ICS 35.040
CCS L 71

团　体　标　准

T/AI XXX.XX—XXXX

信息技术 视觉特征编码
第 4 部分：深度特征图

Information Technology – Visual Feature Coding

Part 4: Deep Feature Map

点击此处添加与国际标准一致性程度的标识

(征求意见稿)

(在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上)

XXXX - XX - XX 实施

XXXX - XX - XX 实施

中关村视听产业技术创新联盟　发布

目　　次

前　　言 II
引　　言 III
信息技术 视觉特征编码 第 4 部分：深度特征图编码 2
1 范围 2
2 规范性引用文件 2
3 术语和定义 2
4 缩略语 2
5 约定 3
5.1 概述 3
5.2 算术运算符 3
5.3 逻辑运算符 3
5.4 关系运算符 4
5.5 位运算符 4
5.6 赋值 4
5.7 位流语法、解析过程和解码过程的描述方法 5
5.7.1 位流语法的描述方法 5
5.7.2 函数 6
5.7.3 描述符 6
5.7.4 保留、禁止和标记位 7
6 语法和语义 7
6.1 特征图序列语法 7
6.2 特征图序列语义 10
7 特征图压缩标准 11
7.1 特征图特指卷积神经网特征图数据 11
7.2 解码框架 11
7.2.1 预量化/反预量化 12
7.2.2 重打包/反重打包 14
7.2.3 视频编码/解码器 17
8 与传统视频编码标准的接口 17
附录 A 19
附录 B 20

前　言

本文件按照 GB/T 1.1-2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件是T/AI XX.XX-XXXX《信息技术 视觉特征编码》的第4部分。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由新一代人工智能产业技术创新战略联盟AI标准工作组提出。

本文件由中关村视听产业技术创新联盟归口。

本部分起草单位：鹏城实验室，北京大学，中新国际联合研究院，青岛海信网络科技股份有限公司，青岛新一代人工智能技术研究院，青岛图灵科技有限公司，浙江邦盛科技股份有限公司

本部分起草人：陈卓，段凌宇，Alex C. Kot，Weisi Lin，杨文瀚，汪维，高峰，冯栋，王雯雯，王新宇，陈伟，赵海英，崔晓冉。

引　言

《信息技术 视觉特征编码》拟由6个部分构成。

——第1部分：系统；

——第2部分：手工设计特征；

——第3部分：深度学习特征；

——第4部分：深度特征图；

——第5部分：语义分割图；

——第6部分：结构点序列。

本文件的发布机构提请注意，声明符合本文件时，可能涉及到XX、XX中如下X项相关专利的使用。专利名称如下：

CNxxxxx.1，xxxxxx；

本文件的发布机构对于该专利的真实性、有效性和范围无任何立场。

该专利持有人已向本文件的发布机构保证，他愿意同任何申请人在合理且无歧视的条款和条件下，就专利授权许可进行谈判。该专利持有人的声明已在本文件的发布机构备案，相关信息可以通过以下联系方式获得：

联 系 人：陈卓

通讯地址：广东省深圳市南山区 万科云城1期 鹏城实验室

邮政编码：518000

电子邮件：chenzh08@pcl.ac.cn

电话：13003034992

传真：

网址：

请注意除上述专利外，本文件的某些内容仍可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

信息技术 视觉特征编码
第 4 部分：深度特征图

1 范围

本标准规范了图像分析任务中，深度网络提取的特征图数据的编码格式和解码工具。
本标准适用于图像数据中目标或场景的分类、检索、识别等应用。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Encoder Description Update 13, ISO/IEC JTC 1/SC 29/WG 11 N19122.

3 术语和定义

下列术语和定义适用于本文件。

3.1

编码器 encoder
完成编码过程的实体。

3.2

像素 pixel
原始图像或转换图像的最小单元，每个像素包含空间坐标信息及其亮度值。

3.3

深度网络 deep neural network
多层神经网络。

3.4

深度特征图 deep feature maps
经过深度卷积网络提取后的具有语义信息的张量。

4 缩略语

下列缩略语适用于本文件。
ResNet：深度残差网络（Deep Residual Network）
VggNet：Karen Simonyan 等 2014 年提出的深度卷积网络结构
ReLU：线性整流函数（Rectified Linear Unit）
HEVC：ITU-T 视频编码专家组 High Efficiency Video Coding 视频压缩标准
YUV：使用 YUV 颜色编码方法的视频数据

AVS：Audio Video Coding Standard 是我国自主制定，拥有自主知识产权的音视频编码标准

5 约定

5.1 概述

本部分中使用的数学运算符和优先级参照C语言和python语言。但对整型除法和算术移位操作进行了特定定义。除特别说明外，约定编号和计数从0开始。

5.2 算术运算符

算术运算符定义见表1。

表 1 算术运算符定义

算术运算符	定义
+	加法运算
-	减法运算（二元运算符）或取反（一元前缀运算符）
×	乘法运算
a^b	幂运算，表示a的b次幂。也可表示上标
/	整除运算，沿向0的取值方向截断。例如，7/4和-7/-4截断至1，-7/4和7/-4截断至-1
÷	除法运算，不做截断或四舍五入
$\frac{a}{b}$	除法运算，不做截断或四舍五入
$\sum_{i=a}^{b} f(i)$	自变量i取由a到b（含b）的所有整数值时，函数$f(i)$的累加和
$a \% b$	模运算，a除以b的余数，其中a与b都是正整数
Sqrt	开平方运算
Arctg	反正切运算
Arcsin	反正弦运算
Arccos	反余弦运算
Abs	取绝对值
Max	取较大值
Min	取较小值
tan	正切运算

5.3 逻辑运算符

逻辑运算符定义见表2。

表 2 逻辑运算符定义

逻辑运算符	定义
a&&*b*	*a*和*b*之间的与逻辑运算
a || *b*	*a*和*b*之间的或逻辑运算
!	逻辑非运算

5.4 关系运算符

关系运算符定义见表3。

表 3 关系运算符定义

关系运算符	定义
>	大于
>=	大于或等于
<	小于
<=	小于或等于
==	等于
!=	不等于

5.5 位运算符

位运算符定义见表4。

表 4 位运算符定义

位运算符	定义
&	与运算
|	或运算
~	取反运算
a>>*b*	将*a*以2的补码整数表示的形式向右移*b*位。仅当*b*取正数时定义此运算
a<<*b*	将*a*以2的补码整数表示的形式向左移*b*位。仅当*b*取正数时定义此运算

5.6 赋值

赋值运算定义见表5。

表 5 赋值运算定义

赋值运算	定义
=	赋值运算符
++	递增，*x*++相当于*x*= *x*+ 1。当用于数组下标时，在自加运算前先求变量值

--	递减，*x*--相当于*x*=*x* – 1。当用于数组下标时，在自减运算前先求变量值
+=	自加指定值，例如*x* += 3相当于*x* = x + 3，*x* += (-3)相当于*x* = *x* + (-3)
-=	自减指定值，例如*x*-= 3相当于*x* = *x*- 3,，*x*-= (-3)相当于*x* = *x*- (-3)

5.7 位流语法、解析过程和解码过程的描述方法

5.7.1 位流语法的描述方法

位流语法描述方法类似C语言。位流的语法元素使用粗体字表示，每个语法元素通过名字（用下划线分割的英文字母组，所有字母都是小写）、语法和语义来描述。语法表和正文中语法元素的值用常规字体表示。

某些情况下，可在语法表中应用从语法元素导出的其他变量值，这样的变量在语法表或正文中用不带下划线的小写字母和大写字母混合命名。大写字母开头的变量用于解码当前以及相关的语法结构，也可用于解码后续的语法结构。小写字母开头的变量只在它们所在的小节内使用。

语法元素值的助记符和变量值的助记符与它们的值之间的关系在正文中说明。在某些情况下，二者等同使用。助记符由一个或多个使用下划线分隔的字母组表示，每个字母组以大写字母开始，也可包括多个大写字母。

位串的长度是4的整数倍时，可使用十六进制符号表示。十六进制的前缀是“0x”，例如“0x1a”表示位串“0001 1010”。

条件语句中0表示FALSE，非0表示TRUE。

语法表描述了所有符合本部分的位流语法的超集，附加的语法限制在相关条中说明。

表6给出了描述语法的伪代码例子。当语法元素出现时，表示从位流中读一个数据单元。

表 6 语法描述的伪代码

伪代码	描述符
/*语句是一个语法元素的描述符，或者说明语法元素的存在、类型和数值，下面给出两个例子。*/	
syntax_element	ue(v)
conditioning statement	
/*花括号括起来的语句组是复合语句，在功能上视作单个语句。*/	
{	
statement	
…	
}	
/*“while”语句测试condition是否为TRUE，如果为TRUE，则重复执行循环体，直到condition不为TRUE。*/	
while (condition)	
statement	
/*“do … while”语句先执行循环体一次，然后测试condition是否为TRUE，如果为TRUE，则重复执行循环体，直到condition不为TRUE。*/	

do	
statement	
while (condition)	
/*“if … else”语句首先测试condition，如果为TRUE，则执行primarY语句，否则执行alternative语句。如果alternative语句不需要执行，结构的“else”部分和相关的alternative语句可忽略。*/	
if (condition)	
primarY statement	
else	
alternative statement	
/*“for”语句首先执行initial语句，然后测试condition，如果conditon为TRUE，则重复执行primarY语句和subsequent语句直到condition不为TRUE。*/	
for (initial statement; condition; subsequent statement)	
primarY statement	

解析过程和解码过程用文字和类似 C 语言的伪代码描述。

5.7.2 函数

以下函数用于语法描述。假定解码器中存在一个位流指针，这个指针指向位流中要读取的下一个二进制位的位置。函数由函数名及左右圆括号内的参数构成。函数也可没有参数。

next_start_code()

在位流中寻找下一个起始码，将位流指针指向起始码前缀的第一个二进制位。

time_tag ()

时间戳，具体过程的定义见章节 6.1 。

feat_map_sequence_header ()

特征图序列头，具体过程的定义见章节 6.1 。

feat_map_data()

特征图序数据，具体过程的定义见章节 6.1 。

quant_partitions ()

