|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 论文题目 | Software Visualization and Deep Transfer Learning for Effective Software Defect Prediction | | | |
| Paper URL | <https://doi.org/10.1145/3377811.3380389> | | | |
| Project URL | https://zenodo.org/record/3373409#.XV0Oy5Mza35 | | | |
| 综述/背景介绍 | 发展状况 | 原因 | 意义 | 关键词（速记词汇、信息索引词汇） |
| 发展状况：目前的软件缺陷预测研究主要集中在两个方向：新的特征提取方法和从大规模数据集中学习的分类方法。  原因：使用机器学习算法进行分类的缺陷预测技术有一些局限性，所需的功能工程非常耗时，需要在上游使用特殊工具，如代码复杂性分析、提交日志挖掘和代码结构分析工具。因此，在某些项目中很难捕捉到许多特性，比如例如隐藏在AST中的信息，这需要用额外的工具来构建和挖掘AST，并且，一旦提取了AST，源代码通常就不会被使用。  意义：提出了一种更直接的利用程序语义信息预测缺陷的方法：将源代码表示为图像，并在这些图像上训练图像分类模型。可以避免中间表示（例如AST）来改进缺陷预测，而是直接获取代码语义信息。 | | | 跨项目缺陷预测、项目内缺陷预测、深度迁移学习、自我注意、软件可视化 |
| 假设 | 通过视觉比较程序图像，可以有效识别两个程序之间的语义和结构相似性。 | | |  |
| 方法描述(含图) | 我们的方法，简而言之是将训练集和测试集的原始程序文件直接作为输入，并从中生成图像，然后用于构建缺陷预测的评估模型。具体来说是，由于基于CNN结构的网络模型的输入数据应该是图像的形式，因此构建了一个映射来将文件转换为图像。然后，使用AlexNet的前5层作为特征网，从图像中生成特征。浅层CNN将缺陷的存在与否与整体代码结构相关联，逐渐加深从函数/循环体到函数名称、标识符等的粒度。这些特征随后被输入注意层，用于分配权重和突出显示特征，哪些在分类中更有用。训练集和测试集的重加权特征用于计算MK-MMD，然后，将训练集的重加权特征输入到完全连通的层中，计算交叉熵作为分类损失。将MMD损失和分类损失的加权和反馈给整个网络，包括特征网、注意层和全连接层。最后，基于源代码可视化方法和DTL模型， 建立、训练和评估缺陷预测模型。（见图2） | | |  |
| 实验设计 |  | | |  |
| 数据处理 | 输入 | 筛除特例 | 处理方式 | 关键词（速记词汇、信息索引词汇） |
| Java源代码 |  |  |  |
| 结论 | 我们在缺陷预测的代码可视化和改进的深度迁移学习模型两方面做了贡献。并且我们在10个开源项目上的实验结果表明，将可视化方法应用于代码后，深度学习可以有效地直接应用于缺陷预测。具体来说，我们的方法DTL-DP是最先进的WPDP方法中的佼佼者。对于CPDP，DTL-DP将基于传统功能的最先进技术TCA+提高了29.0%（在F值中）。它还比基于深度学习的方法DBN-CP、DP-LSTM和DP-CNN分别高出8.8%、24.8%和15.5%。 | | | |
| 局限性分析 | 对于一些项目，出现负迁移问题，导致预测比直接预测更糟。两个项目之间的巨大差异可能会导致这种负迁移。减少负迁移的影响是未来需要解决的问题之一。此外，我们拥有的训练数据量很小，类不平衡是软件缺陷预测中固有的。我们计划在未来获得更多数据。 | | | |

