**YOLOv3-SPP module\_list**

0 Sequential(

(Conv2d): Conv2d(3, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(32, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

1 Sequential(

(Conv2d): Conv2d(32, 64, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

2 Sequential(

(Conv2d): Conv2d(64, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(32, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

3 Sequential(

(Conv2d): Conv2d(32, 64, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

4 weightedFeatureFusion()

5 Sequential(

(Conv2d): Conv2d(64, 128, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

6 Sequential(

(Conv2d): Conv2d(128, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

7 Sequential(

(Conv2d): Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

8 weightedFeatureFusion()

9 Sequential(

(Conv2d): Conv2d(128, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

10 Sequential(

(Conv2d): Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

11 weightedFeatureFusion()

12 Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

13 Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

14 Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

15 weightedFeatureFusion()

16 Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

17 Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

18 weightedFeatureFusion()

19 Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

20 Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

21 weightedFeatureFusion()

22 Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

23 Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

24 weightedFeatureFusion()

25 Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

26 Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

27 weightedFeatureFusion()

28 Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

29 Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

30 weightedFeatureFusion()

31 Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

32 Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

33 weightedFeatureFusion()

34 Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

35 Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

36 weightedFeatureFusion()

37 Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

38 Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

39 Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

40 weightedFeatureFusion()

41 Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

42 Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

43 weightedFeatureFusion()

44 Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

45 Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

46 weightedFeatureFusion()

47 Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

48 Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

49 weightedFeatureFusion()

50 Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

51 Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

52 weightedFeatureFusion()

53 Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

54 Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

55 weightedFeatureFusion()

56 Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

57 Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

58 weightedFeatureFusion()

59 Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

60 Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

61 weightedFeatureFusion()

62 Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

63 Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

64 Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

65 weightedFeatureFusion()

66 Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

67 Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

68 weightedFeatureFusion()

69 Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

70 Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

71 weightedFeatureFusion()

72 Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

73 Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

74 weightedFeatureFusion()

75 Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

76 Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

77 Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

78 MaxPool2d(kernel\_size=5, stride=1, padding=2, dilation=1, ceil\_mode=False)

79 Sequential()

80 MaxPool2d(kernel\_size=9, stride=1, padding=4, dilation=1, ceil\_mode=False)

81 Sequential()

82 MaxPool2d(kernel\_size=13, stride=1, padding=6, dilation=1, ceil\_mode=False)

83 Sequential()

84 Sequential(

(Conv2d): Conv2d(2048, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

85 Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

86 Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

87 Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

88 Sequential(

(Conv2d): Conv2d(1024, 255, kernel\_size=(1, 1), stride=(1, 1))

)

89 YOLOLayer()

90 Sequential()

91 Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

92 Upsample(scale\_factor=2.0, mode=nearest)

93 Sequential()

94 Sequential(

(Conv2d): Conv2d(768, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

95 Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

96 Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

97 Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

98 Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

99 Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

100 Sequential(

(Conv2d): Conv2d(512, 255, kernel\_size=(1, 1), stride=(1, 1))

)

101 YOLOLayer()

102 Sequential()

103 Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

104 Upsample(scale\_factor=2.0, mode=nearest)

105 Sequential()

106 Sequential(

(Conv2d): Conv2d(384, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

107 Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

108 Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

109 Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

110 Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

111 Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

112 Sequential(

(Conv2d): Conv2d(256, 255, kernel\_size=(1, 1), stride=(1, 1))

)

113 YOLOLayer()

**YOLOv3-SPP module\_defs**

0 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 32, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

1 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 64, 'size': 3, 'stride': 2, 'pad': 1, 'activation': 'leaky'}

2 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 32, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

3 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 64, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

4 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

5 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 3, 'stride': 2, 'pad': 1, 'activation': 'leaky'}

6 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 64, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

7 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

8 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

9 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 64, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

10 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

11 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

12 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 3, 'stride': 2, 'pad': 1, 'activation': 'leaky'}

13 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

14 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

15 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

16 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

17 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

18 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

19 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

20 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

21 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

22 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

23 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

24 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

25 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

26 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

27 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

28 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

29 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

30 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

31 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

32 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

33 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

34 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

35 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

36 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

37 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 3, 'stride': 2, 'pad': 1, 'activation': 'leaky'}

38 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

39 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

40 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

41 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

42 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

43 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

44 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

45 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

46 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

47 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

48 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

49 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

50 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

51 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

52 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

53 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

54 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

55 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

56 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

57 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

58 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

59 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

60 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

61 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

62 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 1024, 'size': 3, 'stride': 2, 'pad': 1, 'activation': 'leaky'}

63 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

64 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 1024, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

65 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

66 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

67 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 1024, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

