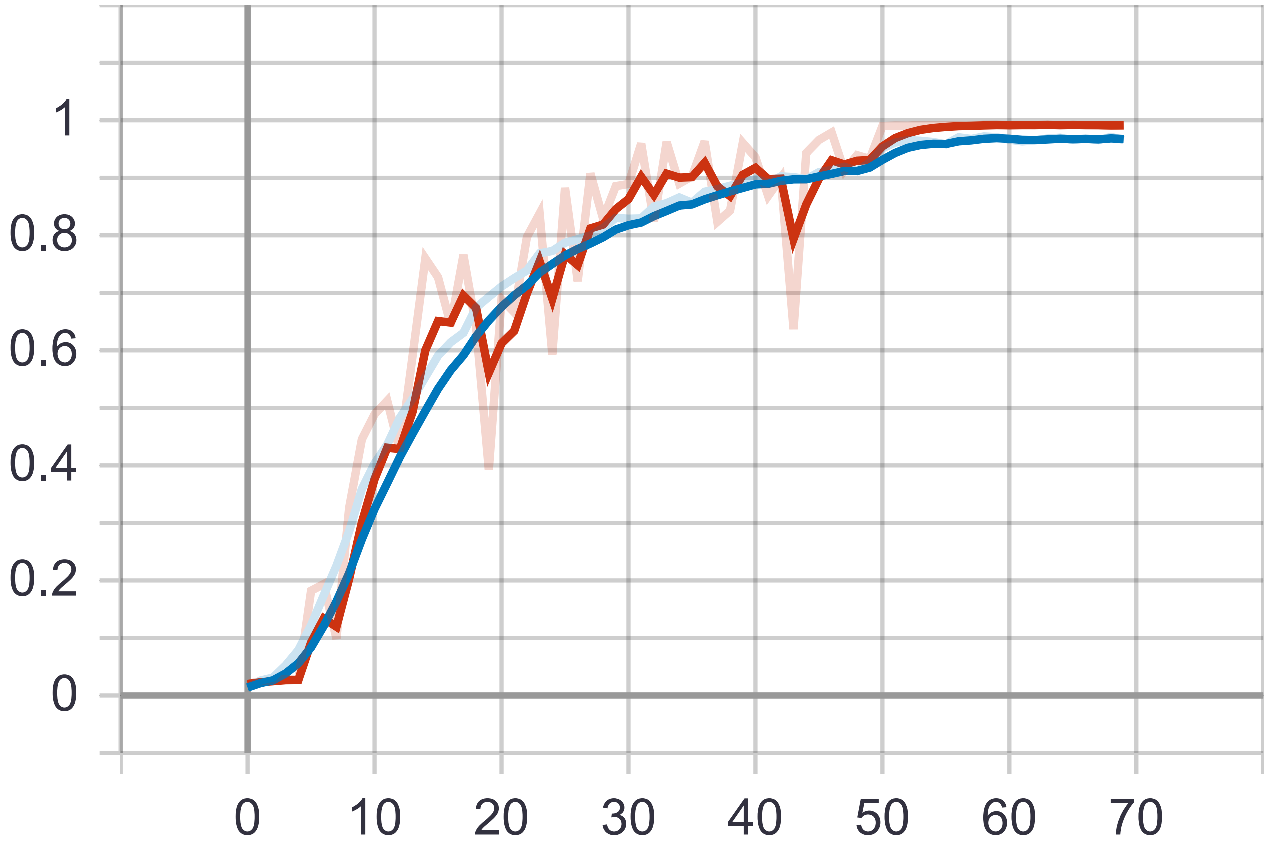
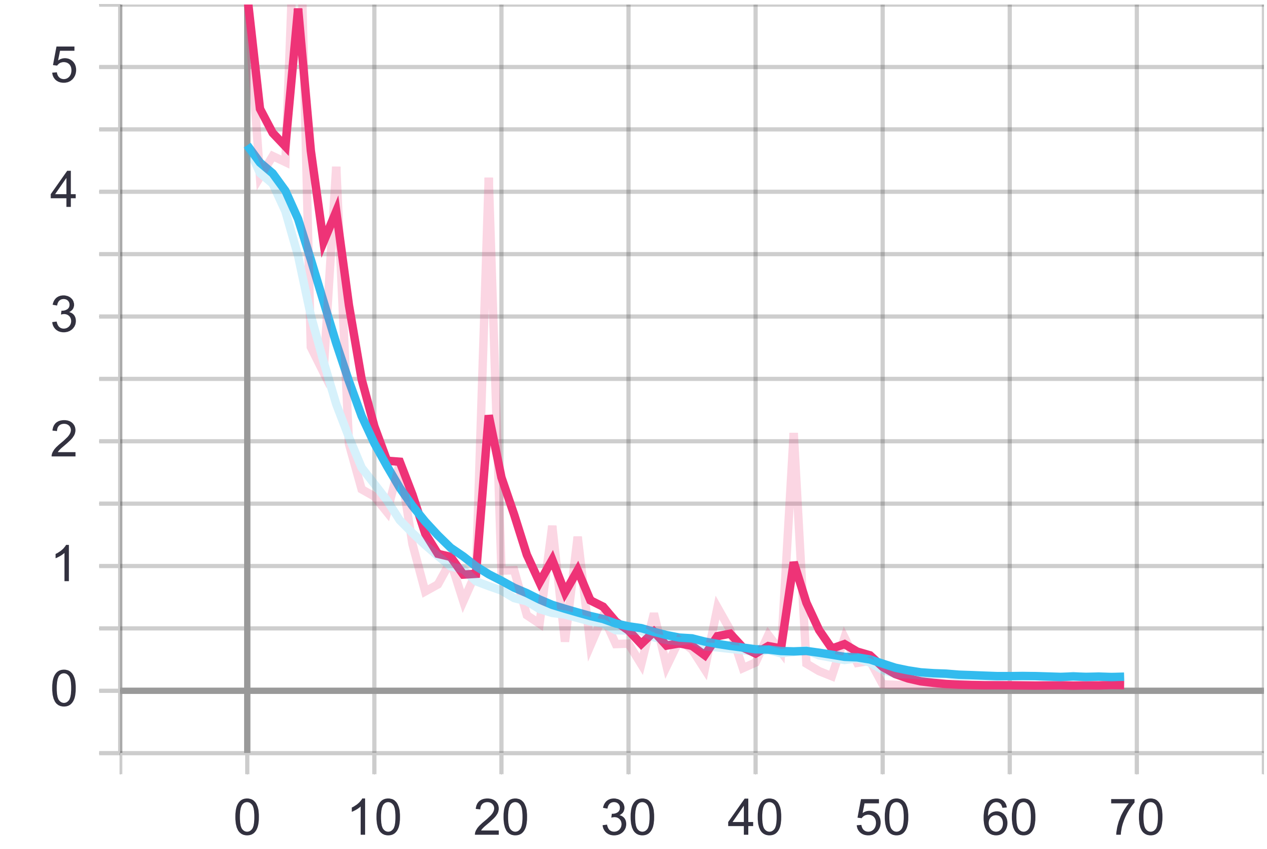
1. 数据目录结构

```
[1-92]                                  # 人员标签
    ├── multi/                          # 多光谱
    │   └── illum[1-3]/, normal/        # 干扰 `1-3`，无干扰
    │       ├── Multi_[1-7]_W1_1/       # 位置 `1-7`        无眼镜        每个位置目录下包括四个照片目录
    │       │   └── [1-4]/              #                                  每个目录下包括25张图片文件
    │       │       └── [1-25].jpg
    │       ├── Multi_4_W1_6            # 位置 `4`          墨镜          目录下包括四个照片目录
    │       │   └── [1-4]/              #                                  每个目录下包括25张图片文件
    │       │       └── [1-25].jpg
    │       └── Multi_[1-7]_W1_5        # 位置 `1-7`        眼镜          每个目录下包括25张图片文件， 部分人员无眼镜，即无该目录
    │           └── [1-25].jpg
    └── rgb                             # 可见光
        └── illum[1-3]/, normal/        # 干扰 `1-3`，无干扰
            ├── RGB_[1-7]_W1_1/         # 位置 `1-7`        无眼镜        每个位置目录下包括四张照片文件
            │   └── [1-4].jpg
            ├── RGB_4_W1_6/             # 位置 `4`          墨镜          目录下包括四张照片文件
            │   └── [1-4].jpg
            └── RGB_[1-7]_W1_5.jpg      # 位置 `1-7`        眼镜          部分人员无眼镜，即无该图片
```

