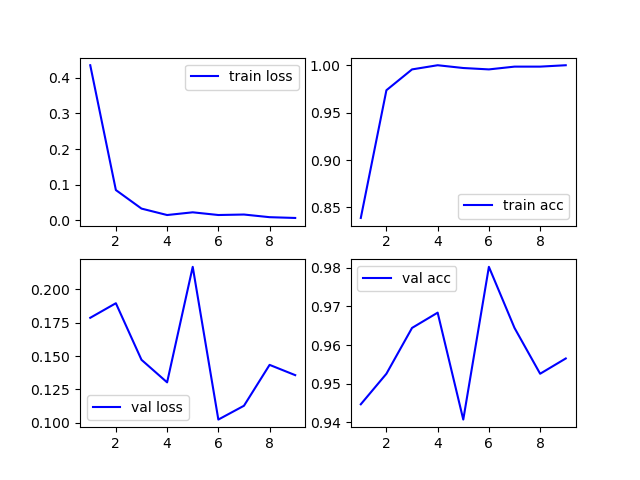
**1.0**

LeftDown

[334 349] train

[124 129] val

[116 122] test

参数：

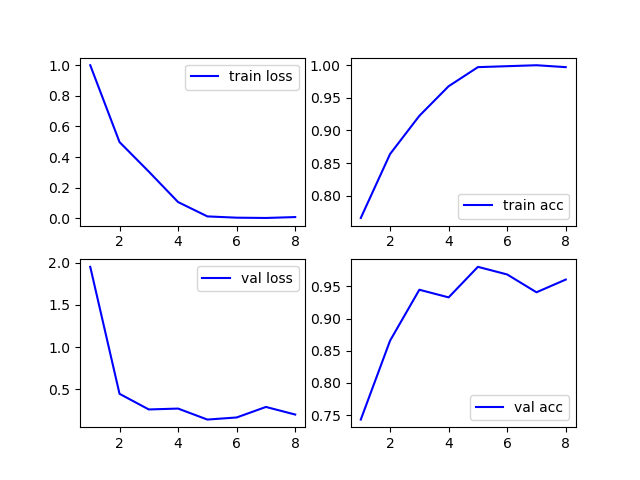
keras.optimizers.Adam(lr=1e-5)

EarlyStopping(monitor='val\_loss', patience=3)

ReduceLROnPlateau('val\_loss',patience=1)

test Result: 232 / 238 0.9747

fault distribution [1, 5]

**1.1**

LeftUp

[334 349]

[124 129]

[116 122]

参数：

接着1.0.h5训练

keras.optimizers.Adam(lr=1e-4)

EarlyStopping(monitor='val\_loss', patience=3)

ReduceLROnPlateau('val\_loss',patience=1)

test Result: 233 / 238 0.9789915966386554

fault distribution [1, 4]

1.2

Middle

[334 349]

[124 129]

[116 122]

参数：

接着1.1.h5训练

keras.optimizers.Adam(lr=1e-4)

EarlyStopping(monitor='val\_loss', patience=3)

ReduceLROnPlateau('val\_loss',patience=1)

test Result: 235 / 238 0.9873949579831933

fault distribution [1, 2]

1.3

Right

[334 349]

[124 129]

[116 122]

参数：

接着1.2.h5训练

keras.optimizers.Adam(lr=1e-4)

EarlyStopping(monitor='val\_loss', patience=3)

ReduceLROnPlateau('val\_loss',patience=1)

test Result: 227 / 238 0.9537815126050421

fault distribution [1, 10]

1.4

Mix

[1368 1450]

[457 467]

[471 483]

参数：

随机初始参数

keras.optimizers.Adam(lr=1e-5)

EarlyStopping(monitor='val\_loss', patience=3)

ReduceLROnPlateau('val\_loss',patience=1)

test Result: 927 / 954 0.9716981132075472

fault distribution [15, 12]

LeftDown：

test Result: 206 / 406 0.5073891625615764

fault distribution [11, 189]

LeftUp：

test Result: 198 / 406 0.4876847290640394

fault distribution [32, 176]

Middle:

test Result: 201 / 406 0.49507389162561577

fault distribution [1, 204]

Right:

test Result: 206 / 406 0.5073891625615764

fault distribution [0, 200]

Mix:

test Result: 810 / 1624 0.4987684729064039

fault distribution [86, 728]