

论文总结

一 论文简介

最近阅读了论文 *Segmentation-based deep-learning approach for surface defect* (复现代码上传到了 <https://github.com/LiangXiMin2046/surface-defect-detect>), 根据名字就能看出来, 这篇论文讲的是基于分割网络来实现表面缺陷检测的深度学习方法。目前针对工业缺陷检测而设计的网络比较少, 这篇文章提出的方法考虑了不少实际工业中遇到的一些问题, 因此值得一读。

这篇文章提出的方法有以下特点: (1)需要的训练样本少, 只需要 25 到 30 个有缺陷的训练样本。在实际工业生产中, 能提供的缺陷样本也比较少, 这个方法比较切合实际, 同时, 由于所需要的训练样本少, 那么计算压力也相应减小了, 相比其他模型成千上万的训练样本。(2) 识别精准度高, 作者比较了文章提出的方法和一些其他主流算法在 KolektorSDD 数据集上的表现, 文章提出的方法性能是更优的。

二 网络架构

这篇文章提出的网络结构由两部分组成, 分别是分割网络和决策网络。训练的时候也是分别训练的。网络的结构图如下所示:

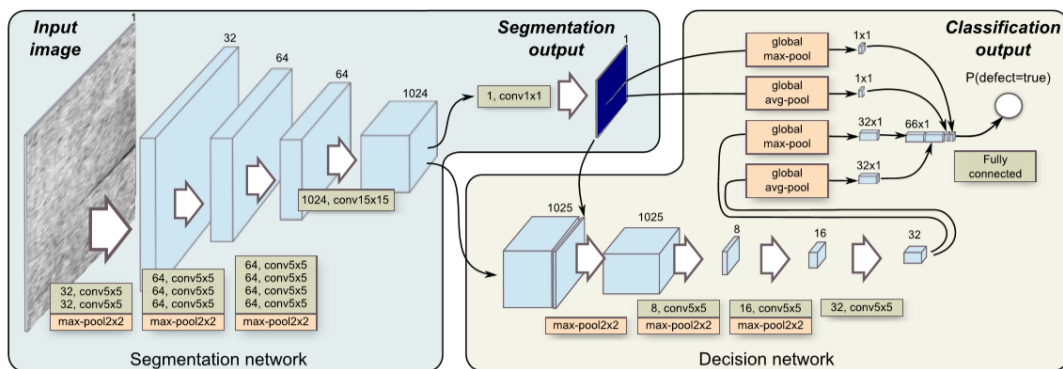


图 1-网络结构

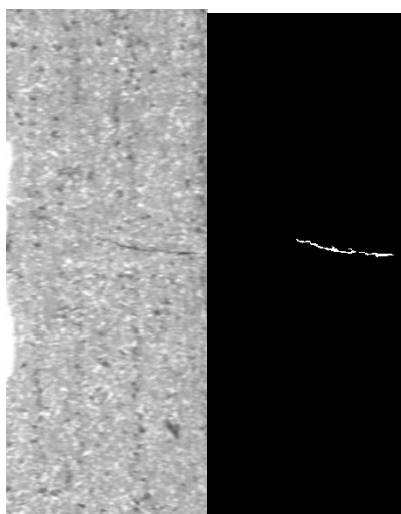
第一部分是分割网络(Segmentation network), 它的作用是实现图像的语义分割, 即将输入图像的缺陷部分显示出来。分割网络包含 11 个卷积层和 3 个最大池化层, 具体的细节如上图所示。第二部分是决策网络(decision net), 它是使用了分割网络的输出, 然后经过一系列卷积、池化、全局池化操作, 将数据降维为维度