2. 实验

首先进行试验，确定合适的`configer`参数，在此基础上进行实验，在配置文件`config.py`中保存的参数下，获得良好的实验结果





```
dsize = (112//2, 96//2)
```

3.1 划分比例的确定

> 确定在何种划分下进行实验，后续实验均以此结果为标准。

- 划分方式与上阶段一致，在每人的数据中，保留`Multi`与`RGB`同时检测出的图片路径，打乱后按一定比例划分；
- 本次实验划分时不做特殊处理，若需要其中指定条件的数据，可在`RecognizeDataset`中指定筛选条件`condition`；

运行

```
python gen_split.py

[split_112x96_[0.10:0.70:0.20]_[1]] n_items: 3796, n_train: 365, n_valid: 2633, n_test: 798, ratio: 0.096: 0.694: 0.210
[split_112x96_[0.10:0.70:0.20]_[2]] n_items: 3796, n_train: 365, n_valid: 2633, n_test: 798, ratio: 0.096: 0.694: 0.210
[split_112x96_[0.10:0.70:0.20]_[3]] n_items: 3796, n_train: 365, n_valid: 2633, n_test: 798, ratio: 0.096: 0.694: 0.210
[split_112x96_[0.10:0.70:0.20]_[4]] n_items: 3796, n_train: 365, n_valid: 2633, n_test: 798, ratio: 0.096: 0.694: 0.210
[split_112x96_[0.10:0.70:0.20]_[5]] n_items: 3796, n_train: 365, n_valid: 2633, n_test: 798, ratio: 0.096: 0.694: 0.210
[split_112x96_[0.20:0.60:0.20]_[1]] n_items: 3796, n_train: 735, n_valid: 2263, n_test: 798, ratio: 0.194: 0.596: 0.210
[split_112x96_[0.20:0.60:0.20]_[2]] n_items: 3796, n_train: 735, n_valid: 2263, n_test: 798, ratio: 0.194: 0.596: 0.210
[split_112x96_[0.20:0.60:0.20]_[3]] n_items: 3796, n_train: 735, n_valid: 2263, n_test: 798, ratio: 0.194: 0.596: 0.210
[split_112x96_[0.20:0.60:0.20]_[4]] n_items: 3796, n_train: 735, n_valid: 2263, n_test: 798, ratio: 0.194: 0.596: 0.210
[split_112x96_[0.20:0.60:0.20]_[5]] n_items: 3796, n_train: 735, n_valid: 2263, n_test: 798, ratio: 0.194: 0.596: 0.210
[split_112x96_[0.30:0.50:0.20]_[1]] n_items: 3796, n_train: 1104, n_valid: 1895, n_test: 797, ratio: 0.291: 0.499: 0.210
[split_112x96_[0.30:0.50:0.20]_[2]] n_items: 3796, n_train: 1104, n_valid: 1895, n_test: 797, ratio: 0.291: 0.499: 0.210
[split_112x96_[0.30:0.50:0.20]_[3]] n_items: 3796, n_train: 1104, n_valid: 1895, n_test: 797, ratio: 0.291: 0.499: 0.210
[split_112x96_[0.30:0.50:0.20]_[4]] n_items: 3796, n_train: 1104, n_valid: 1895, n_test: 797, ratio: 0.291: 0.499: 0.210
[split_112x96_[0.30:0.50:0.20]_[5]] n_items: 3796, n_train: 1104, n_valid: 1895, n_test: 797, ratio: 0.291: 0.499: 0.210
[split_112x96_[0.40:0.40:0.20]_[1]] n_items: 3796, n_train: 1474, n_valid: 1474, n_test: 848, ratio: 0.388: 0.388: 0.223
[split_112x96_[0.40:0.40:0.20]_[2]] n_items: 3796, n_train: 1474, n_valid: 1474, n_test: 848, ratio: 0.388: 0.388: 0.223
[split_112x96_[0.40:0.40:0.20]_[3]] n_items: 3796, n_train: 1474, n_valid: 1474, n_test: 848, ratio: 0.388: 0.388: 0.223
[split_112x96_[0.40:0.40:0.20]_[4]] n_items: 3796, n_train: 1474, n_valid: 1474, n_test: 848, ratio: 0.388: 0.388: 0.223
[split_112x96_[0.40:0.40:0.20]_[5]] n_items: 3796, n_train: 1474, n_valid: 1474, n_test: 848, ratio: 0.388: 0.388: 0.223
[split_112x96_[0.50:0.30:0.20]_[1]] n_items: 3796, n_train: 1895, n_valid: 1104, n_test: 797, ratio: 0.499: 0.291: 0.210
[split_112x96_[0.50:0.30:0.20]_[2]] n_items: 3796, n_train: 1895, n_valid: 1104, n_test: 797, ratio: 0.499: 0.291: 0.210
[split_112x96_[0.50:0.30:0.20]_[3]] n_items: 3796, n_train: 1895, n_valid: 1104, n_test: 797, ratio: 0.499: 0.291: 0.210
[split_112x96_[0.50:0.30:0.20]_[4]] n_items: 3796, n_train: 1895, n_valid: 1104, n_test: 797, ratio: 0.499: 0.291: 0.210
[split_112x96_[0.50:0.30:0.20]_[5]] n_items: 3796, n_train: 1895, n_valid: 1104, n_test: 797, ratio: 0.499: 0.291: 0.210
[split_112x96_[0.60:0.20:0.20]_[1]] n_items: 3796, n_train: 2263, n_valid: 704, n_test: 829, ratio: 0.596: 0.185: 0.218
[split_112x96_[0.60:0.20:0.20]_[2]] n_items: 3796, n_train: 2263, n_valid: 704, n_test: 829, ratio: 0.596: 0.185: 0.218
[split_112x96_[0.60:0.20:0.20]_[3]] n_items: 3796, n_train: 2263, n_valid: 704, n_test: 829, ratio: 0.596: 0.185: 0.218
[split_112x96_[0.60:0.20:0.20]_[4]] n_items: 3796, n_train: 2263, n_valid: 704, n_test: 829, ratio: 0.596: 0.185: 0.218
[split_112x96_[0.60:0.20:0.20]_[5]] n_items: 3796, n_train: 2263, n_valid: 704, n_test: 829, ratio: 0.596: 0.185: 0.218
[split_112x96_[0.70:0.10:0.20]_[1]] n_items: 3796, n_train: 2633, n_valid: 334, n_test: 829, ratio: 0.694: 0.088: 0.218
[split_112x96_[0.70:0.10:0.20]_[2]] n_items: 3796, n_train: 2633, n_valid: 334, n_test: 829, ratio: 0.694: 0.088: 0.218
[split_112x96_[0.70:0.10:0.20]_[3]] n_items: 3796, n_train: 2633, n_valid: 334, n_test: 829, ratio: 0.694: 0.088: 0.218
[split_112x96_[0.70:0.10:0.20]_[4]] n_items: 3796, n_train: 2633, n_valid: 334, n_test: 829, ratio: 0.694: 0.088: 0.218
[split_112x96_[0.70:0.10:0.20]_[5]] n_items: 3796, n_train: 2633, n_valid: 334, n_test: 829, ratio: 0.694: 0.088: 0.218
```