特征图编码预量化模块中的量化划分，具体过程的定义见章节 6.1 。

repack_order_list()

特征图编码重打包模块中的打包通道顺序列表，具体过程的定义见第 6 章 1 节。

5.7.3 描述符

描述符表示不同语法元素的解析过程，见表7。

表 7 描述符

描述符	说明
b(8)	一个任意取值的字节。解析过程由函数read_bits(8)的返回值规定
f(n)	取特定值的连续n个二进制位。解析过程由函数read_bits(n)的返回值规定
i(n)	n位整数。在语法表中，如果n是“v”，其位数由其他语法元素值确定。解析过程由函数read_bits(n)的返回值规定，该返回值用高位在前的2的补码表示
r(n)	连续n个‘0’。解析过程由函数read_bits(n)的返回值规定
u(n)	n位无符号整数。在语法表中，如果n是“v”，其位数由其他语法元素值确定。解析过程由函数read_bits(n)的返回值规定，该返回值用高位在前的二进制表示

5.7.4 保留、禁止和标记位

本部分定义的位流语法中，某些语法元素的值被标注为“保留”（reserved）或“禁止”（forbidden）。

“保留”定义了一些特定语法元素值用于将来对本部分的扩展。这些值不应出现在符合本部分的位流中。

“禁止”定义了一些特定语法元素值，这些值不应出现在符合本部分的位流中。

“标记位”（marker_bit）指该位的值应为‘1’。

位流中的“保留位”（reserved_bits）表明保留了一些语法单元用于将来对本部分的扩展，解码处理应忽略这些位。“保留位”不应出现从任意字节对齐位置开始的21个以上连续的‘0’。

6 语法和语义

6.1 特征图序列语法

特征图序列定义见表 8。

表 8 特征图序列定义

特征图序列定义	描述符
feat_map_stream( ) {	
feat_map_sequence_header( )	
do {	
time_tag( )	
do {	
if (next_bits(32) == feat_map_start_code)	
feat_map_data( )	
} while ((next_bits(32) == feat_map_start_code)	
} while ((next_bits(32) != feat_map_sequence_end_code)	
if (next_bits(32) == feat_map_sequence_end_code)	
feat_map_sequence_end_code	f(32)
}	

其中，特征图序列头定义见表 9。

表 9 特征图序列头定义

特征图序列头定义	描述符
feat_map_sequence_header( ){	
feat_map_sequence_start_code	f(32)
applied_video_codec	u(3)
feat_extractor_id	u(3)
reserved_bits	r(2)
next_start_code( )	
}	

其中，时间戳定义见表 10。

表 10 时间戳定义

时间戳定义	描述符
time_tag ( ){	
time_tag_start_code	f(32)
universal_time_flag	u(1)
if (universal_time_flage)	
universal_time	u(63)
else	
interval_time	u(15)
next_start_code( )	
}	

其中，特征图数据定义见表 11。

表 11 特征图序数据定义

特征图数据定义	描述符
feat_map_data( ){	
feat_map_start_code	f(32)
feat_type_id	u(8)
feat_integer	u(1)
BitDepth_compact	u(2)
if (!feat_integer) {	
pre_quant_mode	u(3)
max_feat_digit	u(32)
if (pre_quant_mode == $(010)_2$) {	
heritage_flag	u(1)
if (!heritage_flag) {	
quant_partitions( )	

reserved_bits	r(2)
}	
else {	
reserved_bits	r(5)
}	
repack_mode	u(2)
feat_map_pad_h	u(3)
feat_map_pad_w	u(3)
if (repack_mode == (00)2) {	
heritage_flag	u(1)
if (!heritage_flag) {	
repack_order_list( )	
}	
}	
if (repack_mode == (01)2) {	
heritage_flag	u(1)
if (!heritage_flag) {	
repack_tile_h	u(12)
repack_tile_w	u(12)
repack_tile_c	u(16)
}	
}	
if (repack_mode == (10)2) {	
heritage_flag	u(1)
if (!heritage_flag) {	
repack_tile_h	u(12)
repack_tile_w	u(12)
repack_order_list( )	
}	
}	
video_codec_stream( )	
next_start_code( )	
}	

其中，量化划分定义见表 12。

表 12 量化划分定义

量化划分	描述符
quant_partitions ( ){	
for(k=0; k<=$2^{BitDepth\_compact+1}$; k++) {	
quant_partitions_bound	u(32)

}	
}	

其中，重打包通道顺序列表定义见表 13。

表 13 重打包通道顺序列表定义

重打包通道顺序列表	描述符
repack_order_list ( ){	
total_order_number	u(16)
for(k=0; k<total_order_number; k++) {	
order_index	u(16)
}	
}	

6.2 特征图序列语义

特征图序列结束码 feat_map_sequence_end_code

位串‘0x000000E0’。标识特征图序列的结束。

特征图序列起始码 feat_map_sequence_start_code

位串‘0x000000E1’。标识特征图序列的开始。

所使用的视频编解码器标号 applied_video_codec

3 位无符号整数。值为‘000’时代表 AVS3，值为‘001’时代表 HEVC，其余值预留。

特征提取器标号 feat_extractor_id

3 位无符号整数。标明即将传输的特征图由此标号代表的网络模型提取。

时间戳起始码 time_tag_start_code

位串‘0x000000E2’。标识时间戳的开始。

全局时间标志 universal_time_flag

二值变量。值为‘1’表示当前传输的时间为全局时间；值为‘0’表示当前传输的时间为当前特征与上一个特征之间的时间间隔。

全局时间 universal_time

63 位无符号整数。补全最高位‘0’之后，转换成 64 位浮点数（数据格式遵循 IEEE754 标准）。表示 1970 纪元后经过的浮点秒数。

间隔时间 interval_time

15 位无符号整数。表示当前特征提取时间与上次提取时间间隔为 interval_time*0.01 秒。

特征图数据起始码 feat_map_start_code

位串‘0x000000E3’。标识特征图数据的开始。

特征类型标号 feat_type_id

8 位无符号整数。标明当前传输的特征图所属类型编号。

整型特征标记 feat_integer

二值变量。值为‘1’表示当前传输特征图为整型特征图；值为‘1’表示当前传输特征图为浮点型特征图。

特征比特深度紧缩表示 bit_depth_compact

2 位无符号整数。特征图/预量化后的特征图比特深度为 $2^{bit\_depth\_compact+1}$。

预量化模式 pre_quant_mode

3 位无符号整数。其数值指定预量化方法。其中数值‘000’代表均匀量化，数值‘001’代表对数量化，数值‘001’代表自定义标量量化，其余值预留。

特征最大值 max_feat_digit

32 位无符号整数。使用时转换成 32 位浮点数（数据格式遵循 IEEE754 标准）。代表原特征图最大值，用于量化及反量化。

量化划分边界 quant_partitions_bound

32 位无符号整数。使用时转换成 32 位浮点数（数据格式遵循 IEEE754 标准）。代表量化划分列表中的边界数值。

重打包模式 repack_mode

2 位无符号整数。其数值指定重打包方法。其中数值‘00’代表特征图通道按指定顺序叠加；数值‘01’代表特征图通道按默认顺序平铺；数值‘10’代表特征图通道按指定顺序平铺；数值‘11’为预留模式。

特征图填充高度 feat_map_pad_h

3 位无符号整数。代表特征图输入视频编码器前填充的像素高度。

特征图填充宽度 feat_map_pad_w

3 位无符号整数。代表特征图输入视频编码器前填充的像素宽度。

继承标记 heritage_flag

二值变量。值为‘1’表示当前传输特征图重打包操作中使用的参数与前一时间点下相同类型特征图重打包参数相同，参数将不再次传输；值为‘0’则会重新传输新的重打包参数。

顺序列表长度 total_order_number

16 位无符号整数。表示重打包通道顺序列表的长度。

顺序索引 order_index

16 位无符号整数。表示重打包通道顺序列表中的引索数值。

重打包平铺个数（高方向） repack_tile_h

12 位无符号整数。表示特征图通道平铺过程中，高方向上平铺的通道个数。

重打包平铺个数（宽方向） repack_tile_w

12 位无符号整数。表示特征图通道平铺过程中，宽方向上平铺的通道个数。

特征图通道数 repack_tile_c

16 位无符号整数。表示特征图总通道数量。

7 特征图压缩标准

7.1 特征图特指卷积神经网特征图数据

络中卷积层及池化层产生的特征数组。其数据形式为形状为[H, W, C]的三维数组，数组元素值为非负数。其中H为特征图高，W为特征图宽，C为特征图通道。其数据类型可为浮点型或整型。其中浮点型多为32位浮点数，整型多为2位、4位、8位、16位整数。

特征图可由深度神经网络提取，其过程可参考附录A。

7.2 解码框架

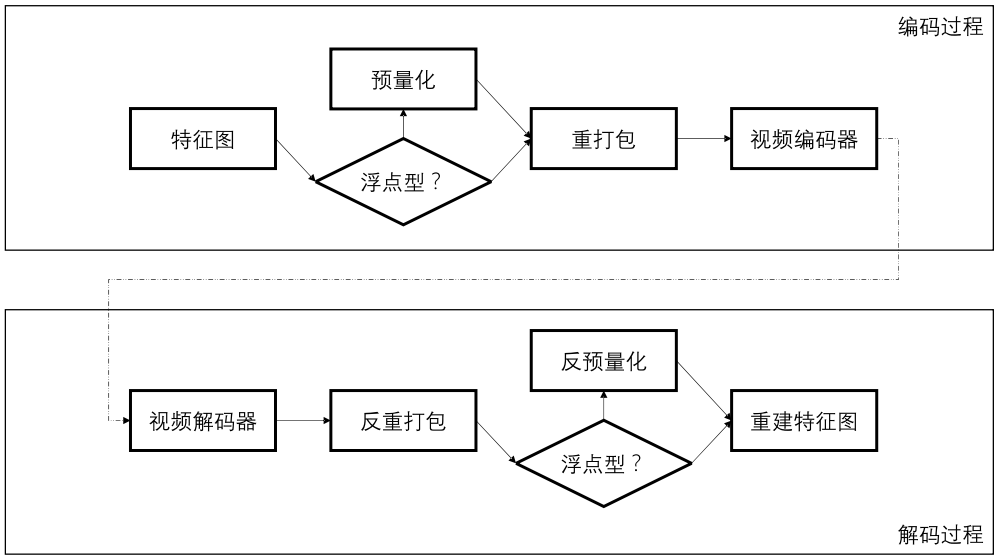

图 1 特征图解码流程

特征图编解码流程见图1，包含三个主要模块：预量化/反预量化；重打包/反重打包；传统视频编码/解码。

7.2.1 预量化/反预量化

当原始输入特征图为浮点型时，需要对特征图进行预量化，使其转化为符合传统视频编码器输入要求的整型数据。预量化模块有多个模式可选，分别为均匀量化、对数量化、自定义标量量化。一些量化重要参数见附录B。

