

100

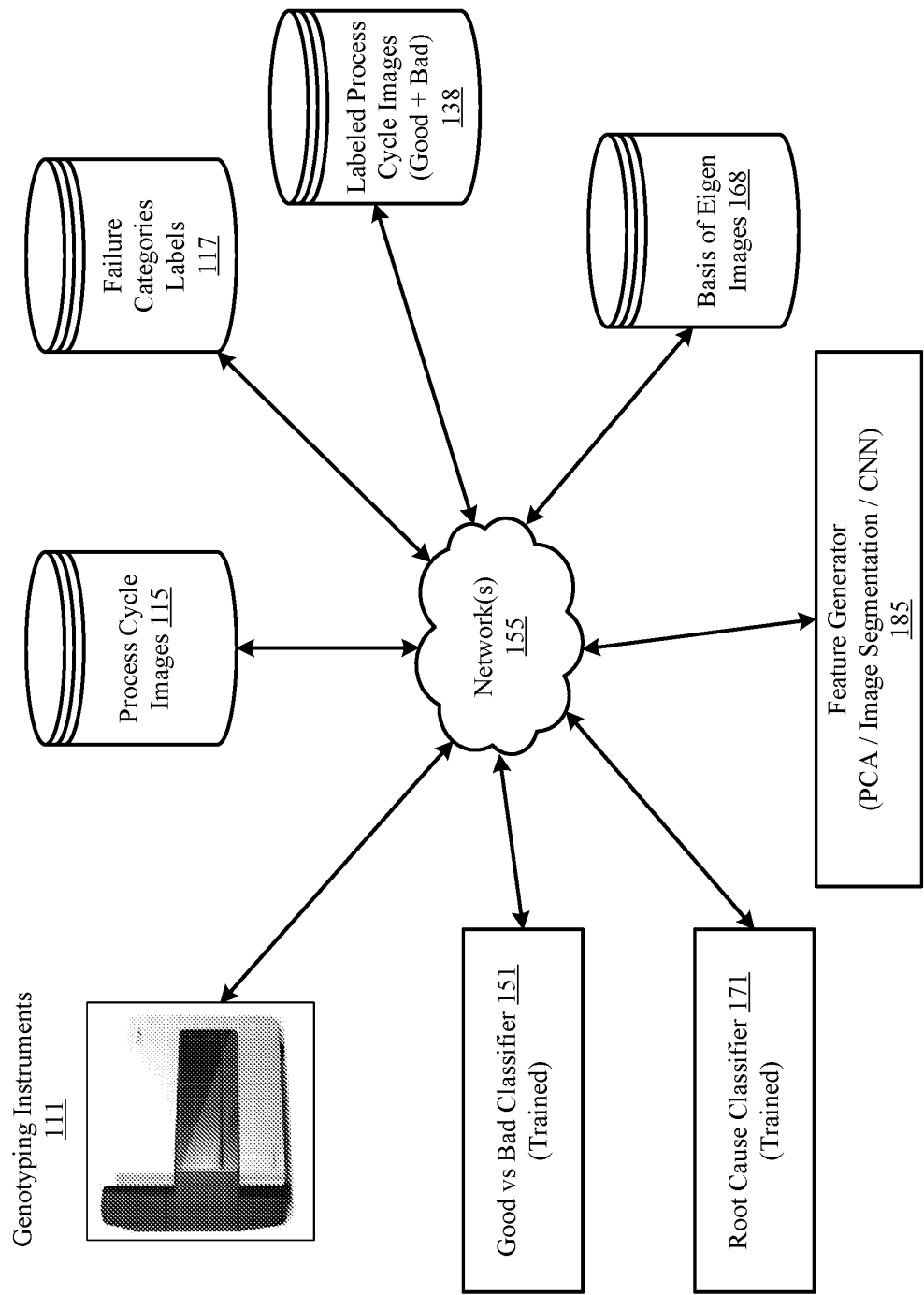


FIG. 1

200

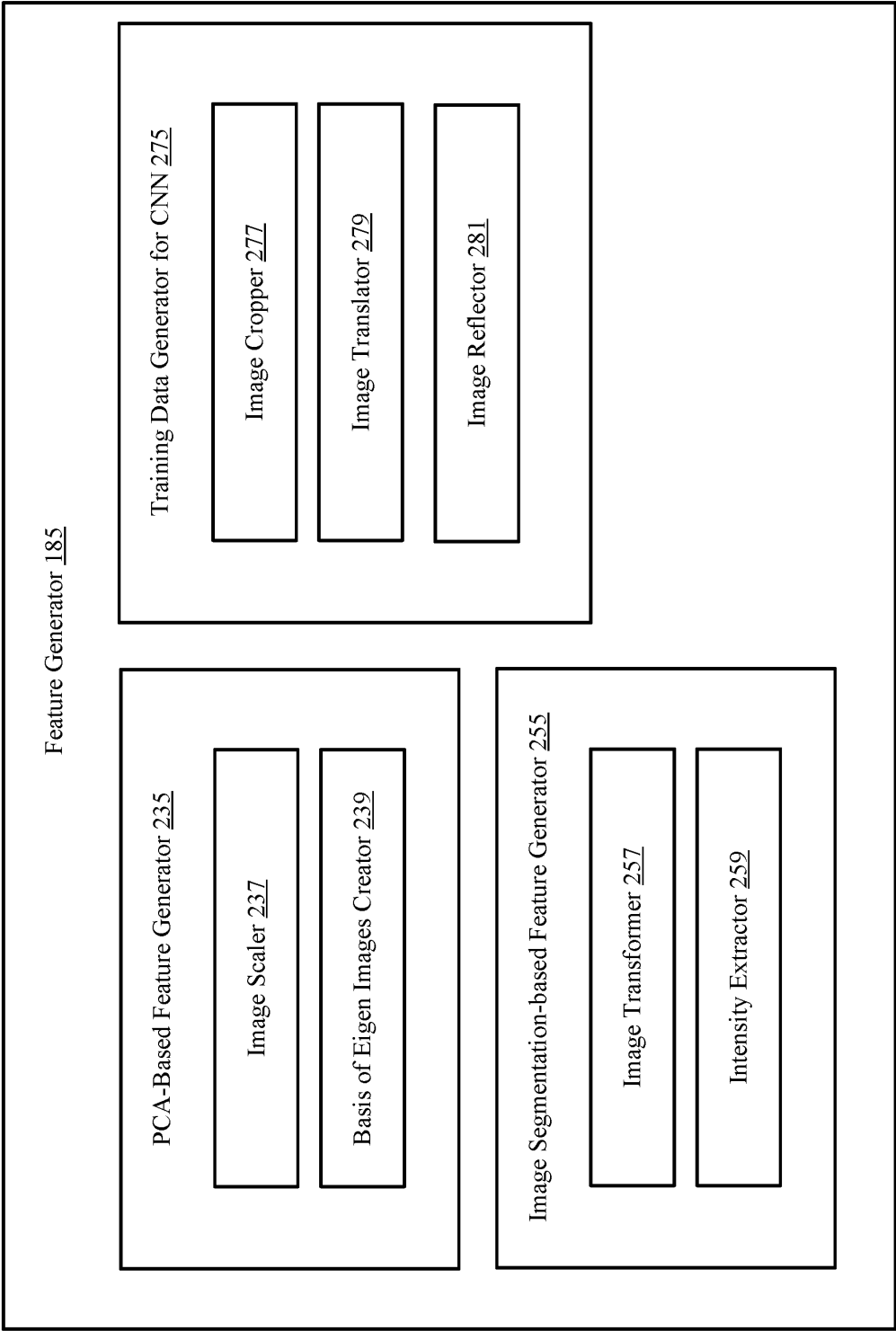


FIG. 2

300

Genotyping Process Steps

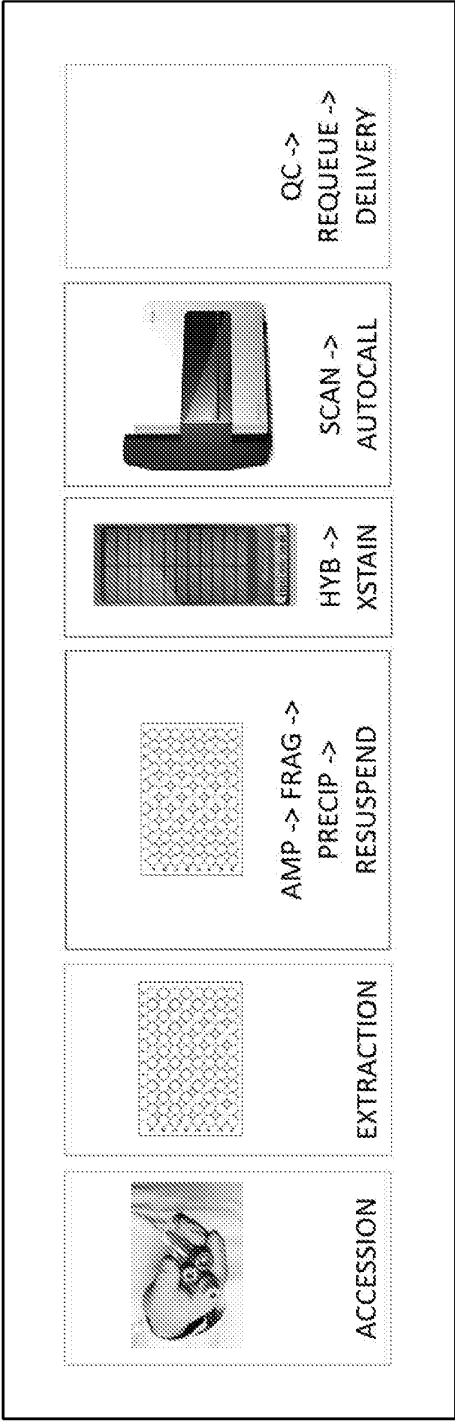


FIG. 3

400

Good Process Cycle Images – Successful Genotyping

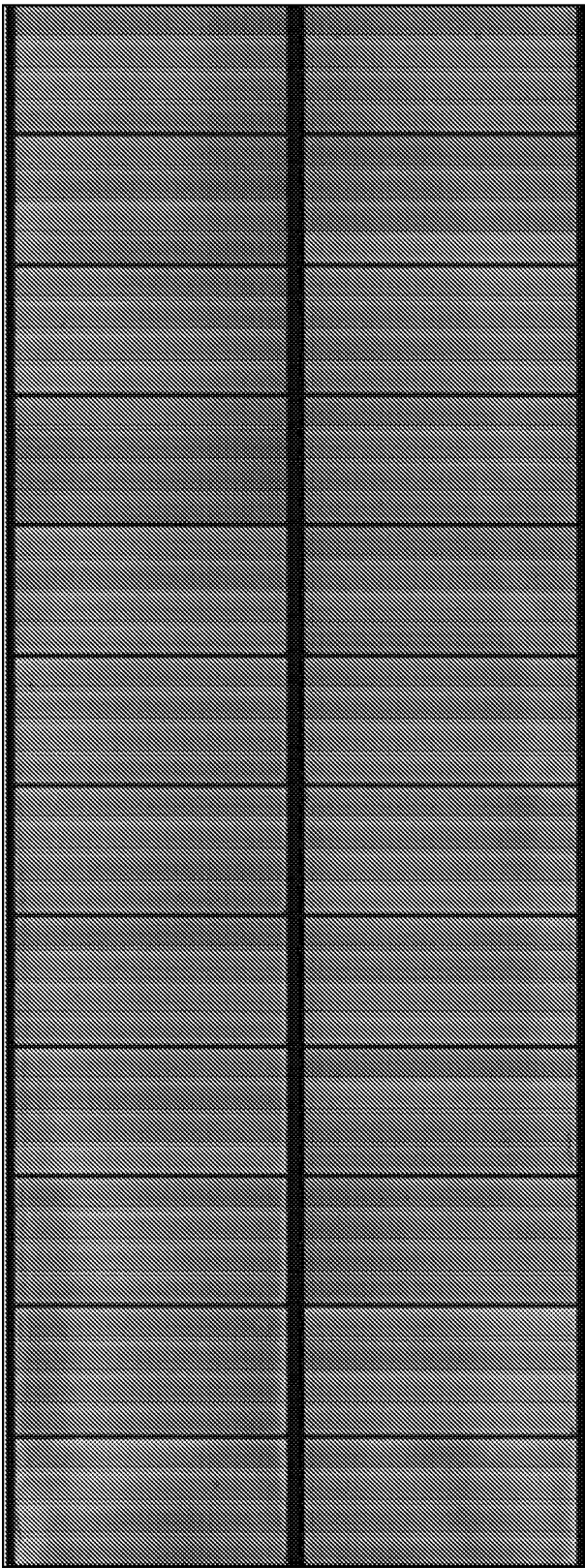
99.912	99.917	98.91	99.902	99.893	99.913	99.915	99.884	98.833	99.875	99.633	99.692
