



中山大學

Computer Vision Experiment Report

计算机视觉实验报告 2

姓名: xxx

学号: xxx

邮箱: xxx

Experiment Report Regulations(实验报告规章)

1. This is a personal assignment. Each student MUST submit only the soft copy of the report via sending an email to cvexpsysu@163.com and the topic of this email should be "Your studentID_name_ Exp_report1", otherwise, your email will be swallowed by hundreds of emails in the mailbox. The soft copy will be used as the ONLY material to mark, if you miss submitting it, you have to take all the responsibility. 这是一次个人独立完成的考核作业，每一位同学都应在提交截止日期前将作业的电子版通过邮件发送至 cvexpsysu@163.com。邮件主题应为“你的学号_姓名_Exp_report1”，否则主题不明的邮件将会被淹没在邮箱中的其他邮件中。该电子版作业将被直接批改计分，对于没有提交电子版的同学，后果请自负。
2. A cover sheet can be created in your own way but the following information must be included: Your (or all the group members') full name, student ID number, and email address. 作业的封面可以设计，但请包含以下重要信息：你（或者全部小组成员）的中文全名、学号和邮箱。
3. You may refer to textbooks, lecture notes, and the Internet to discover approaches to problems, however, the lab report should be your team's own work. Any plagiarism is NOT ACCEPTABLE. Once we find that, then you will get a 0 mark on this lab report. 在撰写报告的过程中你也许需要参考课本、课件和网络，这是可以的。但是请确保这份作业是你独立完成的。任何形式的抄袭或作弊都是不被接受的，一旦发现本作业将按 0 分处置。
4. Assignments may be accepted up to 5 days after the deadline has passed; a late penalty of 5% will apply for each day late without an extension being granted. Submissions over 5 days late will not be marked. Emailed submissions will NOT be accepted without exceptional circumstances. 晚于截止日期 5 天内提交的电子版作业仍可受理，但每晚一天总成绩减少 5%（即晚一天提交满分为 95 分，晚两天提交满分为 90 分），以此类推。晚于截止日期 5 天后的提交作业视同无效提交，将不会被批改和打分。

1 实验目标

- 使用 YOLO 模型在足球比赛的视频中实时检测并标注出人和球。

2 实验原理

2.1 YOLO 模型

“You Only Look Once”，简称YOLO，是一种使用卷积神经网络进行目标检测的算法。与识别算法相比，检测算法不仅预测类别标签，还检测对象的位置。因此，它不仅将图像分类到一个类别中，还可以在图像中检测多个对象。该算法将单个神经网络应用于整个图像。这意味着该网络将图像分成区域，并为每个区域预测边界框和概率。

2.2 评价函数

准确度：模型在给定数据集上正确分类或预测的样本数占总样本数的比例

召回率：真正例(Tp)与所有正样本(Tp+Fn)的比值，即 $Recall = TP / (TP + FN)$

F1函数：F1分数是一种衡量分类模型效果的指标，它综合了模型的精确度和召回率。计算

公式： $F1 = 2 * (Precision * Recall) / (Precision + Recall)$

3 实验步骤

3.1 数据准备

使用实验课程提供的比赛视频作为评估最终结果好坏的测试集，另外选用从Github, Roboflow等网络资源获取的特定的足球比赛数据集对模型进行微调。

3.2 模型选择与训练

- 我首先对测试视频进行了分析，发现其主要特征：

- a) 拍摄角度为远距离视角，这样的视频里足球和球员所占的pixel都比较少，特别是足球，非常小。因此在选择训练集微调的时候我倾向选择相似类型的图片/视频（如图左）而不是下图右的那种类型。



b) 视频的清晰度比较差。考虑这点，用以优化的想法是对测试集进行提升分辨率等预处理或者重裁一下它的尺寸，看看是否能使效果更好。

➤ 模型的选择

由于YOLOv8x在各大数据集下的训练效果都比较好，所以选择了它作为本次实验的核心模型。[\[https://github.com/ultralytics/ultralytics\]](https://github.com/ultralytics/ultralytics)

See [Detection Docs](#) for usage examples with these models trained on [COCO](#), which include 80 pre-trained classes.