7.2.1.1 均匀量化

均匀量化过程见式（1）：

$$D' = rint\left(\frac{D}{max\_feat\_digir} \times \left(2^{BitDepth\_compact+1} - 1\right)\right), \quad (1)$$

式中：

D——原始特征数据；

D'——预量化后的数据；

$rint(\cdot)$——四舍五入函数，用以返回最接近输入数值的整型数值；

max_feat_digit——特征图数组D的最大值，

$2^{BitDepth\_compact+1}$——量化比特精度；

max_feat_digit——定义参见章节6.2；

bit_depth_compact——定义参见章节6.2。

相对应地，均匀量化的反量化过程见式（2）：

$$F = \frac{max\_feat\_digit}{2^{bit\_depth\_compact+1} - 1} \times float(F'), \quad (2)$$

式中：

F——量化后的数据；

F'——反量化恢复的数据；

$float(\cdot)$——将输入的整型数据转化为浮点型数据。

7.2.1.2 对数量化

对数量化过程见式（3）：

$$D' = rint\left(\frac{log_2(D+1)}{max\_feat\_digit} \times \left(2^{BitDepth\_compact+1} - 1\right)\right), \tag{3}$$

式中：

D——原始数据；

D'——预量化后数据；

$log_2(\cdot)$——返回输入以2为底的对数值；

max_feat_digit——经过$log_2(D+1)$对数变换后的特征图数组最大值；

$2^{BitDepth\_compact+1}$——量化比特精度；

max_feat_digit——定义参见章节6.2；

bit_depth_compact——定义参见章节6.2。

相对应地，对数量化的反量化过程见式（4）：

$$F = 2^{\left(\frac{max\_feat\_digit}{2^{BitDepth\_compact+1}-1} \times float(F')-1\right)}, \tag{4}$$

式中：

F'——量化后的数据；

F——反量化恢复的数据。

7.2.1.3 自定义标量量化

自定义标量量化模式下，根据特征图数据统计特性，通过手工设计、机器学习等方法得出量化划分quant_partitions。量化过程将输入特征图D中属于特定量化区间的元素映赋予相应数值。量化划分包含若干个连续且不重叠的数值范围（量化区间）。为指定量化区间，量化划分quant_partitions为一个包含量化区间端点的列表。

例如，四个量化区间将将实数轴分为{x: x ≤ 0}，{x: 0< x ≤ 1}，{x: 1 < x ≤ 3}，{x: 3 < x}四段，则此时，量化划分$\mathbf{quant_partitions}$可记作数组[0,1,3]。

反量化过程由预量化后的数据D_q结合量化划分$\mathbf{quant_partitions}$恢复出特征矩阵$D_r$。该过程用C语言伪代码表示见表14。

表14 反量化过程伪代码

```
D_r[len(D_q)] = {0};
for(i=0; i<len(quant_partitions); i++) {
    if(i == 0) {
        continue;
    }
    else {
        value = (quant_partitions[i]+quant_partitions[i-1]) / 2;
```

```
            for(j=0; j<len(D_q); j++) {
                if(D_q[j] == i) {
                    D_r[j] == value;
                }
            }
        }
}
for(j=0; j<len(D_q); j++) {
    if(D_q[j] > i) {
        D_r[j] = value;
    }
}
```

7.2.2 重打包/反重打包

重打包模块将原始特征图三维数组变换为符合传统视频编码器输入要求的YUV400颜色编码格式。同时通过改变特征图的组合方式，提高传统视频编码器对待特征图数据的编码效率。重打包/反重打包有多个模式可选，分别为特征图指定顺序叠加、特征图默认顺序平铺、特征图指定顺序平铺。

7.2.2.1 特征图指定顺序叠加

在该模式下，特征图的每个通道对应传统视频编码器输入数据中的一帧。特征图的高、宽被填充至符合传统视频编码器输入要求的高度与宽度。特征图通道顺序由**repack_order_list**记录。解码过程中，设反重打包模块的输入D'是形状为$[H', W', C]$的三维数组。首先去除重打包过程中填充的数值，数组变为$D_c = D'[0: H' - \mathbf{feat_map_pad_h}, 0: W' - \mathbf{feat_map_pad_w}, C']$，其中**feat_map_pad_h**和**feat_map_pad_w**为重打包过程中特征图填充的像素长度和宽度，参见章节6.2 。之后对去除填充值后的数组进行重新排序得到该模块的输出数组D，该过程用C语言伪代码表示见表15。

表15 特征图指定顺序叠加过程伪代码

```
D[H'-feat_map_pad_h][W'-feat_map_pad_w][C'] = {0};
for(c=0; c<length(reapck_order_list); c++) {
    for(i=0; i<H'-feat_map_pad_h; i++) {
        for(j=0; j<W'-feat_map_pad_w; j++) {
            D[i][j][repack_order_list[c]] =  D_c[i][j][c];
        }
    }
}
```

其中，**repack_order_list**保存至下一次解码过程。根据heritage_flag标志进行判断。该标志用于判断是否继承前一次解码过程的repack_order_list，若下一次解码中，heritage_flag为真，则沿用本次的repack_order_list参数，heritage_flag为假，则会按照新传输的

repack_order_list参数进行反重打包。若解码过程中发现repack_order_list缺省，则默认使用顺序数组，即重打包前原始特征图数据通道顺序。

7.2.2.2 特征图默认顺序平铺

在该模式下,特征图多个通道平铺拼接成一个二维数组作为传统视频编码器输入数据中的一帧。拼接后数组的高、宽被填充至符合传统视频编码器输入要求的高度与宽度。见图2,拼接顺序为原始特征图通道顺序，由数组宽方向优先，高方向其次依次排列，当前帧铺满后再创造下一帧继续平铺，直到特征图所有通道均平铺完毕。

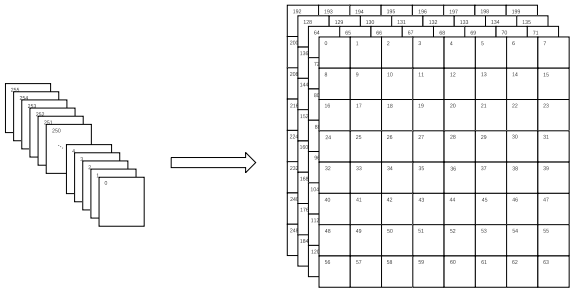

图2 特征图默认顺序平铺

解码过程中，设反重打包模块的输入D'为形状为$[H', W', C']$的三维数组。首先去除重打包过程中填充的数值，数组变为$D_t = D'[0:H' - \mathbf{feat_map_pad_h}, 0:W' - \mathbf{feat_map_pad_w}, C']$，其中**feat_map_pad_h**和**feat_map_pad_w**为重打包过程中特征图填充的像素长度和宽度，参见章节6.2。之后对去除填充值后的数组进行重新排序得到该模块的输出数组D，该过程用python伪代码表示见表16。

表16 特征图默认顺序平铺过程伪代码

```
feat_h = (H'-feat_map_pad_h) / repack_tile_h;
feat_w = (W'-feat_map_pad_w) / repack_tile_w;
D[feat_h][feat_w][repack_tile_c] = {0};
cnt = h = w = 0;
for(c=0; c<C'; c++) {
    for(i=0; i<feat_h; i++) {
        for(j=0; j<feat_w; j++) {
            if(cnt>repacl_tile_c || h>repack_tile_h || w>repack_tile_w) {
                break;
            }
            D[i][j][cnt] =  D_t[h*feat_h+i][w*feat_w+j][c];
        }
    }
    cnt = cnt+1
    if(w<repack_tile_w-1) {
        w = w+1;
```

```
    }
    else {
        w = 0;
        h = h+1;
        if(h>=repack_tile_h-1) {
            break;
        }
    }
}
```

其中，**repack_tile_h**和**repack_tile_w**分别为特征图通道平铺过程中高方向和宽方向上平铺的通道个数，**repack_tile_c**为特征图总通道数量，参见章节6.2 。

7.2.2.3 特征图指定顺序平铺

该模式解码过程先对平铺过程（参见章节7.2.2.2）进行解码，再对特征图通道顺序解码（参见章节7.2.2.1）。详细解码过程如下：

设反重打包模块的输入D'为形状为$[H', W', C']$的三维数组。首先去除重打包过程中填充的数值，数组变为$D_t = D'[0: H' - \mathbf{feat_map_pad_h}, 0: W' - \mathbf{feat_map_pad_w}, C']$，其中**feat_map_pad_h**和**feat_map_pad_w**为重打包过程中特征图填充的像素长度和宽度，参见章节6.2。之后过程用C语言伪代码表示见表17。

表17 特征图指定顺序平铺过程伪代码

```
feat_h = H'-feat_map_pad_h / repack_tile_h;
feat_w = W'-feat_map_pad_w / repack_tile_w;
D_c[feat_h][feat_w][repack_tile_c] = {0};
cnt = h = w = 0;
for(c=0; c<C'; c++) {
    for(i=0; i<feat_h; i++) {
        for(j=0; j<feat_w; j++) {
        D_c[i][j][cnt] = D_t[h*feat_h+i][w*feat_w+j][c];
        }
    }
    cnt = cnt+1;
    if(w<repack_tile_w-1) {
        w = w+1;
    else {
        w = 0;
        h = h+1;
        if(h>=repack_tile_h-1) {
            break;
        }
    }
}
```

```
D[feat_h][feat_w][repack_tile_c] = {0};
for(c=0; c<length(repack_order_list); c++) {
    for(i=0; i<feat_h; i++) {
        for(j=0; j<feat_w; j++) {
            D[i][j][repack_order_list[c]] = D_c[i][j][c];
        }
    }
}
```