2.论文总结：

训练基础写作能力，请依据上面的表格再写一段话来总结（有点类似于你来为这篇文章写摘要，可以参考头条作者：（一张图读论文），如附录的参考）：

论文总结参考格式：

**（论文名）：**有效预测软件缺陷的软件可视化和深度迁移学习

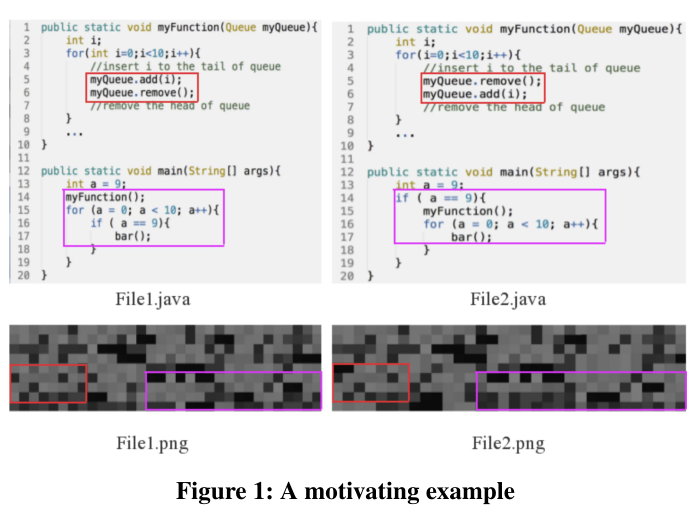
**（题目）**：Software Visualization and Deep Transfer Learning for Effective Software Defect Prediction

**（论文URL）：**论文URL:<https://doi.org/10.1145/3377811.3380389>

**（项目URL）：**https://zenodo.org/record/3373409#.XV0Oy5Mza35

**（总结）：**本文提出了一种端到端的深度学习框架DTL-DP,它是一种更直接的利用程序语义信息预测缺陷的方法，通过将源代码表示为图像，并在这些图像上训练图像分类模型。提出该方法的动机是通过避免中间表示（例如AST）来改进缺陷预测，而是直接获取代码语义信息。DTL-DP是将训练集和测试集的原始程序文件直接作为输入，并从中生成图像，然后用于构建缺陷预测的评估模型。我们在10个开源的Java项目上的实验结果表明，，我们的方法和其他项目内缺陷预测（WPDP）的方法相比性能更好一点，和跨项目缺陷预测（CPDP）相比，DTL-DP比基于传统功能的最先进技术TCA+提高了29.0%。比基于深度学习的方法DBN-CP、DP-LSTM和DP-CNN分别高出8.8%、24.8%和15.5%。 未来，我们将研究其他工具同样出现的的问题，减少两个项目之间的巨大差异导致的负迁移的影响和扩充训练数据量。

