RGB Plan v1 架構和合併方法

1. RGB Plan v1:

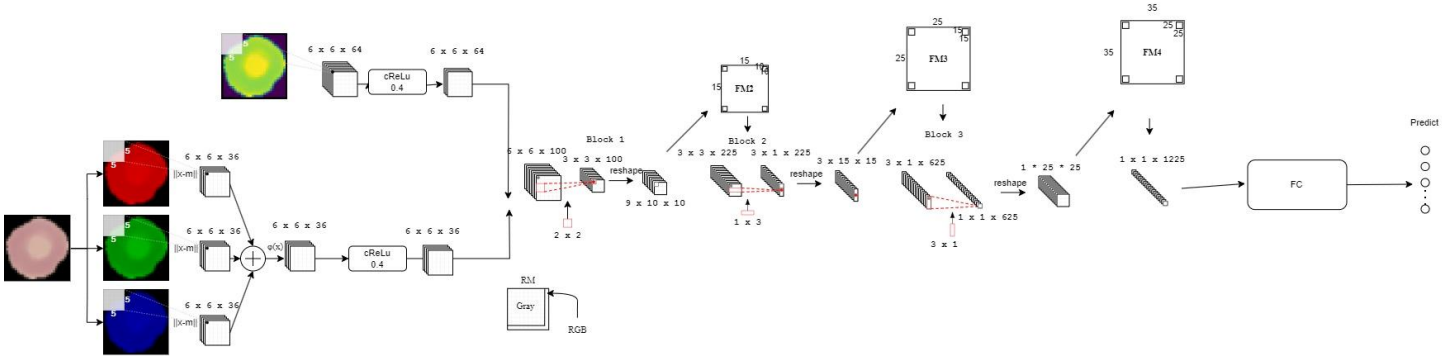
```
Input shape = (3, 28, 28)
SOMNetwork(
    (RGB_preprocess): Sequential(
      (0): RGB_Conv2d(weight shape=torch.Size([36, 3]), kernel size=(5, 5))
       (1): cReLU(bias = 0.4)
    )
    (GRAY_preprocess): Sequential(
      (0): RBF_Conv2d(weight shape=torch.Size([64, 1, 5, 5]), kernel size=(5, 5))
      (1): cReLU(bias = 0.4)
   )
  (layer1): Sequential(
      (0): SFM(filter=(2, 2), alpha=0.9)
   )
   (layer2): Sequential(
      (0): RBF_Conv2d(weight shape=torch.Size([225, 1, 10, 10]), kernel size=(10, 10))
      (1): cReLU(bias = 0.1)
      (2): SFM(filter=(1, 3), alpha=0.8999999761581421)
      (3): RBF_Conv2d(weight shape=torch.Size([625, 1, 15, 15]), kernel size=(15, 15))
      (4): cReLU(bias = 0.01)
      (5): SFM(filter=(3, 1), alpha=0.9)
      (6): RBF_Conv2d(weight shape=torch.Size([1225, 1, 25, 25]), kernel size=(25, 25))
      (7): cReLU(bias = 0.01)
   )
   (fc1): Sequential(
      (0): Linear(in_features=1225, out_features=15, bias=True)
   )
)
```

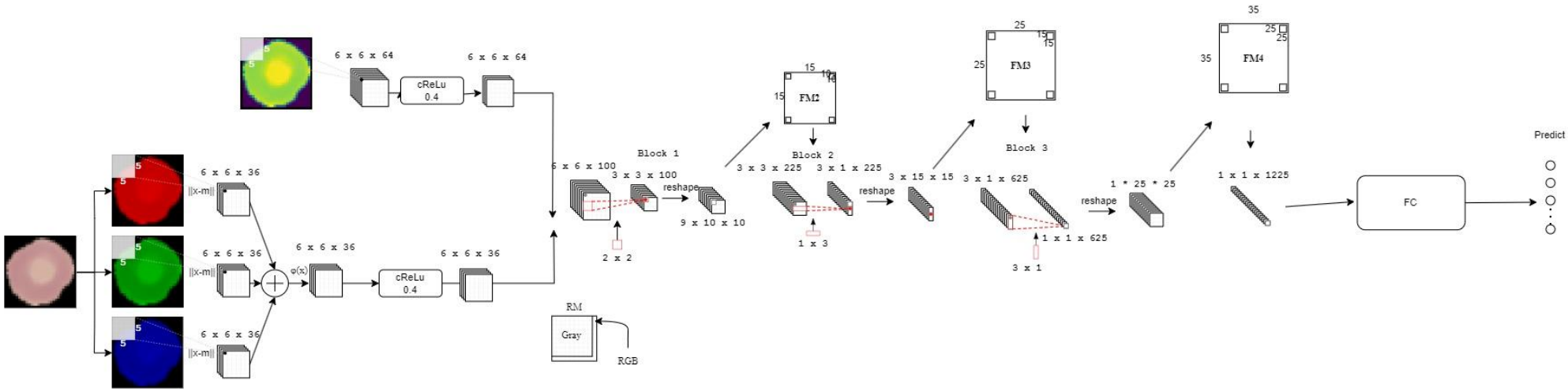
每層的 output shape:

Input shape = [-1(batch_size), 3, 28, 28]

Layer (type)	Output Shape	Param #
RGB_Conv2d-1	[-1, 36, 6, 6]	0
cReLU-2	[-1, 36, 6, 6]	1
RBF_Conv2d-3	[-1, 64, 6, 6]	1,600
cReLU-4	[-1, 64, 6, 6]	1
SFM-5	[-1, 1, 10, 10]	0
RBF_Conv2d-6	[-1, 225, 1, 1]	22,500
cReLU-7	[-1, 225, 1, 1]	1
SFM-8	[-1, 1, 15, 15]	0
RBF_Conv2d-9	[-1, 625, 1, 1]	140,625
cReLU-10	[-1, 625, 1, 1]	1
SFM-11	[-1, 1, 25, 25]	0
RBF_Conv2d-12	[-1, 1225, 1, 1]	765,625
cReLU-13	[-1, 1225, 1, 1]	1
Linear-14	[-1, 15]	18,390

架構圖




6 x 6 x 64
cReLu
0.4
6 x 6 x 64
6 x 6 x 36
||x-m||
6 x 6 x 36
||x-m||
6 x 6 x 36
φ(x)
6 x 6 x 36
cReLu
0.4
6 x 6 x 36
||x-m||
RM
Gray
RGB
6 x 6 x 100
Block 1
3 x 3 x 100
reshape
9 x 10 x 10
2 x 2
15
FM2
Block 2
3 x 3 x 225
3 x 1 x 225
reshape
1 x 3
3 x 15 x 15
25
FM3
Block 3
3 x 1 x 625
1 x 1 x 625
3 x 1
reshape
1 * 25 * 25
35
FM4
1 x 1 x 1225
FC
Predict

2. **RGB Plan v1 in (64, 64)**:

```
SOMNetwork(
  (RGB_preprocess): Sequential(
    (0): RGB_Conv2d(weight shape=torch.Size([36, 3]), kernel size=(5, 5))
    (1): cReLU(bias = 0.4)
  )
  (GRAY_preprocess): Sequential(
    (0): RBF_Conv2d(weight shape=torch.Size([64, 1, 5, 5]), kernel size=(5, 5))
    (1): cReLU(bias=0.4)
  )
  (layer1): Sequential(
    (0): SFM(filter=(5, 5), alpha=0.9)
  )
  (layer2): Sequential(
    (0): RBF_Conv2d(weight shape=torch.Size([225, 1, 10, 10]), kernel size=(10, 10))
    (1): cReLU(bias=0.1)
    (2): SFM(filter=(2, 2), alpha=0.9)
    (3): RBF_Conv2d(weight shape=torch.Size([625, 1, 15, 15]), kernel size=(15, 15))
    (4): cReLU(bias=0.01)
    (5): SFM(filter=(1, 3), alpha=0.9)
    (6): RBF_Conv2d(weight shape=torch.Size([1225, 1, 25, 25]), kernel size=(25, 25))
    (7): cReLU(bias=0.01)
    (8): SFM(filter=(3, 1), alpha=0.8999999761581421)
    (9): RBF_Conv2d(weight shape=torch.Size([2025, 1, 35, 35]), kernel size=(35, 35))
    (10): cReLU(bias= 0.0100)
  (fc1): Sequential(
    (0): Linear(in_features=2025, out_features=15, bias=True)
  )
)
```

每層的 output shape:

Input shape = [-1, 3, 64, 64]

Layer (type)	Output Shape	Param #
RGB_Conv2d-1	[-1, 36, 30, 30]	0
cReLU-2	[-1, 36, 30, 30]	1
RBF_Conv2d-3	[-1, 64, 30, 30]	1,600
cReLU-4	[-1, 64, 30, 30]	1
SFM-5	[-1, 1, 10, 10]	0
RBF_Conv2d-6	[-1, 225, 1, 1]	22,500
cReLU-7	[-1, 225, 1, 1]	1
SFM-8	[-1, 1, 15, 15]	0
RBF_Conv2d-9	[-1, 625, 1, 1]	140,625
cReLU-10	[-1, 625, 1, 1]	1
SFM-11	[-1, 1, 25, 25]	0
RBF_Conv2d-12	[-1, 1225, 1, 1]	765,625
cReLU-13	[-1, 1225, 1, 1]	1
SFM-14	[-1, 1, 35, 35]	0
RBF_Conv2d-15	[-1, 2025, 1, 1]	2,480,625
cReLU-16	[-1, 2025, 1, 1]	1
Linear-17	[-1, 15]	30,390

架構圖同 RGB Plan v1 in (28, 28)