R01C02	R02C02	R03C02	R04C02	R05C02	R06C02	R07C02	R08C02	R09C02	R10C02	R11C02	R12C02
											
R01C01	R02C01	R03C01	R04C01	R05C01	R06C01	R07C01	R08C01	R09C01	R10C01	R11C01	R12C01
99.9	99.914	99.767	99.862	99.913	99.922	99.884	99.621	99.79	9.843	99.814	99.797

FIG. 4

510

Bad Process Cycle Image – Hybridization (Hyb) Failure (1/2)

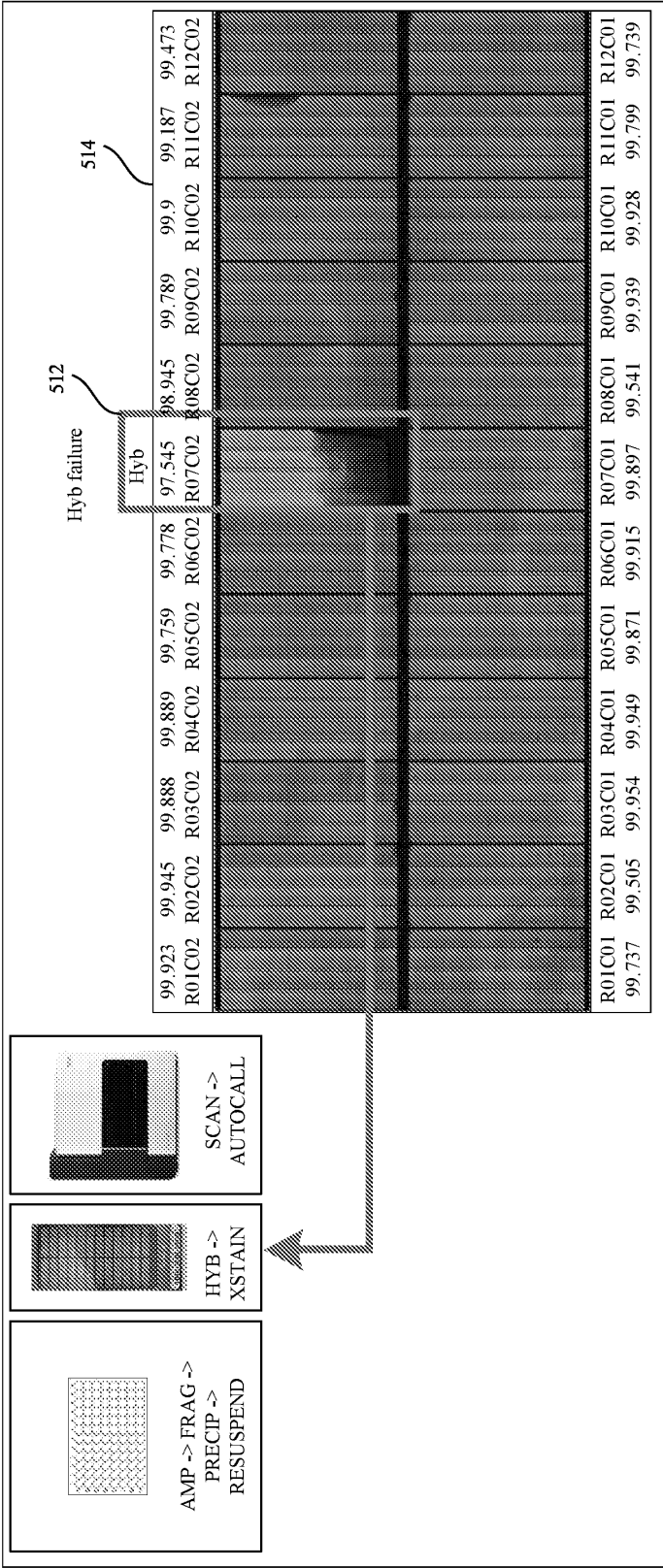


FIG. 5A

515

Bad Process Cycle Image – Hybridization (Hyb) Failure (2/2)