**（附图）：**



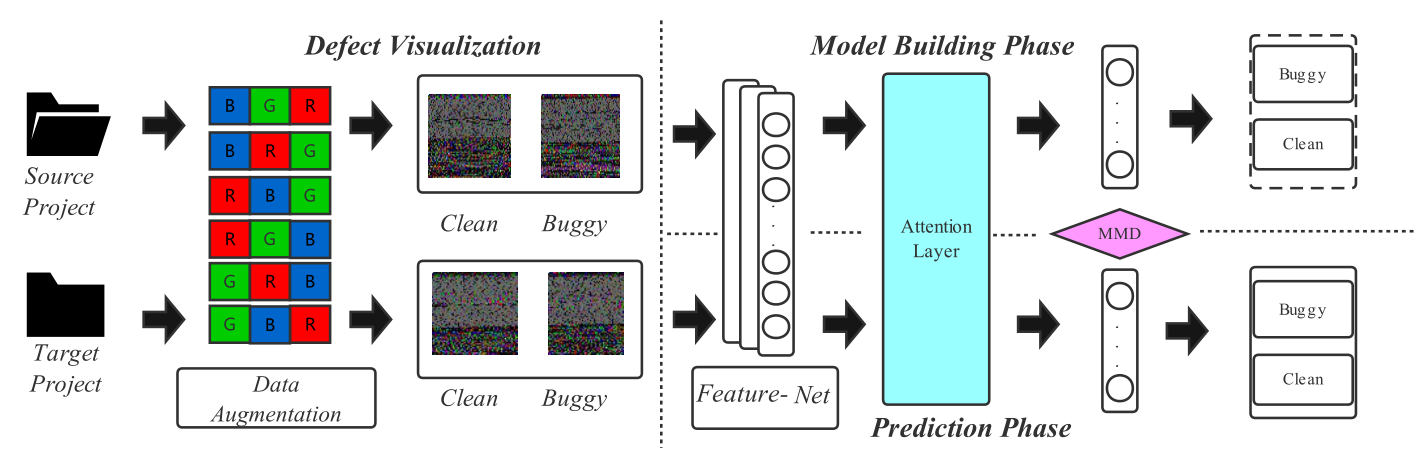


图2:DTL发展计划的总体框架

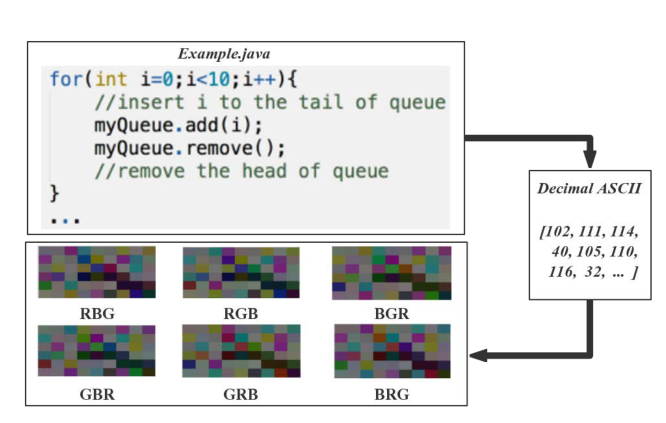


图3：将代码转换为彩色图像的过程

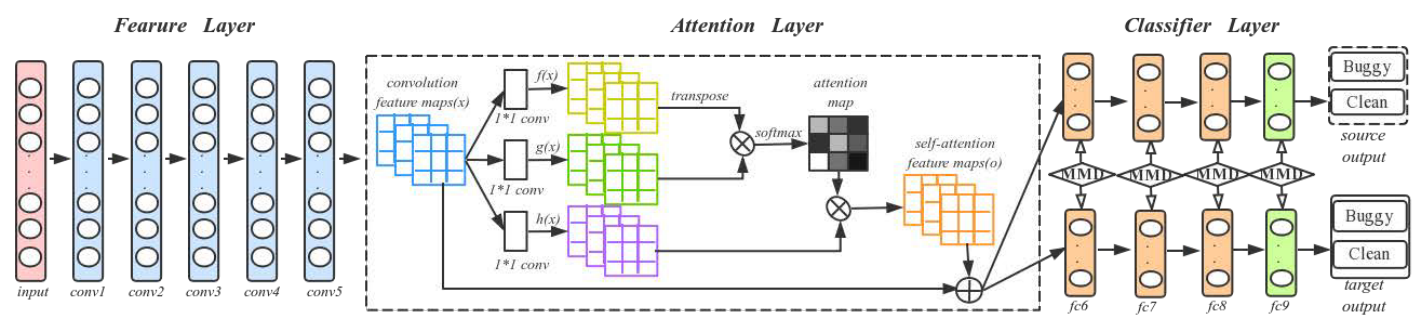
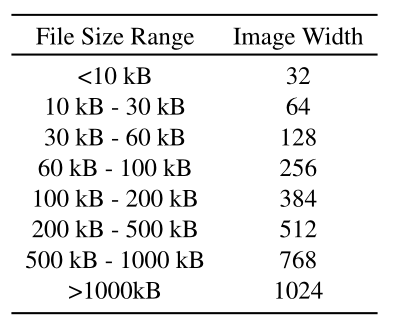


