4.x 验证实验

4.x.1 视频预处理

我们在台式机上通过Xvid进行了视频的原始处理，将MXF格式的视频原始文件编码为avi文件。Xvid是一款主流的MPEG-4视频编码器，可以将多种输入格式的文件编码为多种输出格式。主要编码选项包括目标帧率，目标比特率，目标文件大小，最大最小瞬时比特率，量化类型，运动估计等初高级选项。编码原理如下图所示

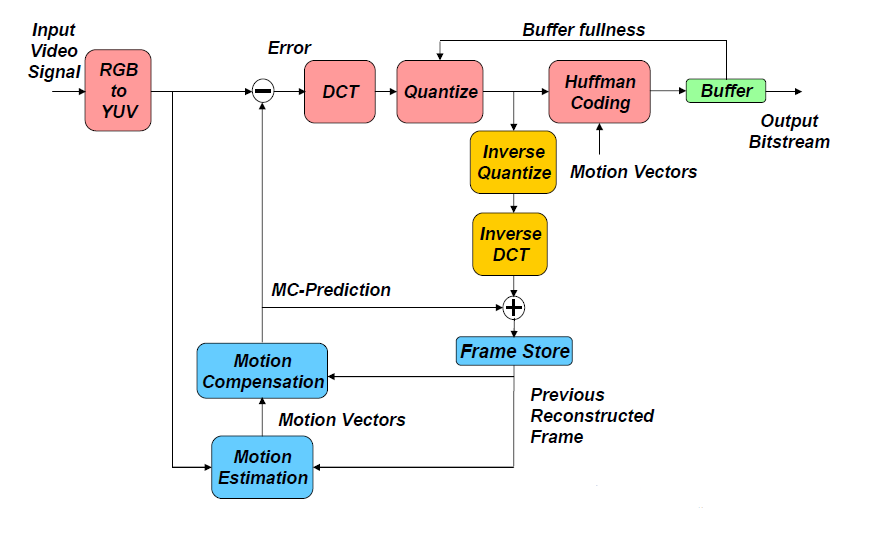
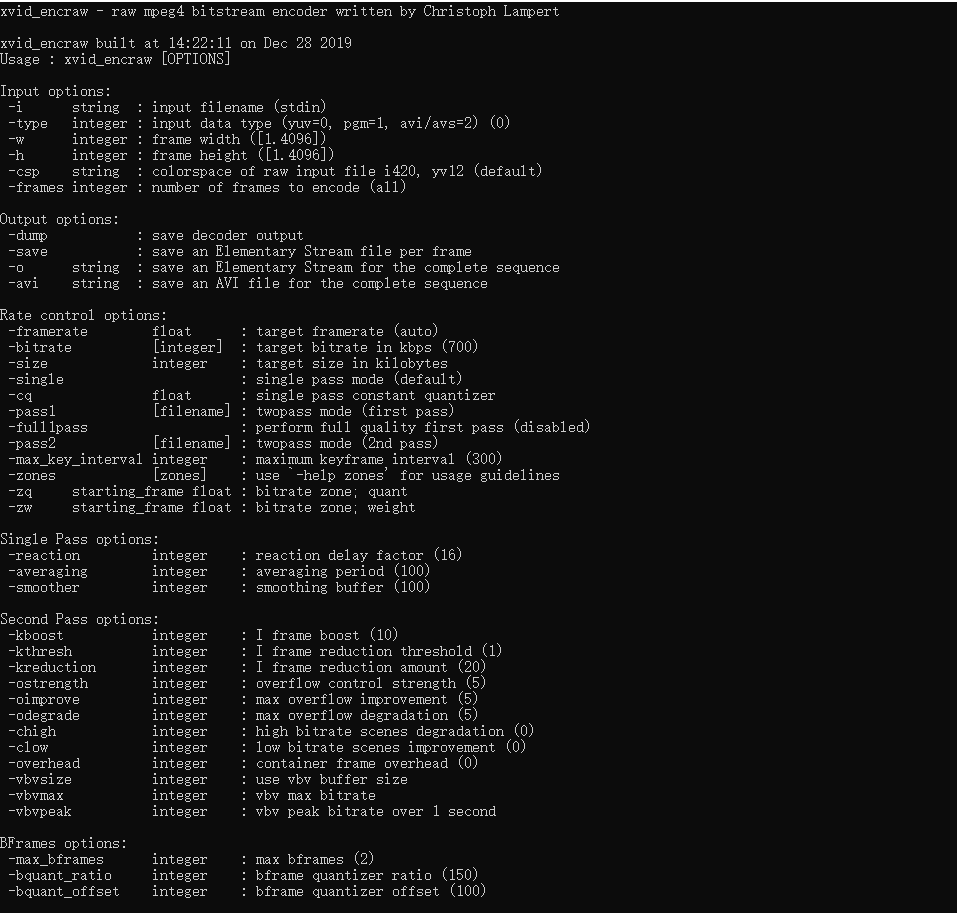