其中，**repack_order_list**保存至下次一解码过程。根据heritage_flag标志进行判断。该标志用于判断是否继承前一次解码过程的repack_order_list，若下一次解码中，heritage_flag为真，则沿用本次的repack_order_list参数，heritage_flag为假，则会按照新传输的repack_order_list参数进行反重打包。若解码过程中发现**repack_order_list**缺省，则默认使用顺序数组，即重打包前原始特征图数据通道顺序。**repack_tile_h**和**repack_tile_w**分别为特征图通道平铺过程中高方向和宽方向上平铺的通道个数，**repack_tile_c**为特征图总通道数量，参见章节6.2 。

7.2.3 视频编码/解码器

经过预量化、重打包后的特征图数组数据以YUV视频数据形式送入传统视频编码器进行压缩编码。

在解码端收到特征码流后，首先对其中包含的视频码流video_codec_stream( )进行解码。视频码流以表11描述的语法形式包含在特征图数据码流中，符合相应视频编解码器的语法规范。使用对应视频解码器解码后，获得YUV格式的视频数据，作为反重打包模式的输入。

8 与传统视频编码标准的接口

本部分旨在规定如何扩展传统编码标准（例如HEVC、AVS等），以便支持本标准所定义的视频特征编码。对HEVC或者AVS的帧码流均可通过video_codec_stream( )进行扩展，见表18。

表18 与传统视频编码标准的接口

特征图数据定义	描述符
feat_map_data( ){	
feat_map_start_code	f(32)
feat_type_id	u(8)
feat_integer	u(1)
BitDepth_compact	u(2)
if (!feat_integer) {	
pre_quant_mode	u(3)
maxFeature	u(32)
reserved_bits	r(2)
}	
else {	

reserved_bits	r(5)
}	
repack_mode	u(2)
feat_map_pad_h	u(3)
feat_map_pad_w	u(3)
if (repack_mode == $(00)_2$) {	
heritage_flag	u(1)
if (!heritage_flag) {	
repack_order_list	ae(v)
}	
}	
if (repack_mode == $(01)_2$) {	
heritage_flag	u(1)
if (!heritage_flag) {	
repack_tile_h	u(12)
repack_tile_w	u(12)
repack_tile_c	u(12)
}	
}	
if (repack_mode == $(10)_2$) {	
heritage_flag	u(1)
if (!heritage_flag) {	
repack_tile_h	u(12)
repack_tile_w	u(12)
repack_order_list	ae(v)
}	
}	
video_codec_stream( )	
next_start_code( )	
}	

附录 A

（资料性附录）
深度特征图提取过程

深度特征图一般指深度卷积神经网络中间层的输出。

深度卷积神经网络主要由多个卷积层和池化层叠加组成。卷积层和池化层的输入为形状为[H', W', C']的张量，其中H'为张量高度，W'为张量宽度，C'为张量通道数；其输出为卷积层和池化层的输入为形状为[H, W, C]的张量，其中H为张量高度，W为张量宽度，C为张量通道数。前一个卷积层/池化层的输出可以作为下一个卷积层/池化层的输入，网络的起始输入一般为图像数据的张量[H_i, W_i, C_i],其中H_i为图像高度，W_i为图像宽度，C_i为图像通道数，RGB图像的通道数一般为3。

任意卷积层或池化层输出的形状为[H, W, C]的张量均可以作为特征图数据，成为本标准中拟编码的原始数据。

附录 B

（资料性附录）
量化重要参数

量化重要参数为在AlexNet、VGGNet以及ResNet不同层上对ImageNet2012数据集中1000个类别图像进行分析得到的参数数据，具有一定的参考性，具体参数信息见表B.1-B.6。

表B.1 2bit量化重要参数

量化值	网络模型中间层					
	AlexNet 4th	AlexNet 5th	VGG 9th	VGG 12th	ResNet18 5th	ResNet18 13th
0	[0,7.3351)	[0,4.8656)	[0,12.0838)	[0,4.4912)	[0,0.9336)	[0,0.5403)
1	[7.3351,22.0053)	[4.8656,14.5968)	[12.0838,36.2514)	[4.4912,13.4735)	[0.9336,2.8007)	[0.5403,1.6208)
2	[22.0053,36.6755)	[14.5968,24.3279)	[36.2514,60.4190)	[13.4735,22.4558)	[2.8007,4.6678)	[1.6208,2.7013)
3	[36.6755,44.0106]	[24.3280,29.1936]	[60.4190,72.5028]	[22.4558,26.9470]	[4.6678,5.6013]	[2.7013,3.2416]

表B.2 3bit量化重要参数

量化值	网络模型中间层					
	AlexNet 4th	AlexNet 5th	VGG 9th	VGG 12th	ResNet18 5th	ResNet18 13th
0	[0,3.1436)	[0,2.0853)	[0.5.1788)	[0,,1.9248)	[0,0.4001)	[0,0.2315)
1	[3.1436,9.4308)	[2.0853,6.2558)	[5.1788,15.5363)	[1.9248,5.7744)	[0.4001,1.2003)	[0.2315,0.6946)
2	[9.4308,15.7181)	[6.2558,10.4263)	[15.5363,25.8939)	[5.7744,9.6239)	[1.2003,2.0005)	[0.6946,1.1577)
3	[15.7181,22.0053)	[10.4263,14.5968)	[25.8939,36.2514)	[9.6239,13.4735)	[2.0005,2.8007)	[1.1577,1.6208)
4	[22.0053,28.2925)	[14.5968,18.7673)	[36.2514,46.6089)	[13.4735,17.3231)	[2.8007,3.6009)	[1.6208,2.0839)
5	[28.2925,34.5798)	[18.7673,22.9378)	[46.6089,56.9665)	[17.3231,21.1727)	[3.6009,4.4010)	[2.0839,2.5469)
6	[34.5798,40.8670)	[22.9378,27.1083)	[56.9665,67.3240)	[21.1727,25.0222)	[4.4010,5.2012)	[2.5469,3.0101)
7	[40.8670,44.0106]	[27.1083,29.1936]	[67.3240,72.5028]	[25.0222,26.9470]	[5.2012,5.6013]	[3.0101,3.2416]

表 B.3 4bit 量化重要参数

量化值	网络模型中间层					
	AlexNet 4th	AlexNet 5th	VGG 9th	VGG 12th	ResNet18 5th	ResNet18 13th
0	[0,1.4670)	[0,0.9731)	[0,2.4168)	[0,0.8982)	[0,0.1867)	[0,0.1081)
1	[1.4670,4.4011)	[0.9731,2.9194)	[2.4168,7.2503)	[0.8982,2.6947)	[0.1867,0.5601)	[0.1081,0.3242)
2	[4.4011,7.3351)	[2.9194,4.8656)	[7.2503,12.0838)	[2.6847,4.4912)	[0.5601,0.9336)	[0.3242,0.5403)
3	[7.3351,10.2691)	[4.8656,6.8118)	[12.0838,16.9173)	[4.4912,6.2876)	[0.9336,1.3070)	[0.5403,0.7564)
4	[10.2691,13.2032)	[6.8118,8.7581)	[16.9173,21.7508)	[6.2876,8.0841)	[1.3070,1.6804)	[0.7564,0.9725)
5	[13.2032,16.1372)	[8.7581,10.7043)	[21.7508,26.5844)	[8.0841,9.8806)	[1.6804,2.0538)	[0.9725,1.1886)
6	[16.1372,19.0713)	[10.7043,12.6505)	[26.5844,31.4179)	[9.8806,11.6770)	[2.0538,2.4272)	[1.1886,1.4047)
7	[19.0713,22.0053)	[12.6505,14.5968)	[31.4179,36.2514)	[11.6770,13.4735)	[2.4272,2.8007)	[1.4047,1.6208)
8	[22.0053,24.9394)	[14.5968,16.5430)	[36.2514,41.0849)	[13.4735,15.2699)	[2.8007,3.1741)	[1.6208,1.8369)
9	[24.9394,27.8734)	[16.5430,18.4892)	[41.0849,45.9184)	[15.2699,17.0664)	[3.1741,3.5475)	[1.8369,2.0530)
10	[27.8734,30.8074)	[18.4892,20.4355)	[45.9184,50.7520)	[17.0664,18.8629)	[3.5475,3.9209)	[2.0530,2.2691)
11	[30.8074,33.7415)	[20.4355,22.3817)	[50.7520,55.5855)	[18.8629,20.6594)	[3.9209,4.2943)	[2.2691,2.4852)
12	[33.7415,36.6755)	[22.3817,24.3280)	[55.5855,60.4190)	[20.6594,22.4558)	[4.2943,4.6678)	[2.4852,2.7013)

13	[36.6755,39.6096)	[24.3280,26.2742)	[60.4190,65.2525)	[22.4558,24.2523)	[4.6678,5.0412)	[2.7013,2.9174)
14	[39.6096,42.5436)	[26.2742,28.2204)	[65.2525,70.0860)	[24.2523,26.0488)	[5.0412,5.4146)	[2.9174,3.1335)
15	[42.5436,44.0106]	[28.2204,29.1936]	[70.0860,72.5028]	[26.0488,26.9470]	[5.4146,5.6013]	[3.1335,3.2416]