图4:我们的方法DTL-DP的训练和测试过程，灵感来自于DAN论文

表1:不同文件大小的图像宽度



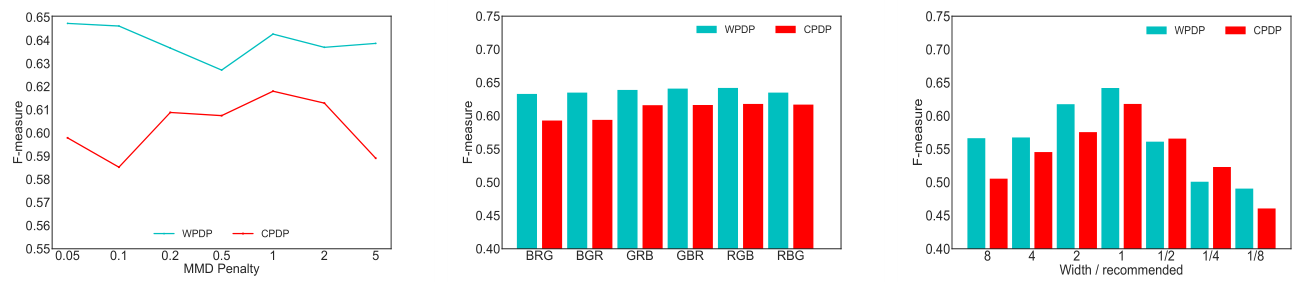


图5：帮助选择模型参数的敏感性研究

表2：数据集描述

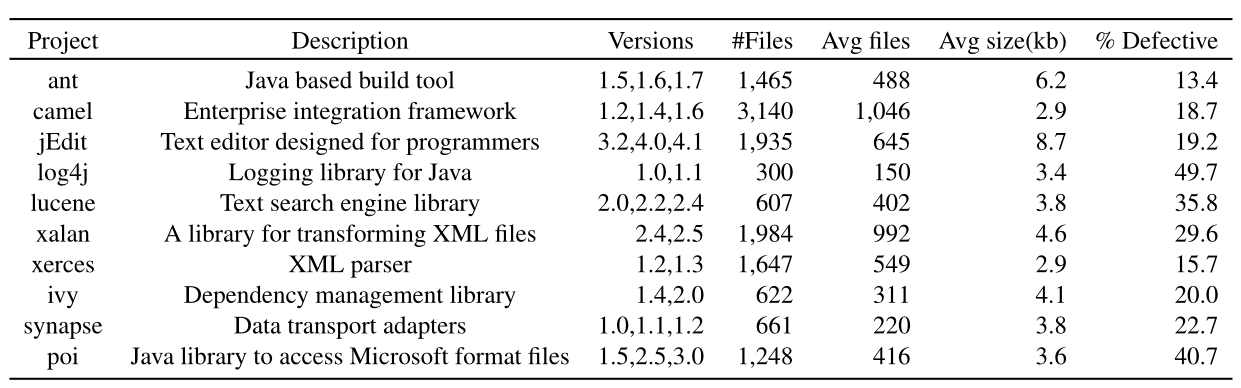
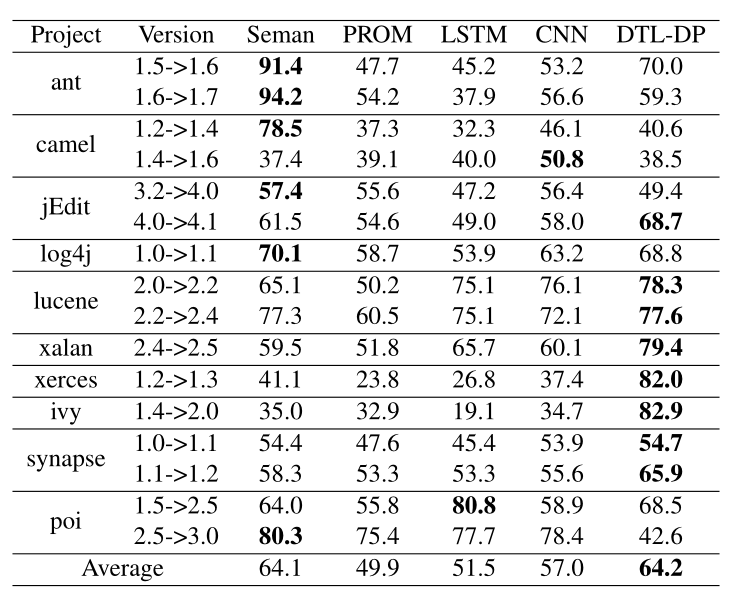