图 .视频编码原理

（<http://site.ieee.org/scv-ces/files/2015/06/video_coding_overview_IEEESantaClara_Sept05.pdf>）

在第一步编码过程中我们均使用默认参数。



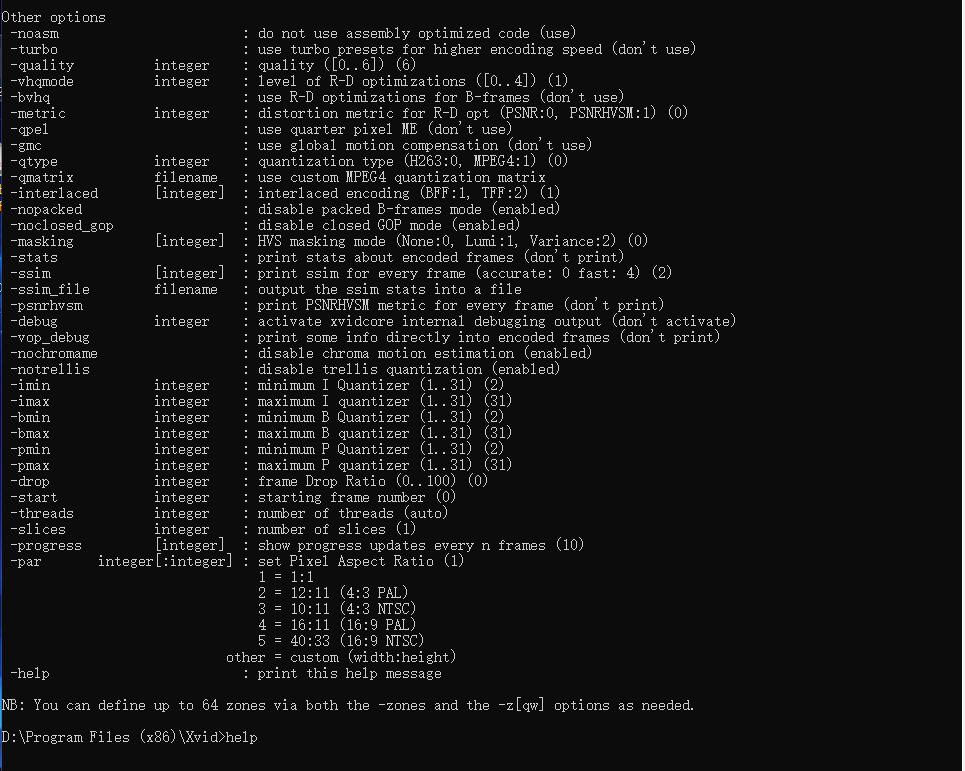


图 .Xvid编码可选及默认参数

之所以要选择原始视频进行手动编码，是因为编码本质上就是对视频的压缩，压缩就会造成信息的损失。自行编码可以控制信息的损失程度，以何种形式损失，确保后续实验结果不会因为使用了未知编码过程的文件而产生不必要的偏差。

文件所使用的视频数据集来自Browstow等人2008年于剑桥大学拍摄的行车数据集，分辨率为720\*960，由固定位置的闭路电视式摄像机从车辆主视角拍摄。该数据集比较符合我们预期的使用场景，并且具有很好的物体类型多样性。编码后视频效果如下图所示



图 .视频数据集示意

之后使用Xvid无损截取其中图像较为丰富的一个片段，再通过Xvid压缩为20%-90%不等，每10%为一个梯度的avi视频文件，用于测试YOLO5算法对于不同压缩率视频的处理精度。压缩过程调整了预定压缩文件大小的参数，其余均使用Xvid默认参数，压缩前后视频分辨率不变，均为720\*960。压缩过程如图所示

