利用ffmpeg，提取视频特征进行帧内预测模式的隐写分析

原创

顾哥NET 于 2018-05-02 16:17:06 发布 2628 收藏 5
分类专栏： 视频编码 隐写分析 文章标签： 视频隐写 隐写分析

本文链接：https://blog.csdn.net/qq_37366877/article/details/80165142

前言

终于写到这个毕设比较核心的部分了，虽然这个部分比较简单。

我这里参考的论文是：

Zhao Y, Zhang H, Cao Y, et al. Video Steganalysis Based on Intra Prediction Mode Calibration[J]. 2015 International Workshop on Digital-forensics and Watermarking (IWDW), 2015(IWDW).

是一篇中国科学院信息工程研究所当时的一名硕士生发表在Springer上的一篇论文，由于当时写论文的学生已经毕业，当时试验用的视频并没有找到，以至于论文中的实验难以复现。但还是感谢一下论文作者之一的曹纭老师提供了未隐写和修改了10%IPM隐写后的264视频作为我这次课题的部分实验样例。

为了保证我文章中的实验可以复现，在课题结题后我还是会公开所有代码以及实验用的视频（尽管代码很丑陋）。

正文

这篇文章中我基本不考虑讲代码了，如果看了我写的如何实现隐写算法的文章后，提取特征的代码应该不会太难。

1.视频隐写分析

视频隐写分析其实就是给一段视频，判断视频是否被嵌入过隐藏信息。当然对隐藏信息的具体内容是不必要提取出来的，因为隐藏的信息很有可能被加密过。

H264的视频隐写域主要有四个：1. 帧内预测模式 2. 运动向量 3. DCT系数直方图 4. 熵编码

当然每个隐写域的隐写算法和隐写分析算法都不一样。

