

基于深度学习的视频转码快速算法

徐婧瑶 王祖林 徐 迈

(北京航空航天大学电子信息工程学院 北京 100191)

摘要 由于良好的率失真表现,新一代视频压缩标准 HEVC(High Efficiency Video Coding)得到了越来越多终端设备的支持。然而目前仍有大量的 H.264 码流存在,因此 H.264 到 HEVC 的高效视频转码具有重要的实际意义。实现 H.264 到 HEVC 转码最简单的方法,是将 H.264 解码端和 HEVC 编码端直接级联起来。由于 HEVC 编码过程的复杂度较高,这种方法的转码时间较长。针对 H.264 到 HEVC 转码耗时的问题,文中提出一种基于深度学习的方法来预测 HEVC 的 CTU(Coding Tree Unit)块划分结果,从而避开 HEVC 对 CTU 所有块划分情况循环遍历以寻找率失真最优划分结构的过程,实现 H.264 到 HEVC 的快速转码。首先建立了一个 H.264 到 HEVC 转码的大型数据库,为训练深度学习模型提供数据保障;随后对 H.264 压缩域特征和 HEVC 的 CTU 块划分模式进行相关性分析,并发掘了 CTU 块划分模式在时序上的相似性,进而提出基于时间递归神经网络 LSTM(Long Short-Term Memory)的三级分类器来预测 HEVC 的 CTU 划分。实验结果表明,与直接级联转码器相比,文中提出的 H.264 到 HEVC 快速转码算法实现了 60% 的时间节省,同时峰值信噪比仅下降了 0.039 dB,其性能胜过近年来的转码算法的性能。

关键词 HEVC, H.264, 视频转码, 深度学习

中图分类号 TN919.81 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.03.016

Deep Learning Based Fast Video Transcoding Algorithm

XU Jing-yao WANG Zu-lin XU Mai

(College of Electronics and Information Engineering, Beihang University, Beijing 100191, China)

Abstract Due to the good rate-distortion performance, as the latest video compression standard, high efficiency video coding (HEVC) has been adopted by more and more terminals. However, there are still a large number of H.264 streams in the field of video compression. Therefore, H.264 to HEVC video transcoding is a meaningful research issue. The simplest way to achieve H.264 to HEVC transcoding is to directly cascade the H.264 decoder and the HEVC encoder. Due to high complexity of the HEVC coding process, this transcoding method is time-consuming. Therefore, this paper proposed a fast H.264 to HEVC transcoding method based on deep learning to predict the CTU(Coding Tree Unit) partition of HEVC, avoiding the brute-force search of CTU partition for rate-distortion optimization(RDO). First, a large-scale database of H.264 to HEVC transcoding is built for ensuring the training of deep learning model. Second, the correlation between HEVC CTU partition and H.264 domain features is analyzed, and the similarity of CTU partition across frames is found out. Then, a three-level classifier based on LSTM (Long Short-Term Memory) is designed to predict the CTU partition. The experimental results show that the H.264 to HEVC fast transcoding algorithm proposed in this paper achieves 60% reduction in complexity compared to the original transcoder, while the peak signal-to-noise ratio is only reduced by 0.039 dB, so the proposed method outperforms the state-of-the-art transcoding methods.

Keywords HEVC, H.264, Video transcoding, Deep learning

1 引言

随着多媒体技术的发展,新一代视频压缩标准 HEVC 应运而生。与上一代视频压缩标准 H.264 相比,HEVC 在实现相同的重建视频质量的前提下,将压缩率提高了一倍左右,这使得 HEVC 得到越来越多终端设备的支持。然而,在过去的

20 多年里, H.264 在视频压缩领域几乎占据着主导地位,目前仍有大量的 H.264 码流存在,因此, H.264 到 HEVC 的视频高效转码具有重要的实际意义。

实现 H.264 到 HEVC 转码最简单的方法是将 H.264 解码器和 HEVC 编码器级联起来, H.264 码流经过完全解码后再重新编码成 HEVC 码流。然而,这种方法非常耗时,这主

到稿日期:2018-07-11 返修日期:2018-09-15 本文受国家自然科学基金(61573037)资助。

徐婧瑶(1994-),女,硕士生,主要研究方向为视频转码和深度学习;王祖林(1965-),男,博士,教授,主要研究方向为图像处理、通信信号处理等;徐 迈(1981-),男,博士,教授,CCF 会员,主要研究方向为视频通信、图像处理、计算机视觉,E-mail:maixu@buaa.edu.cn(通信作者)。