Model	size (pixels)	mAP ^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

➤ 模型训练过程

先使用通用的足球比赛数据集对YOLOv8x模型进行微调，训练了150个epochs。

核心代码：

```
from ultralytics import YOLO

# Load a pretrained YOLOv8n model
model = YOLO('yolov8x.pt')
model.train(data='/content/drive/MyDrive/labs_baogao/data.yaml', epochs=150,)
```

最后可以用以下代码评估模型：

```
model.val(data='/content/drive/MyDrive/labs_baogao/valid')
```

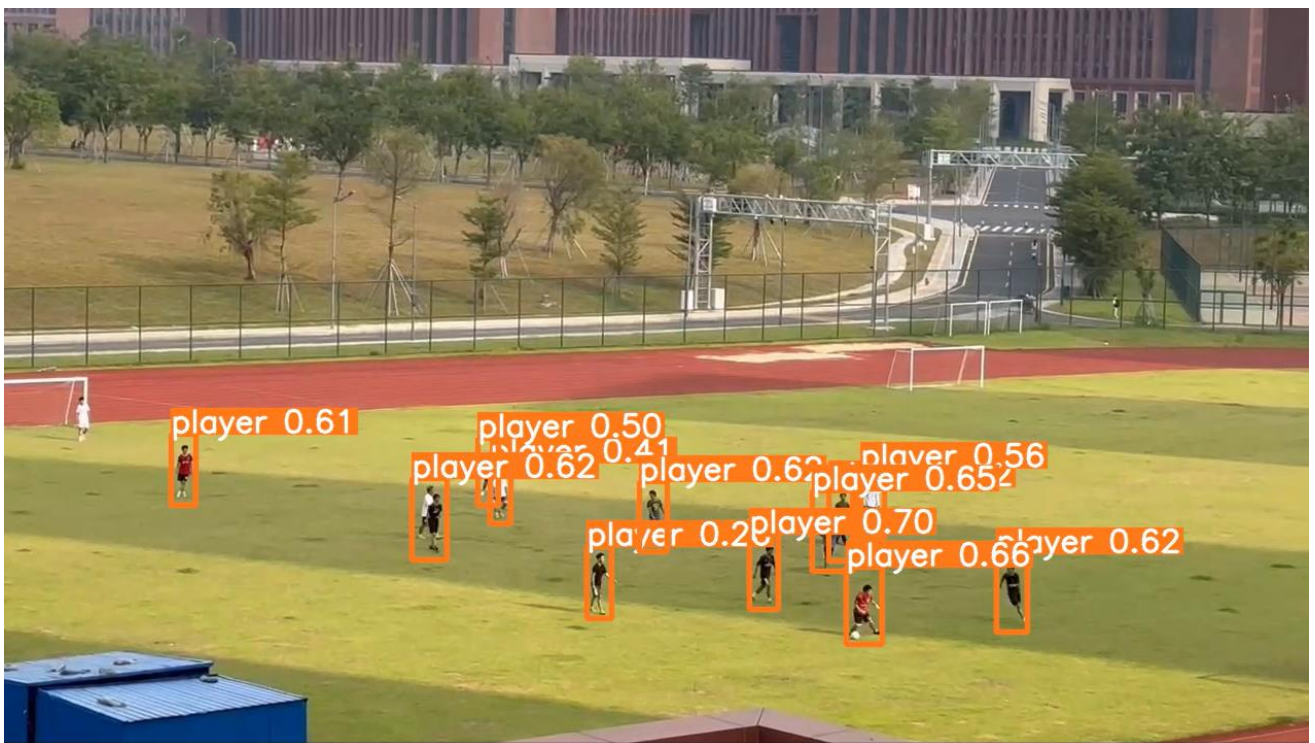
3.3 视频处理与目标检测

读取视频，并对每一帧应用YOLO模型进行目标检测，标注出检测到的人和球。核心代码如下：

```
model.predict(source='/content/drive/MyDrive/try1/football.mp4', save=True, show=True, save_txt=False)
```