表 B.4 5bit 量化重要参数

量化值	网络模型中间层					
	AlexNet 4th	AlexNet 5th	VGG 9th	VGG 12th	ResNet18 5th	ResNet18 13th
0	[0,0.7098)	[0,0.4709)	[0,1.1694)	[0,0.4346)	[0,0.0903)	[0,0.0523)
1	[0.7098,2.1295)	[0.4709,1.4126)	[1.1694,3.5082)	[0.4346,1.3039)	[0.0903,0.2710)	[0.0523,0.1569)
2	[2.1295,3.5492)	[1.4126,2.3543)	[3.5082,5.8470)	[1.3039,2.1731)	[0.2710,0.4517)	[0.1569,0.2614)
3	[3.5492,4.9689)	[2.3543,3.2960)	[5.8470,8.1858)	[2.1731,3.0424)	[0.4517,0.6324)	[0.2614,0.3660)
4	[4.9689,6.3886)	[3.2960,4.2378)	[8.1858,10.5246)	[3.0424,3.9117)	[0.6324,0.8131)	[0.3660,0.4706)
5	[6.3886,7.8083)	[4.2378,5.1795)	[10.5246,12.8634)	[3.9117,4.7809)	[0.8131,0.9938)	[0.4706,0.5751)
6	[7.8083,9.2280)	[5.1795,6.1212)	[12.8634,15.2022)	[4.7809,5.6502)	[0.9938,1.1745)	[0.5751,0.6797)
7	[9.2280,10.6477)	[6.1212,7.0630)	[15.2022,17.5410)	[5.6502,6.5194)	[1.1745,1.3552)	[0.6797,0.7843)
8	[10.6477,12.0674)	[7.0630,8.0047)	[17.5410,19.8798)	[6.5194,7.3887)	[1.3552,1.5358)	[0.7843,0.8888)
9	[12.0674,13.4871)	[8.0047,8.9464)	[19.8798,22.2186)	[7.3887,8.2580)	[1.5358,1.7165)	[0.8888,0.9934)
10	[13.4871,14.9068)	[8.9464,9.8881)	[22.2186,24.5574)	[8.2580,9.1272)	[1.7165,1.8972)	[0.9934,1.0980)
11	[14.9068,16.3265)	[9.8881,10.8299)	[24.5574,26.8962)	[9.1272,9.9965)	[1.8972,2.0779)	[1.0980,1.2025)
12	[16.3265,17.7462)	[10.8299,11.7716)	[26.8962,29.2350)	[9.9965,10.8657)	[2.0779,2.2586)	[1.2025,1.3071)
13	[17.7462,19.1659)	[11.7716,12.7133)	[29.2350,31.5738)	[10.8657,11.7350)	[2.2586,2.4393)	[1.3071,1.4117)
14	[19.1659,20.5856)	[12.7133,13.6550)	[31.5738,33.9126)	[11.7350,12.6042)	[2.4393,2.6200)	[1.4117,1.5162)
15	[20.5856,22.0053)	[13.6550,14.5968)	[33.9126,36.2514)	[12.6042,13.4735)	[2.6200,2.8007)	[1.5162,1.6208)
16	[22.0053,23.4250)	[14.5968,15.5385)	[36.2514,38.5902)	[13.4735,14.3428)	[2.8007,2.9814)	[1.6208,1.7254)
17	[23.4250,24.8447)	[15.5385,16.4802)	[38.5902,40.9290)	[14.3428,15.2120)	[2.9814,3.1620)	[1.7254,1.8299)
18	[24.8447,26.2644)	[16.4802,17.4220)	[40.9290,43.2678)	[15.2120,16.0813)	[3.1620,3.3427)	[1.8299,1.9345)
19	[26.2644,27.6841)	[17.4220,18.3637)	[43.2678,45.6066)	[16.0813,16.9505)	[3.3427,3.5234)	[1.9345,2.0391)
20	[27.6841,29.1038)	[18.3637,19.3054)	[45.6066,47.9454)	[16.9505,17.8198)	[3.5234,3.7041)	[2.0391,2.1436)
21	[29.1038,30.5235)	[19.3054,20.2471)	[47.9454,50.2842)	[17.8198,18.6891)	[3.7041,3.8848)	[2.1436,2.2482)
22	[30.5235,31.9432)	[20.2471,21.1889)	[50.2842,52.6230)	[18.6891,19.5583)	[3.8848,4.0655)	[2.2482,2.3528)
23	[31.9432,33.3629)	[21.1889,22.1306)	[52.6230,54.9618)	[19.5583,20.4276)	[4.0655,4.2462)	[2.3528,2.4573)
24	[33.3629,34.7826)	[22.1306,23.0723)	[54.9618,57.3006)	[20.4276,21.2968)	[4.2462,4.4269)	[2.4573,2.5619)
25	[34.7826,36.2023)	[23.0723,24.0140)	[57.3006,59.6394)	[21.2968,22.1661)	[4.4269,4.6075)	[2.5619,2.6665)
26	[36.2023,37.6220)	[24.0140,24.9558)	[59.6394,61.9782)	[22.1661,23.0353)	[4.6075,4.7882)	[2.6665,2.7710)
27	[37.6220,39.0417)	[24.9558,25.8975)	[61.9782,64.3170)	[23.0353,23.9046)	[4.7882,4.9689)	[2.7710,2.8756)
28	[39.0417,40.4614)	[25.8975,26.8392)	[64.3170,66.6558)	[23.9046,24.7739)	[4.9689,5.1496)	[2.8756,2.9802)
29	[40.4614,41.8811)	[26.8392,27.7810)	[66.6558,68.9946)	[24.7739,25.6431)	[5.1496,5.3303)	[2.9802,3.0847)
30	[41.8811,43.3008)	[27.7810,28.7227)	[68.9946,71.3334)	[25.6431,26.5124)	[5.3303,5.5110)	[3.0847,3.1893)
31	[43.3008,44.0106]	[28.7227,29.1936]	[71.3334,72.5028]	[26.5124,26.9470]	[5.5110,5.6013]	[3.1893,3.2416]

表 B.5 6bit 量化重要参数

量化值	网络模型中间层

	AlexNet 4th	AlexNet 5th	VGG 9th	VGG 12th	ResNet18 5th	ResNet18 13th
0	[0,0.3493)	[0,0.2317)	[0,0.5754)	[0,0.2139)	[0,0.0445)	[0,0.0257)
1	[0.3493,1.0479)	[0.2317,0.6951)	[0.5754,1.7263)	[0.2139,0.6416)	[0.0445,0.1334)	[0.0257,0.0772)
2	[1.0479,1.7465)	[0.6951,1.1585)	[1.7263,2.8771)	[0.6416,1.0693)	[0.1334,0.2223)	[0.0772,0.1286)
3	[1.7465,2.4450)	[1.1585,1.6219)	[2.8771,4.0279)	[1.0693,1.4971)	[0.2223,0.3112)	[0.1286,0.1801)
4	[2.4450,3.1436)	[1.6219,2.0853)	[4.0279,5.1788)	[1.4971,1.9248)	[0.3112,0.4001)	[0.1801,0.2315)
5	[3.1436,3.8422)	[2.0853,2.5486)	[5.1788,6.3296)	[1.9248,2.3525)	[0.4001,0.4890)	[0.2315,0.2830)
6	[3.8422,4.5408)	[2.5486,3.0120)	[6.3296,7.4804)	[2.3525,2.7802)	[0.4890,0.5779)	[0.2830,0.3345)
7	[4.5408,5.2394)	[3.0120,3.4754)	[7.4804,8.6313)	[2.7802,3.2080)	[0.5779,0.6668)	[0.3345,0.3859)
8	[5.2394,5.9379)	[3.4754,3.9388)	[8.6313,9.7821)	[3.2080,3.6357)	[0.6668,0.7557)	[0.3859,0.4374)
9	[5.9379,6.6365)	[3.9388,4.4022)	[9.7821,10.9330)	[3.6357,4.0634)	[0.7557,0.8446)	[0.4374,0.4888)
10	[6.6365,7.3351)	[4.4022,4.8656)	[10.9330,12.0838)	[4.0634,4.4912)	[0.8446,0.9336)	[0.4888,0.5403)
11	[7.3351,8.0337)	[4.8656,5.3290)	[12.0838,13.2346)	[4.4912,4.9189)	[0.9336,1.0225)	[0.5403,0.5917)
12	[8.0337,8.7323)	[5.3290,5.7924)	[13.2346,14.3855)	[4.9189,5.3466)	[1.0225,1.1114)	[0.5917,0.6432)
13	[8.7323,9.4308)	[5.7924,6.2558)	[14.3855,15.5363)	[5.3466,5.7744)	[1.1114,1.2003)	[0.6432,0.6946)
14	[9.4308,10.1294)	[6.2558,6.7192)	[15.5363,16.6872)	[5.7744,6.2021)	[1.2003,1.2892)	[0.6946,0.7461)
15	[10.1294,10.8280)	[6.7192,7.1825)	[16.6872,17.8380)	[6.2021,6.6298)	[1.2892,1.3781)	[0.7461,0.7975)
16	[10.8280,11.5266)	[7.1825,7.6459)	[17.8380,18.9888)	[6.6298,7.0576)	[1.3781,1.4670)	[0.7975,0.8490)
17	[11.5266,12.2252)	[7.6459,8.1093)	[18.9888,20.1397)	[7.0576,7.4853)	[1.4670,1.5559)	[0.8490,0.9004)
18	[12.2252,12.9238)	[8.1093,8.5727)	[20.1397,21.2905)	[7.4853,7.9130)	[1.5559,1.6448)	[0.9004,0.9519)
19	[12.9238,13.6223)	[8.5727,9.0361)	[21.2905,22.4413)	[7.9130,8.3407)	[1.6448,1.7337)	[0.9519,1.0034)
20	[13.6223,14.3209)	[9.0361,9.4995)	[22.4413,23.5922)	[8.3407,8.7685)	[1.7337,1.8227)	[1.0034,1.0548)
21	[14.3209,15.0195)	[9.4995,9.9629)	[23.5922,24.7430)	[8.7685,9.1962)	[1.8227,1.9116)	[1.0548,1.1063)
22	[15.0195,15.7181)	[9.9629,10.4263)	[24.7430,25.8939)	[9.1962,9.6239)	[1.9116,2.0005)	[1.1063,1.1577)
23	[15.7181,16.4167)	[10.4263,10.8897)	[25.8939,27.0447)	[9.6239,10.0517)	[2.0005,2.0894)	[1.1577,1.2092)
24	[16.4167,17.1152)	[10.8897,11.3530)	[27.0447,28.1955)	[10.0517,10.4794)	[2.0894,2.1783)	[1.2092,1.2606)
25	[17.1152,17.8138)	[11.3530,11.8164)	[28.1955,29.3464)	[10.4794,10.9071)	[2.1783,2.2672)	[1.2606,1.3121)
26	[17.8138,18.5124)	[11.8164,12.2798)	[29.3464,30.4972)	[10.9071,11.3349)	[2.2672,2.3561)	[1.3121,1.3635)
27	[18.5124,19.2110)	[12.2798,12.7432)	[30.4972,31.6480)	[11.3349,11.7626)	[2.3561,2.4450)	[1.3635,1.4150)
28	[19.2110,19.9096)	[12.7432,13.2066)	[31.6480,32.7989)	[11.7626,12.1903)	[2.4450,2.5339)	[1.4150,1.4664)
29	[19.9096,20.6081)	[13.2066,13.6700)	[32.7989,33.9497)	[12.1903,12.6180)	[2.5339,2.6228)	[1.4664,1.5179)
30	[20.6081,21.3067)	[13.6700,14.1334)	[33.9497,35.1006)	[12.6180,13.0458)	[2.6228,2.7118)	[1.5179,1.5693)
31	[21.3067,22.0053)	[14.1334,14.5968)	[35.1006,36.2514)	[13.0458,13.4735)	[2.7118,2.8007)	[1.5693,1.6208)
32	[22.0053,22.7039)	[14.5968,15.0602)	[36.2514,37.4022)	[13.4735,13.9012)	[2.8007,2.8896)	[1.6208,1.6723)
33	[22.7039,23.4025)	[15.0602,15.5236)	[37.4022,38.5531)	[13.9012,14.3290)	[2.8896,2.9785)	[1.6723,1.7237)
34	[23.4025,24.1011)	[15.5236,15.9869)	[38.5531,39.7039)	[14.3290,14.7567)	[2.9785,3.0674)	[1.7237,1.7752)
35	[24.1011,24.7996)	[15.9869,16.4503)	[39.7039,40.8547)	[14.7567,15.1844)	[3.0674,3.1563)	[1.7752,1.8266)
36	[24.7996,25.4982)	[16.4503,16.9137)	[40.8547,42.0056)	[15.1844,15.6122)	[3.1563,3.2452)	[1.8266,1.8781)
37	[25.4982,26.1968)	[16.9137,17.3771)	[42.0056,43.1564)	[15.6122,16.0399)	[3.2452,3.3341)	[1.8781,1.9295)
38	[26.1968,26.8954)	[17.3771,17.8405)	[43.1564,44.3073)	[16.0399,16.4676)	[3.3341,3.4230)	[1.9295,1.9810)
39	[26.8954,27.5940)	[17.8405,18.3039)	[44.3073,45.4581)	[16.4676,16.8953)	[3.4230,3.5119)	[1.9810,2.0324)
40	[27.5940,28.2925)	[18.3039,18.7673)	[45.4581,46.6089)	[16.8953,17.3231)	[3.5119,3.6009)	[2.0324,2.0839)
41	[28.2925,28.9911)	[18.7673,19.2307)	[46.6089,47.7598)	[17.3231,17.7508)	[3.6009,3.6898)	[2.0839,2.1353)