为 66 的特征向量，最后使用一个全连接层分类，判定是否有缺陷，具体的操作流程如上图所示。

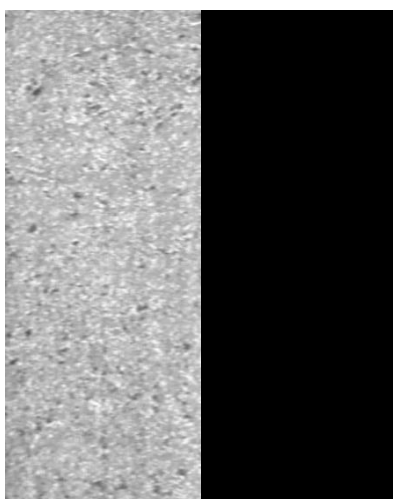
三 训练细节

数据集

训练所使用的数据集是 KolektorSDD 数据集，这个数据集一共有 50 个部分，每部分分为大概八张灰度图片和样本。如下图所示：



灰度图片和样本(有缺陷)



灰度图片和样本(无缺陷)

在数据集中大概有 50 个有缺陷的样本，每张图片的分辨率为 1200×500 左右。

训练参数方法

在分割网络中，选用的损失函数为逐像素交叉熵，作者通过实验证明了使用交叉熵的效果要好于逐像素均方差(mse)。在决策网络中同样是交叉熵。在训练中，比较有意思的是，一共要跑 50 个 epoch,在每个 epoch 中，采用的交替训练的方法，也就是一个缺陷样本和一个无缺陷样本，每个 epoch 中将训练集中的缺陷样本过一遍。在训练中，将数据集分为训练集(30 组)，测试集(20 组)。训练的 batch 设置为 1，也就是每个 step 每次只需要输入 1 张灰度图片。

评估指标

我们可以将网络输出结果分为四类：(1)有缺陷样本正确分类(TP) (2)有缺陷样本误认为无缺陷(FP) (3)无缺陷样本正确分类(TN) (4)无缺陷样本误认为有缺陷(FN)。在测试集上测完所有样本后，统计所有结果。

使用三个指标完成对模型的评估，首先定义 $accuracy = \frac{TP+TN}{N}$, N 是样本总数，

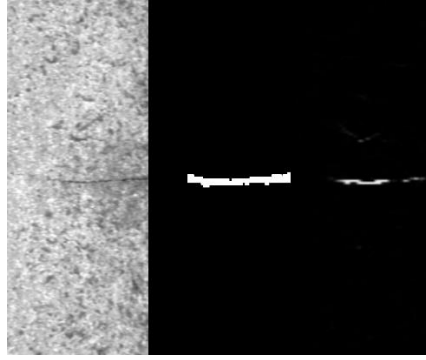
这个指标反应的是模型一个总体的正确率。定义 $precision = \frac{TP}{TP+FP}$ ，这个指标反应

的是有缺陷样本中有多少被成功检测了。定义 $recall = \frac{TP}{TP+FN}$ ，这个指标反映的是分

类为有缺陷的样本中真正有缺陷的样本。

四 结果分析

训练结束之后，在测试集上测试。测试集一共包含 160 幅图片，其中包括 22 幅有缺陷图片，138 幅无缺陷图片。使用测试集测试结果如下：(1)有缺陷样本正确分类(TP)为 21 (2)有缺陷样本误认为无缺陷(FP)为 1 (3)无缺陷样本正确分类(TN)为 138 (4)无缺陷样本误认为有缺陷(FN)为 0，那么可以计算出相关参数 $accuracy = 0.9938$ ， $precision = 0.9545$ ， $recall = 1.00$ ，效果不错。下图展示了分割网络的效果。从左向右依次是输入图片、分割网络输出以及 label。



输入图片-分割网络输出-label

五 总结

从总体上说，这篇文章提出的方法是很有可取之处的，它针对工业中遇到的很多问题进行了解决，尤其是在训练样本少和计算力有限的情况下，达到一个很高的正确率。同时这篇文章的提出的方法的不足也是有的，比如这个方法不涉及缺陷所在的位置的确定，只是一个单纯的分类，缺陷具体的位置需要观察分割网络的输出，如果说有指出缺陷位置的需求，这个方法是无能为力的。