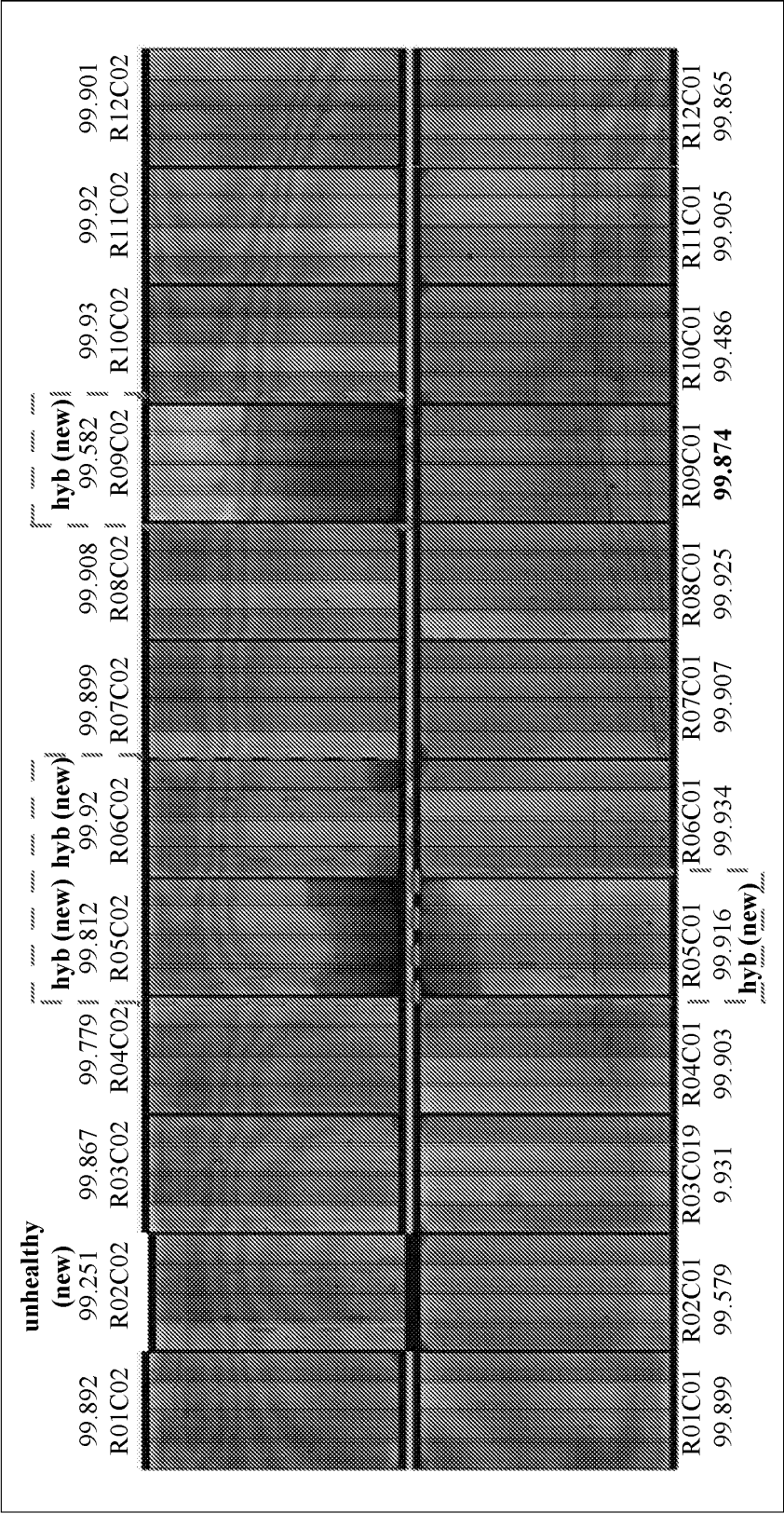


FIG. 5B

520

Bad Process Cycle Images – Spacer Shift Failure (1/2)

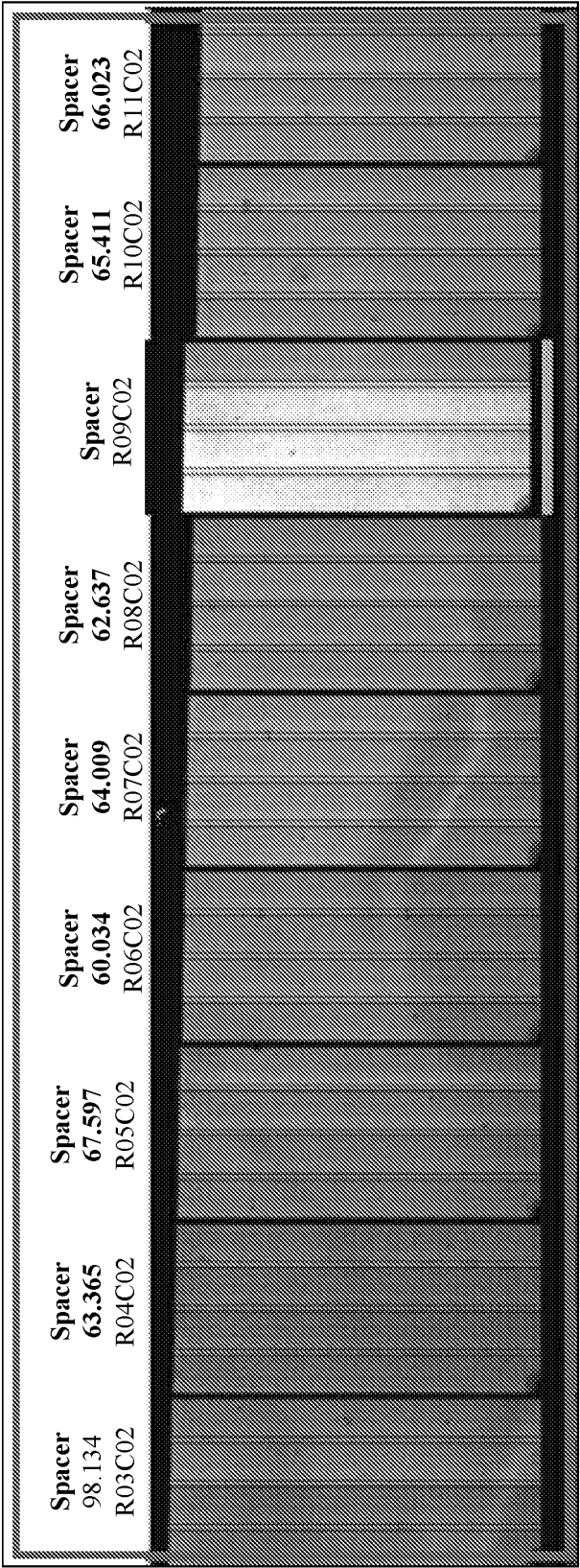


FIG. 5C

Bad Process Cycle Images – Spacer Shift Failure (2/2)

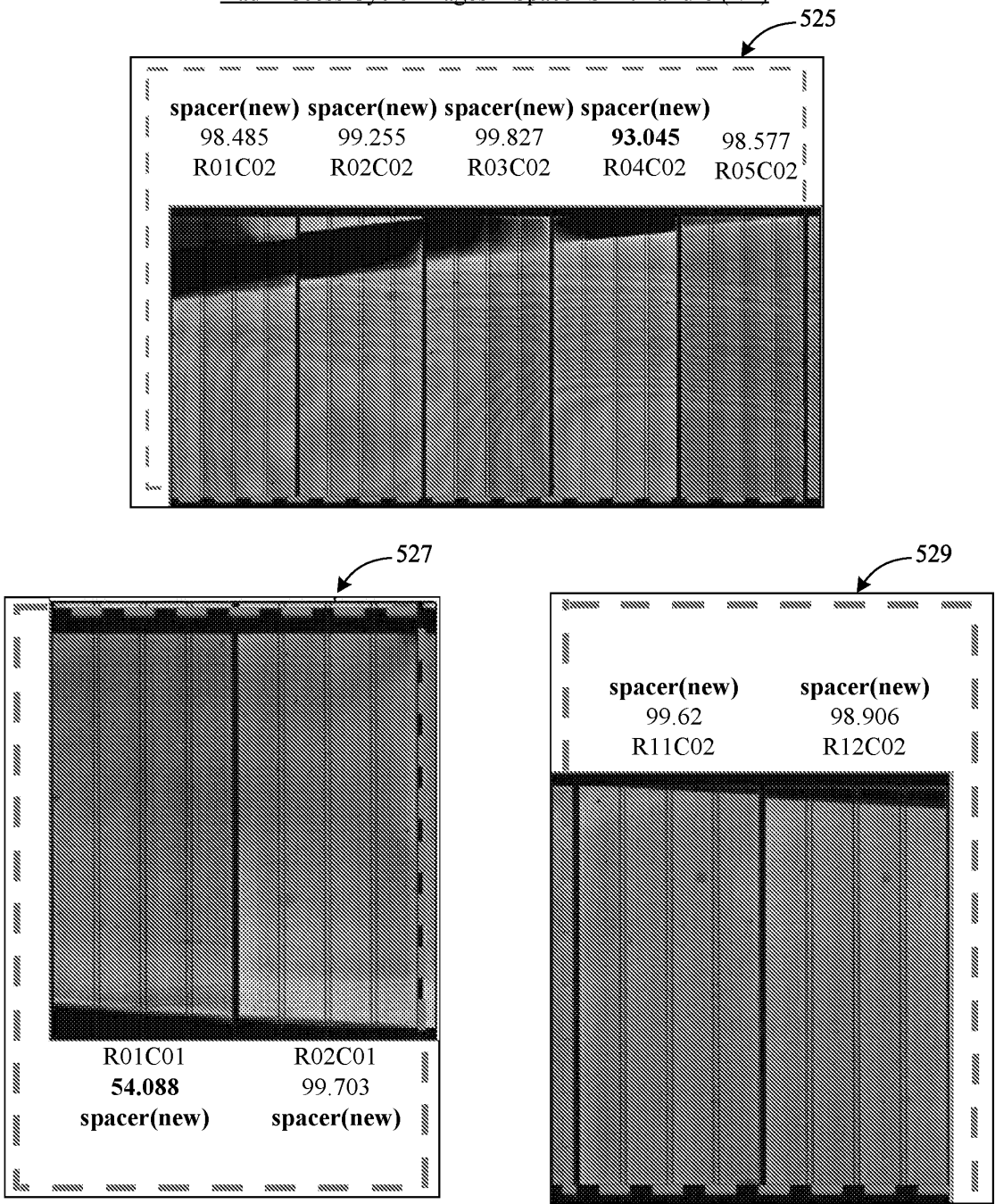


FIG. 5D

530

Bad Process Cycle Images – Offset Failure

offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)
96.996	96.969	96.944	96.962	97.077	97.277	97.304	97.531	97.522	97.492	97.625	97.251	
R01C02	R02C02	R03C02	R04C02	R05C02	R06C02	R07C02	R08C02	R09C02	R10C02	R11C02	R12C02	
R01C01	R02C01	R03C01	R04C01	R05C01	R06C01	R07C01	R08C01	R09C01	R10C01	R11C01	R12C01	
97.246	96.976	97.177	97.311	97.503	97.652	97.792	97.905	97.876	97.044	98.005	97.906	
offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	offset (new)	