要是因为 HEVC 编码过程中需要对所有 CTU 的块划分情况循环遍历以寻找率失真最优划分结构,时间复杂度较高。近年来,已经有学者提出了一些预测 CTU 块划分模式的方法^[1-2]来避免 HEVC 编码过程中的暴力式搜索,从而实现 HEVC 的快速编码。类似地,在设计 H.264 到 HEVC 快速转码算法的过程中,有必要充分利用 H.264 码流信息来预测 HEVC 的 CTU 块划分模式。

本文首先建立了一个 H.264 到 HEVC 转码的大型数据库,为深度学习模型的训练提供数据保证;随后提出一种基于深度学习的 H.264 到 HEVC 转码方法,利用 CTU 块划分的帧间相似性,以及 H.264 压缩域特征与 CTU 块划分的相关性,建立基于时间递归神经网络 LSTM 的三级分类器来预测 CTU 的最优划分结果,从而避免 HEVC 编码过程中对 CTU 块划分模式的暴力式搜索,实现转码速度的大幅提升。

2 相关工作

目前,实现 H.264 到 HEVC 的高效转码方法主要分为两类:启发式方法和数据驱动式方法。

启发式方法主要利用人们在 H.264 到 HEVC 转码过程中总结出的先验知识来简化 HEVC 的某些编码步骤。其中,Zhang 等^[3]采用一种基于功率谱的率失真优化模型(PS-RDO),并通过减少循环遍历的备选 CU(Coding Unit)、PU(Prediction Unit)划分模式,实现了 70%~80%的时间节省,但也导致了 30%的率失真性能损失。Peixoto 等^[4]提出了一种 H.264 运动矢量重用的方法,决定对哪些 CU 模式和 PU 模式进行遍历,转码速度最高提升 4.13 倍,同时 BD-RATE (Bjonteggard delta bit-rate)最高达到 10.92%。Nagaraghatta 等^[5]则提出了基于运动矢量和条件概率的模式映射方法。其中,H.264 宏块的运动矢量方差被用来预测 HEVC 中编码深度为 0 和 1 的可能的 CU 模式。他们还提出了一个基于条件概率的数学模型来预测 HEVC 中编码深度为 2 和 3 的最可能的两种 CU 块划分模式,实现了 50.33%的时间节省,同时平均 BD-RATE 为 4.72%。有别于 HEVC 编码过程中对所有 CU 划分情况进行自顶向下的二叉树遍历方式,Franche 等^[6]用自底向上的遍历方式以及 H.264 码流信息减少了待遍历的 CU 块划分模式,并通过提前终止遍历实现了平均 7.89 倍的速度提升,同时,平均 BD-RATE 为 3.28%。总的来说,由于依赖人类由先验知识总结出的人为规则,启发式的转码方法对于 CTU 块划分的预测准确度具有一定的局限性。

相比于启发式的方法,数据驱动式方法则更充分地利用训练样本完成 H.264 到 HEVC 转码过程中 CTU 块划分模式的预测。目前,数据驱动式转码主要用到的机器学习方法包括:线性判别分类器、支持向量机、决策树、朴素贝叶斯分类器和聚类等。其中,Peixoto 等首先提出用基于线性判别的方法来实现 H.264 宏块模式到 HEVC 的 CTU 块划分模式的映射^[7],并提出了两种模型训练模式:离线式的模型训练和动态式的模型训练;接着又提出了动态门限的方法,将视频序列的前 K 帧用于模型训练,以计算线性判别分类器的参数,同

时确定出判别 CTU 分割结果的门限^[8]。这种方法训练出来的模型更能适应当前序列的特点,从而达到更优的转码性能。随后,Peixoto 等^[9]在之前工作的基础上进一步改善了线性判别模型,提出了鲁棒性更强的特征选择算法和更优的 CU 分类策略,实现了 3.8 倍的速度提升,同时只增加了 4%的率失真。Jiang 等^[10]依据图像复杂度和编码比特数之间的相关性,将视频序列的每一帧分成 3 个等级的区域。根据 CTU 所属区域,决定 CTU 搜寻的深度,再利用运动矢量聚合分析 CU 的区域特征,进而选择出 CU 深度和 PU 模式,实现了 1.93 倍的转码加速,同时 BD-RATE 只增加了 1.73%。后来,Díaz-Honrubia 利用贝叶斯分类器进行 CTU 模式分类^[11],并采用一种自适应的门限更新算法^[12]减少了 56.7%的转码时间,同时 BD-RATE 惩罚约为 3.4%。Correa 等^[13]通过挖掘 H.264 解码过程中与 CTU 划分相关性较高的属性,并利用 C4.5 机器学习算法建立决策树模型,减少了 44%的转码时间,同时 BD-PSNR (Bjonteggard delta PSNR)只降低了 0.04dB。Zhu 等^[14]提出了一种特征选择算法,并使用在线门限更新方法,基于支持向量机构建了三级分类器模型,以此实现 CTU 块划分模式的预测,达到了约 50%的时间节省,同时 BD-PSNR 低于 0.1dB。