表3:WPDP的DTL-DP、语义（Seman）、PROMISEDP（PROM）、DP-LSTM（LSTM）和DP-CNN（CNN）的F值



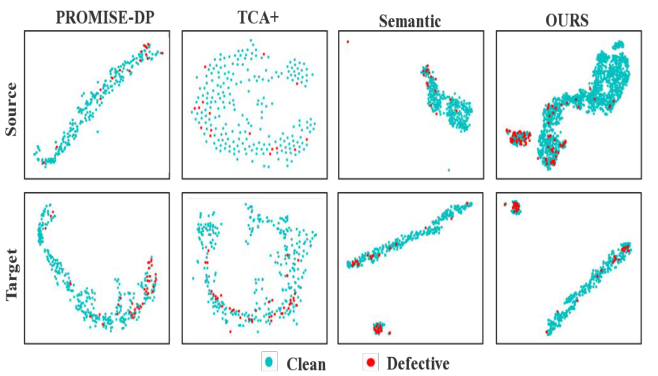
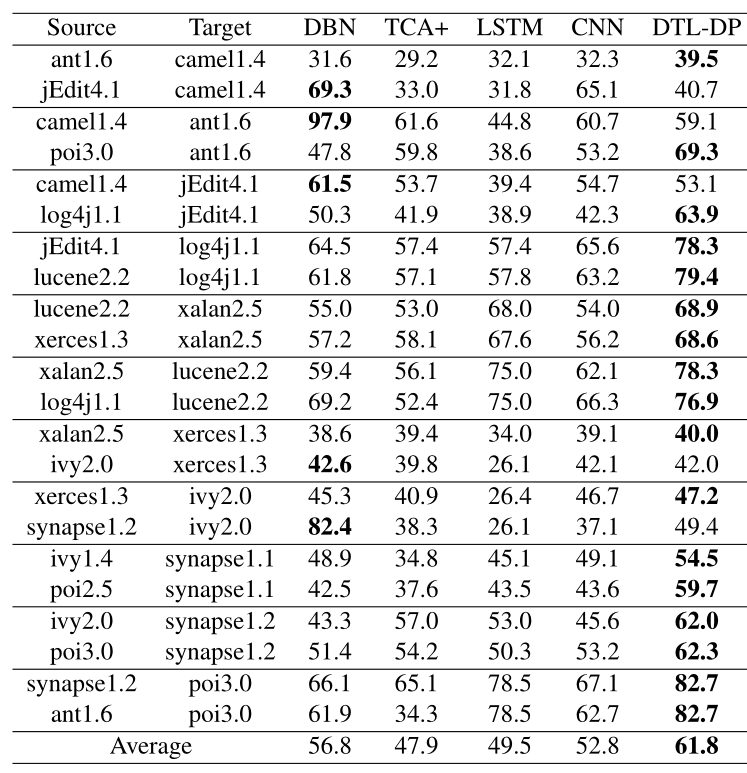