在当前目录下，生成文件夹`split`，其目录结构如下

```
split
└── split_112x96_[比例]_[划分计数]
    ├── note.txt
    ├── test_Multi.txt
    ├── test_RGB.txt
    ├── train_Multi.txt
    ├── train_RGB.txt
    ├── valid_Multi.txt
    └── valid_RGB.txt
```

其中比例形式为训练集：验证集：测试集，划分计数为1~5。

- 各比例下进行5次随机划分，依次在比例为以下情况时进行实验；
- 统计各情况下5次准确率、损失值，并计算均值；
- 做出曲线；

```
cd Ecust/louishsu/recognize_stage_2
python
>>> from main_update_config import main_3_1
>>> main_3_1()       # 训练、测试
>>> main_3_1(True)   # 输出文件到`images`
```

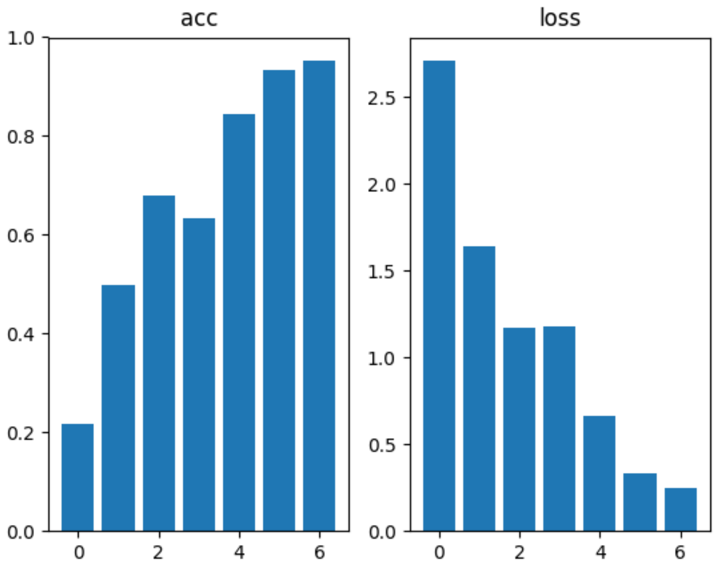
Multi

准确率：

count/ 比例	0.10: 0.70: 0.2	0.20: 0.60: 0.2	0.30: 0.50: 0.2	0.40: 0.40: 0.2	0.50: 0.30: 0.2	0.60: 0.20: 0.2	
1	0.24170000000000003	0.48100000000000004	0.6811	0.5521	0.8656	0.9469	
2	0.155	0.4457	0.6248	0.5312	0.8007	0.9239	
3	0.175	0.4806	0.6361	0.7095	0.8759999999999999	0.9384999999999999	
4	0.24789999999999998	0.6162	0.7094	0.7569	0.8581	0.9144	0.9
5	0.2614	0.47009999999999996	0.7382	0.6214999999999999	0.8151	0.9445999999999999	0.9
average	0.21620000000000003	0.49872000000000005	0.6779200000000001	0.6342399999999999	0.8431	0.93366	

损失值：

count/比例	0.10: 0.70: 0.2	0.20: 0.60: 0.2	0.30: 0.50: 0.2	0.40: 0.40: 0.2	0.50: 0.30: 0.2	0.60: 0.20: 0.2	0.70: 0.10: 0.2
1	2.5304	1.6541	1.1972	1.4315	0.638	0.299	0.2006
2	3.0033	1.6708	1.3105	1.479	0.85	0.357	0.1667
3	2.9661	1.6299	1.2551	1.002	0.5376	0.31	0.2357
4	2.4419	1.379	1.1182	0.8872	0.5724	0.4153	0.3248
5	2.6015	1.8503	0.9732	1.1133	0.707	0.2783	0.3129
average	2.70864	1.6368200000000002	1.17084	1.1825999999999999	0.6609999999999999	0.33192	0.24813999999999997