68 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

69 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

70 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 1024, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

71 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

72 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

73 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 1024, 'size': 3, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

74 {'type': 'shortcut', 'from': [-3], 'activation': 'linear'}

75 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

76 {'type': 'convolutional', 'batch\_normalize': 1, 'size': 3, 'stride': 1, 'pad': 1, 'filters': 1024, 'activation': 'leaky'}

77 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

78 {'type': 'maxpool', 'stride': 1, 'size': 5}

79 {'type': 'route', 'layers': [-2]}

80 {'type': 'maxpool', 'stride': 1, 'size': 9}

81 {'type': 'route', 'layers': [-4]}

82 {'type': 'maxpool', 'stride': 1, 'size': 13}

83 {'type': 'route', 'layers': [-1, -3, -5, -6]}

84 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

85 {'type': 'convolutional', 'batch\_normalize': 1, 'size': 3, 'stride': 1, 'pad': 1, 'filters': 1024, 'activation': 'leaky'}

86 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 512, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

87 {'type': 'convolutional', 'batch\_normalize': 1, 'size': 3, 'stride': 1, 'pad': 1, 'filters': 1024, 'activation': 'leaky'}

88 {'type': 'convolutional', 'batch\_normalize': 0, 'size': 1, 'stride': 1, 'pad': 1, 'filters': 255, 'activation': 'linear'}

89 {'type': 'yolo', 'mask': [6, 7, 8], 'anchors': array([[ 10, 13],

[ 16, 30],

[ 33, 23],

[ 30, 61],

[ 62, 45],

[ 59, 119],

[116, 90],

[156, 198],

[373, 326]]), 'classes': 80, 'num': 9, 'jitter': '.3', 'ignore\_thresh': '.7', 'truth\_thresh': 1, 'random': 1}

90 {'type': 'route', 'layers': [-4]}

91 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

92 {'type': 'upsample', 'stride': 2}

93 {'type': 'route', 'layers': [-1, 61]}

94 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

95 {'type': 'convolutional', 'batch\_normalize': 1, 'size': 3, 'stride': 1, 'pad': 1, 'filters': 512, 'activation': 'leaky'}

96 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

97 {'type': 'convolutional', 'batch\_normalize': 1, 'size': 3, 'stride': 1, 'pad': 1, 'filters': 512, 'activation': 'leaky'}

98 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 256, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

99 {'type': 'convolutional', 'batch\_normalize': 1, 'size': 3, 'stride': 1, 'pad': 1, 'filters': 512, 'activation': 'leaky'}

100 {'type': 'convolutional', 'batch\_normalize': 0, 'size': 1, 'stride': 1, 'pad': 1, 'filters': 255, 'activation': 'linear'}

101 {'type': 'yolo', 'mask': [3, 4, 5], 'anchors': array([[ 10, 13],

[ 16, 30],

[ 33, 23],

[ 30, 61],

[ 62, 45],

[ 59, 119],

[116, 90],

[156, 198],

[373, 326]]), 'classes': 80, 'num': 9, 'jitter': '.3', 'ignore\_thresh': '.7', 'truth\_thresh': 1, 'random': 1}

102 {'type': 'route', 'layers': [-4]}

103 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

104 {'type': 'upsample', 'stride': 2}

105 {'type': 'route', 'layers': [-1, 36]}

106 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

107 {'type': 'convolutional', 'batch\_normalize': 1, 'size': 3, 'stride': 1, 'pad': 1, 'filters': 256, 'activation': 'leaky'}

108 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

109 {'type': 'convolutional', 'batch\_normalize': 1, 'size': 3, 'stride': 1, 'pad': 1, 'filters': 256, 'activation': 'leaky'}

110 {'type': 'convolutional', 'batch\_normalize': 1, 'filters': 128, 'size': 1, 'stride': 1, 'pad': 1, 'activation': 'leaky'}

111 {'type': 'convolutional', 'batch\_normalize': 1, 'size': 3, 'stride': 1, 'pad': 1, 'filters': 256, 'activation': 'leaky'}

112 {'type': 'convolutional', 'batch\_normalize': 0, 'size': 1, 'stride': 1, 'pad': 1, 'filters': 255, 'activation': 'linear'}

113 {'type': 'yolo', 'mask': [0, 1, 2], 'anchors': array([[ 10, 13],

[ 16, 30],

[ 33, 23],

[ 30, 61],

[ 62, 45],

[ 59, 119],

[116, 90],

[156, 198],

[373, 326]]), 'classes': 80, 'num': 9, 'jitter': '.3', 'ignore\_thresh': '.7', 'truth\_thresh': 1, 'random': 1}