图6：源和目标特征的t-SNE映射（WPDP）

表4:CPDP中DTL-DP、DBN-CP（DBN）、TCA+、DPLSTM（LSTM）和DP-CNN（CNN）的F值



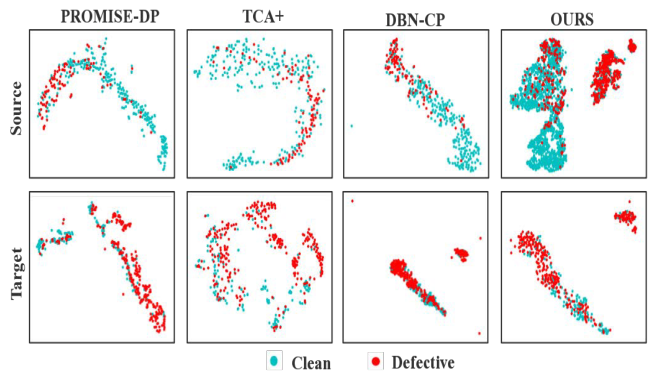


图7：源和目标特征的t-SNE映射（CPDP）

表5：三种机制对WPDP的贡献

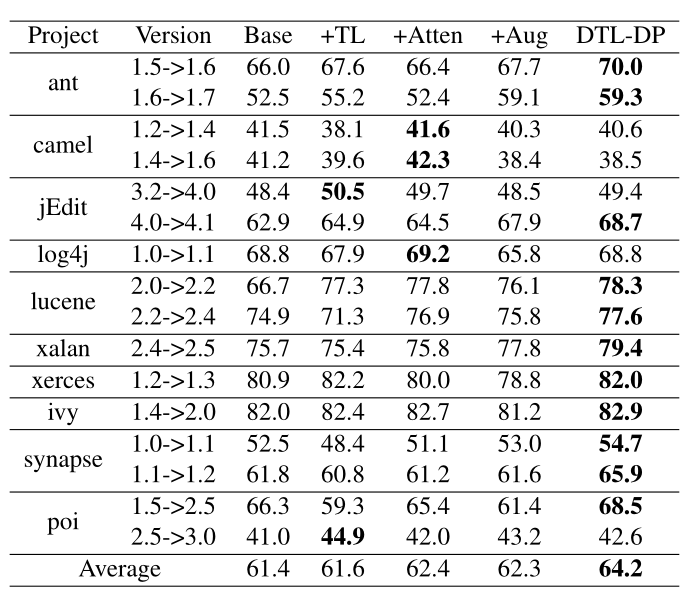


表6：三种机制对CPDP的贡献

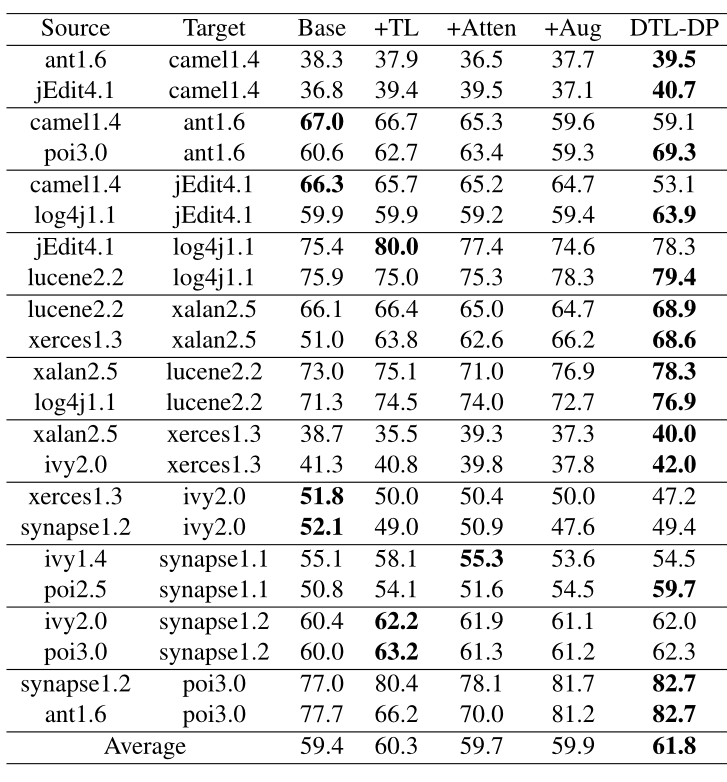


表7：三种机制的时间成本：缺陷可视化（Visualize.），自我关注（Attent.）迁移学习（TL）