作图如下

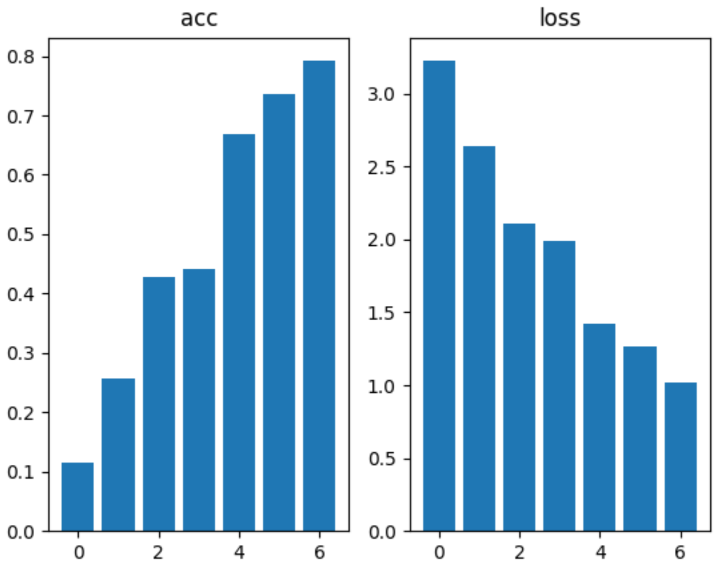
RGB

准确率：

count/ 比例	0.10: 0.70: 0.2	0.20: 0.60: 0.2	0.30: 0.50: 0.2	0.40: 0.40: 0.2	0.50: 0.30: 0.2	0.60: 0.20: 0.2
1	0.13269999999999998	0.2094	0.5321	0.43060000000000004	0.6828	0.7371
2	0.11410000000000001	0.24780000000000002	0.35969999999999996	0.478	0.5544	0.7216
3	0.1025	0.2878	0.44270000000000004	0.46299999999999997	0.7265999999999999	0.6829000000000001
4	0.105	0.2492	0.4264	0.4086	0.685	0.7553
5	0.12619999999999998	0.2957	0.3789	0.4225	0.6983	0.7884
average	0.11610000000000001	0.25798	0.42796	0.44054000000000004	0.66942	0.7370599999999999

损失值：

count/比例	0.10: 0.70: 0.2	0.20: 0.60: 0.2	0.30: 0.50: 0.2	0.40: 0.40: 0.2	0.50: 0.30: 0.2	0.60: 0.20: 0.2	0.70: 0.10: 0.2
1	2.9865	2.6234	1.8837	1.86	1.4542	1.2498	1.0729
2	3.2924	2.6482	2.2293	1.9742	1.5704	1.4082	0.9855
3	3.3243	2.6682	2.0879	1.8662	1.2611	1.4295	1.0614
4	3.391	2.6835	2.0477	2.244	1.5035	1.1464	1.0414
5	3.1297	2.5649	2.3028	1.9972	1.3184	1.1135	0.9489
average	3.2247800000000004	2.63764	2.11028	1.9883199999999999	1.42152	1.2694800000000002	1.02202



作图如下

可知比例为`0.50: 0.30: 0.2`时，效果最佳。

3.2 波段对比实验

- 根据实验3.1得到的最优划分，在5次随机划分进行实验；
- 依次选择单个波段的数据进行实验；
- 统计各情况下5次准确率、损失值，并计算均值；
- 做出曲线；

```
cd Ecust/louishsu/recognize_stage_2
python
>>> from main_update_config import main_3_2
>>> main_3_2()        # 训练、测试
>>> main_3_2(True)   # 输出文件到`images`
```

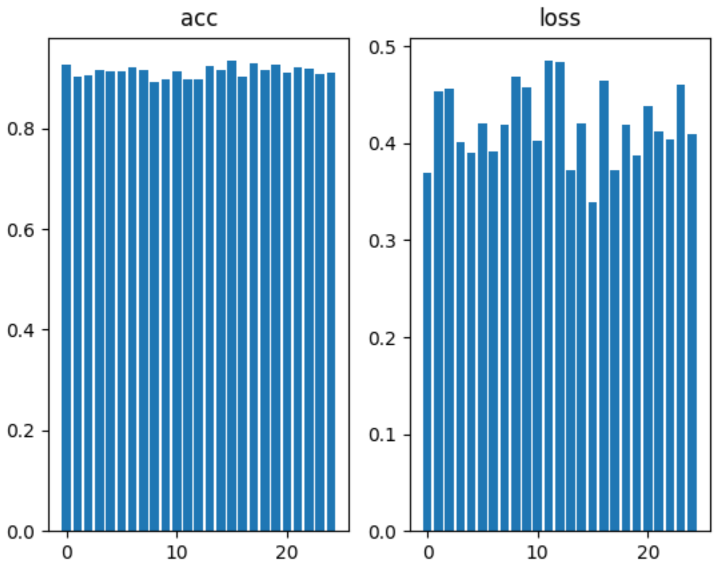
Multi

准确率：

count/波段索引	**1**	**2**	**3**	**4**	**5**	**6**	
1	0.8976000000000001	0.9334	0.8924	0.9193000000000001	0.9164	0.9109	
2	0.9323	0.8806	0.9164	0.9264	0.9264	0.9238	
3	0.941	0.8903	0.9094	0.9024	0.9362	0.9131	
4	0.9254000000000001	0.9178000000000001	0.9276000000000001	0.9203	0.8752	0.8817	
5	0.9406	0.8923000000000001	0.8833	0.9057999999999999	0.9167000000000001	0.9309999999999999	0.876
average	0.9273800000000001	0.90288	0.90582	0.9148400000000001	0.91418	0.9120999999999999	0.920

损失值：

count/波段索引	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
1	0.4638	0.3356	0.5563	0.3971	0.4195	0.4511	0.3445	0.4993	0.5026	
2	0.3163	0.4093	0.3634	0.2681	0.302	0.3416	0.3541	0.3355	0.5748	
3	0.3265	0.5748	0.4633	0.4776	0.3349	0.4342	0.3156	0.2937	0.3668	
4	0.3815	0.3812	0.3955	0.3735	0.4768	0.5282	0.3635	0.5451	0.452	
5	0.3581	0.5672	0.503	0.4922	0.4197	0.3478	0.5796	0.4191	0.4472	
average	0.36924	0.45362	0.4563	0.40170000000000006	0.39058	0.42058	0.39146000000000003	0.41853999999999997	0.46868	0.4579800