FIG. 5E

Bad Process Cycle Images – Surface Abrasion Failure

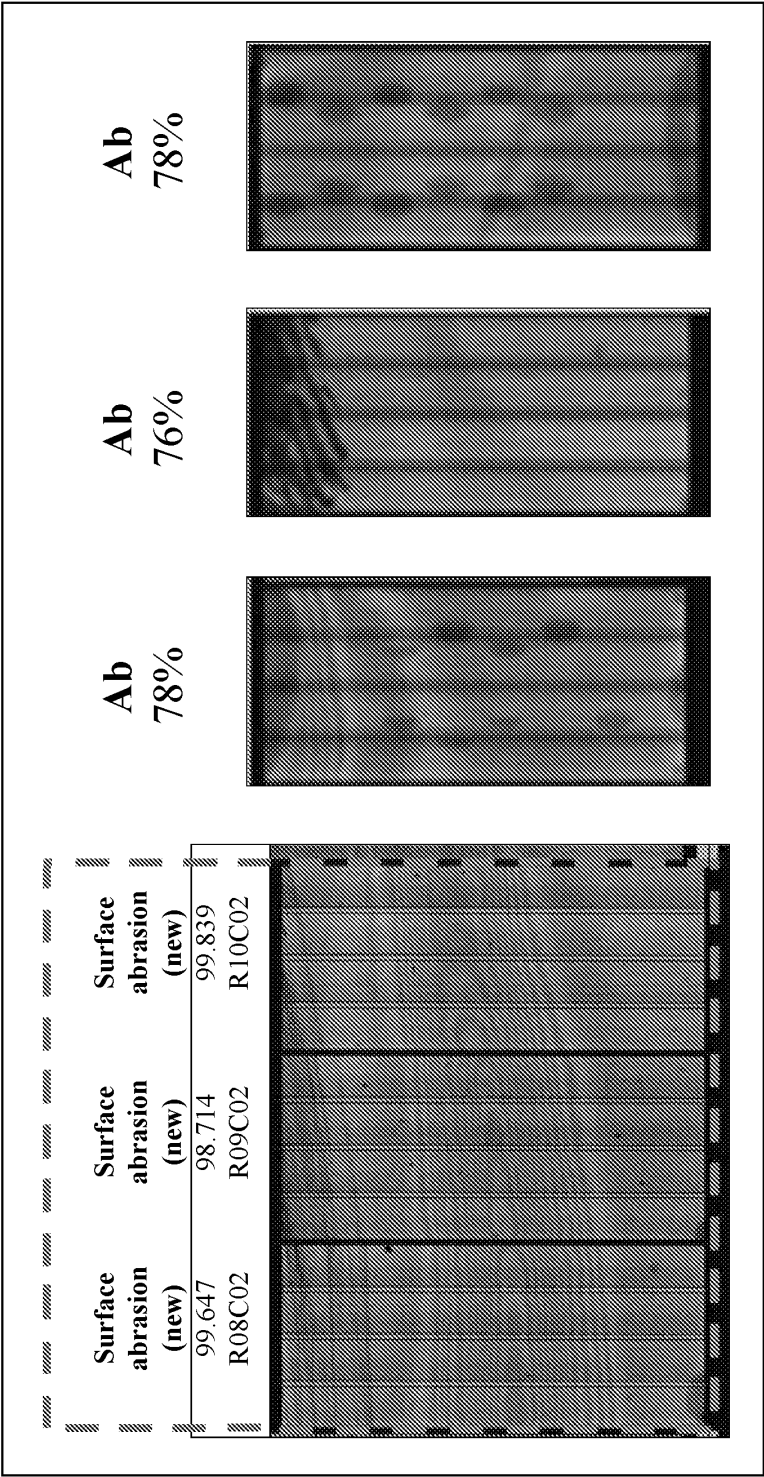


FIG. 5F

[illegible]

FIG. 5G

545

Bad Process Cycle Images – Reagent Flow Failure (2/2)