3.4 结果展示

使用标注后的新视频效果如下：

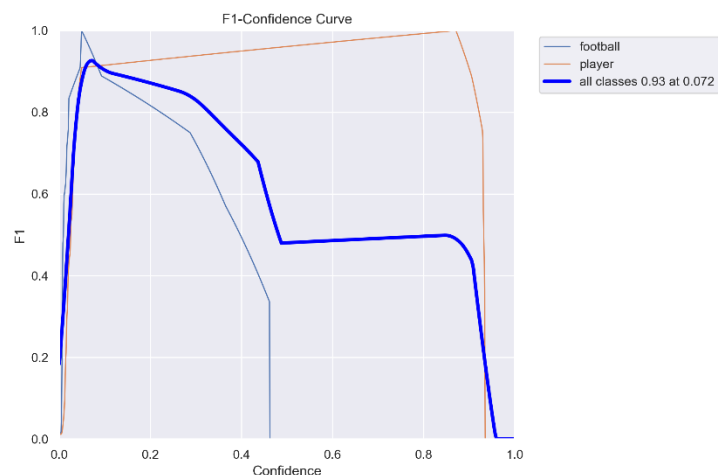


完整视频的文件：

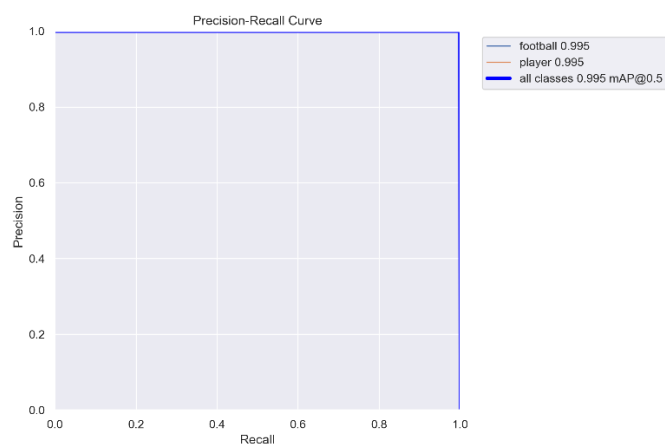


football.mp4

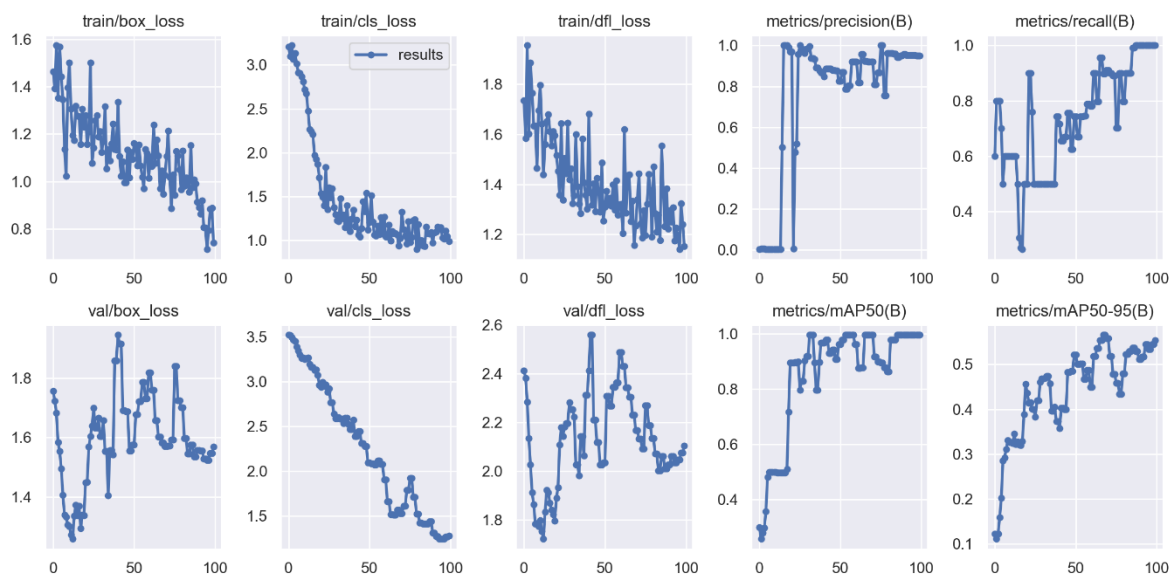
3.5 性能评估



F1曲线的最大值达到了93%，而AUC也有不错的水平。另外该曲线表现出这个模型更加关注高召回率。



P-R曲线越靠近右上角，则学习器的效果越好，这里我们可见该模型拥有优越的性能。



以上是模型训练和验证过程中的各损失变化曲线，可以看到曲线的趋势均在向使模型更优越的方向发展。

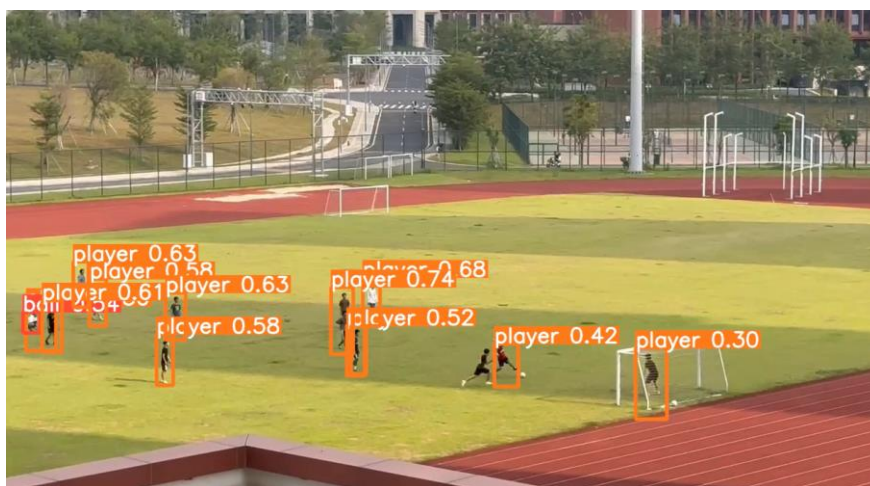
3.6 优化与挑战

说明：因为测试视频文件过大，所以只放最终效果的视频文件，而优化过程仅以截图形式展示。

- 考虑3.2中对测试视频的分析a)，寻找情况更相似的足球比赛数据集对模型进行微调。微调代码与报告最开始使用的微调模型代码相似

这里找到了来自roboflow的数据集SoccerBall Detector Computer Vision