**YOLOv3-SPP named\_modules()**

**0** ('', Darknet(

(module\_list): ModuleList(

(0): Sequential(

(Conv2d): Conv2d(3, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(32, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(1): Sequential(

(Conv2d): Conv2d(32, 64, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(2): Sequential(

(Conv2d): Conv2d(64, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(32, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(3): Sequential(

(Conv2d): Conv2d(32, 64, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(4): weightedFeatureFusion()

(5): Sequential(

(Conv2d): Conv2d(64, 128, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(6): Sequential(

(Conv2d): Conv2d(128, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(7): Sequential(

(Conv2d): Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(8): weightedFeatureFusion()

(9): Sequential(

(Conv2d): Conv2d(128, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(10): Sequential(

(Conv2d): Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(11): weightedFeatureFusion()

(12): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(13): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(14): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(15): weightedFeatureFusion()

(16): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(17): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(18): weightedFeatureFusion()

(19): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(20): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(21): weightedFeatureFusion()

(22): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(23): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(24): weightedFeatureFusion()

(25): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(26): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(27): weightedFeatureFusion()

(28): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(29): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(30): weightedFeatureFusion()

(31): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(32): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(33): weightedFeatureFusion()

(34): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(35): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(36): weightedFeatureFusion()

(37): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(38): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(39): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(40): weightedFeatureFusion()

(41): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(42): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(43): weightedFeatureFusion()

(44): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(45): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(46): weightedFeatureFusion()

(47): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(48): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(49): weightedFeatureFusion()

(50): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(51): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(52): weightedFeatureFusion()

(53): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(54): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(55): weightedFeatureFusion()

(56): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(57): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(58): weightedFeatureFusion()

(59): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(60): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(61): weightedFeatureFusion()

(62): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(63): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(64): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(65): weightedFeatureFusion()

(66): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(67): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(68): weightedFeatureFusion()

(69): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(70): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(71): weightedFeatureFusion()

(72): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(73): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(74): weightedFeatureFusion()

(75): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(76): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(77): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(78): MaxPool2d(kernel\_size=5, stride=1, padding=2, dilation=1, ceil\_mode=False)

(79): Sequential()

(80): MaxPool2d(kernel\_size=9, stride=1, padding=4, dilation=1, ceil\_mode=False)

(81): Sequential()

(82): MaxPool2d(kernel\_size=13, stride=1, padding=6, dilation=1, ceil\_mode=False)

(83): Sequential()

(84): Sequential(

(Conv2d): Conv2d(2048, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(85): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(86): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(87): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(88): Sequential(

(Conv2d): Conv2d(1024, 255, kernel\_size=(1, 1), stride=(1, 1))

)

(89): YOLOLayer()

(90): Sequential()

(91): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(92): Upsample(scale\_factor=2.0, mode=nearest)

(93): Sequential()

(94): Sequential(

(Conv2d): Conv2d(768, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(95): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(96): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(97): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(98): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(99): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(100): Sequential(

(Conv2d): Conv2d(512, 255, kernel\_size=(1, 1), stride=(1, 1))

)

(101): YOLOLayer()

(102): Sequential()

(103): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(104): Upsample(scale\_factor=2.0, mode=nearest)

(105): Sequential()

(106): Sequential(

(Conv2d): Conv2d(384, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(107): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(108): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(109): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(110): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(111): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(112): Sequential(

(Conv2d): Conv2d(256, 255, kernel\_size=(1, 1), stride=(1, 1))

)

(113): YOLOLayer()

)

))

**1** ('module\_list', ModuleList(

(0): Sequential(

(Conv2d): Conv2d(3, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(32, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(1): Sequential(

(Conv2d): Conv2d(32, 64, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(2): Sequential(

(Conv2d): Conv2d(64, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(32, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(3): Sequential(

(Conv2d): Conv2d(32, 64, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(4): weightedFeatureFusion()

(5): Sequential(

(Conv2d): Conv2d(64, 128, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(6): Sequential(

(Conv2d): Conv2d(128, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(7): Sequential(

(Conv2d): Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(8): weightedFeatureFusion()

(9): Sequential(

(Conv2d): Conv2d(128, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(10): Sequential(

(Conv2d): Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(11): weightedFeatureFusion()

(12): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(13): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(14): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(15): weightedFeatureFusion()

(16): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(17): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(18): weightedFeatureFusion()

(19): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(20): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(21): weightedFeatureFusion()

(22): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(23): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(24): weightedFeatureFusion()

(25): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(26): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(27): weightedFeatureFusion()

