

烟火检测与识别

实验方案(中期)

无 91 李天骜 2019011559 无 97 张凯 2019011159

一、实验目的

在大作业给出的两个任务当中，我们计划选择任务一：辨别图片是否有火，并设计算法完成烟火识别与分类。对于烟火检测任务来说，现有的任务多是直接根据图片预测出烟火位置所在的掩膜(Mask)，从而直接得到烟火所在的位置，此种任务已经有多种比较好的实现，如 YOLO 算法，改进的 SSD 算法及更多的新颖的基于深度学习的网络方法。作业中给出的任务与之稍有不同，要求只需要判断是否有火即可，未明确要求给出烟火图像的掩膜，实际上降低了任务难度，但同时因为与现有的主流的烟火图像直接分割任务有所不同，也给予了自由发挥的空间，因此仍然是一个值得自主探索的图像分类和识别任务。

尽管实验的任务相对比较简洁易懂，本质上也是一个二分类问题，但是我们认为实验的目的绝不在于仅仅是取得了好的实验效果，因为如果只追求好的实验效果，使用更加深层的网络和更加复杂的现有的结构即可达到目的，但是，我们并不能够通过实验收获多少真正的课程所学的知识和方法。因此，我们认为，**实验的首要目标是能够灵活运用课堂所讲的大量数字图像处理的方法和技巧，包括与此任务强相关的物体检测一节的方法和现有的主流算法思想，并做适当的拓展，辅助以合理的创新。**其次需要追求的是算法的效率和准确率。当然，实验也将证明，如果达到第一个目标，第二个目标也会比较容易地实现。

实验的目的即主要包括以下几点：一，结合课程所学的知识，灵活运用课程所讲的传统和非传统方法，设计合理的算法框架并通过代码实现；二，在整体框架实现的基础上，通过模型的微调和算法的优化，改善算法效率，提高算法准确度；三，总结和梳理算法过程，以及实际实验中结果和预期值的差异即原因，并对算法的优缺点和改进空间做比较全面的分析。

二、实验规划与流程框架

经过前期的讨论和相关资料梳理，我们目前计划的算法流程如下：

1.对于数据前期的处理

(1)颜色变换预处理

将每一个彩色图片在 RGB 空间中的值 $P_i = (R_i, G_i, B_i)$ 转化为 YCbCr 空间中的值 $P_i' = (Y_i, Cb_i, Cr_i)$ ，此后再对 P_i' 进行分析。这样的好处在于 YCbCr 空间

中颜色成分更有利于从颜色角度区分出属于火焰的像素和其他像素。

(2)纹理分析预处理

将图片通过 super pixel 算法提取图像中的边界,并图片按照边界进行分割得到图像块。采用边缘提取算法对图像增强,便于后续纹理的提取和分类。

(3)其他预处理和增强方法

在实验的过程中,也会采取其他的大量的图像预处理和增强的方法,比如图像旋转、尺度变换、边缘锐化、直方图均衡等等方法,争取能够提高算法的鲁棒性。

2.对于区域的选取策略

由于真正测试的图像都是包含烟火和背景的图像,比训练集仅仅包含烟火的图片要大得多,因此,如何分割图像区域,就成了一个必须要解决和考虑的问题。

目前,我们的想法有以下几点,并准备应用到不同的分类器上:

(1)借鉴课堂所讲 YOLO_v1 算法,将图像分成 $S \times S$ 个格子,每个格子预测 B 个 bounding boxes,以此将图像分割成有烟火和无烟火的区域,并用于之后的训练和检测环节。

(2)对于纹理分类器,可以借鉴超像素(super pixel)的方法进行分割,将不同的区域按目前成熟的超像素方法进行分割,从而得到纹理相近的不同连通区域,进而提取每个区域的局部纹理特征,与火焰的局部纹理特征做比对,得到分类结果。

(3)对于像素级分类器,无需做太多的分割,可以直接以像素为单位进行判别,并最后采用并查集算法确定连通区域的大小(设置大小阈值防止噪声干扰),进而判断是否有火焰存在。

(4)对于其他的图像分割和分块方法,目前还未确定,需要实验中进一步尝试。

3.Cascade 级联策略

如果可以,将会借鉴 Cascade 级联的策略,在算法正式分类之前加入简单的前置过滤层,以滤除“绝对不可能”是烟火的图片,并不送入接下来的分类器和网络进行预测。设计此前置过滤层也比较容易实现,只要满足所有的“有火”图片不被判断为“无火”即可,而不关心“无火”图片的判断结果,即假阴性(FN)为 0 即可,这样即可实现部分“无火”图片在前置过滤层即被滤除,不再进行之后的算法流程,主要是提高了算法的效率。关于此策略是否是能够提高算法准确

率的问题，则有待之后的消融实验验证。

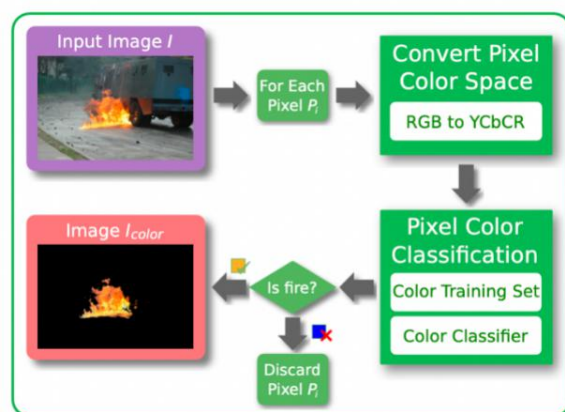
4. 对于不同弱分类器的实现

4.1 浅层 CNN

使用一个 4 层的浅层 CNN 网络对图片进行分类。首先在训练集的 50×50 图片上进行训练，接下来借鉴 YOLO_v1 算法，将 dataset 中的图像分成 50×50 的格子，来检测是否存在火焰的特征。