作图如下

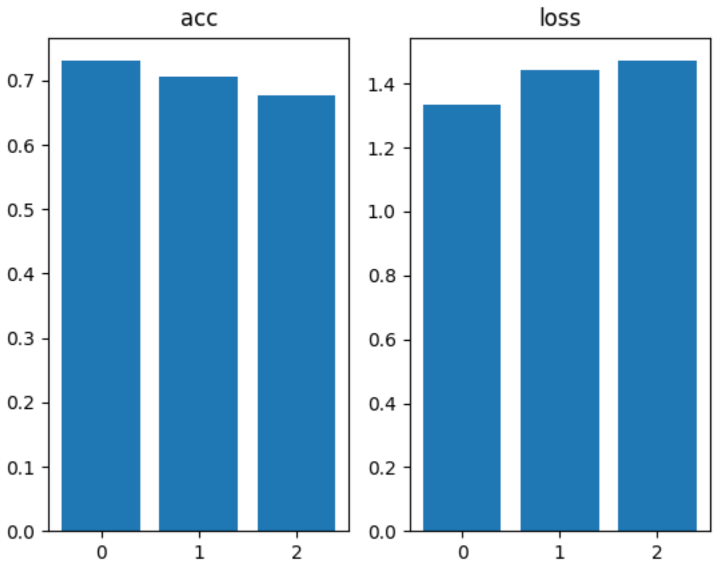
RGB

准确率：

count/波段索引	R	G	B
1	0.7515000000000001	0.7297	0.6979000000000001
2	0.7223	0.7382	0.7167
3	0.7303000000000001	0.6995999999999999	0.6639
4	0.7192000000000001	0.7217	0.6263000000000001
5	0.7286	0.6377	0.6806
average	0.73038	0.70538	0.67708

损失值：

count/波段索引	R	G	B
1	1.2973	1.5255	1.3572
2	1.2896	1.3162	1.4087
3	1.5191	1.5242	1.5751
4	1.2566	1.321	1.5907
5	1.3011	1.5228	1.4258
average	1.3327399999999998	1.44194	1.4715



作图如下

根据图3.2.1.1，按准确率将波段排序，降序排序如下：

```
Generating tables and figures [Multi]...
Best:  [23 19 24 16  7  8 21 13 17  3  1 14 22 11  9 20 15 12  5  6  4 25 18  2
 10]
Generating tables and figures [RGB]...
Best:  [3 1 2]
```

3.3 波段组合实验

> 该部分实验仅针对多光谱数据。

- 根据实验3.1得到的最优划分，在5次随机划分进行实验；
- 根据实验3.2得到的最优排序，依次选择最前1, 2, ..., 25个波段进行组合实验；
- 统计各情况下5次准确率、损失值，并计算均值；
- 做出曲线；

```
cd Ecust/louishsu/recognize_stage_2
python
>>> from main_update_config import main_3_3
>>> main_3_3()       # 训练、测试
>>> main_3_3(True)   # 输出文件到`images`
```

准确率：

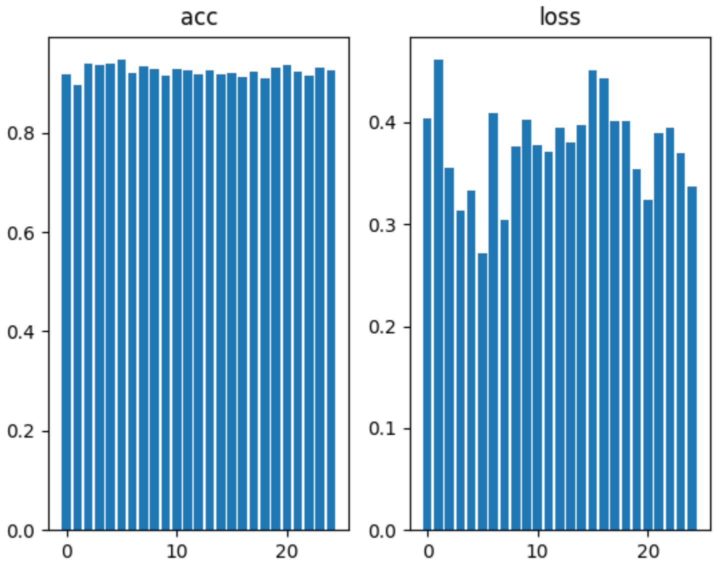
acc

count/ 组合数	**1**	**2**	**3**	**4**	**5**	**6**	
1	0.9055	0.9056000000000001	0.9662000000000001	0.9251999999999999	0.9601999999999999	0.9601000000000001	
2	0.8892	0.9007999999999999	0.9299	0.9517	0.9286	0.9431	0.959
3	0.9251	0.888	0.935	0.9083	0.9386	0.9469	
4	0.9359999999999999	0.9384999999999999	0.9422	0.9542	0.9348000000000001	0.9420999999999999	
5	0.9397	0.8473	0.9262999999999999	0.9405	0.9347	0.9420999999999999	0.922
average	0.9191	0.89604	0.9399200000000001	0.93598	0.9393800000000001	0.9468599999999998	0.919

损失值：

count/ 组合数	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
1	0.469	0.4575	0.2135	0.3826	0.2113	0.19	0.2932	0.205	0.3
2	0.4931	0.4039	0.3125	0.2469	0.332	0.2266	0.2035	0.3268	0.22

count/ 组合数	1	2	3	4	5	6	7	8	
3	0.386	0.492	0.3718	0.4195	0.3882	0.3321	0.4802	0.3112	0.41
4	0.3259	0.302	0.4243	0.2579	0.3819	0.282	0.6014	0.4338	0.57
5	0.3424	0.65	0.455	0.2567	0.351	0.3234	0.4652	0.2455	0.29
average	0.40327999999999997	0.46108000000000005	0.35542	0.31272	0.33288	0.27081999999999995	0.40869999999999995	0.30446	0.375



作图如下

3.4 光谱分辨率实验

> 该部分实验仅针对多光谱数据。

- 根据实验3.1得到的最优划分，在5次随机划分进行实验；
- 依次选择步长为1, 2, ..., 25，进行组合波段实验
- 统计各情况下5次准确率、损失值，并计算均值；
- 做出曲线；

```
cd Ecust/louishsu/recognize_stage_2
python
>>> from main_update_config import main_3_4
>>> main_3_4()       # 训练、测试
>>> main_3_4(True)   # 输出文件到`images`
```