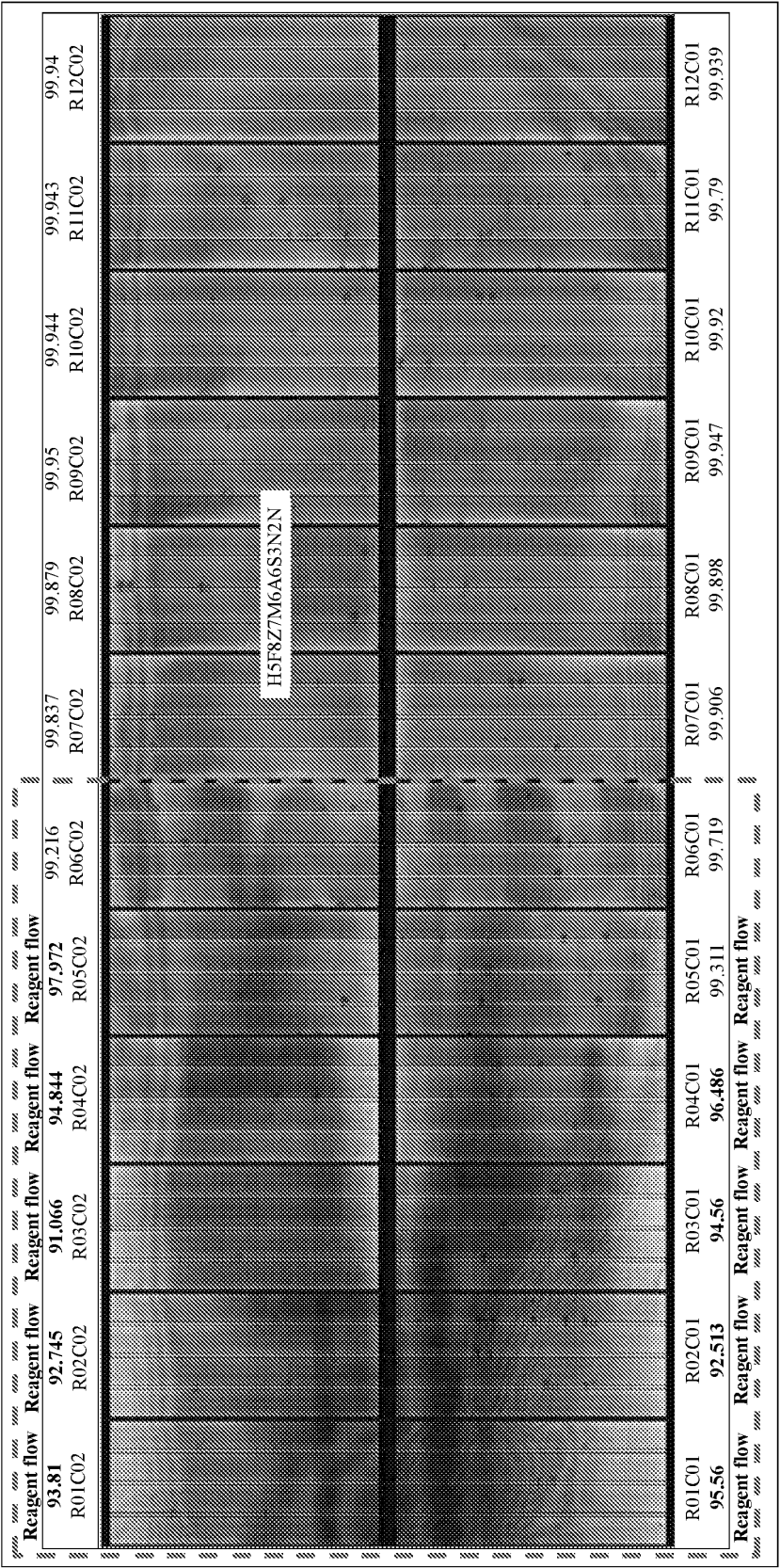
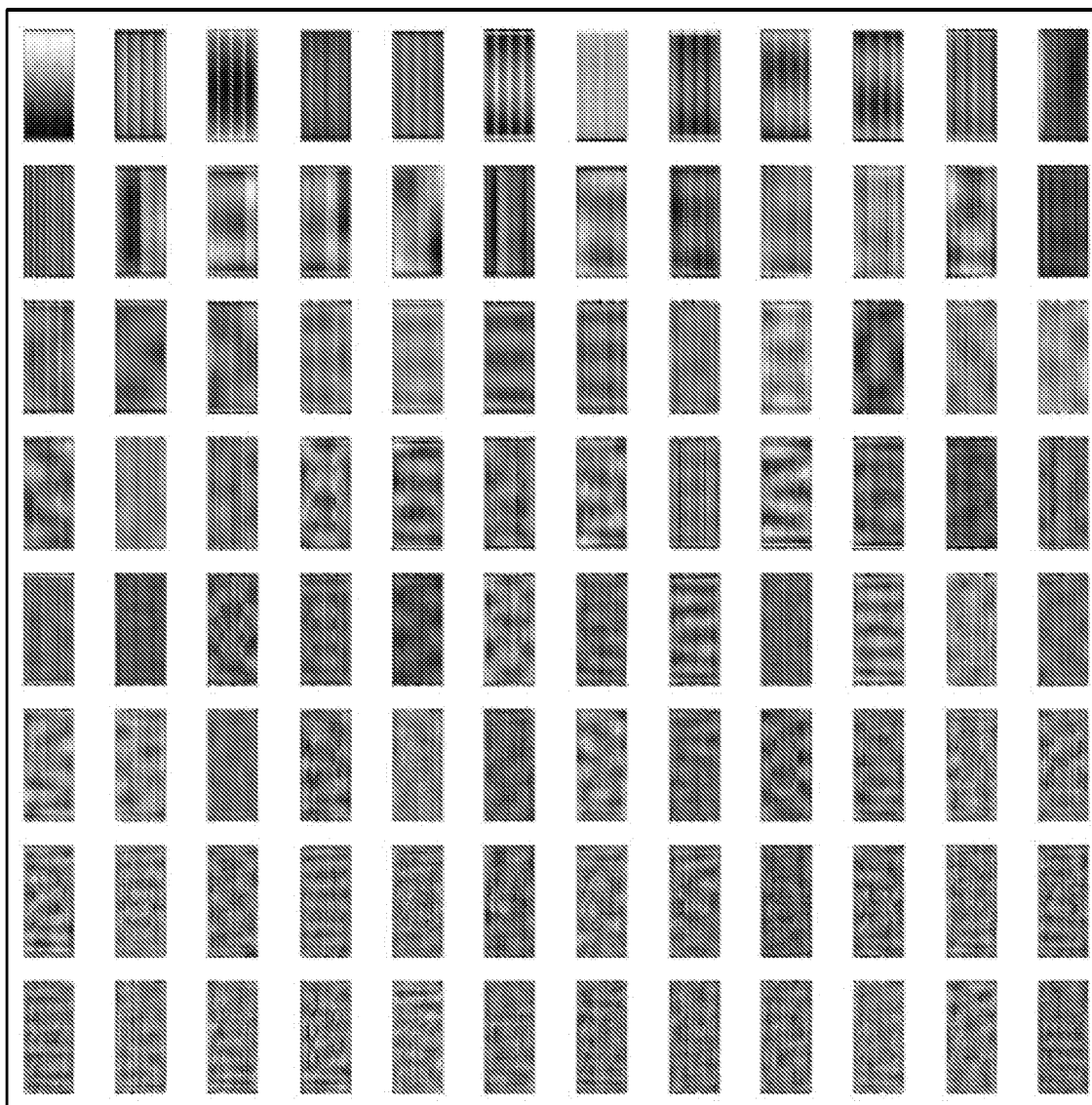
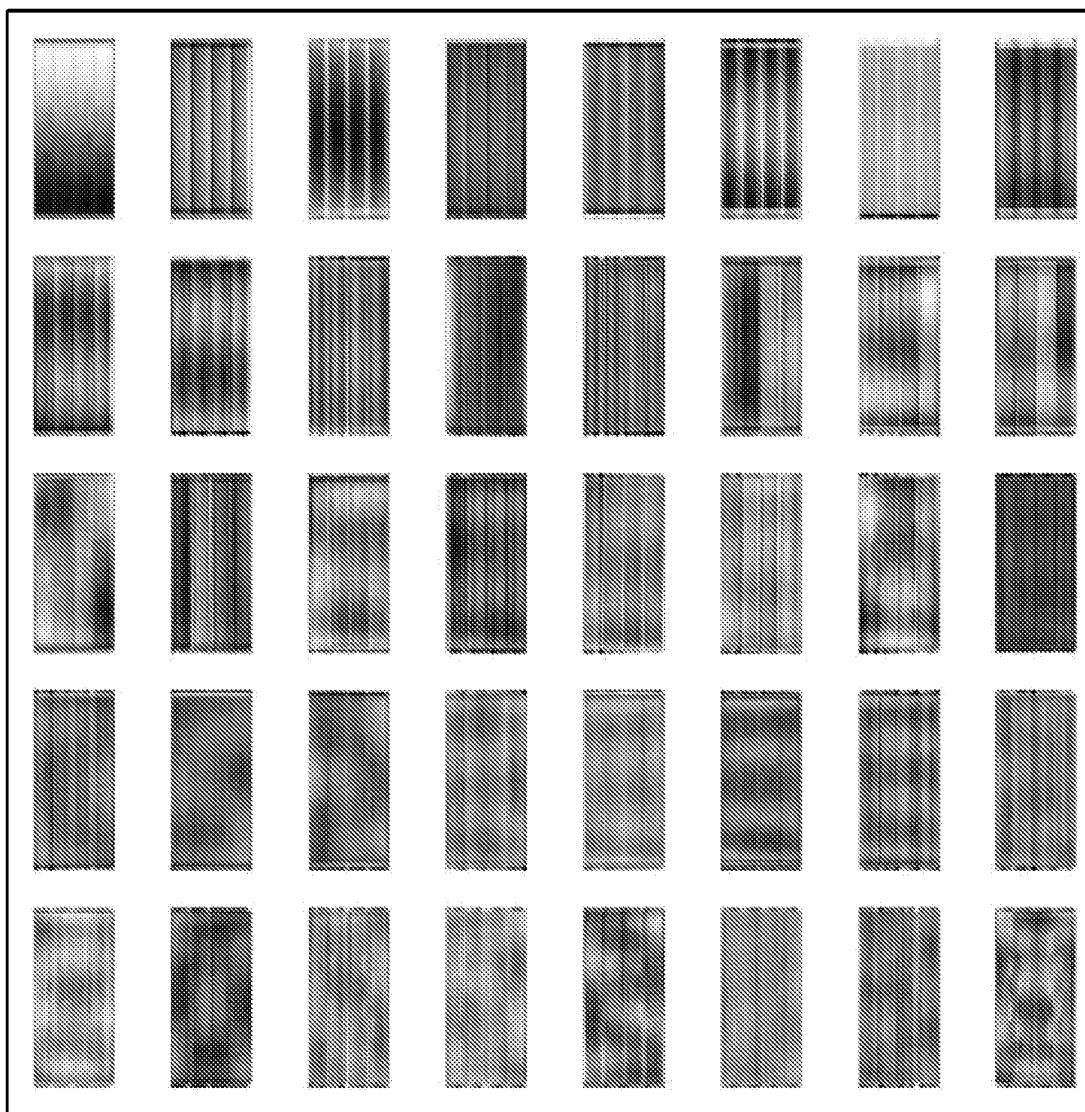


FIG. 5H

Principal Component Analysis – Example of 96 Eigen Images**FIG. 6A**