(28): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(29): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(30): weightedFeatureFusion()

(31): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(32): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(33): weightedFeatureFusion()

(34): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(35): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(36): weightedFeatureFusion()

(37): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(38): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(39): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(40): weightedFeatureFusion()

(41): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(42): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(43): weightedFeatureFusion()

(44): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(45): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(46): weightedFeatureFusion()

(47): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(48): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(49): weightedFeatureFusion()

(50): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(51): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(52): weightedFeatureFusion()

(53): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(54): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(55): weightedFeatureFusion()

(56): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(57): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(58): weightedFeatureFusion()

(59): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(60): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(61): weightedFeatureFusion()

(62): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(63): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(64): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(65): weightedFeatureFusion()

(66): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(67): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(68): weightedFeatureFusion()

(69): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(70): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(71): weightedFeatureFusion()

(72): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(73): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(74): weightedFeatureFusion()

(75): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(76): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(77): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(78): MaxPool2d(kernel\_size=5, stride=1, padding=2, dilation=1, ceil\_mode=False)

(79): Sequential()

(80): MaxPool2d(kernel\_size=9, stride=1, padding=4, dilation=1, ceil\_mode=False)

(81): Sequential()

(82): MaxPool2d(kernel\_size=13, stride=1, padding=6, dilation=1, ceil\_mode=False)

(83): Sequential()

(84): Sequential(

(Conv2d): Conv2d(2048, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(85): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(86): Sequential(

(Conv2d): Conv2d(1024, 512, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(87): Sequential(

(Conv2d): Conv2d(512, 1024, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(1024, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(88): Sequential(

(Conv2d): Conv2d(1024, 255, kernel\_size=(1, 1), stride=(1, 1))

)

(89): YOLOLayer()

(90): Sequential()

(91): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(92): Upsample(scale\_factor=2.0, mode=nearest)

(93): Sequential()

(94): Sequential(

(Conv2d): Conv2d(768, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(95): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(96): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(97): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(98): Sequential(

(Conv2d): Conv2d(512, 256, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(99): Sequential(

(Conv2d): Conv2d(256, 512, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(512, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(100): Sequential(

(Conv2d): Conv2d(512, 255, kernel\_size=(1, 1), stride=(1, 1))

)

(101): YOLOLayer()

(102): Sequential()

(103): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(104): Upsample(scale\_factor=2.0, mode=nearest)

(105): Sequential()

(106): Sequential(

(Conv2d): Conv2d(384, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(107): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(108): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(109): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(110): Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(111): Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

)

(112): Sequential(

(Conv2d): Conv2d(256, 255, kernel\_size=(1, 1), stride=(1, 1))

)

(113): YOLOLayer()

))

**2** ('module\_list.0', Sequential(

(Conv2d): Conv2d(3, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(32, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

**3** ('module\_list.0.Conv2d', Conv2d(3, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False))

**4** ('module\_list.0.BatchNorm2d', BatchNorm2d(32, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

**5** ('module\_list.0.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

**6** ('module\_list.1', Sequential(

(Conv2d): Conv2d(32, 64, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

**7** ('module\_list.1.Conv2d', Conv2d(32, 64, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False))

**8** ('module\_list.1.BatchNorm2d', BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

**9** ('module\_list.1.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

**10** ('module\_list.2', Sequential(

(Conv2d): Conv2d(64, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(32, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

11 ('module\_list.2.Conv2d', Conv2d(64, 32, kernel\_size=(1, 1), stride=(1, 1), bias=False))

12 ('module\_list.2.BatchNorm2d', BatchNorm2d(32, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

13 ('module\_list.2.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

14 ('module\_list.3', Sequential(

(Conv2d): Conv2d(32, 64, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

15 ('module\_list.3.Conv2d', Conv2d(32, 64, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False))

16 ('module\_list.3.BatchNorm2d', BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

17 ('module\_list.3.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

18 ('module\_list.4', weightedFeatureFusion())

19 ('module\_list.5', Sequential(

(Conv2d): Conv2d(64, 128, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

20 ('module\_list.5.Conv2d', Conv2d(64, 128, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False))

21 ('module\_list.5.BatchNorm2d', BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

22 ('module\_list.5.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

23 ('module\_list.6', Sequential(

(Conv2d): Conv2d(128, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

24 ('module\_list.6.Conv2d', Conv2d(128, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False))

25 ('module\_list.6.BatchNorm2d', BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

26 ('module\_list.6.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

27 ('module\_list.7', Sequential(

(Conv2d): Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

28 ('module\_list.7.Conv2d', Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False))