准确率：

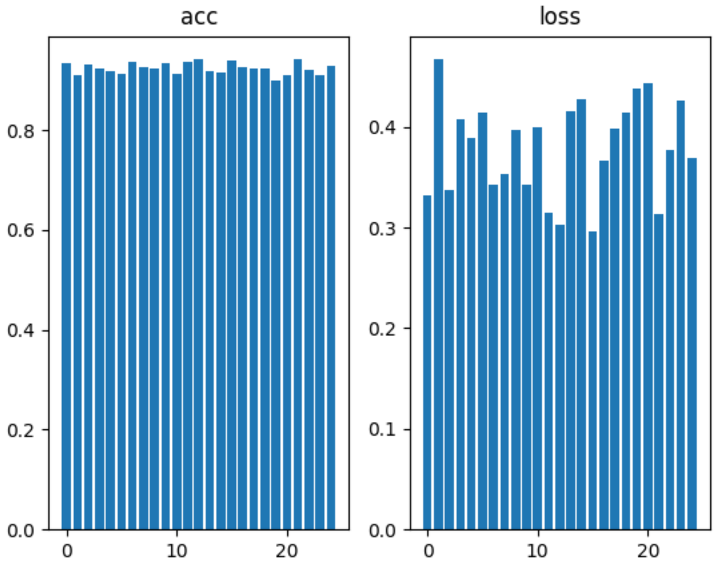
acc

count/ 波段步 长	1	2	3	4	5	6	7	
1	0.9469	0.941	0.9529000000000001	0.9155	0.8863	0.9192	0.9131	
2	0.9239	0.9347	0.9167000000000001	0.9420999999999999	0.9505	0.883	0.9467	
3	0.9384999999999999	0.9142	0.9096	0.9335	0.894	0.935	0.9433	
4	0.9144	0.9081999999999999	0.9457	0.9168000000000001	0.9312	0.9566	0.9299	
5	0.9445999999999999	0.855	0.9354	0.912	0.9322	0.8717	0.9445	
average	0.93366	0.91062	0.9320600000000001	0.92398	0.91884	0.9131	0.9354999999999999	0.925879

损失值：

count/ 波段步 长	1	2	3	4	5	6
1	0.299	0.3127	0.2382	0.5272	0.5782	0.4466

count/波段步长	1	2	3	4	5	6	
2	0.357	0.2748	0.3659	0.2627	0.2206	0.4472	
3	0.31	0.598	0.422	0.3964	0.4888	0.3422	
4	0.4153	0.5128	0.2903	0.4133	0.3558	0.2237	
5	0.2783	0.6369	0.3728	0.4414	0.3042	0.6111	
average	0.33192	0.46703999999999999	0.33784000000000003	0.40820000000000006	0.38952000000000003	0.41416000000000003	0.3423799999



作图如下

3.5 鲁棒性实验

- 根据实验3.1得到的最优划分，在5次随机划分进行实验；
- 选用全部波段进行实验；
- 统计5次实验中，改变条件得到表格；
- 做出曲线

```
cd Ecust/louishsu/recognize_stage_2
python
>>> from main_update_config import main_3_5
>>> main_3_5()        # 训练、测试
>>> main_3_5(True)    # 输出文件到`images`
```

3.5.1 干扰种类

统计无干扰、干扰1、干扰2、干扰3下，每次实验的准确率、损失

Multi

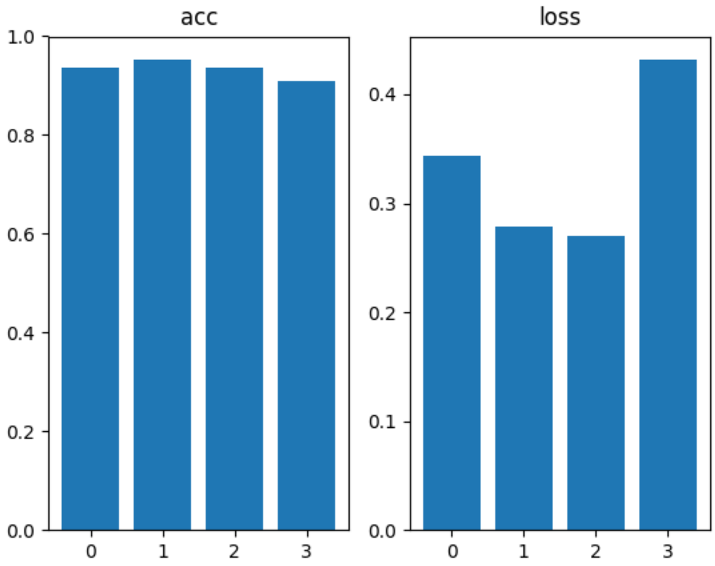
准确率

count/光照	illum1	illum2	illum3	normal
1	0.9585253596305847	0.96517413854599	0.9556650519371033	0.9086538553237915
2	0.9095237851142883	0.9599999785423279	0.9146919250488281	0.9134615659713745
3	0.9436619877815247	0.9508928656578064	0.9154228568077087	0.9424083828926086
4	0.9278846383094788	0.9227272868156433	0.9455445408821106	0.8592965006828308
5	0.9435897469520569	0.9583333134651184	0.9537037014961243	0.9207921028137207
average	0.9366371035575867	0.9514255166053772	0.937005615234375	0.9089224815368653

损失值

count/光照	illum1	illum2	illum3	normal

count/光照	illum1	illum2	illum3	normal
1	0.29575470089912415	0.19055938720703125	0.24191363155841827	0.46396496891975403
2	0.506862223148346	0.21154668927192688	0.3321232497692108	0.36441123485565186
3	0.22924265265464783	0.3756282329559326	0.34654149413108826	0.27989837527275085
4	0.44631752371788025	0.35346367955207825	0.25616833567619324	0.609917163848877
5	0.2411019206047058	0.26365926861763	0.17424382269382477	0.4398614168167114
average	0.3438558042049408	0.2789714515209198	0.27019810676574707	0.431610631942749



