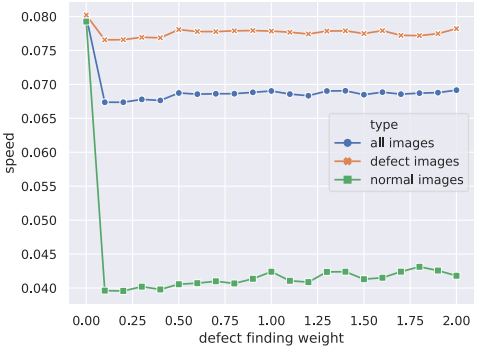
normal images and defect images 的单张检测速度几乎保持不变，但是都比权重 0 的耗时要少，其中对于正常图像，所花费的检测时间几乎少了一倍。从以上的结果中，我们建议 defect finding weight 设为 0.1。




(a)

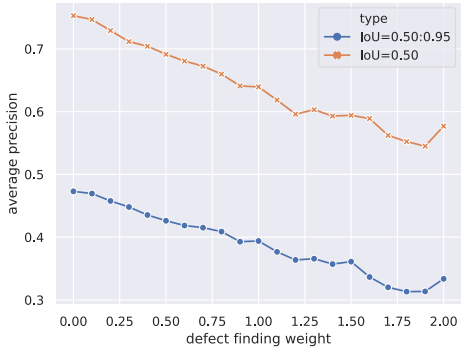



(b)

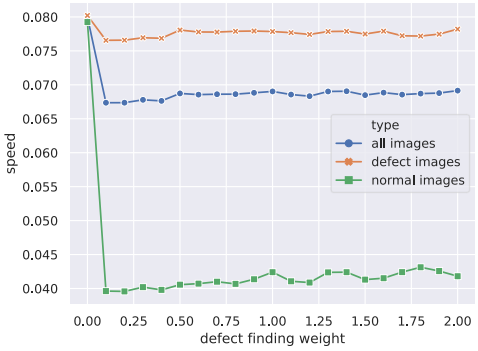



(c)

**Different number of normal images in test set.** 从图？(c)中我们可以看出，我们对速度的提升来自于对正常图像的快捷判断，作为验证，我们设置不同比例的正常图像：瑕疵图像的测试集做一次推理实验。图？展示了我们的实验结果。从图？中我们可以看出，我们的 DefectNet 的 aIT 和测试集中正常图像的数量呈负相关，当测试集不包含正常图像时，DefectNet 的推理时间和 One-model 几乎没有区别，这可以佐证 DefectNet 对速度的提升来自于对正常图像的快捷判断。



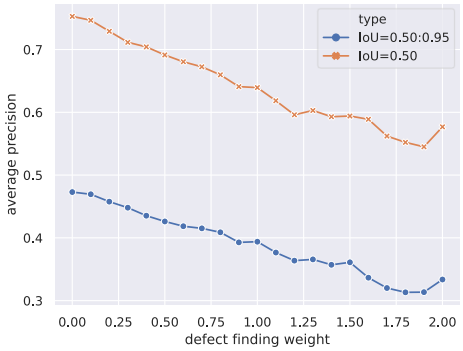

(a)



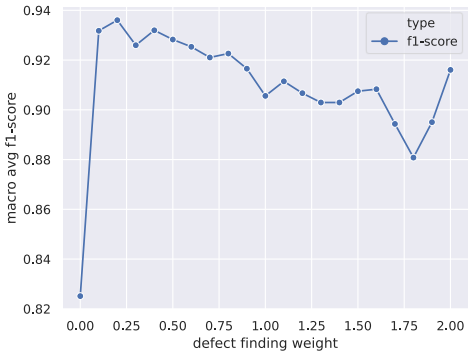

(c)

**Different number of normal images in train set.** 从图(c)？中可以看出，数据集中不包含瑕疵的图像的数量对测试结果有较大的影

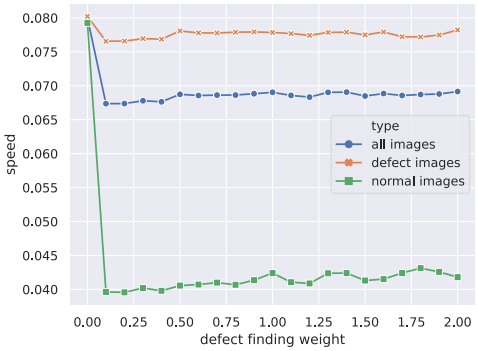
响，于是我们猜想，它会不会对网络的训练过程有影响呢？从而影响测试集的评价结果。于是我们设置了一组不同数量比例的正常图像和瑕疵图像的数据集实验，这里我们设置 defect finding weight 为 0.1。图？展示了测试集结果。从图？中我们可以看出。吧啦吧啦哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩吧啦吧啦哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩吧啦吧啦哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩吧啦吧啦哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩吧啦吧啦哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩哔哩。