相比于启发式的方法,数据驱动式的方法在转码性能上有所提升,但是仍然有改进的空间。由于 CTU 块划分本身具有时序相关性,而以上所有数据驱动式的转码方法都没有考虑到这一点,因此,本文提出一种基于 LSTM 的 H.264 到 HEVC 转码方法,通过学习 HEVC 中 CTU 块划分的帧间相关性,以及 H.264 压缩域特征与 HEVC 中 CTU 块划分的相关性,建立三级分类器来预测 CTU 的最优划分结果。所提方法在保证视频质量的前提下,大幅节省了转码时间。

3 转码数据库建立及特征相关性分析

3.1 数据库建立

深度学习模型的训练往往需要大量的样本,因此,本文建立了一个大型的 H.264 到 HEVC 转码数据库,包括训练数据库和测试数据库。

训练数据库包括来自文献^[15]和文献^[16]的 93 个无损视频序列,涵盖了 352×240 , 352×288 , 720×480 , 704×576 , 1280×720 , 1920×1080 , 2480×1080 这 7 种分辨率的室内和室外视频。同时,室内视频和室外视频又都包含摄像机运动类视频和摄像机静止类视频。分别对这 93 个无损视频进行 H.264 编码、H.264 解码和 HEVC 编码。本文利用 JM19.0 进行 H.264 编码,编码模式设置为 IPPP,量化步长设置为 27。随后,在修改 JM19.0 解码端程序的基础上,提取出 H.264 码流的压缩域特征,包括残差、运动矢量、分配比特数和宏块划分模式。同时,本文使用 HM16.0 进行 HEVC 编码,编码模式为 LDP,使用码率控制,量化步长设置为 27,在编码过程中同时提取出 HEVC 的 CU 块划分结果。如表 1 所列,训练数据库一共有 76 309 527 个 CU 划分样本,对应 64×64 , 32×32 , 16×16 这 3 种 CU 尺寸,其中正样本表示对当前尺寸的 CU 块进行分割,负样本则表示不进行分割。训练数据库的建立为深度学习模型的训练提供了强有力的数据支撑。

表 1 训练数据库的样本数统计

Table 1 Statistics of samples in training database

CU 尺寸	正样本数	负样本数	总样本数
64×64	6011500	4083411	10094911
32×32	10542154	13503846	24046000
16×16	11700893	30467723	42168616
总计	28254547	48054980	76309527
占总样本比/%	37.03	62.97	—

测试数据库包括来自文献[17]的 18 个标准测试序列,涵盖了 2560×1080(A 类)、1920×1080(B 类)、832×480(C 类)、416×240(D 类)、1280×720(E 类)5 种分辨率的视频。测试数据库也包括室内场景和室外场景,同时也存在摄像机运动和摄像机静止这两类视频。与训练数据库操作类似,分别对这 18 个视频进行 H.264 编码、H.264 解码和 HEVC 编码。如表 2 所列,测试数据库中一共有 16332436 个 CU 划分样本,对应 64×64,32×32,16×16 这 3 种 CU 尺寸。对比表 1 和表 2 的结果可知,训练集和测试集的正负样本比例大体接近。

表 2 测试数据库的样本数统计

Table 2 Statistics of samples in test database

CU 尺寸	正样本数	负样本数	总样本数
64×64	1173646	727854	1901500
32×32	2434088	2260496	4694584
16×16	3146531	6589821	9736352
总计	6754265	9578171	16332436
占总样本比/%	41.35	58.65	—

3.2 特征相关性分析

基于 3.1 节对 H.264 码流特征和 HEVC 的 CU 块划分结果的提取,本节进行特征相关性分析。

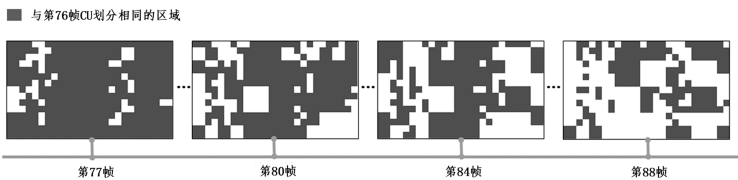


图 2 CU 块划分的帧间时序相关性的举例

Fig. 2 Examples of the temporal similarity of CU partition across video frames

进一步地,分别统计在 64×64,32×32 和 16×16 这 3 种 CU 尺寸上不同 GOP(Group of Pictures)间隔之间 CU 块划分的线性相关系数,得到图 3 所示的结果。