Principal Component Analysis – Top 40 Eigen Images**FIG. 6B**

Dimensionality Reduction and Creating a Basis of Eigen Images (1/2)

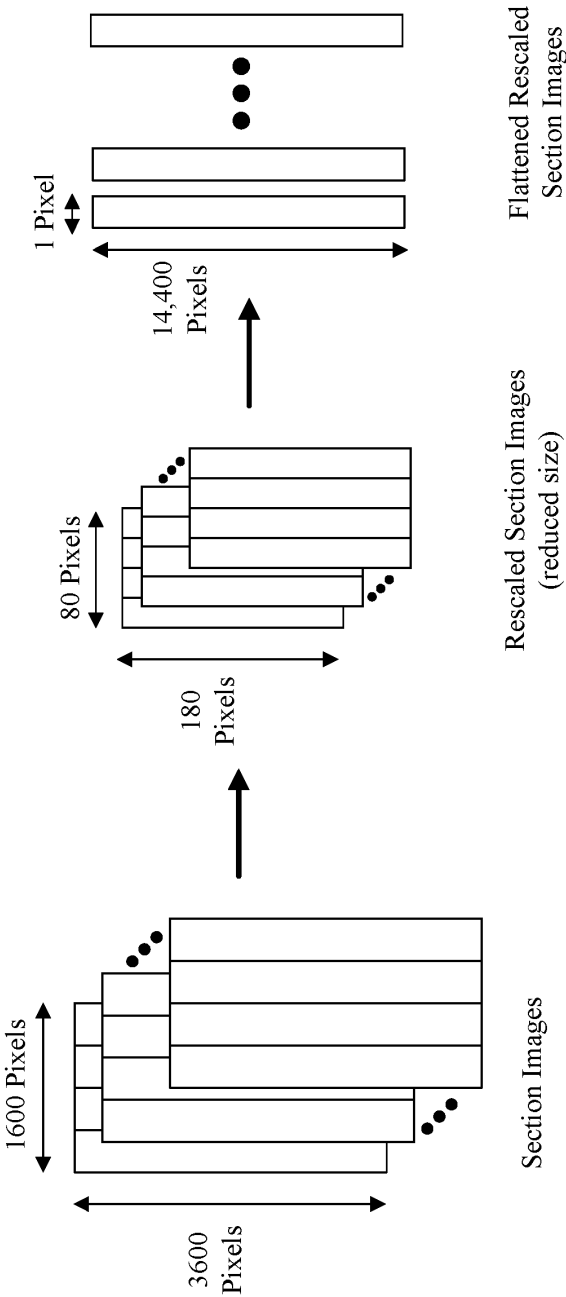


FIG. 7A

Dimensionality Reduction and Creating a Basis of Eigen Images (2/2)