Project[<https://universe.roboflow.com/quince-imaging/soccerballdetector>]，该数据集包含更广视角的带标签的图片。因为该数据集仅对球做了标签，所以本次训练出来的模型仅用于识别球，之后再用一般的模型去识别人，效果如下。



原测试效果



本优化模型效果（能较好地识别出足球）

- 考虑3.2中对测试视频的分析b)，使用锐化、对比度调整等方法对测试视频进行清晰度上的优化



处理前->处理后

通过锐化、提高对比度、降低亮度的方法，人和球的轮廓被凸显的更明显了，但是很可惜，同一模型在原视频和处理过的视频下表现**差异不明显**。经反思，我认为我对模型的训练集整体的特征掌握的不是很好，不知道模型本身的偏好，因此盲目的对测试集预处理带来的效果是无法预知的。



➤ 对测试视频重裁，观察是否有更好的效果

在识别的过程中，球场外的事物不时会被误识别，因此我识别之前会对视频**选取ROI区域**。实现方法：裁剪为球场内和球场外两部分，仅对球场内的事物进行识别，最后再将两部分重新拼接到一起，这样就能只对感兴趣的区域进行识别了。

裁剪与合并：





效果对比如下：



➤ 尝试使用不同版本的YOLO，比较分析：

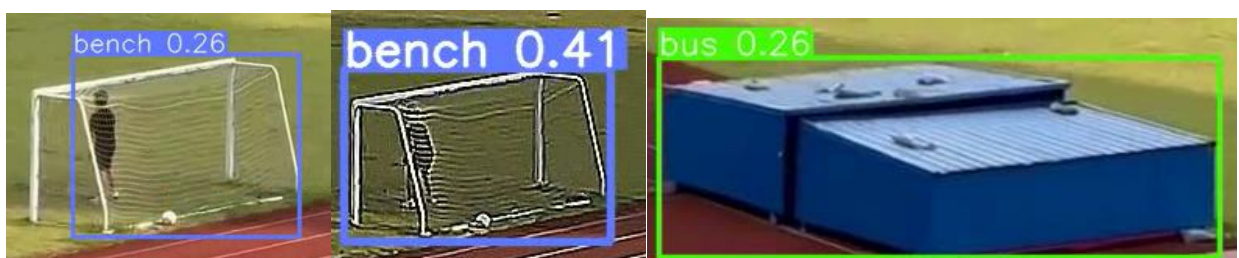
在测试视频中，并未发现不同的YOLO模型带来的效果有明显的差异，结合这点考虑，可以在实验中使用运行速度较快的YOLOv8n和YOLOv8s，以下是尝试过的YOLO模型：