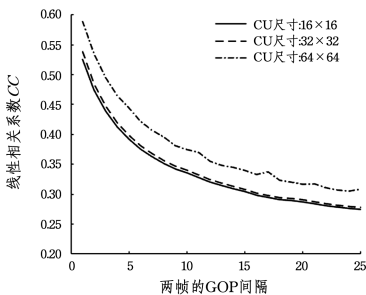


图 3 CU 块划分帧间相关性统计分析图

Fig. 3 CC values of CU partition between two frames

考虑到 H.264 码流和 HEVC 的 CU 块划分之间存在一定的相关性,本节首先对 H.264 的特征(残差、运动矢量、分配比特数和宏块划分模式)和 HEVC 的 CU 块划分进行定性分析。分别计算出 H.264 的 4 个特征与随机产生的 CU 块划分结果的线性相关系数(CC),并取平均值作为基准。如图 1 所示,64×64,32×32,16×16 尺寸的 CU 块划分与对应的分配比特数、宏块模式、残差和运动矢量之间的相关系数均远高于基准,说明了这 4 个 H.264 特征与 HEVC 中 CU 块划分之间具有较高的相关性。同时,4 个特征在 64×64 和 32×32 尺寸上与 CU 块划分的相关性均高于 16×16 尺寸上的情况,因此 16×16 尺寸的 CU 块划分的预测难度明显更大。

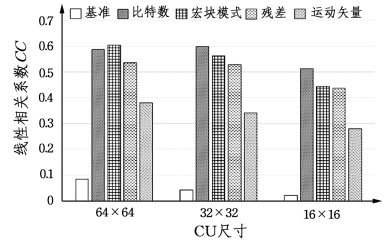


图 1 H.264 特征与 CU 块划分的相关性统计图

Fig. 1 CC values between H.264 features and CU partition

前人的研究工作中通常只考虑到 H.264 码流中某些特征与 HEVC 中 CU 块划分之间的相关性,忽略了 CU 块划分在帧间其实也存在着一定的联系。图 2 以视频序列 BasketballPass 为例,将第 76 帧作为参考帧,比较了第 77 帧、第 80 帧、第 84 帧、第 88 帧与参考帧 CU 块划分的差异。其中,黑色块代表当前帧与第 76 帧 CU 块划分相同的区域。从图 2 可以看出,HEVC 的 CU 块划分在时序上存在着相似性,并且相似性随着帧数间隔的增加而逐渐减小。

从图 3 可以看出,对于 3 种 CU 尺寸,相邻帧之间的 CU 块划分存在着很强的相关性,并且随着帧间间隔的增加,相关性逐渐降低。同时,在同等帧间间隔下,16×16 和 32×32 尺寸下的 CU 块划分的帧间相关性明显低于 64×64 尺寸下的帧间相关性,这意味着利用帧间相关性来预测 16×16 和 32×32 尺寸下的 CU 块划分的难度可能会高于 64×64 尺寸的。

4 基于 LSTM 的快速转码算法

基于 H.264 码流特征与 CU 块划分之间的相关性,以及 CU 块划分的时序相关性,本文构建了一种基于 LSTM 的 CU 块划分预测网络,结构如图 4 所示。该网络由三级 LSTM 分类器构成,分别对尺寸为 64×64,32×32 和 16×16 的 CU 块划分进行预测。当某一级分类器判定当

前尺寸的 CU 块进行分割,则下一级分类器被激活,否则提前终止。例如,第一级分类器判定将 64×64 尺寸的 CU 分割成 4 个 32×32 尺寸的 CU,则二级分类器被激活,进而进行 32×32 尺寸的 CU 块划分预测,否则提前终止。激活/终止机制的使用能避免对不存在的 CU 块进行预测,进一步节省网络的预测时间。三级分类器的输入分别是对应尺寸的 H.264 特征,包括残差、运动矢量、分配比特数和宏块划分模式。由于残差和运动矢量的原始维度较高,本文将每 8×8 尺寸的 CU 块内的残差进行绝对值求和,并对每 16×16 尺寸的 CU 块内的运动矢量取最大值,替代原始特征以实现