42	[28.9911,29.6897)	[19.2307,19.6941)	[47.7598,48.9106)	[17.7508,18.1785)	[3.6898,3.7787)	[2.1353,2.1868)
43	[29.6897,30.3883)	[19.6941,20.1575)	[48.9106,50.0615)	[18.1785,18.6063)	[3.7787,3.8676)	[2.1868,2.2382)
44	[30.3883,31.0869)	[20.1575,20.6208)	[50.0615,51.2123)	[18.6063,19.0340)	[3.8676,3.9565)	[2.2382,2.2897)
45	[31.0869,31.7854)	[20.6208,21.0842)	[51.2123,52.3631)	[19.0340,19.4617)	[3.9565,4.0454)	[2.2897,2.3412)
46	[31.7854,32.4840)	[21.0842,21.5476)	[52.3631,53.5140)	[19.4617,19.8895)	[4.0454,4.1343)	[2.3412,2.3926)
47	[32.4840,33.1826)	[21.5476,22.0110)	[53.5140,54.6648)	[19.8895,20.3172)	[4.1343,4.2232)	[2.3926,2.4441)
48	[33.1826,33.8812)	[22.0110,22.4744)	[54.6648,55.8156)	[20.3172,20.7449)	[4.2232,4.3121)	[2.4441,2.4955)
49	[33.8812,34.5798)	[22.4744,22.9378)	[55.8156,56.9665)	[20.7449,21.1727)	[4.3121,4.4010)	[2.4955,2.5470)
50	[34.5798,35.2784)	[22.9378,23.4012)	[56.9665,58.1173)	[21.1727,21.6004)	[4.4010,4.4900)	[2.5470,2.5984)
51	[35.2784,35.9769)	[23.4012,23.8646)	[58.1173,59.2682)	[21.6004,22.0281)	[4.4900,4.5789)	[2.5984,2.6499)
52	[35.9769,36.6755)	[23.8646,24.3280)	[59.2682,60.4190)	[22.0281,22.4558)	[4.5789,4.6678)	[2.6499,2.7013)
53	[36.6755,37.3741)	[24.3280,24.7913)	[60.4190,61.5698)	[22.4558,22.8836)	[4.6678,4.7567)	[2.7013,2.7528)
54	[37.3741,38.0727)	[24.7913,25.2547)	[61.5698,62.7207)	[22.8836,23.3113)	[4.7567,4.8456)	[2.7528,2.8042)
55	[38.0727,38.7713)	[25.2547,25.7181)	[62.7207,63.8715)	[23.3113,23.7390)	[4.8456,4.9345)	[2.8042,2.8557)
56	[38.7713,39.4698)	[25.7181,26.1815)	[63.8715,65.0223)	[23.7390,24.1668)	[4.9345,5.0234)	[2.8557,2.9071)
57	[39.4698,40.1684)	[26.1815,26.6449)	[65.0223,66.1732)	[24.1668,24.5945)	[5.0234,5.1123)	[2.9071,2.9586)
58	[40.1684,40.8670)	[26.6449,27.1083)	[66.1732,67.3240)	[24.5945,25.0222)	[5.1123,5.2012)	[2.9586,3.0101)
59	[40.8670,41.5656)	[27.1083,27.5717)	[67.3240,68.4749)	[25.0222,25.4500)	[5.2012,5.2901)	[3.0101,3.0615)
60	[41.5656,42.2642)	[27.5717,28.0351)	[68.4749,69.6257)	[25.4500,25.8777)	[5.2901,5.3791)	[3.0615,3.1130)
61	[42.2642,42.9627)	[28.0351,28.4985)	[69.6257,70.7765)	[25.8777,26.3054)	[5.3791,5.4680)	[3.1130,3.1644)
62	[42.9627,43.6613)	[28.4985,28.9619)	[70.7765,71.9274)	[26.3054,26.7331)	[5.4680,5.5569)	[3.1644,3.2159)
63	[43.6613,44.0106]	[28.9619,29.1935]	[71.9274,72.5028]	[26.7331,26.9470]	[5.5569,5.6013]	[3.2159,3.2416]

表 B.6 7bit 量化重要参数

量化值	网络模型中间层					
	AlexNet 4th	AlexNet 5th	VGG 9th	VGG 12th	ResNet18 5th	ResNet18 13th
0	[0,0.1733)	[0,0.1149)	[0,0.2854)	[0,0.1061)	[0,0.0221)	[0,0.0128)
1	[0.1733,0.5198)	[0.1149,0.3448)	[0.2854,0.8563)	[0.1061,0.3183)	[0.0221,0.0662)	[0.0128,0.0383)
2	[0.5198,0.8664)	[0.3448,0.5747)	[0.8563,1.4272)	[0.3183,0.5305)	[0.0662,0.1103)	[0.0383,0.0638)
3	[0.8664,1.2129)	[0.5747,0.8045)	[1.4272,1.9981)	[0.5305,0.7426)	[0.1103,0.1544)	[0.0638,0.0893)
4	[1.2129,1.5594)	[0.8045,1.0344)	[1.9981,2.5690)	[0.7426,0.9548)	[0.1544,0.1985)	[0.0893,0.1149)
5	[1.5594,1.9060)	[1.0344,1.2643)	[2.5690,3.1399)	[0.9548,1.1670)	[0.1985,0.2426)	[0.1149,0.1404)
6	[1.9060,2.2525)	[1.2643,1.4942)	[3.1399,3.7108)	[1.1670,1.3792)	[0.2426,0.2867)	[0.1404,0.1659)
7	[2.2525,2.5991)	[1.4942,1.7240)	[3.7108,4.2817)	[1.3792,1.5914)	[0.2867,0.3308)	[0.1659,0.1914)
8	[2.5991,2.9456)	[1.7240,1.9539)	[4.2817,4.8525)	[1.5914,1.8035)	[0.3308,0.3749)	[0.1914,0.2170)
9	[2.9456,3.2921)	[1.9539,2.1838)	[4.8525,5.4234)	[1.8035,2.0157)	[0.3749,0.4190)	[0.2170,0.2425)
10	[3.2921,3.6387)	[2.1838,2.4136)	[5.4234,5.9943)	[2.0157,2.2279)	[0.4190,0.4631)	[0.2425,0.2680)
11	[3.6387,3.9852)	[2.4136,2.6435)	[5.9943,6.5652)	[2.2279,2.4401)	[0.4631,0.5072)	[0.2680,0.2935)
12	[3.9852,4.3318)	[2.6435,2.8734)	[6.5652,7.1361)	[2.4401,2.6523)	[0.5072,0.5513)	[0.2935,0.3191)
13	[4.3318,4.6783)	[2.8734,3.1033)	[7.1361,7.7070)	[2.6523,2.8644)	[0.5513,0.5954)	[0.3191,0.3446)
14	[4.6783,5.0248)	[3.1033,3.3331)	[7.7070,8.2779)	[2.8644,3.0766)	[0.5954,0.6395)	[0.3446,0.3701)