- 📄 yolov8l.pt
- 📄 yolov8m.pt
- 📄 yolov8n.pt
- 📄 yolov8s.pt
- 📄 yolov8x.pt

3.7 扩展

➤ 多目标检测

除了球员和球，尝试检测其他目标，如门柱。但是很难找到比较合适的带标签的训练集，训练集标签太杂了的话**效果不好**，不管视频是否经过了预处理。具体效果可见下图（把门柱识别成了bench等等）。



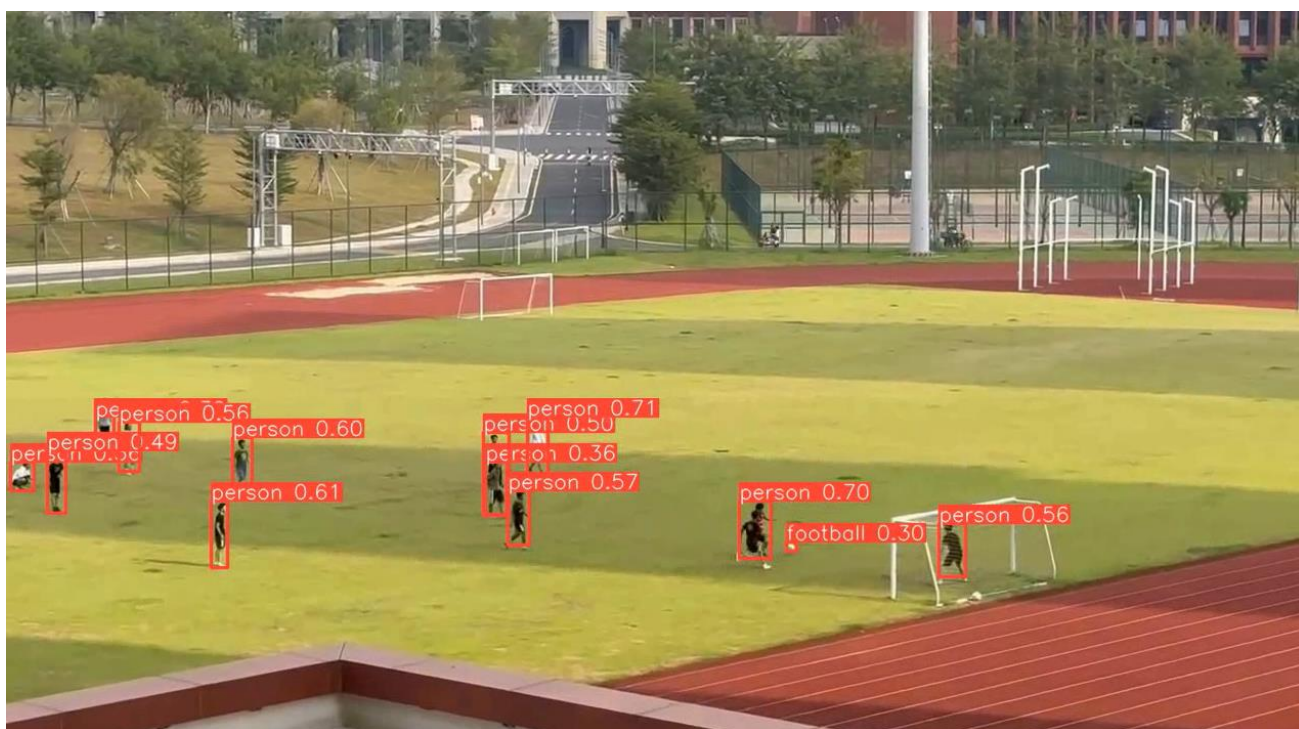
➤ 事件检测

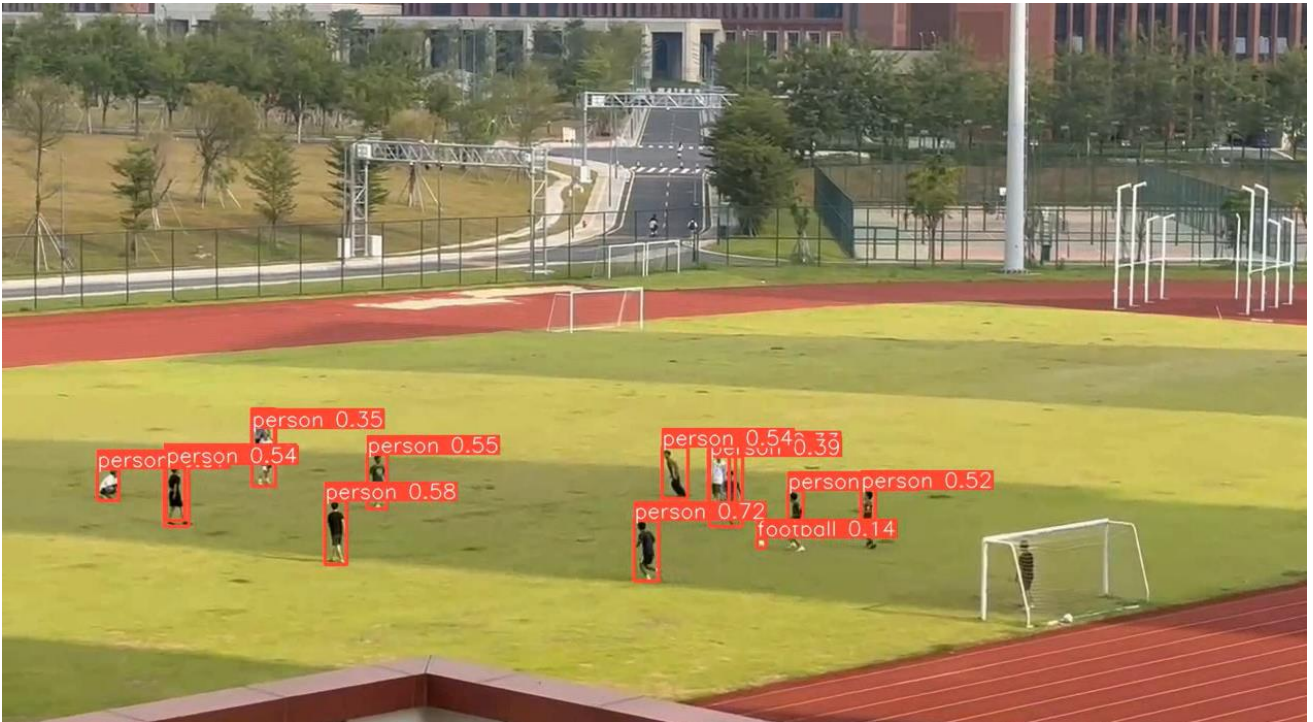
在原先的对球员和球检测的基础上，增加进球、碰撞等事件的检测。对事件的检测相比于对物体的检测存在一些困难，主要是因为事件通常具有更高的复杂性和动态性。为了做好事件检测，需要考虑多目标跟踪方法，以在不同帧之间维护目标的标识并处理目标之间的交互。以下是碰撞检测的识别效果。[<https://universe.roboflow.com/aibuilder/violence-detection-s9acq>]



➤ 采用表现更优异的模型提高性能

网络上有一些经过coco训练集预训练过的YOLO模型，对视频检测的效果有一些提升。综合以上所有优化策略，最终视频detect的效果如下，还是不错的：





标注后的视频:



football_pinjie2.
mp4

3.8 附录文件

yolov8n.pt: 已预训练的yolov8n模型, 可用于测人, 但是测球效果不好

model_final.pt: 单用来测球的模型

test_video.py: 测试模型所用代码

train.py: 训练模型所用代码

裁剪.py: 裁剪视频所用代码

还原.py: 还原、拼接视频所用代码

锐化.py: 对视频进行锐化、调整对比度和亮度等操作所用代码