降维处理。经过归一化后,特征被分别送入三级分类器,经过 LSTM 单元和全连接层,输出以 CTU 为基本单元的划分结果。对于第 t 帧的某个 CTU 块, $\hat{y}_1(U, t)$ 表示第 t 帧 CTU 块的第一级分类结果,若值为 1,表示 64×64 尺寸的 CU 块被分成 4 个 32×32 的 CU 块,若值为 0,则表示不分割。同理, $\{\hat{y}_2(U_i, t)\}_{i=1}^4$ 和 $\{\hat{y}_3(U_i, t)\}_{i=1}^{16}$ 分别表示第二级和第三级的分类结果。由 $\hat{y}_1(U, t), \{\hat{y}_2(U_i, t)\}_{i=1}^4$ 和 $\{\hat{y}_3(U_i, t)\}_{i=1}^{16}$ 组合而成的 21 维向量构成网络输出的最小单元,表示一个完整 CTU 的块划分结果。

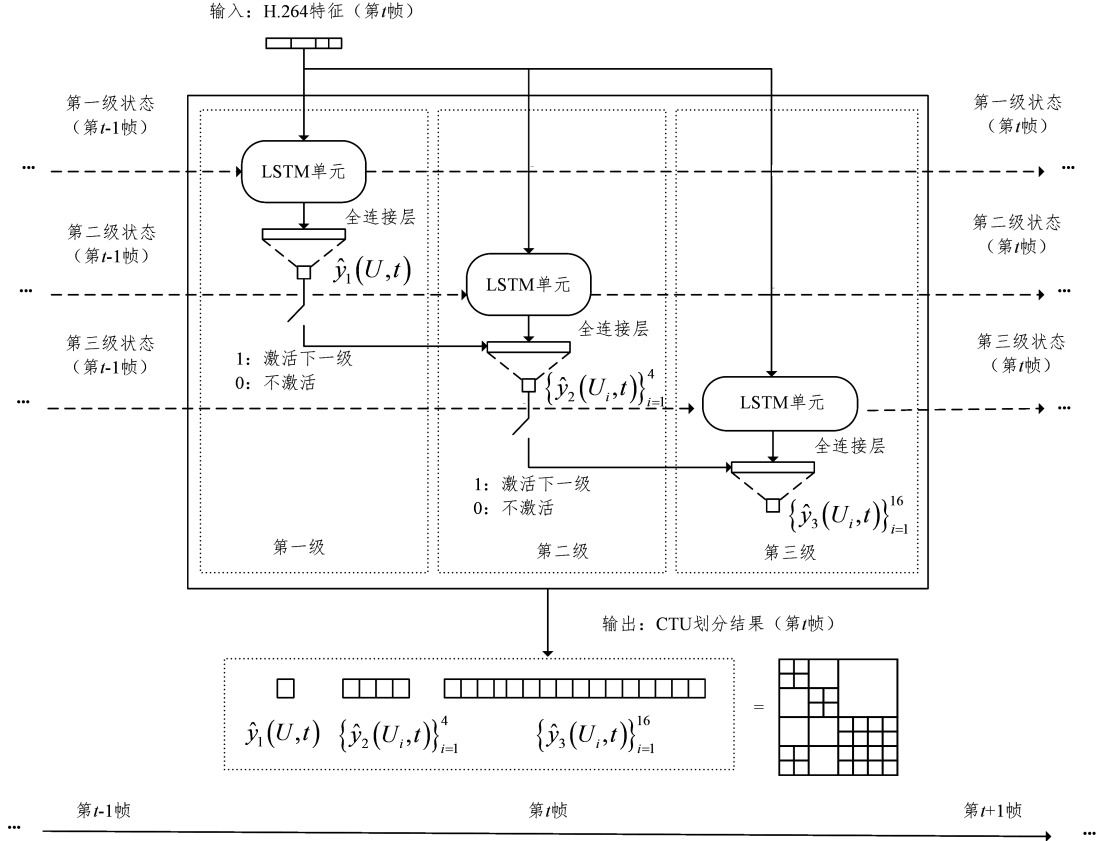


图 4 CTU 预测网络的结构

Fig. 4 Structure of predicting CTU partition

通过 LSTM 单元,本网络能充分利用 CTU 块划分的帧间相关性对 CTU 块划分进行预测。从图 4 中可以看到, LSTM 单元的状态在时序上随着帧数的向后推移而不断更新,同时,当前帧的输出不仅取决于当前帧的特征输入,还与前一帧的状态有关。下面以第 l 级分类器为例来介绍本网络的内部机制。LSTM 单元具有 3 个门,即输入门 $i_l(t)$ 、遗忘门 $f_l(t)$ 和输出门 $o_l(t)$,定义如下:

$$i_l(t) = \sigma(W_i \cdot [h_l(t-1), g_l(t)] + b_i) \quad (1)$$

$$f_l(t) = \sigma(W_f \cdot [h_l(t-1), g_l(t)] + b_f) \quad (2)$$

$$o_l(t) = \sigma(W_o \cdot [h_l(t-1), g_l(t)] + b_o) \quad (3)$$

其中, W_i, W_f 和 W_o 分别表示 3 个门的权重参数, b_i, b_f 和 b_o 分别表示 3 个门的偏置量, $h_l(t-1)$ 表示第 l 级 LSTM 单元在 $t-1$ 帧的输出结果, $g_l(t)$ 表示第 l 级 LSTM 单元在 t 帧的输入, σ 表示 sigmoid 函数。第 l 级 LSTM 单元在 t 帧时的状态 $C_l(t)$ 按式(4)进行更新:

$$C_l(t) = f_l(t) \cdot C_l(t-1) + i_l(t) \cdot \tanh(W_c \cdot [h_l(t-1), g_l(t)] + b_c) \quad (4)$$

其中,“ \cdot ”表示点乘操作, W_c 和 b_c 是 $C_l(t)$ 的可学习参数。进而,第 l 级 LSTM 单元在 t 帧时的输出结果可由下式计算:

$$h_l(t) = o_l(t) \cdot \tanh(C_l(t)) \quad (5)$$

通过 LSTM 单元的门机制,该网络能充分利用 HEVC 中 CTU 块划分的长短期依赖对 CTU 划分进行有效预测。

需要注意的是,三级 LSTM 分类器是分开进行训练的。本文采用基于交叉熵函数的损失函数 L 进行模型训练,其表达式如下:

$$L = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T L_n(t) \quad (6)$$

其中, $L_n(t)$ 表示第 t 帧内第 n 个 CU 块的交叉熵损失, T 表示 LSTM 的长度。在本文中,采用 Adam 优化器^[18]来训练给定损失函数的分类器模型。最后,利用训练好的 LSTM 分类器

模型,即可对 HEVC 的 CTU 块划分模式进行预测。

5 实验结果

本文提出的转码方法基于 H.264 官方参考软件 JM19.0 和 HEVC 官方参考软件 HM16.0 实现。H.264 采用 IPPP 模式进行编码, JM19.0 端对应的配置文件为 encoder_base-line.cfg。HEVC 采用 LDP 模式进行编码, HM16.0 端对应的配置文件为 encoder_low_delay_P_main.cfg。视频序列量化步长设置为 27。分别用 ΔT 和 $\Delta PSNR$ 衡量转码算法的时间复杂度节省情况和质量损失情况,表达式如下:

$$\Delta T = \frac{T_o - T_p}{T_o} \quad (7)$$

$$\Delta PSNR = PSNR_p - PSNR_o \quad (8)$$

其中, T_o 和 $PSNR_o$ 分别表示直接级联转码器的转码时间和峰值信噪比, T_p 和 $PSNR_p$ 分别表示改进的转码器的转码时间和峰值信噪比。需要强调的是, 本文转码实验使用了码率控制, 以保证在不同转码方法下同一视频的比特率固定, 因此 $\Delta PSNR$ 反映了固定码率下的质量损失。

另外, 本文所有的实验均在具有 Windows 10 64 位操作系统、16GB 内存和英特尔 i7-6700 @3.40GHz 处理器的计算机上进行。在模型训练阶段, 使用 GeForce GTX 1080GPU 进行加速, 但在测试转码时间时并未使用该 GPU。

在训练阶段, 训练数据源于 3.1 节建立的大型转码数据库。对于数据库中的每一个视频序列, 每连续 T 帧(等于 LSTM 长度)取相同位置上的 HEVC CTU 块划分及对应的

H.264 特征作为一个训练样本。本文中 LSTM 的长度 T 设为 30。同时, 为了得到充足的训练样本, 相邻样本间允许有 10 帧重叠。因此, 本文从数据库的 93 个视频序列中得到了 451 424 个样本。随机抽取 400 000 个样本作为训练集, 剩下的 51 424 个样本作为验证集。利用训练集进行模型训练, 并在验证集上调整模型的超参数。所有的训练参数都采用零均值、标准差为 1.0 的截断正态分布随机初始化方式。样本批处理大小设为 200, 学习率设为 0.001, 优化器选用 Adam。