显然我做的主要工作都是在帧内预测模式这一块，而帧内预测模式隐写都是通过修改4*4亮度宏块预测模式达到信息嵌入的目的，所以在分析的对象也主要是4*4的亮度宏块。

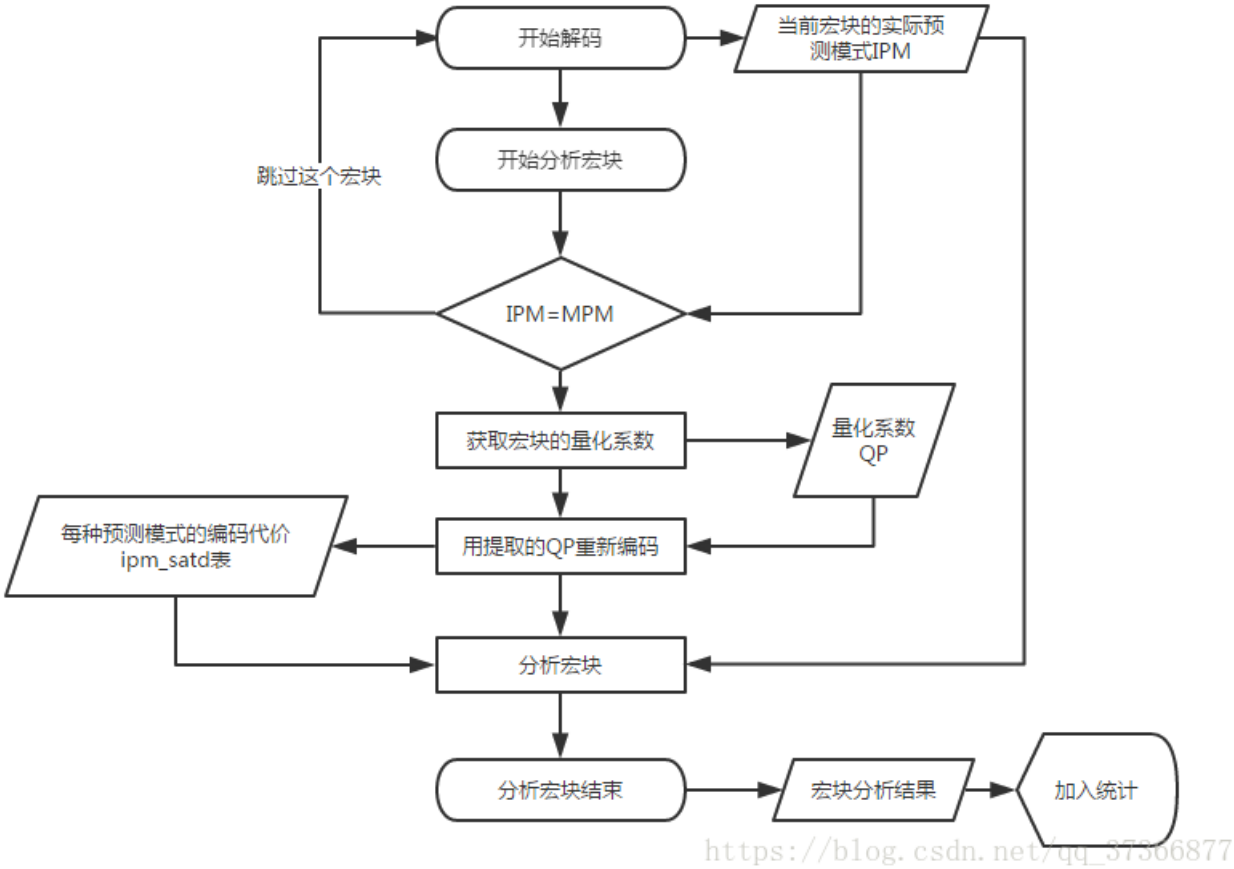
2. 帧内预测模式隐写分析算法

2.1 论文中的算法

这里我先介绍一下前言中所述论文的的算法，也给个这篇论文在百度学术的传送门。

注：以下提及宏块不特殊说明都认为是4*4宏块

这篇文章主要提出的算法的流程如下图



注：ipm_satd表反映了用不同预测模式的编码代价

我们接着来讲这篇论文中如何利用ipm_satd表分析宏块，以及统计的内容具体是什么。

ipm_satd表中记录了所有预测模式相应的编码代价，论文中主要提取了其中两个数据

1. 当前实际的IPM在ipm_satd表中对应的satd值（编码代价）从小到大排第几。（取值1-9）

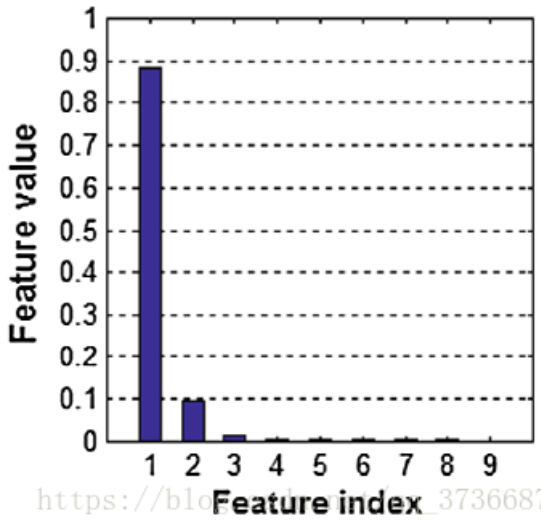
2. 当前实际的IPM对应的satd值相比最小的satd值增加了百分之多少。

如果实际IPM的satd值不是表中最小的，我们就把这个宏块称为参与校正(calibration)的宏块。（“校正”为论文中的表述）

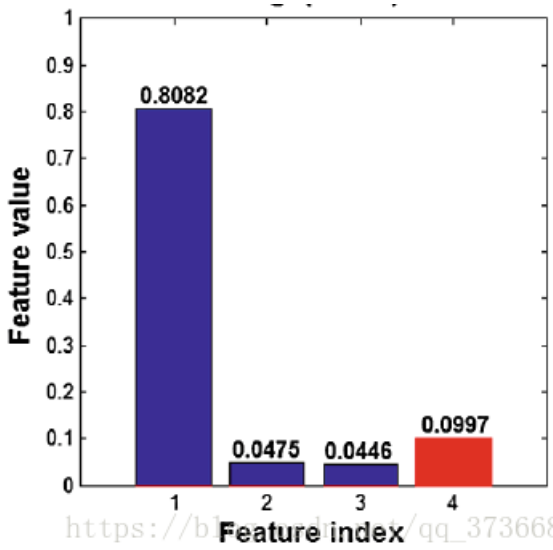
理想情况中，IPM的satd值在ipm_satd表中都是最小的，如果是这样，增加率也为0%。

实际情况中，如果是未隐写过的视频，参与校正的宏块比例非常少，而隐写过的视频，参与校正的宏块比例明显高于未隐写的视频。

对于提取出来的第一个数据（排名）进行统计，我们很简单地就能构造出一组9维特征，分别表示相应排名占所有宏块的比例。



对于第二组数据（satd增加率），由于是一组连续的数据，我们很难直接对它进行统计。论文中所采用的方法为将数据离散化，控制一个变量β，分别统计增加率在0β-β，β-2β，2β-3β，大于3β，四种情况的宏块占所有宏块的比例。