作图如下

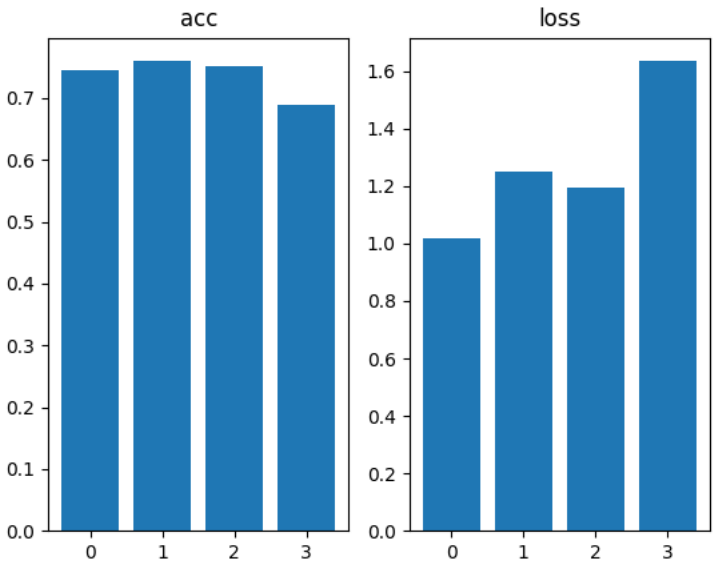
RGB

准确率

count/光照	illum1	illum2	illum3	normal
1	0.774193525314331	0.746268630027771	0.7635468244552612	0.6634615659713745
2	0.7047619223594666	0.7699999809265137	0.7298578023910522	0.682692289352417
3	0.7042253613471985	0.6919642686843872	0.6815920472145081	0.6492146849632263
4	0.7836538553237915	0.7909091114997864	0.7475247383117676	0.6934673190116882
5	0.764102578163147	0.8009259104728699	0.8333333134651184	0.7524752616882324
average	0.7461874485015869	0.7600135803222656	0.7511709451675415	0.6882622241973877

损失值

count/光照	illum1	illum2	illum3	normal
1	0.8474841117858887	1.2554359436035156	1.2438684701919556	1.6745892763137817
2	1.0703822374343872	1.474153995513916	1.2931591272354126	1.8105355501174927
3	1.2859147787094116	1.4190694093704224	1.3152297735214233	1.7251135110855103
4	0.8544989228248596	1.0853160619735718	1.1699944734573364	1.5005419254302979
5	1.032544732093811	1.0119115114212036	0.9546257853507996	1.4654229879379272
average	1.0181649565696715	1.249177384376526	1.1953755259513854	1.635240650177002



作图如下

3.5.2 偏转角度

统计各角度下，每次实验的准确率、损失

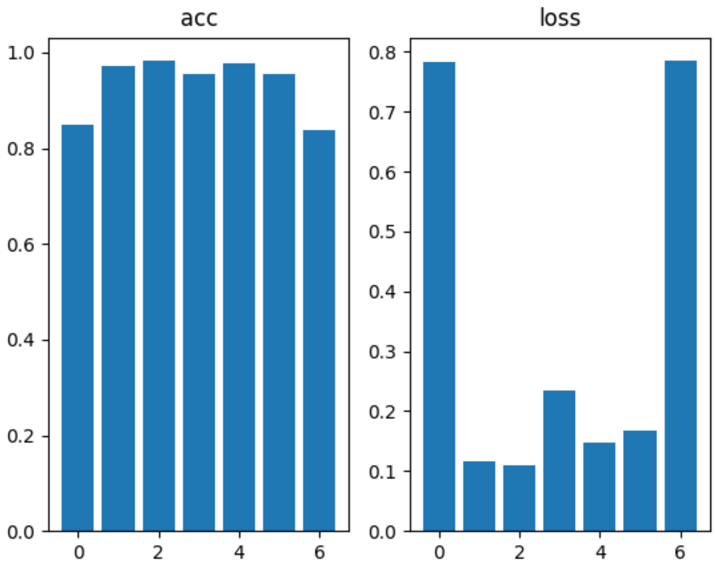
Multi

准确率

count/ 位置	1	2	3	4	5	6	
1	0.9304347634315491	0.9890109896659851	0.9897959232330322	0.9597989916801453	0.9727272987365723	0.9433962106704712	0.845
2	0.8272727131843567	0.9622641801834106	0.9818181991577148	0.9384615421295166	0.9893617033958435	0.9357798099517822	0.828
3	0.8073394298553467	0.9769230484962463	1.0	0.9659090638160706	0.9902912378311157	0.9523809552192688	0.851
4	0.8363636136054993	0.9528301954269409	0.9587628841400146	0.9528796076774597	0.9482758641242981	0.9489796161651611	0.783
5	0.8407079577445984	0.9797979593276978	0.982300877571106	0.9617486596107483	0.9807692170143127	0.9895833134651184	0.884
average	0.84842369556427	0.9721652746200562	0.9825355768203735	0.9557595729827881	0.9762850642204285	0.9540239810943604	0.838

损失值

count/ 位置	1	2	3	4	5	
1	0.46516335010528564	0.030804017558693886	0.036316562443971634	0.3195633292198181	0.23156127333641052	0.1781840473413467
2	0.8053836226463318	0.17389707267284393	0.15099187195301056	0.2555968761444092	0.0724916085600853	0.2233729511499400
3	0.901444673538208	0.09998495876789093	0.0629013180732727	0.15705424547195435	0.20752596855163574	0.1832218766212463
4	0.7416712045669556	0.23027673363685608	0.21197019517421722	0.2374773621559143	0.18385827541351318	0.2158989757299423
5	1.0006636381149292	0.04727545753121376	0.08522555232048035	0.1996830552816391	0.04238962382078171	0.0427315980195999
average	0.782865297794342	0.11644764803349972	0.1094810999929905	0.23387497365474702	0.1475653499364853	0.1686818897724151



作图如下

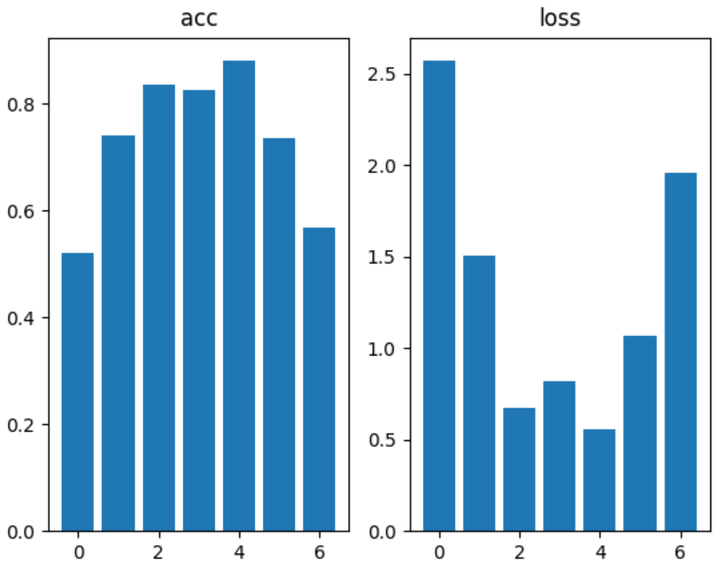
RGB

准确率

count/位置	1	2	3	4	5	6	
1	0.47826087474823	0.692307710647583	0.8673469424247742	0.7939698696136475	0.918181836605072	0.801886796951294	0.5
2	0.581818163394928	0.7452830076217651	0.7909091114997864	0.8153846263885498	0.8829787373542786	0.60550457239151	0.57
3	0.40366971492767334	0.7384615540504456	0.7714285850524902	0.7670454382896423	0.844660222530365	0.7333333492279053	0.45
4	0.5636363625526428	0.7264150977134705	0.8247422575950623	0.8795811533927917	0.8620689511299133	0.7755101919174194	0.56
5	0.5752212405204773	0.7979797720909119	0.9292035102844238	0.868852436542511	0.8942307829856873	0.7604166865348816	0.66
average	0.5205212712287903	0.7400894284248352	0.8367260813713073	0.8249667048454284	0.8804241061210633	0.735330319404602	0.56