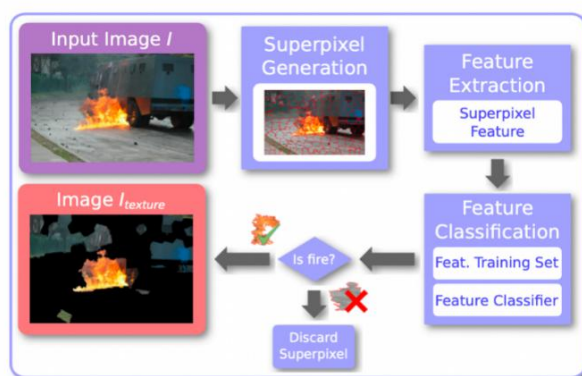
4.2 色彩成分分析

使用色彩变换将图片转化为，再将每一个像素通过颜色变换转化到 YCbCr 空间，并将每一个像素通过训练好的颜色分类器，以分类的到属于火焰的像素。最终可以根据分类结果确定火焰区域。算法流程如下：



4.3 纹理分析

将图片通过 superpixel 算法提取图像中的纹理边界，并图片按照边界进行分割得到图像块。训练模型对分割后的图像块进行分类以确定图像块是否为火焰。最后综合火焰图像块的位置来确定是否有火焰以及火焰区域，算法如下：

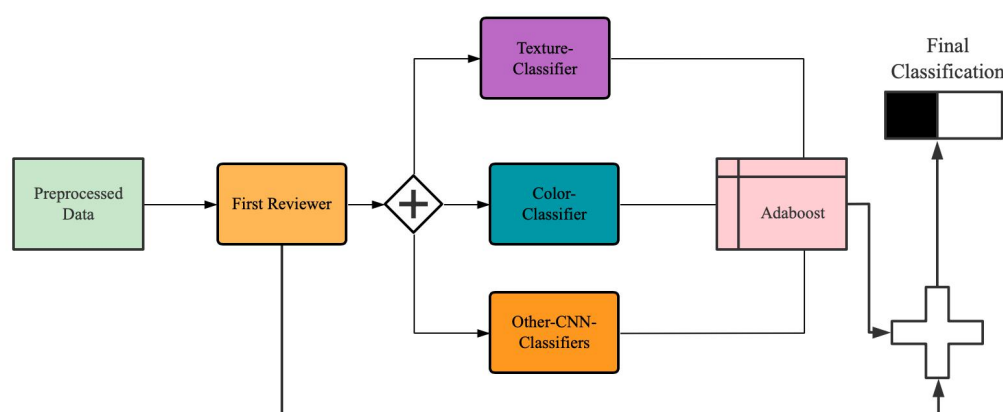


5. Adaboost 集成算法框架

前文所述分类器未必有好的效果，因此，我们借鉴课堂所讲的 Adaboost 集成算法，在最后增加集成判断模块。

当上文所述分类器都训练结束之后，并且都取得了比“盲猜”效果好的分类结果，则可以选择此分类器。最后 Adaboost 算法实际相当于训练了不同分类器的分数权重，进一步提高了总的算法准确度，Adaboost 算法实现按课件所讲进行设计和编写。

以上即为目前设计的整体算法流程框架，为了更加清晰地表示算法的流程，完整的算法流程框图如下：



三、实验任务安排

1. 进行简单分类器的训练尝试(简单分类器设计)

首先设计的是简单的分类器，比如尝试 LeNet-5 浅层网络分类器、基于火焰像素值的简单分类器、基于图像纹理的简单分类器等，尝试并改进这些简单分类器的分类性能，当训练到准确率大于盲分类的准确率时，按照最终 Adaboost 集成的算法思想，即可选取此分类器。当然，在实际的实验流程中，也希望能够尽可能高地提高分类器的准确率。

2. 尝试多种数据预处理与增强方法(提高算法鲁棒性)

采取其他的大量的图像预处理和增强的方法，比如图像旋转、尺度变换、边缘锐化、直方图均衡等等方法，以提高算法的鲁棒性。

3.展开消融实验分析(验证各模块独立的功能)

由于我们目前基本还是在初步的设计阶段，还缺乏大量的实验验证，因此，所设计的框架虽然有很多模块，但并不能保证每个模块对最终算法结果的实际贡献的正负，因此，在完成整个算法流程框架的搭建和基本的测试调整之后，需要做一定的消融实验，即单独摘除算法中的某个模块，并对照验证算法的结果。根据消融实验，我们可以较好地分析每个算法模块的功能，并用于之后的模型调整和算法流程修正。

4.对网络结构和算法流程进行合理简化(提高算法效率)

根据消融实验的结果，对作用不大或者冗余的模块进行合理的改进和删减，简化算法流程，提高算法效率。

5.总结算法流程与实验过程出现的问题及改善方法

通过前期的实验设计和后期的实验中实际出现的问题，进行总结和梳理，并记录算法改进的方法和策略。最后，简要分析算法的优点和局限性，对算法的可拓展性做一定的展望。

参考文献

BoWFire: Detection of Fire in Still Images by Integrating Pixel Color and Texture Analysis, Daniel Y. T. Chino, Letricia P. S. Avallhais, Jose F. Rodrigues Jr., Agma J. M. Traina Institute of Mathematics and Computer Science, University of Sao Paulo.