(a)




(b)




(c)

## 3.7. Other Similar Dataset

为了充分验证我们的方法的有效性，我们在其他的数据集也做了实验。这里我们设置 defect finding weight 为依然为 0.1。

瓶装白酒数据集来自于数智重庆.全球产业赋能创新大赛（初赛），均拍摄于企业真实生产的产品，该数据集总共有 4516 张图像，其中，正常图像有 1146 张，含有瑕疵的图像有 3370 张。瓶装酒的瑕疵可分为五个大类，但是该数据集只有三个大类，总共十个小类，特别地，还包括一个特殊的类别——背景，不包含在瑕疵类别里面，它的作用只是为了区分正常的图像和包含瑕疵的图像，当一张图像不含有任何一类瑕疵类别时，该图像才为正常的图像。我们将该数据集按照 8：2 的比例划分为训练集和测试集，其中，训练集总共有 3612 张图像，包含 921 正常图像和 2691 张瑕疵图像；测试集总共有 904 张图像，包含 225 正常图像和 679 张瑕疵图像。

| 背景是否参与训练 | IoU:0.50:0.95 | Iou:0.50 | F1-score | aIT | aIT of normal image | aIT of defect images |
|---|---|---|---|---|---|---|
| Yes | | | | | | |
| N | | | | | | |

| o | | | | | | |
|---|---|---|---|---|---|---|

## 4. Discussion

为什么要提出瑕疵网络?
为什么瑕疵网络要快?
瑕疵网络

## 5. Related Works

## 6. Conclusions

## 7. Acknowledgment

Garbage:
Fabric Defects Dataset
In order to verify the efficiency of our models, we introduce a fabric defect dataset which was collected in the real textile workshop. The fabric defect dataset is composed of plain fabrics and patterned fabrics. The plain fabrics have ???? normal images and ???? defect images, the patterned fabrics have ???? normal images and ???? defect images. The number of plain fabrics defects categories and patterned fabrics defects categories is ?? and ?? respectively. Table ? and Table ? shows their name and number of per defects categories severally in detail.
For the purpose of
Table ? The name and number of per defects categories of plain fabrics

| Name | Number | buttonhole selvage | 13 | take marks | 26 |
|---|---|---|---|---|---|
| hole | 1??? | coarse picks | 14 | singeing | 27 |
| water stain | 2??? | looped weft | 15 | crinked | 28 |
| oil stain | 3??? | hard size | 16 | uneven weaving | 29??? |
| soiled | 4??? | warping knot | 17 | double pick | 30??? |
| three silk | 5??? | stitch | 18 | double end | 31??? |
| knots | 6 | skips | 19 | felter | 32??? |
| card skip | 7 | broken spandex | 20 | reediness | 33??? |
| mispick | 8 | thin thick place | 21 | bad weft yarn | 34??? |
| card neps | 9 | buckling place | 22 | | |
| coarse end | 10 | color shading | 23??? | | |
| loose warp | 11 | smash | 24 | | |
| cracked ends | 12 | roll marks | 25 | | |

~~从图? 中，我们可以看出，布匹疵点数据集的长宽比分布不正常。~~

Append Index

Table ? The name and number of per defects categories of patterned fabrics

| Name | Contamination | Mis-pattern | Watermark | Variegated wool | Sewing | Sewing head seal | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

| Name | Number |
|---|---|
| stain | 1 |
| broken figures | 2 |
| water stain | 3 |
| variegated wool | 4 |
| seam allowance | 5 |
| seam allowance marks | 6 |
| chongnian | 7 |

| hole | 8 |
|---|---|
| pleat | 9 |
| knit fault | 10 |
| through printing | 11 |
| wax spot | 12 |
| color shading | 13 |
| broken silk | 14 |
| others | 15 |