在测试阶段, 本文的转码模型用数据库中来自 JCT-VC 的 18 个标准测试序列进行测试。与训练阶段每 30 帧预测一次不同, 模型在测试阶段是单步进行预测的, 即每次只产生一帧的 CTU 块划分结果。当上一帧 CTU 块划分预测完成, H.264 解码端对当前帧完成解码并将相应的压缩域特征送入 CTU 预测网络后, 该网络才能产生该帧的 CTU 块划分预测结果, 并送入 HEVC 编码端辅助编码。紧接着, 进行下一帧的 H.264 解码、CTU 预测、HEVC 编码, 以此类推后面的所有帧。

将本文提出的算法与近年来性能较优的两个算法^[14,19]在 18 个标准测试序列下进行比较, 得到如表 3 所列的结果。从表中可知, 本文算法的平均 PSNR 降低了 0.039 dB, 低于对比算法 1 的 0.048 dB, 远低于对比算法 2 的 0.138 dB。同时, 本文算法节省了 60% 的转码时间, 多于对比算法 1 的 48% 和对比算法 2 的 51%。因此, 本文算法在质量损失和复杂度降低两方面均优于对比算法。

表 3 本文算法和对比算法的性能比较

Table 3 Performance comparison between proposed method and state-of-the-art methods

类别	视频序列	对比算法 1 ^[14]		对比算法 2 ^[19]		本文算法	
		$\Delta T/\%$	$\Delta PSNR/-dB$	$\Delta T/\%$	$\Delta PSNR/-dB$	$\Delta T/\%$	$\Delta PSNR/-dB$
A	Traffic	0.53	0.056	0.48	0.116	0.66	0.040
	PeopleOnstreet	0.41	0.096	0.40	0.351	0.53	0.027
	BasketballDrive	0.56	0.045	0.65	0.137	0.66	0.042
B	BQTerrance	0.49	0.027	0.52	0.041	0.67	0.026
	Cactus	0.56	0.045	0.63	0.094	0.65	0.029
	Kimono	0.63	0.044	0.66	0.111	0.73	0.048
C	Parkscene	0.60	0.046	0.46	0.093	0.63	0.042
	BasketballDrill	0.48	0.045	0.57	0.170	0.60	0.038
	BQMall	0.41	0.042	0.47	0.252	0.63	0.059
D	PartyScene	0.36	0.032	0.34	0.107	0.57	0.047
	RaceHorsesC	0.38	0.030	0.43	0.177	0.55	0.021
	BasketballPass	0.34	0.055	0.47	0.196	0.38	0.037
E	BlowingBubbles	0.39	0.082	0.33	0.063	0.28	0.029
	RaceHorsesD	0.31	0.056	0.32	0.223	0.47	0.024
	BQSquare	0.26	0.023	0.18	0.067	0.48	0.029
平均结果	FourPeople	0.64	0.069	0.74	0.112	0.72	0.061
	Johnny	0.68	0.038	0.82	0.062	0.77	0.050
	KristenAndSara	0.58	0.039	0.78	0.112	0.74	0.059
平均结果		0.48	0.048	0.51	0.138	0.60	0.039

另一方面, 经过统计, 本文提出的 CTU 预测网络的计算时间少于转码总时间的 2%, 说明了网络的高效性。此外, 将网络中的 LSTM 单元去掉, 仅利用简单的三级全连接网络进行模型训练并测试其性能, 平均 PSNR 降低了 0.1052 dB, 远差于本文算法的 0.039 dB, 从而证明了本文提出的 CTU 预测网络结构的有效性。

HEVC 快速转码算法, 其利用 H.264 压缩域特征来预测 HEVC 的 CTU 块划分, 取代 HEVC 编码中对 CTU 块划分所有可能情况循环遍历以寻找率失真最优的 CTU 块划分的过程, 从而大幅提升了转码速度。首先, 本文建立了一个 H.264 到 HEVC 的大型转码数据库, 为深度学习模型的训练提供了强有力的数据支撑。随后, 本文挖掘了 H.264 压缩域特征和 HEVC 的 CTU 块划分之间的相关性, 同时分析了

结束语 本文提出了一种基于深度学习的 H.264 到

CTU块划分的帧间相关性,进而提出了基于LSTM的三级分类器来预测HEVC的CTU块划分。实验结果表明,相比于对比算法,本文提出的转码方法在更大程度地降低时间复杂度的前提下,能保证更小的质量损失。