具体β的取值根据不同码率而定，论文中给出的参考值如下

Table 1. The reference value of β at different bit rate

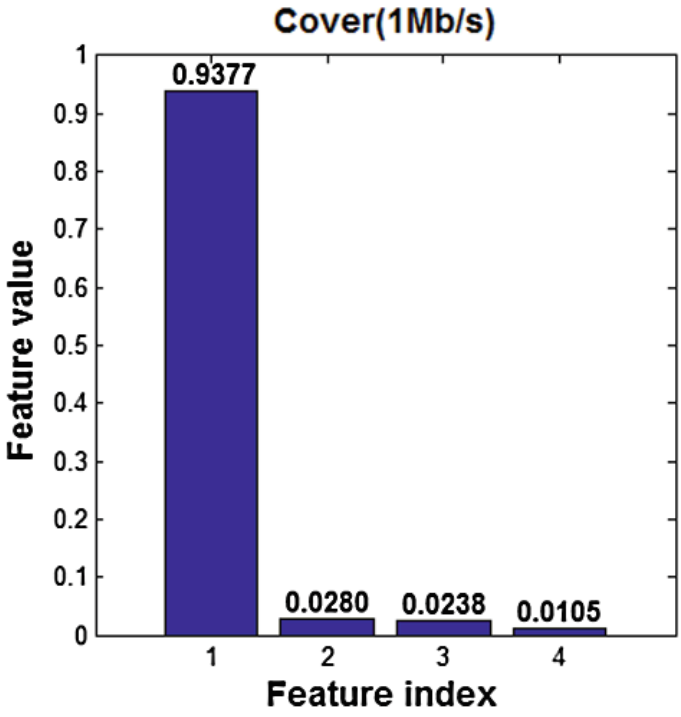
The reference value of β at different bit rate						
Bit rate(Mb/s)	0.1	0.2	0.5	1.0	2.0	>2.0
β	0.08	0.06	0.05	0.04	0.03	0.025

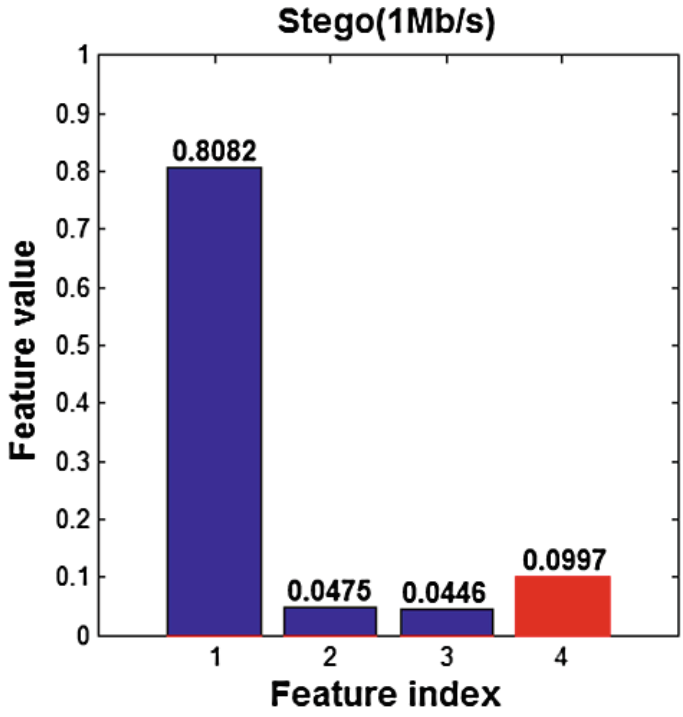
经过对所有数据的统计，我们最终一共得到了一组9维，一组4维供13维的特征。将其送进SVM中进行训练，分类。

