|  |  |  |  |
| --- | --- | --- | --- |
|  | MSE | MS-SSIM | 备注 |
| 10 | 178.550 | 0.964 | 去掉特征复用 |
| 10\_1  2组编解码器的编码器输出均为  64\*16\*16 | 310.334 | 0.957 | *# 编解码器1 输入trainData 输出decData1 逼近trainData*  *# 编解码器2 输入decData1*  *输出decData2 逼近trainData - decData1* |
| 10\_2 | 200.783 | 0.964 | 3组编解码器 |
| 10\_3 | 19694.562 | 0.588 | 1组编解码器  1个残差修复器 |
| 10\_4 | 123.890 | 0.982 | 与10\_1相同  但编解码器2尺寸128\*32\*32 |
| 10\_5 | 98.790 | 0.985 | 同上  编解码2尺寸  128\*64\*64 |
| 10\_6 | 113.900 | 0.981 | 同10\_5  但增加了编解码器2中的卷积层 |
| 10\_7 | 189.634 | 0.941 | 与10\_1相同  但编解码器1的输出为64\*8\*8 |
| 10\_8 | 185.774 | 0.967 | 与10\_1相同  但编解码器1的输出为32\*16\*16 |
| 10\_9 | 283.849 | 0.936 | 与10\_1相同  但2组编码器都输出64\*8\*8 |
| 10\_10 | 152.669 | 0.988 | 与10\_1相同  编解码器1仍然是64\*16\*16  编解码器2是64\*8\*8 |
| 10\_11 | 365.559 | 0.905 | 与10\_1相同  编解码器1是64\*8\*8  编解码器2是128\*64\*64 |
| 10\_12 | 效果很差 | 效果很差 | 与10\_11相同  但10\_11是分别训练编解码1和2  10\_12是一起训练 |