以压缩至20%为例，使用指令xvid\_encraw -type 2 -i Cam.avi -w 960 -h 720 -size 4784 -o Cam20.avi将Cam.avi压缩，得到新文件Cam20.avi，其预计大小为4784kB，实际大小可能略有出入。最终窗口显示如下汇总信息，包括总用时，总处理字节数，平均每帧用时，平均每秒处理帧数，平均每秒处理字节数等信息。

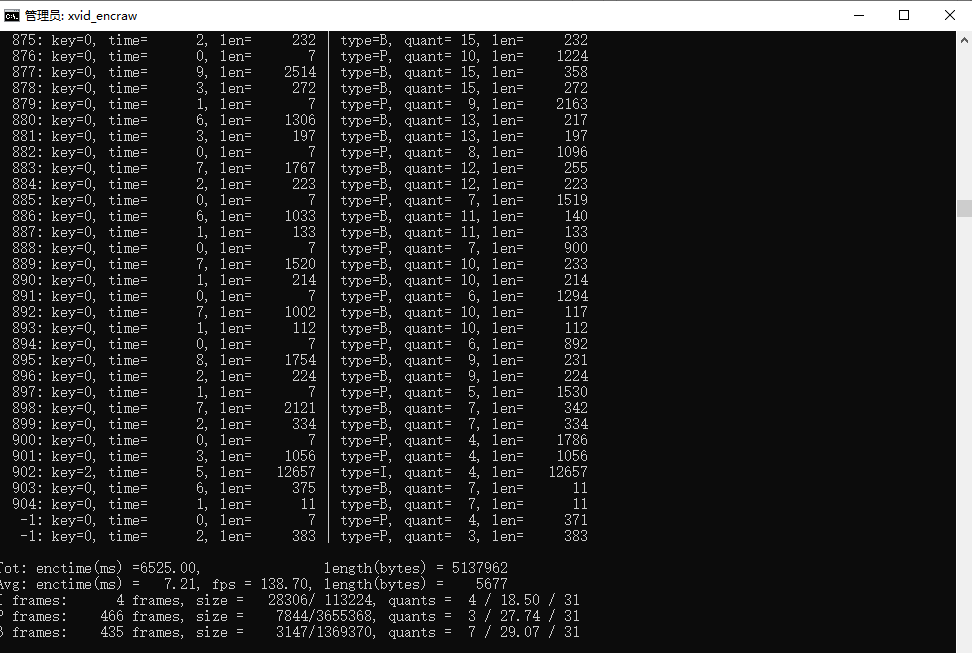


图 .Xvid压缩视频示例

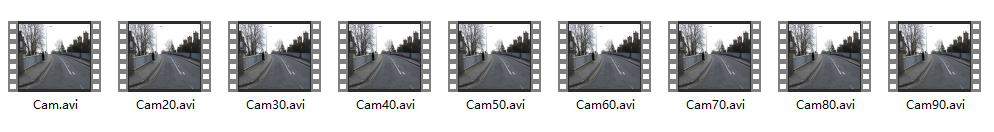


图 .以不同压缩率压缩后的视频

我们记录了同一设备上对于不同大小视频压缩率与压缩用时的关系，以作为考量压缩成本的标准之一。结果如下图所示，对于不同视频，压缩用时与压缩后大小基本都呈线性关系。压缩后大小越小，压缩用时越短。

图 .压缩后文件大小与压缩用时的关系

4.x.2 不同压缩率下的精度测试

在获得了不同压缩率的视频之后，我们基于YOLO5物体检测算法进行精度测试。测试平台为NVIDIA Xavier Jetson开发板，装载ubuntu21与OpenCV4。物体检测运行效果如图所示

图 .YOLO5运行示例

出于对预计使用场景的估计，我们挑选了所有识别的物体中的车辆和人体纳入精度计算范围之内。在约26秒的视频中，我们每隔1s截取一张图片进行识别并将结果纳入精度统计。精度指标采用IOU精度指标，GroundTruth为手动标定。



精度测试结果如下图所示

图 .检测精度与压缩后文件大小的关系

虚线为对数函数拟合结果。可以看到，压缩率与检测精度大致呈凸函数关系，但即使视频压缩至20%，检测精度也保持在了65%以上。**这证明了向边缘AI设备传输视频数据的过程中，压缩造成的检测精度损失是完全可以接受的**。压缩用时在分辨率相同的情况下并无明显区别，且CPU占用率也无明显区别。因此我们认为，检测相同分辨率视频时，无论文件大小，CPU的开销基本相同。