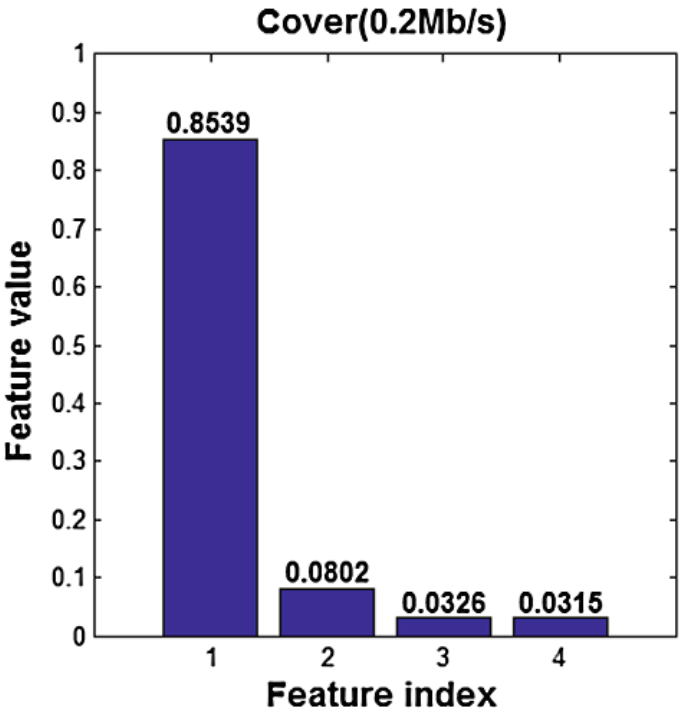
论文中将这13维的特征称为IPMC特征

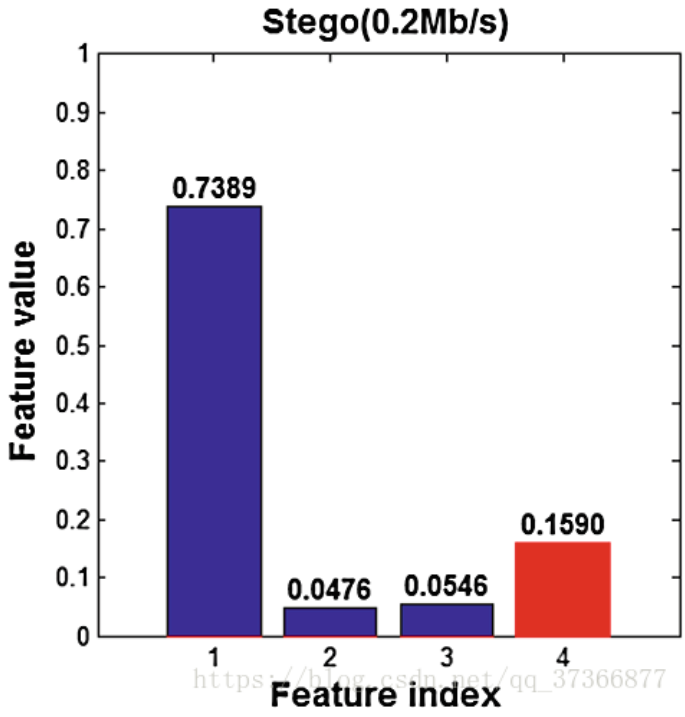
2.2 上述算法的改进

上述算法的特征其实有一定局限性。由于当控制不同码率，不同gop大小下的视频参与校正的宏块比例会有较大不同，所以当用上述论文中提出的特征进行分类时，必须建立对应码率以及gop大小的数据集。尽管第二组特征中为不同码率设定了不同的β值，但是我们会发现不同码率下的特征大小依旧有相当大的不同