29 ('module\_list.7.BatchNorm2d', BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

30 ('module\_list.7.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

31 ('module\_list.8', weightedFeatureFusion())

32 ('module\_list.9', Sequential(

(Conv2d): Conv2d(128, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

33 ('module\_list.9.Conv2d', Conv2d(128, 64, kernel\_size=(1, 1), stride=(1, 1), bias=False))

34 ('module\_list.9.BatchNorm2d', BatchNorm2d(64, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

35 ('module\_list.9.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

36 ('module\_list.10', Sequential(

(Conv2d): Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

37 ('module\_list.10.Conv2d', Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False))

38 ('module\_list.10.BatchNorm2d', BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

39 ('module\_list.10.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

40 ('module\_list.11', weightedFeatureFusion())

41 ('module\_list.12', Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

42 ('module\_list.12.Conv2d', Conv2d(128, 256, kernel\_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False))

43 ('module\_list.12.BatchNorm2d', BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

44 ('module\_list.12.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

45 ('module\_list.13', Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

46 ('module\_list.13.Conv2d', Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False))

47 ('module\_list.13.BatchNorm2d', BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

48 ('module\_list.13.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

49 ('module\_list.14', Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

50 ('module\_list.14.Conv2d', Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False))

51 ('module\_list.14.BatchNorm2d', BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

52 ('module\_list.14.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

53 ('module\_list.15', weightedFeatureFusion())

54 ('module\_list.16', Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

55 ('module\_list.16.Conv2d', Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False))

56 ('module\_list.16.BatchNorm2d', BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

57 ('module\_list.16.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

58 ('module\_list.17', Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

59 ('module\_list.17.Conv2d', Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False))

60 ('module\_list.17.BatchNorm2d', BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

61 ('module\_list.17.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

62 ('module\_list.18', weightedFeatureFusion())

63 ('module\_list.19', Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

64 ('module\_list.19.Conv2d', Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False))

65 ('module\_list.19.BatchNorm2d', BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

66 ('module\_list.19.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

67 ('module\_list.20', Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

68 ('module\_list.20.Conv2d', Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False))

69 ('module\_list.20.BatchNorm2d', BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

70 ('module\_list.20.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

71 ('module\_list.21', weightedFeatureFusion())

72 ('module\_list.22', Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

73 ('module\_list.22.Conv2d', Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False))

74 ('module\_list.22.BatchNorm2d', BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

75 ('module\_list.22.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

76 ('module\_list.23', Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

77 ('module\_list.23.Conv2d', Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False))

78 ('module\_list.23.BatchNorm2d', BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

79 ('module\_list.23.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

80 ('module\_list.24', weightedFeatureFusion())

81 ('module\_list.25', Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

82 ('module\_list.25.Conv2d', Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False))

83 ('module\_list.25.BatchNorm2d', BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

84 ('module\_list.25.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

85 ('module\_list.26', Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

86 ('module\_list.26.Conv2d', Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False))

87 ('module\_list.26.BatchNorm2d', BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

88 ('module\_list.26.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

89 ('module\_list.27', weightedFeatureFusion())

90 ('module\_list.28', Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

91 ('module\_list.28.Conv2d', Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False))

92 ('module\_list.28.BatchNorm2d', BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

93 ('module\_list.28.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

94 ('module\_list.29', Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

95 ('module\_list.29.Conv2d', Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False))

96 ('module\_list.29.BatchNorm2d', BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

97 ('module\_list.29.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

98 ('module\_list.30', weightedFeatureFusion())

99 ('module\_list.31', Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

100 ('module\_list.31.Conv2d', Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False))

101 ('module\_list.31.BatchNorm2d', BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

102 ('module\_list.31.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

103 ('module\_list.32', Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

104 ('module\_list.32.Conv2d', Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False))

105 ('module\_list.32.BatchNorm2d', BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

106 ('module\_list.32.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

107 ('module\_list.33', weightedFeatureFusion())

108 ('module\_list.34', Sequential(

(Conv2d): Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

109 ('module\_list.34.Conv2d', Conv2d(256, 128, kernel\_size=(1, 1), stride=(1, 1), bias=False))

110 ('module\_list.34.BatchNorm2d', BatchNorm2d(128, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True))

111 ('module\_list.34.activation', LeakyReLU(negative\_slope=0.1, inplace=True))

112 ('module\_list.35', Sequential(

(Conv2d): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

(BatchNorm2d): BatchNorm2d(256, eps=0.0001, momentum=0.003, affine=True, track\_running\_stats=True)

(activation): LeakyReLU(negative\_slope=0.1, inplace=True)

))

**113** ('module\_list.35.Conv2d', Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False))