默认的尺寸64\*16\*16

11.py尝试了分层训练，但发现效果不如各层一起训练

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | MSE | MS-SSIM | 尺寸 | 备注 |
| 12\_0 | 161.751 | 0.980 | 64\*16\*16 | 对照样本 |
| 12 | 343.998 | 0.894 | 64\*16\*16 | 每个降维后加入1\*1卷积 |
| 12\_1 | 388.398 | 0.921 | 64\*16\*16 | 每个降维前和后都加入1\*1卷积 |
| 12\_2 | 1178.415 | 0.484 | 64\*16\*16 | 在12基础上改为3\*3卷积 |
| 12\_3 | 396.100 | 0.905 | 32\*16\*16 | 增加了卷积通道先升再降 |
| 12\_4 | 816.906 | 0.682 | 64\*16\*16\*0.5 | 编解码器都加入dropout层 |
| 12\_5 | 385.528 | 0.852 | 64\*16\*16\*0.5 | 只有编码器加入了dropout |
| 12\_6 | 455.073 | 0.850 | 64\*16\*16\*0.5 | 只在编码器的最后一层加入dropout |
| 12\_7 | 214.862 | 0.957 | 32\*16\*16 | 与12\_0相同 |
| 13 | 179.365 | 0.979 | 64\*16\*16 | 在12\_0基础上，数据归一化到[0,1]  但编码器的输出不进行归一化  有量化  归一化后发现收敛很快 |
| 13\_1 |  |  |  | 增加编码器输出与近似零向量的KL距离（稀疏项）损失 |
| 13\_2 | 160.674 | 0.932 | 64\*16\*16 | 基于13，但编码器输出从[0,1]乘7而不是255 |
| 13\_3 | 196.568 | 0.981 | 64\*16\*16 | 去掉量化，同时将编码器输出归一化 |
| 13\_4 | 107.535 | 0.981 | 64\*16\*16 | 12\_0的编码器输出\*255  而13没有这一操作  13的结果比12差很多  增加一个可以学习的参数k  编码器的输出\*k |
| 13\_5 | 197.277 | 0.976 | 32\*16\*16 | 与13\_3相同 |
| 13\_6 | 257.777 | 0.924 | 64\*8\*8 | 与13\_3相同 |
| 13\_7 | 375.910 | 0.885 | 64\*8\*8 | 与13\_6相同但使用MCE损失 |
| 14\_1\_1 | 23.176 | 0.978 | 64\*8\*8 | 操作对象是均值滤波后的图像  加入了8级量化 |
| 14\_1\_2 |  | 0.883 |  | 修复14\_1\_1的高频部分 |
| 14\_1\_3 | 5.137 | 0.995 | 64\*8\*8 | 与14\_1\_1相同，去掉了编码器输出\*255 |
| 15\_1 | 240.505 | 0.965 | 32\*16\*16 | 基于13\_5  13\_5本身没有量化  编码器输出归一化到[0,1]  15\_1将量化等级设置为16 |
| 15\_2 | 244.273 | 0.955 | 32\*16\*16 | 量化等级8 |
| 15\_3 | 223.410 | 0.974 | 32\*16\*16 | 量化等级128 |
| 16 | 223.724 | 0.968 | 32\*16\*16 | 在15\_1基础上  编码器第一层加入5\*5步长1的卷积  块效应的问题在此得到解决  量化等级是16 |
| 16\_1 | 192.114 | 0.970 | 32\*16\*16 | 第一层并行1\*1和5\*5的卷积层 |
| 16\_2 |  | 0.968 | 32\*16\*16 | 第一层并行1\*1 3\*3 5\*5 7\*7 9\*9的卷积核 |
| 16\_3 |  | 0.930 |  | 在第一层加入深层3\*3网络 |
| 17 |  | 0.789 | 32\*8\*8 | 基于16  量化等级16 |
| 17\_1 |  | 0.840 | 64\*8\*8 | 量化等级16 |
| 17\_2 |  | 0.873 | 16\*16\*16 | 量化等级16 |
| 17\_3 |  | 0.929 | 16\*16\*16 | 逐步降低通道数为16  量化等级16  17\_3的效果好于17\_2 |
| 17\_4 |  | 0.795 | 16\*16\*16 | 类似17\_3  增加了更多卷积层来缓慢降低通道数 |
| 18 |  | 0.900 | 256\*16\*16 | 编码器直接输出二进制值，也就是量化等级为2  为了防止数值0反向传播出问题，把量化结果+1，也就是01变成了12 |
| 19 |  |  |  | 同16  但量化等级为1024 |