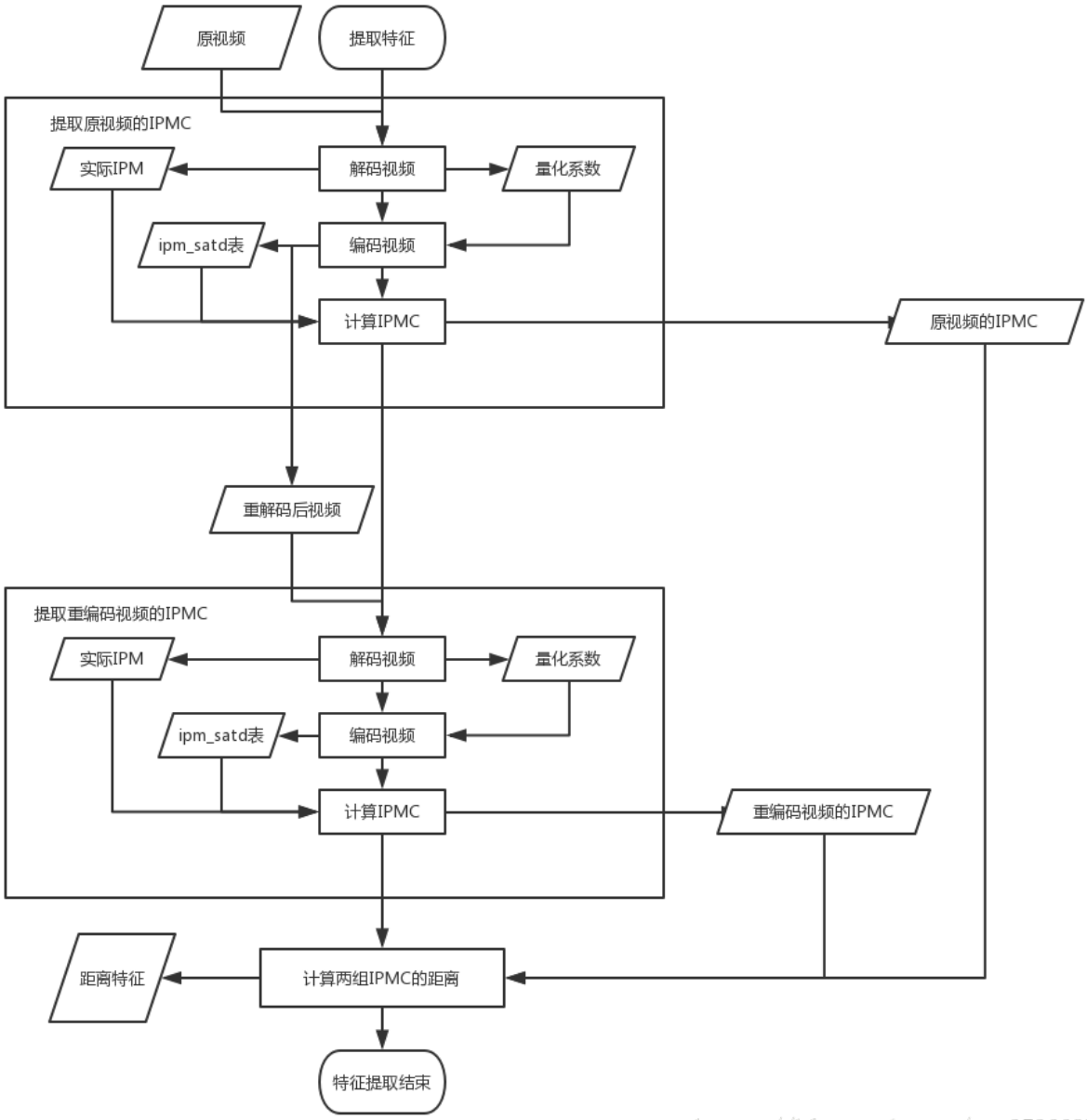






所以我这里提出一种改进算法，能很大程度消除码率不同给特征带来的影响。

大致流程图如下：



由于重编码的视频一定是未经过隐写过的，所以可以成为与原视频对比的一个非常好的参照，我们可以通过提取出原视频和冲重编码视频两组IPMC，取其每一维度的距离，作为特征送进SVM中训练。

接着我们稍微讲一下距离的计算：

由于IPMC中每个维度都是概率，两个概率之间直接取差也不是非常明智的做法。显然90%到99%的跨度和0%到9%的跨度比50%到59%来的重要得多。所以我们这里用sigmod的反函数将0-1的概率投影到整个实数域。

注：sigmod(x)=1/(1+e^-x),sigmod^-1(x)=-ln(1/y-1)

经过投影的概率直接作差能更好地反应概率变化的重要性。所以我们将每一维投影后的差值作为特征，送进SVM训练分类。

3. 分类结果

2. 帧内预测模式隐写分析算法

实验用数据集是自己实验室学长给的30短视频，切分为166个100帧的视频，以及曹纭老师提供的30个隐写过，30个未隐写过的60段视频

训练分类器：台湾大学开发的libSVM

分类器参数：

```
svm_type = C_SVC
kernel_type = POLY//核函数
degree = 3
gamma = 4
coef0 = -0.117
nu = 0.1
cache_size = 1000000
C = 2.5//惩罚因子
eps = 1e-6//松弛变量
p = 0.1
shrinking = 1
probability = 1
weight_label = NULL
weight = NULL
```

训练集包括：

切分的166段视频分别按照隐写/不隐写，不控制码率/控制码率1Mbps/控制码率0.2Mbps，166*6段视频

切分的166段视频按照不控制码率50%概率修改IPM，不控制码率25%概率修改IPM 166*2段视频

共166*8=1328段视频。

测试集为曹纭老师提供的10%IPM修改的30段隐写后视频与30段未隐写视频

TP：22/30=73%

TN：29/30=97%

总正确率85%

最后

最后还是呼吁一下在机器学习/深度学习相关方向的研究者大胆公开自己的源代码和训练集，使实验可以复现，这样才是真正促进这个领域的发展。