15	[5.0248,5.3714)	[3.3331,3.5630)	[8.2779,8.8488)	[3.0766,3.2888)	[0.6395,0.6836)	[0.3701,0.3956)
16	[5.3714,5.7179)	[3.5630,3.7929)	[8.8488,9.4197)	[3.2888,3.5010)	[0.6836,0.7277)	[0.3956,0.4212)
17	[5.7179,6.0645)	[3.7929,4.0227)	[9.4197,9.9905)	[3.5010,3.7132)	[0.7277,0.7718)	[0.4212,0.4467)
18	[6.0645,6.4110)	[4.0227,4.2526)	[9.9905,10.5614)	[3.7132,3.9254)	[0.7718,0.8159)	[0.4467,0.4722)
19	[6.4110,6.7575)	[4.2526,4.4825)	[10.5614,11.1323)	[3.9254,4.1375)	[0.8159,0.8600)	[0.4722,0.4977)
20	[6.7575,7.1041)	[4.4825,4.7123)	[11.1323,11.7032)	[4.1375,4.3497)	[0.8600,0.9042)	[0.4977,0.5232)
21	[7.1041,7.4506)	[4.7123,4.9422)	[11.7032,12.2741)	[4.3497,4.5619)	[0.9042,0.9483)	[0.5232,0.5488)
22	[7.4506,7.7972)	[4.9422,5.1721)	[12.2741,12.8450)	[4.5619,4.7741)	[0.9483,0.9924)	[0.5488,0.5743)
23	[7.7972,8.1437)	[5.1721,5.4020)	[12.8450,13.4159)	[4.7741,4.9863)	[0.9924,1.0365)	[0.5743,0.5998)
24	[8.1437,8.4902)	[5.4020,5.6318)	[13.4159,13.9868)	[4.9863,5.1984)	[1.0365,1.0806)	[0.5998,0.6253)
25	[8.4902,8.8368)	[5.6318,5.8617)	[13.9868,14.5576)	[5.1984,5.4106)	[1.0806,1.1247)	[0.6253,0.6509)
26	[8.8368,9.1833)	[5.8617,6.0916)	[14.5576,15.1285)	[5.4106,5.6228)	[1.1247,1.1688)	[0.6509,0.6764)
27	[9.1833,9.5299)	[6.0916,6.3214)	[15.1285,15.6994)	[5.6228,5.8350)	[1.1688,1.2129)	[0.6764,0.7019)
28	[9.5299,9.8764)	[6.3214,6.5513)	[15.6994,16.2703)	[5.8350,6.0472)	[1.2129,1.2570)	[0.7019,0.7274)
29	[9.8764,10.2229)	[6.5513,6.7812)	[16.2703,16.8412)	[6.0472,6.2593)	[1.2570,1.3011)	[0.7274,0.7530)
30	[10.2229,10.5695)	[6.7812,7.0110)	[16.8412,17.4121)	[6.2593,6.4715)	[1.3011,1.3452)	[0.7530,0.7785)
31	[10.5695,10.9160)	[7.0110,7.2409)	[17.4121,17.9830)	[6.4715,6.6837)	[1.3452,1.3893)	[0.7785,0.8040)
32	[10.9160,11.2626)	[7.2409,7.4708)	[17.9830,18.5539)	[6.6837,6.8959)	[1.3893,1.4334)	[0.8040,0.8295)
33	[11.2626,11.6091)	[7.4708,7.7007)	[18.5539,19.1248)	[6.8959,7.1081)	[1.4334,1.4775)	[0.8295,0.8551)
34	[11.6091,11.9556)	[7.7007,7.9305)	[19.1248,19.6956)	[7.1081,7.3203)	[1.4775,1.5216)	[0.8551,0.8806)
35	[11.9556,12.3022)	[7.9305,8.1604)	[19.6956,20.2665)	[7.3203,7.5324)	[1.5216,1.5657)	[0.8806,0.9061)
36	[12.3022,12.6487)	[8.1604,8.3903)	[20.2665,20.8374)	[7.5324,7.7446)	[1.5657,1.6098)	[0.9061,0.9316)
37	[12.6487,12.9953)	[8.3903,8.6201)	[20.8374,21.4083)	[7.7446,7.9568)	[1.6098,1.6539)	[0.9316,0.9572)
38	[12.9953,13.3418)	[8.6201,8.8500)	[21.4083,21.9792)	[7.9568,8.1690)	[1.6539,1.6980)	[0.9572,0.9827)
39	[13.3418,13.6883)	[8.8500,9.0799)	[21.9792,22.5501)	[8.1690,8.3812)	[1.6980,1.7421)	[0.9827,1.0082)
40	[13.6883,14.0349)	[9.0799,9.3098)	[22.5501,23.1210)	[8.3812,8.5933)	[1.7421,1.7863)	[1.0082,1.0337)
41	[14.0349,14.3814)	[9.3098,9.5396)	[23.1210,23.6919)	[8.5933,8.8055)	[1.7863,1.8304)	[1.0337,1.0593)
42	[14.3814,14.7280)	[9.5396,9.7695)	[23.6919,24.2627)	[8.8055,9.0177)	[1.8304,1.8745)	[1.0593,1.0848)
43	[14.7280,15.0745)	[9.7695,9.9994)	[24.2627,24.8336)	[9.0177,9.2299)	[1.8745,1.9186)	[1.0848,1.1103)
44	[15.0745,15.4210)	[9.9994,10.2292)	[24.8336,25.4045)	[9.2299,9.4421)	[1.9186,1.9627)	[1.1103,1.1358)
45	[15.4210,15.7676)	[10.2292,10.4591)	[25.4045,25.9754)	[9.4421,9.6542)	[1.9627,2.0068)	[1.1358,1.1614)
46	[15.7676,16.1141)	[10.4591,10.6890)	[25.9754,26.5463)	[9.6542,9.8664)	[2.0068,2.0509)	[1.1614,1.1869)
47	[16.1141,16.4607)	[10.6890,10.9188)	[26.5463,27.1172)	[9.8664,10.0786)	[2.0509,2.0950)	[1.1869,1.2124)
48	[16.4607,16.8072)	[10.9188,11.1487)	[27.1172,27.6881)	[10.0786,10.2908)	[2.0950,2.1391)	[1.2124,1.2379)
49	[16.8072,17.1537)	[11.1487,11.3786)	[27.6881,28.2590)	[10.2908,10.5030)	[2.1391,2.1832)	[1.2379,1.2635)
50	[17.1537,17.5003)	[11.3786,11.6085)	[28.2590,28.8299)	[10.5030,10.7152)	[2.1832,2.2273)	[1.2635,1.2890)
51	[17.5003,17.8468)	[11.6085,11.8383)	[28.8299,29.4007)	[10.7152,10.9273)	[2.2273,2.2714)	[1.2890,1.3145)
52	[17.8468,18.1934)	[11.8383,12.0682)	[29.4007,29.9716)	[10.9273,11.1395)	[2.2714,2.3155)	[1.3145,1.3400)
53	[18.1934,18.5399)	[12.0682,12.2981)	[29.9716,30.5425)	[11.1395,11.3517)	[2.3155,2.3596)	[1.3400,1.3656)
54	[18.5399,18.8864)	[12.2981,12.5279)	[30.5425,31.1134)	[11.3517,11.5639)	[2.3596,2.4037)	[1.3656,1.3911)
55	[18.8864,19.2330)	[12.5279,12.7578)	[31.1134,31.6843)	[11.5639,11.7761)	[2.4037,2.4478)	[1.3911,1.4166)
56	[19.2330,19.5795)	[12.7578,12.9877)	[31.6843,32.2552)	[11.7761,11.9882)	[2.4478,2.4919)	[1.4166,1.4421)
57	[19.5795,19.9261)	[12.9877,13.2176)	[32.2552,32.8261)	[11.9882,12.2004)	[2.4919,2.5360)	[1.4421,1.4677)