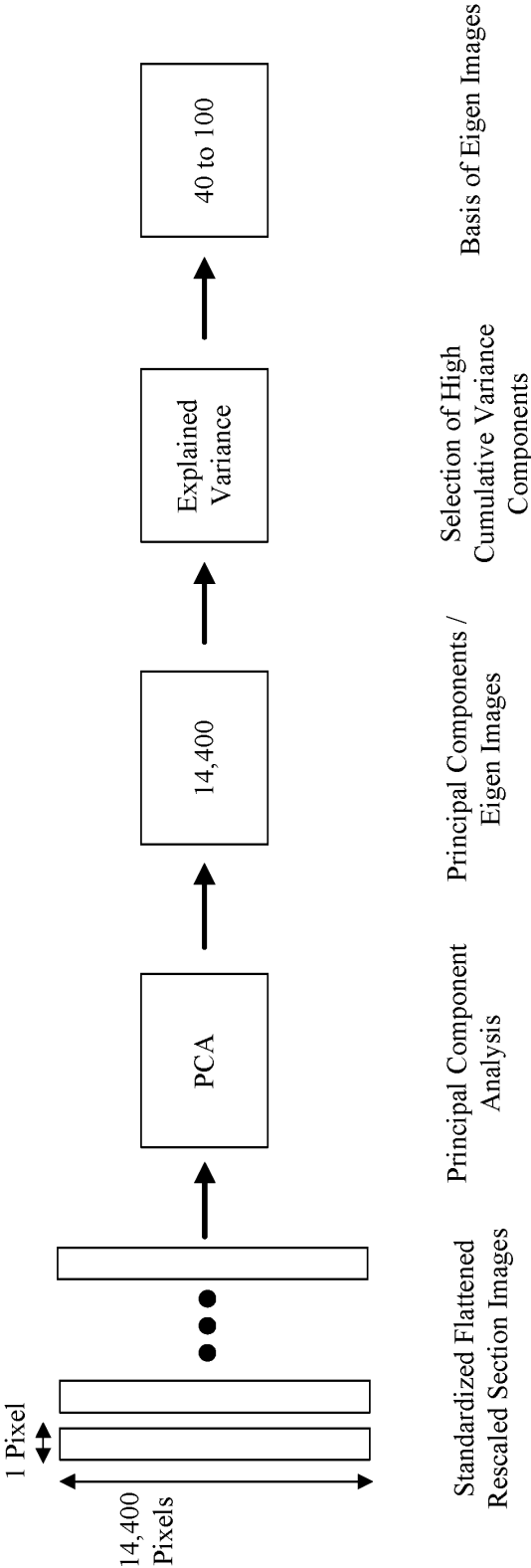


FIG. 7B

810

Feature Generation using Intensities of Areas of Process Cycle Images

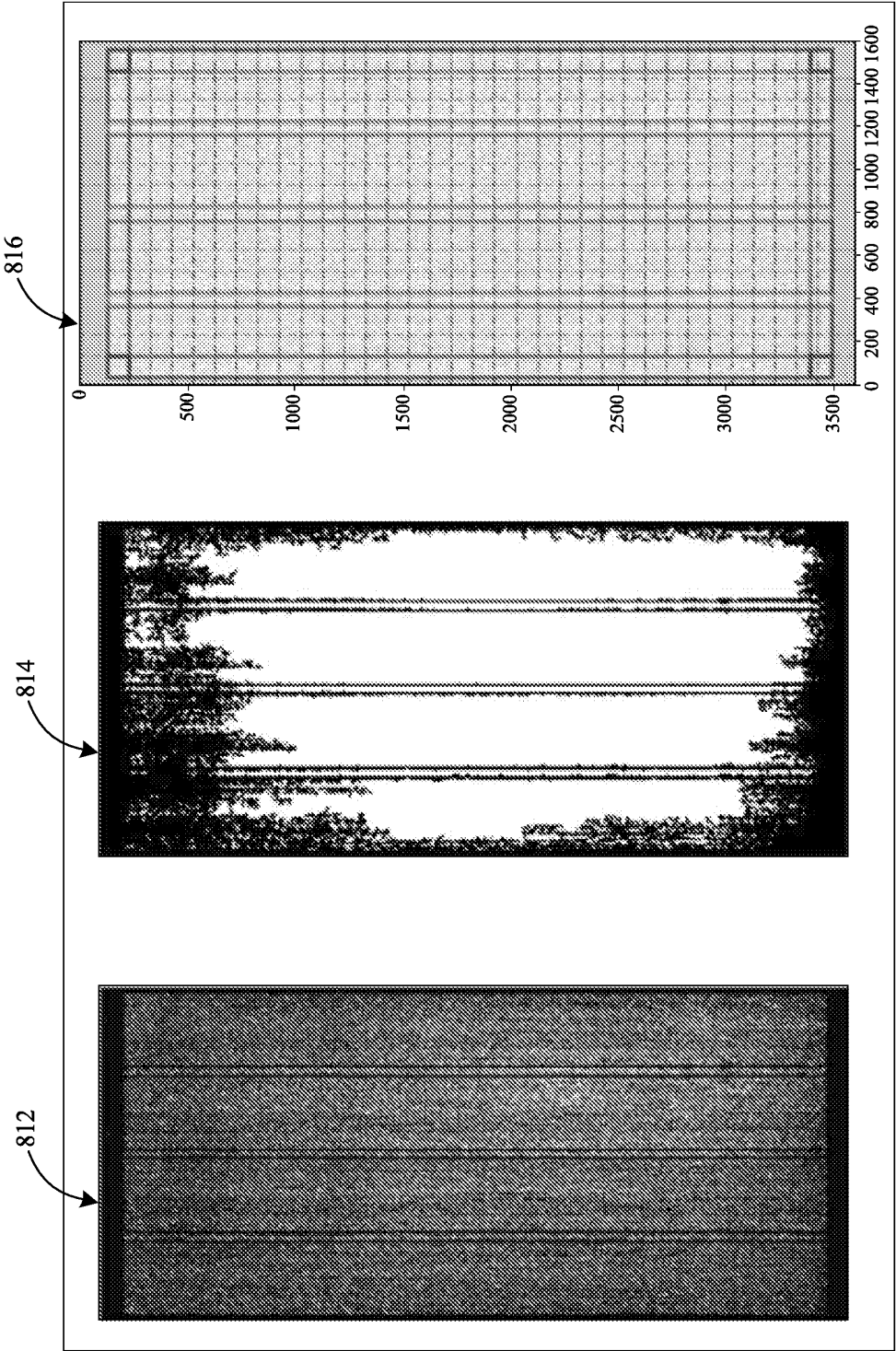


FIG. 8A

820

Feature Generation using Intensities of Areas of Process Cycle Images

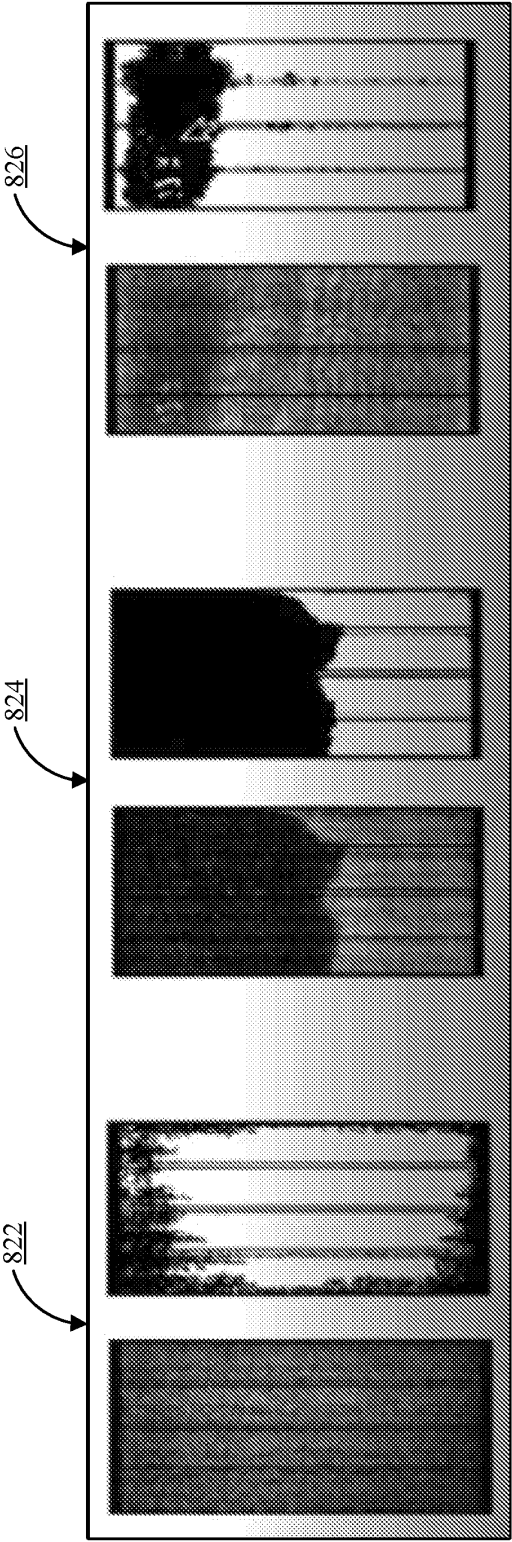
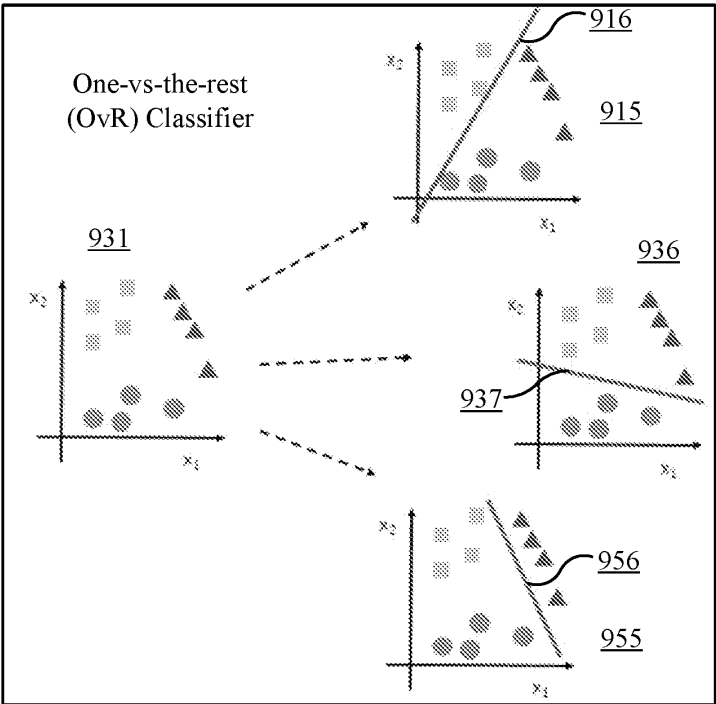


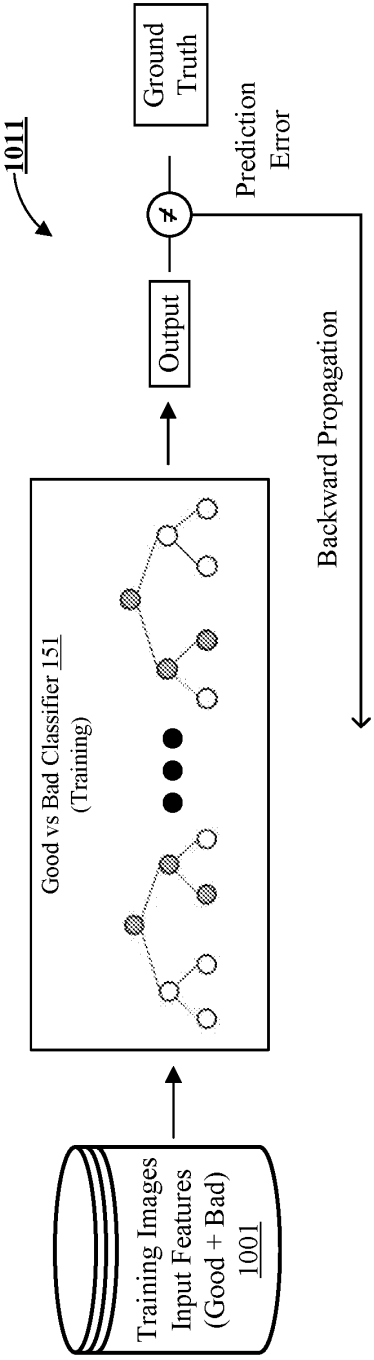
FIG. 8B



(Prior art)