直接对13\_5的输出进行不同等级的量化，结果如下（量化后再训练 MS-SSIM也会增加）

2 0.2122164717875421

3 0.3347523109987378

4 0.4240416716784239

5 0.44770978949964046

6 0.514722540974617

7 0.5834479331970215

8 0.6444179303944111

9 0.6504920478910208

10 0.6914476007223129

11 0.700916800647974

12 0.7171695604920387

13 0.7618770785629749

14 0.7926715947687626

15 0.790420338511467

16 0.8142396211624146

17 0.8176802173256874

18 0.8254399783909321

19 0.8362446092069149

20 0.857216227799654

21 0.8647113256156445

22 0.868688628077507

23 0.8798284716904163

24 0.8728291317820549

25 0.8962120600044727

26 0.8880509473383427

27 0.8972621038556099

28 0.8963146470487118

29 0.9063464403152466

30 0.9059347249567509

31 0.9085631854832172

32 0.9107104502618313

33 0.9271579757332802

34 0.9159930124878883

35 0.9260938316583633

36 0.9266144745051861

37 0.9282824881374836

38 0.939138475805521

39 0.9391078613698483

40 0.9316939748823643

41 0.938141155987978

42 0.9387544766068459

43 0.9432066343724728

44 0.9452606067061424

45 0.9468578547239304

46 0.9477444067597389

47 0.9436961337924004

48 0.9511394575238228

49 0.9417685642838478

50 0.9544493854045868

51 0.9478612951934338

52 0.9530409947037697

53 0.9512216486036777

54 0.942072682082653

55 0.9507274851202965

56 0.9515467844903469

57 0.9545086026191711

58 0.9596833400428295

59 0.9504790045320988

60 0.9563823007047176

61 0.9489657580852509

62 0.9591906145215034

63 0.9581166356801987

64 0.9576096087694168

65 0.9619701467454433

66 0.9603796973824501

67 0.9611040912568569

68 0.95696771889925

69 0.95707568526268

70 0.9570441320538521

71 0.9597596973180771

72 0.9548206552863121

73 0.9532208852469921

74 0.9646593071520329

75 0.9614358469843864

76 0.9618994519114494

77 0.9622093290090561

78 0.9534751400351524

79 0.9539902023971081

80 0.9646591879427433

81 0.9613641388714314

82 0.9603927806019783

83 0.9630109295248985

84 0.9622479826211929

85 0.9568761102855206

86 0.9576408304274082

87 0.9631662257015705

88 0.9684610553085804

89 0.9631713069975376

90 0.9649619087576866

91 0.9651700519025326

92 0.966075461357832

93 0.9588764384388924

94 0.9714235626161098

95 0.9639593213796616

96 0.9665895700454712

97 0.9706461913883686

98 0.9693386591970921

99 0.967614933848381

100 0.9674573242664337

101 0.967912781983614

102 0.9651571549475193

103 0.9608196094632149