58	[19.9261,20.2726)	[13.2176,13.4474)	[32.8261,33.3970)	[12.2004,12.4126)	[2.5360,2.5801)	[1.4677,1.4932)
59	[20.2726,20.6191)	[13.4474,13.6773)	[33.3970,33.9678)	[12.4126,12.6248)	[2.5801,2.6242)	[1.4932,1.5187)
60	[20.6191,20.9657)	[13.6773,13.9072)	[33.9678,34.5387)	[12.6248,12.8370)	[2.6242,2.6684)	[1.5187,1.5442)
61	[20.9657,21.3122)	[13.9072,14.1370)	[34.5387,35.1096)	[12.8370,13.0491)	[2.6684,2.7125)	[1.5442,1.5697)
62	[21.3122,21.6588)	[14.1370,14.3669)	[35.1096,35.6805)	[13.0491,13.2613)	[2.7125,2.7566)	[1.5697,1.5953)
63	[21.6588,22.0053)	[14.3669,14.5968)	[35.6805,36.2514)	[13.2613,13.4735)	[2.7566,2.8007)	[1.5953,1.6208)
64	[22.0053,22.3518)	[14.5968,14.8266)	[36.2514,36.8223)	[13.4735,13.6857)	[2.8007,2.8448)	[1.6208,1.6463)
65	[22.3518,22.6984)	[14.8266,15.0565)	[36.8223,37.3932)	[13.6857,13.8979)	[2.8448,2.8889)	[1.6463,1.6718)
66	[22.6984,23.0449)	[15.0565,15.2864)	[37.3932,37.9641)	[13.8979,14.1101)	[2.8889,2.9330)	[1.6718,1.6974)
67	[23.0449,23.3915)	[15.2864,15.5163)	[37.9641,38.5349)	[14.1101,14.3222)	[2.9330,2.9771)	[1.6974,1.7229)
68	[23.3915,23.7380)	[15.5163,15.7461)	[38.5349,39.1058)	[14.3222,14.5344)	[2.9771,3.0212)	[1.7229,1.7484)
69	[23.7380,24.0846)	[15.7461,15.9760)	[39.1058,39.6767)	[14.5344,14.7466)	[3.0212,3.0653)	[1.7484,1.7739)
70	[24.0846,24.4311)	[15.9760,16.2059)	[39.6767,40.2476)	[14.7466,14.9588)	[3.0653,3.1094)	[1.7739,1.7995)
71	[24.4311,24.7776)	[16.2059,16.4357)	[40.2476,40.8185)	[14.9588,15.1710)	[3.1094,3.1535)	[1.7995,1.8250)
72	[24.7776,25.1242)	[16.4357,16.6656)	[40.8185,41.3894)	[15.1710,15.3831)	[3.1535,3.1976)	[1.8250,1.8505)
73	[25.1242,25.4707)	[16.6656,16.8955)	[41.3894,41.9603)	[15.3831,15.5953)	[3.1976,3.2417)	[1.8505,1.8760)
74	[25.4707,25.8173)	[16.8955,17.1253)	[41.9603,42.5312)	[15.5953,15.8075)	[3.2417,3.2858)	[1.8760,1.9016)
75	[25.8173,26.1638)	[17.1253,17.3552)	[42.5312,43.1021)	[15.8075,16.0197)	[3.2858,3.3299)	[1.9016,1.9271)
76	[26.1638,26.5103)	[17.3552,17.5851)	[43.1021,43.6729)	[16.0197,16.2319)	[3.3299,3.3740)	[1.9271,1.9526)
77	[26.5103,26.8569)	[17.5851,17.8150)	[43.6729,44.2438)	[16.2319,16.4440)	[3.3740,3.4181)	[1.9526,1.9781)
78	[26.8569,27.2034)	[17.8150,18.0448)	[44.2438,44.8147)	[16.4440,16.6562)	[3.4181,3.4622)	[1.9781,2.0037)
79	[27.2034,27.5500)	[18.0448,18.2747)	[44.8147,45.3856)	[16.6562,16.8684)	[3.4622,3.5063)	[2.0037,2.0292)
80	[27.5500,27.8965)	[18.2747,18.5046)	[45.3856,45.9565)	[16.8684,17.0806)	[3.5063,3.5505)	[2.0292,2.0547)
81	[27.8965,28.2430)	[18.5046,18.7344)	[45.9565,46.5274)	[17.0806,17.2928)	[3.5505,3.5946)	[2.0547,2.0802)
82	[28.2430,28.5896)	[18.7344,18.9643)	[46.5274,47.0983)	[17.2928,17.5049)	[3.5946,3.6387)	[2.0802,2.1058)
83	[28.5896,28.9361)	[18.9643,19.1942)	[47.0983,47.6692)	[17.5049,17.7171)	[3.6387,3.6828)	[2.1058,2.1313)
84	[28.9361,29.2827)	[19.1942,19.4241)	[47.6692,48.2400)	[17.7171,17.9293)	[3.6828,3.7269)	[2.1313,2.1568)
85	[29.2827,29.6292)	[19.4241,19.6539)	[48.2400,48.8109)	[17.9293,18.1415)	[3.7269,3.7710)	[2.1568,2.1823)
86	[29.6292,29.9757)	[19.6539,19.8838)	[48.8109,49.3818)	[18.1415,18.3537)	[3.7710,3.8151)	[2.1823,2.2079)
87	[29.9757,30.3223)	[19.8838,20.1137)	[49.3818,49.9527)	[18.3537,18.5659)	[3.8151,3.8592)	[2.2079,2.2334)
88	[30.3223,30.6688)	[20.1137,20.3435)	[49.9527,50.5236)	[18.5659,18.7780)	[3.8592,3.9033)	[2.2334,2.2589)
89	[30.6688,31.0154)	[20.3435,20.5734)	[50.5236,51.0945)	[18.7780,18.9902)	[3.9033,3.9474)	[2.2589,2.2844)
90	[31.0154,31.3619)	[20.5734,20.8033)	[51.0945,51.6654)	[18.9902,19.2024)	[3.9474,3.9915)	[2.2844,2.3100)
91	[31.3619,31.7084)	[20.8033,21.0331)	[51.6654,52.2363)	[19.2024,19.4146)	[3.9915,4.0356)	[2.3100,2.3355)
92	[31.7084,32.0550)	[21.0331,21.2630)	[52.2363,52.8072)	[19.4146,19.6268)	[4.0356,4.0797)	[2.3355,2.3610)
93	[32.0550,32.4015)	[21.2630,21.4929)	[52.8072,53.3780)	[19.6268,19.8389)	[4.0797,4.1238)	[2.3610,2.3865)
94	[32.4015,32.7481)	[21.4929,21.7228)	[53.3780,53.9489)	[19.8389,20.0511)	[4.1238,4.1679)	[2.3865,2.4121)
95	[32.7481,33.0946)	[21.7228,21.9526)	[53.9489,54.5198)	[20.0511,20.2633)	[4.1679,4.2120)	[2.4121,2.4376)
96	[33.0946,33.4411)	[21.9526,22.1825)	[54.5198,55.0907)	[20.2633,20.4755)	[4.2120,4.2561)	[2.4376,2.4631)
97	[33.4411,33.7877)	[22.1825,22.4124)	[55.0907,55.6616)	[20.4755,20.6877)	[4.2561,4.3002)	[2.4631,2.4886)
98	[33.7877,34.1342)	[22.4124,22.6422)	[55.6616,56.2325)	[20.6877,20.8998)	[4.3002,4.3443)	[2.4886,2.5142)
99	[34.1342,34.4808)	[22.6422,22.8721)	[56.2325,56.8034)	[20.8998,21.1120)	[4.3443,4.3884)	[2.5142,2.5397)

100	[34.4808,34.8273)	[22.8721,23.1020)	[56.8034,57.3743)	[21.1120,21.3242)	[4.3884,4.4326)	[2.5397,2.5652)
101	[34.8273,35.1738)	[23.1020,23.3319)	[57.3743,57.9451)	[21.3242,21.5364)	[4.4326,4.4767)	[2.5652,2.5907)
102	[35.1738,35.5204)	[23.3319,23.5617)	[57.9451,58.5160)	[21.5364,21.7486)	[4.4767,4.5208)	[2.5907,2.6162)
103	[35.5204,35.8669)	[23.5617,23.7916)	[58.5160,59.0869)	[21.7486,21.9608)	[4.5208,4.5649)	[2.6162,2.6418)
104	[35.8669,36.2135)	[23.7916,24.0215)	[59.0869,59.6578)	[21.9608,22.1729)	[4.5649,4.6090)	[2.6418,2.6673)
105	[36.2135,36.5600)	[24.0215,24.2513)	[59.6578,60.2287)	[22.1729,22.3851)	[4.6090,4.6531)	[2.6673,2.6928)
106	[36.5600,36.9065)	[24.2513,24.4812)	[60.2287,60.7996)	[22.3851,22.5973)	[4.6531,4.6972)	[2.6928,2.7183)
107	[36.9065,37.2531)	[24.4812,24.7111)	[60.7996,61.3705)	[22.5973,22.8095)	[4.6972,4.7413)	[2.7183,2.7439)
108	[37.2531,37.5996)	[24.7111,24.9409)	[61.3705,61.9414)	[22.8095,23.0217)	[4.7413,4.7854)	[2.7439,2.7694)
109	[37.5996,37.9462)	[24.9409,25.1708)	[61.9414,62.5122)	[23.0217,23.2338)	[4.7854,4.8295)	[2.7694,2.7949)
110	[37.9462,38.2927)	[25.1708,25.4007)	[62.5122,63.0831)	[23.2338,23.4460)	[4.8295,4.8736)	[2.7949,2.8204)
111	[38.2927,38.6392)	[25.4007,25.6306)	[63.0831,63.6540)	[23.4460,23.6582)	[4.8736,4.9177)	[2.8204,2.8460)
112	[38.6392,38.9858)	[25.6306,25.8604)	[63.6540,64.2249)	[23.6582,23.8704)	[4.9177,4.9618)	[2.8460,2.8715)
113	[38.9858,39.3323)	[25.8604,26.0903)	[64.2249,64.7958)	[23.8704,24.0826)	[4.9618,5.0059)	[2.8715,2.8970)
114	[39.3323,39.6789)	[26.0903,26.3202)	[64.7958,65.3667)	[24.0826,24.2947)	[5.0059,5.0500)	[2.8970,2.9225)
115	[39.6789,40.0254)	[26.3202,26.5500)	[65.3667,65.9376)	[24.2947,24.5069)	[5.0500,5.0941)	[2.9225,2.9481)
116	[40.0254,40.3719)	[26.5500,26.7799)	[65.9376,66.5085)	[24.5069,24.7191)	[5.0941,5.1382)	[2.9481,2.9736)
117	[40.3719,40.7185)	[26.7799,27.0098)	[66.5085,67.0794)	[24.7191,24.9313)	[5.1382,5.1823)	[2.9736,2.9991)
118	[40.7185,41.0650)	[27.0098,27.2397)	[67.0794,67.6502)	[24.9313,25.1435)	[5.1823,5.2264)	[2.9991,3.0246)
119	[41.0650,41.4116)	[27.2397,27.4695)	[67.6502,68.2211)	[25.1435,25.3557)	[5.2264,5.2705)	[3.0246,3.0502)
120	[41.4116,41.7581)	[27.4695,27.6994)	[68.2211,68.7920)	[25.3557,25.5678)	[5.2705,5.3147)	[3.0502,3.0757)
121	[41.7581,42.1046)	[27.6994,27.9293)	[68.7920,69.3629)	[25.5678,25.7800)	[5.3147,5.3588)	[3.0757,3.1012)
122	[42.1046,42.4512)	[27.9293,28.1591)	[69.3629,69.9338)	[25.7800,25.9922)	[5.3588,5.4029)	[3.1012,3.1267)
123	[42.4512,42.7977)	[28.1591,28.3890)	[69.9338,70.5047)	[25.9922,26.2044)	[5.4029,5.4470)	[3.1267,3.1523)
124	[42.7977,43.1443)	[28.3890,28.6189)	[70.5047,71.0756)	[26.2044,26.4166)	[5.4470,5.4911)	[3.1523,3.1778)
125	[43.1443,43.4908)	[28.6189,28.8487)	[71.0756,71.6465)	[26.4166,26.6287)	[5.4911,5.5352)	[3.1778,3.2033)
126	[43.4908,43.8373)	[28.8487,29.0786)	[71.6465,72.2173)	[26.6287,26.8409)	[5.5352,5.5793)	[3.2033,3.2288)
127	[43.8373,44.0106]	[29.0786,29.1935]	[72.2173,72.5028]	[26.8409,26.9470]	[5.5793,5.6013]	[3.2288,3.2416]