参 考 文 献

- [1] LIU Z, YU X, GAO Y, et al. CU Partition Mode Decision for HEVC Hardwired Intra Encoder Using Convolution Neural Network[J]. *IEEE Transactions on Image Processing*, 2016, 25(11):5088-5103.
- [2] SHEN L, LIU Z, ZHANG X, et al. An Effective CU Size Decision Method for HEVC Encoders[J]. *IEEE Transactions on Multimedia*, 2013, 15(2):465-470.
- [3] ZHANG D, TONG J, ZAND D. Fast CU partition for H. 264/AVC to HEVC transcoding based on fisher discriminant analysis [C] // *Visual Communications and Image Processing*. IEEE, 2017:1-4.
- [4] PEIXOTO E, IZQUIERDO E. A complexity-scalable transcoder from H. 264/AVC to the new HEVC codec[C] // *IEEE International Conference on Image Processing*. IEEE, 2012:737-740.
- [5] NAGARAGHATTA A, ZHAO Y, MAXWELL G, et al. Fast H. 264/AVC to HEVC transcoding using mode merging and mode mapping[C] // *IEEE International Conference on Consumer Electronics*. Berlin: IEEE, 2016:165-169.
- [6] FRANCHE J F, COULOMBE S. Fast H. 264 to HEVC transcoder based on post-order traversal of quadtree structure [C] // *IEEE International Conference on Image Processing*. IEEE, 2015:477-481.
- [7] PEIXOTO E, MACCHIAVELLO B, HUNG E M, et al. An H. 264/AVC to HEVC video transcoder based on mode mapping [C] // *IEEE International Conference on Image Processing*. IEEE, 2014:1972-1976.
- [8] PEIXOTO E, SHANABLEH T, IZQUIERDO E. H. 264/AVC to HEVC Video Transcoder Based on Dynamic Thresholding and Content Modeling[J]. *IEEE Transactions on Circuits & Systems for Video Technology*, 2014, 24(1):99-112.
- [9] PEIXOTO E, MACCHIAVELLO B, QUEIROZ R L D, et al. Fast H. 264/AVC to HEVC transcoding based on machine learning[C] // *Telecommunications Symposium*. IEEE, 2014:1-4.
- [10] JIANG W, CHEN Y, TIAN X. Fast transcoding from H. 264 to HEVC based on region feature analysis[J]. *Multimedia Tools & Applications*, 2014, 73(3):2179-2200.
- [11] DÍAZ-HONRUBIA A J, MARTÍNEZ J L, PUERTA J M, et al. Fast quadtree level decision algorithm for H. 264/HEVC transcoder[C] // *IEEE International Conference on Image Processing*. IEEE, 2015:2497-2501.
- [12] DÍAZ-HONRUBIA A J, MARTÍNEZ J L, CUENCA P, et al. Adaptive Fast Quadtree Level Decision Algorithm for H. 264 to HEVC Video Transcoding[J]. *IEEE Transactions on Circuits & Systems for Video Technology*, 2016, 26(1):154-168.
- [13] CORREA G, AGOSTINI L, CRUZ L A D S. Fast H. 264/AVC to HEVC transcoder based on data mining and decision trees[C] // *IEEE International Symposium on Circuits and Systems*. IEEE, 2016:2539-2542.
- [14] ZHU L, ZHANG Y, LI N, et al. Machine learning based fast H. 264/AVC to HEVC transcoding exploiting block partition similarity[J]. *Journal of Visual Communication & Image Representation*, 2016, 38(C):824-837.
- [15] Xiph.org. Xiph.org video test media[OL]. <https://media.xiph.org/video/derf/>.
- [16] XU M, DENG X, LI S, et al. Region-of-Interest Based Conversational HEVC Coding with Hierarchical Perception Model of Face[J]. *IEEE Journal of Selected Topics in Signal Processing*, 2014, 8(3):475-489.
- [17] OHM J R, SULLIVAN G J, TAN T K, et al. Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC) [J]. *IEEE Transactions on Circuits & Systems for Video Technology*, 2012, 22(12):1669-1684.
- [18] INGMA D P, BA J. Adam: A Method for Stochastic Optimization[J]. *arXiv preprint arXiv:1412.06980*, 2014.
- [19] CORREA G, ASSUNCAO P A, AGOSTINI L V, et al. Fast HEVC Encoding Decisions Using Data Mining[J]. *IEEE Transactions on Circuits & Systems for Video Technology*, 2015, 25(4):660-673.