104 0.9715779647231102

105 0.9664700292050838

106 0.9663199670612812

107 0.9580566473305225

108 0.9678641185164452

109 0.9706529341638088

110 0.9624094069004059

111 0.9702966473996639

112 0.9593665935099125

113 0.9658819250762463

114 0.9682883583009243

115 0.9645749405026436

116 0.9733687490224838

117 0.960750512778759

118 0.9553022123873234

119 0.9684527404606342

120 0.9749170020222664

121 0.9677282832562923

122 0.9681240133941174

123 0.9734884463250637

124 0.9665904566645622

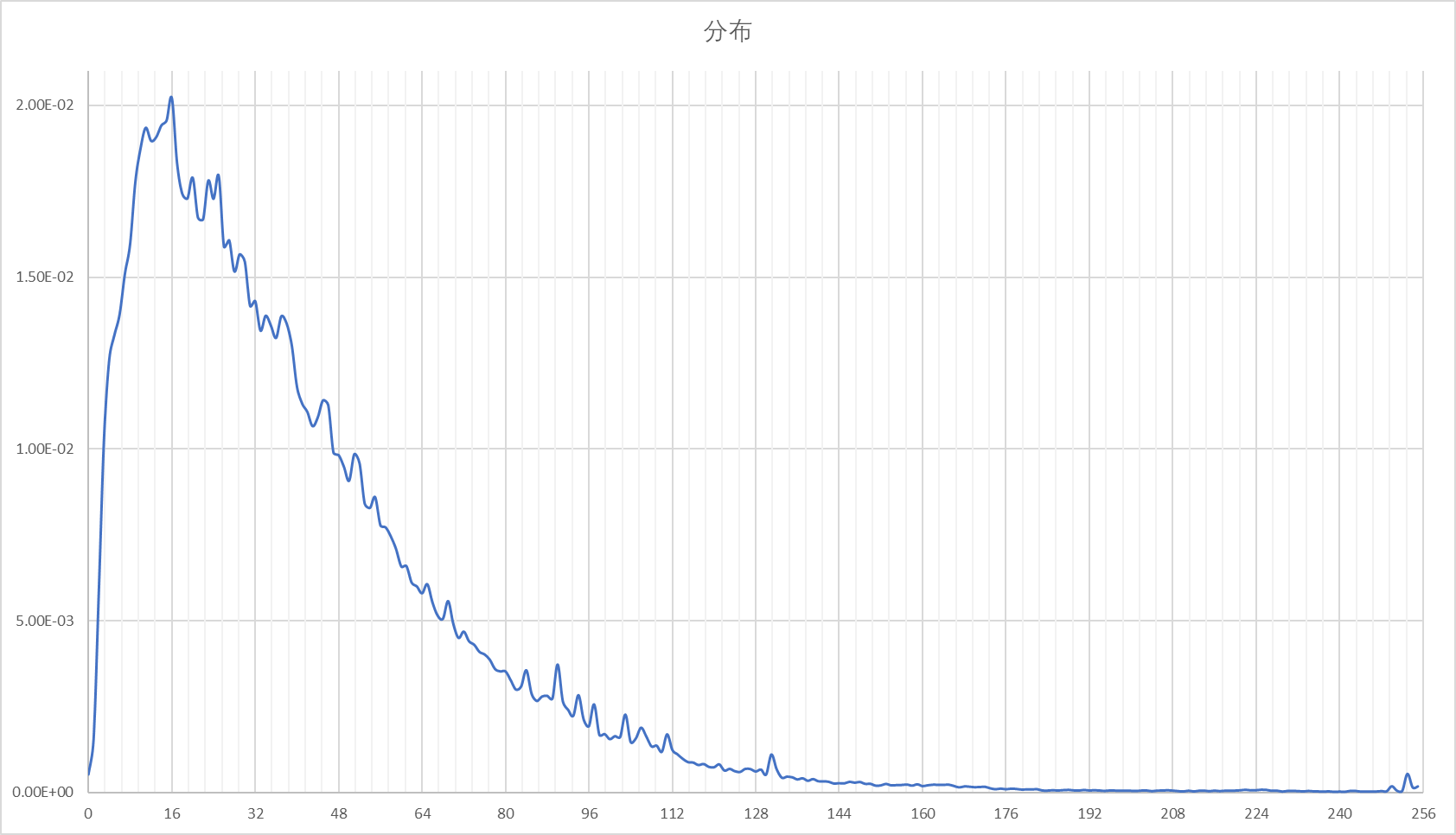
125 0.9683772325515747

126 0.9655206426978111

127 0.9709970578551292

128 0.9709646627306938

256级量化后，数据的分布如下



去除不同通道后 按重要性排序如下

第一次实验

(30, 0.5095305573195219)

(12, 0.623749203979969)

(15, 0.6933158934116364)

(4, 0.6937920525670052)

(14, 0.7050833255052567)

(28, 0.715040247887373)

(24, 0.725745927542448)

(29, 0.7499634735286236)

(13, 0.756662167608738)

(20, 0.763075016438961)

(26, 0.790147203952074)

(10, 0.7995071150362492)

(16, 0.8043824955821037)

(27, 0.8090166561305523)

(11, 0.8178820796310902)

(8, 0.8273916691541672)

(7, 0.8292747251689434)

(19, 0.8331594094634056)

(22, 0.843607559800148)

(23, 0.8439974412322044)

(6, 0.8515211828052998)

(0, 0.853408046066761)

(25, 0.857738483697176)

(5, 0.8614766150712967)

(21, 0.8824301771819592)

(18, 0.8864684961736202)

(9, 0.8949568942189217)

(31, 0.8952393308281898)

(2, 0.9100119508802891)

(3, 0.9136431701481342)

(1, 0.9266495443880558)

(17, 0.9425630122423172)

(-1, 0.9731135070323944)

第二次实验

(30, 0.4722857754677534)

(12, 0.6358377747237682)

(15, 0.672402199357748)

(4, 0.6799484305083752)

(28, 0.6984723880887032)

(14, 0.7075389847159386)

(24, 0.7170288637280464)

(29, 0.7539073079824448)

(13, 0.7564674541354179)

(20, 0.7603367604315281)

(16, 0.7980533614754677)

(26, 0.7989716418087482)

(10, 0.8126741386950016)

(27, 0.8183471709489822)

(11, 0.81957021728158)

(8, 0.827064260840416)

(7, 0.8313435353338718)

(19, 0.8354809172451496)

(22, 0.8462581597268581)

(0, 0.8463009111583233)

(23, 0.8500323779881)

(6, 0.8533016741275787)

(5, 0.8609485141932964)

(25, 0.8616165928542614)

(21, 0.8842720054090023)

(9, 0.8909971937537193)

(18, 0.8916000165045261)

(31, 0.8934141062200069)

(2, 0.9063855670392513)

(3, 0.918744970113039)

(1, 0.9263919591903687)

(17, 0.9417113065719604)

(-1, 0.9732156619429588)