损失值

count/位置	1	2	3	4	5	6	
1	3.0169320106506348	1.458120346069336	0.813869297504425	0.7641111612319946	0.49475374817848206	0.7900811433792114	1.7
2	2.539004325866699	1.6696488857269287	0.988752007484436	0.815473198890686	0.492302805185318	1.8571171760559082	1.8
3	2.5576670169830322	1.5919281244277954	0.675086259841919	1.001042366027832	0.584235668182373	0.9687958359718323	2.8
4	2.532008171081543	1.1013211011886597	0.5062957406044006	0.7470343708992004	0.5797130465507507	0.8773159384727478	1.9
5	2.1972715854644775	1.7188125848770142	0.38396337628364563	0.7599490880966187	0.6170359253883362	0.854550302028656	1.4
average	2.568576622009277	1.5079662084579468	0.6735933363437653	0.8175220370292664	0.553608238697052	1.0695720791816712	1.9



作图如下

3.5.3 遮挡实验

统计无眼镜、近视眼镜、太阳镜下，每次实验的准确率、损失

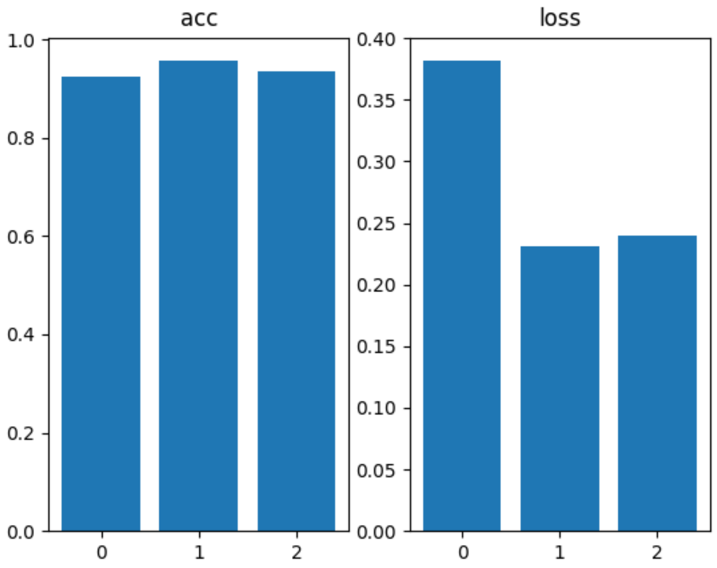
Multi

准确率

count/眼镜	1	5	6
1	0.9381044507026672	0.9734513163566589	0.930232584476471
2	0.9178571701049805	0.9621621370315552	0.8809523582458496
3	0.9425926208496094	0.9248826503753662	0.9473684430122375
4	0.8935361504554749	0.9495412707328796	0.9529411792755127
5	0.9326047301292419	0.9708737730979919	0.9594594836235046
average	0.9249390244483948	0.9561822295188904	0.9341908097267151

损失值

count/眼镜	1	5	6
1	0.3519522547721863	0.2069748342037201	0.22510196268558502
2	0.42510712146759033	0.11289246380329132	0.42477288842201233
3	0.29311615228652954	0.4096846282482147	0.13867482542991638
4	0.47649845480918884	0.31510424613952637	0.28687840700149536
5	0.36150237917900085	0.11045077443122864	0.1248135194182396
average	0.3816352725028992	0.23102138936519623	0.24004832059144973



作图如下

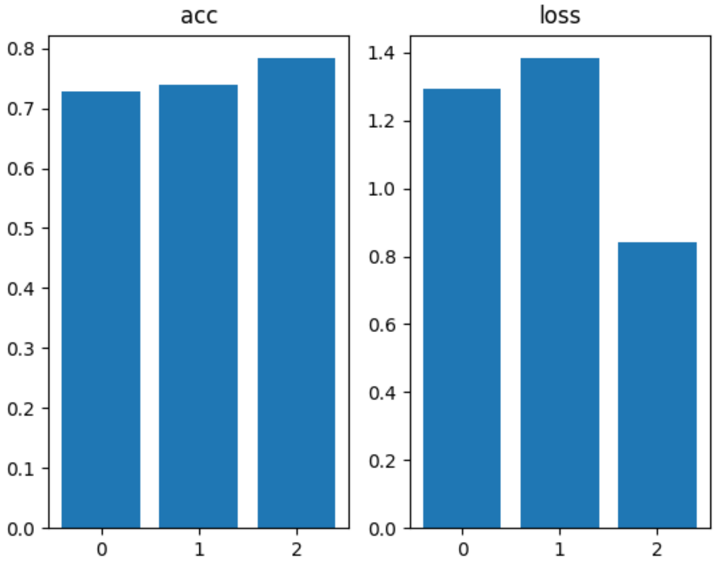
RGB

准确率

count/眼镜	1	5	6
1	0.7272727489471436	0.76106196641922	0.7325581312179565
2	0.7250000238418579	0.6972972750663757	0.75
3	0.6777777671813965	0.6760563254356384	0.7368420958518982
4	0.7414448857307434	0.747706413269043	0.8588235378265381
5	0.7723132967948914	0.8155339956283569	0.837837815284729
average	0.7287617444992065	0.7395311951637268	0.7832123160362243

损失值

count/眼镜	1	5	6
1	1.2996947765350342	1.265355110168457	0.9203951358795166
2	1.4102486371994019	1.6275476217269897	0.9312302470207214
3	1.3472990989685059	1.778964877128601	1.0416909456253052
4	1.139885425567627	1.3269877433776855	0.7363423109054565
5	1.2593384981155396	0.9146506190299988	0.5721426010131836
average	1.2912932872772216	1.3827011942863465	0.8403602480888367



作图如下