FIG. 9

1000



1061

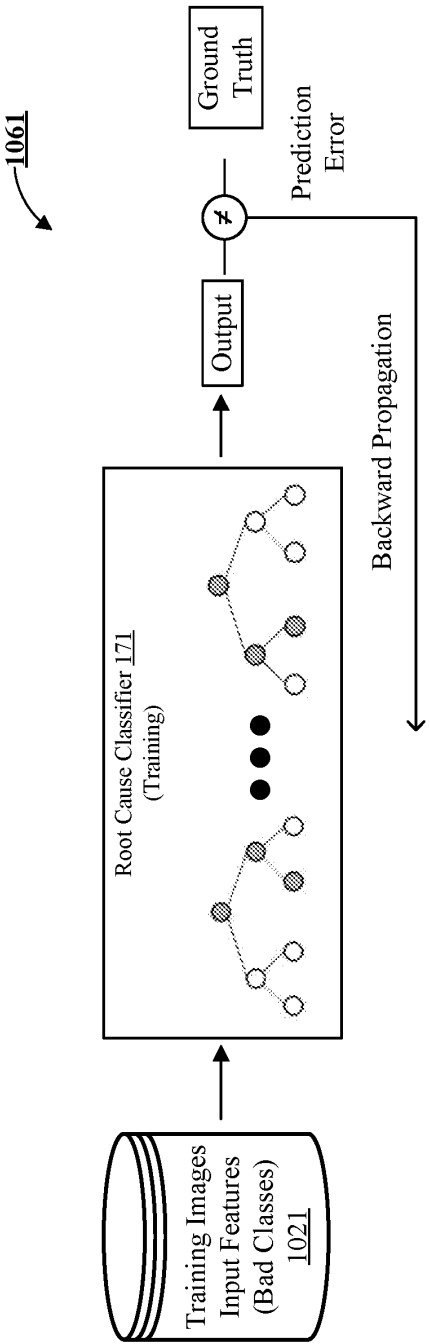


FIG. 10A

1080

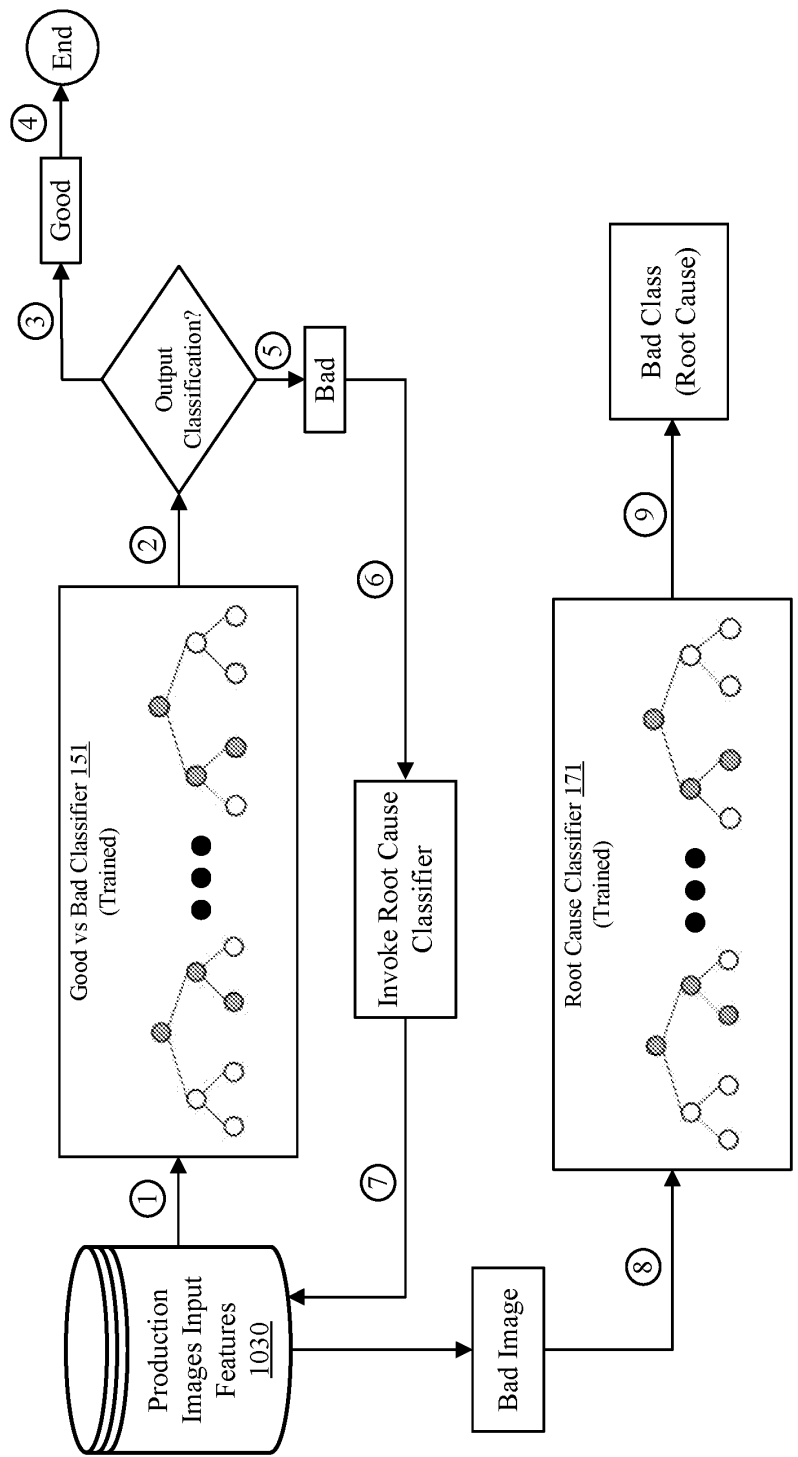


FIG. 10B

1100

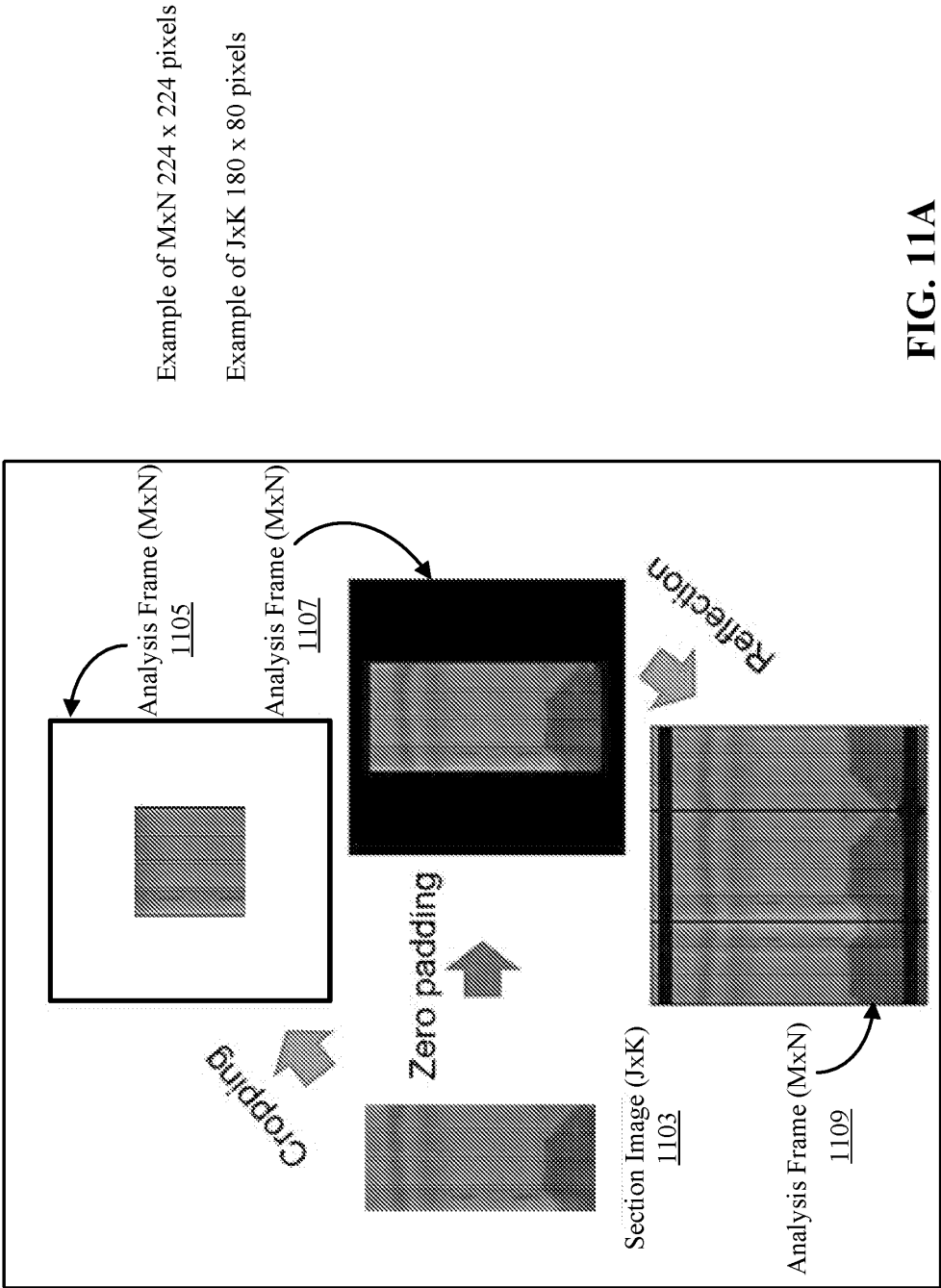


FIG. 11A

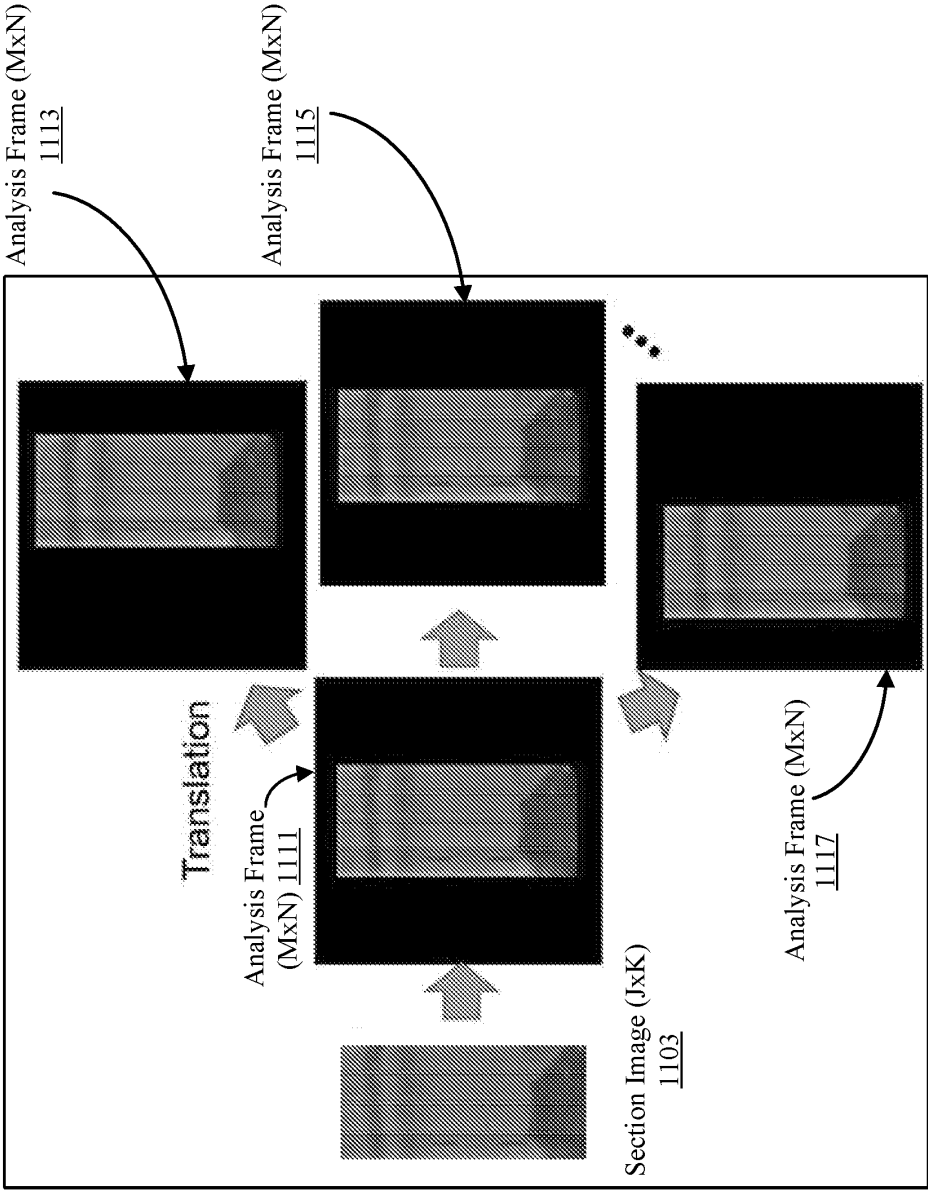


FIG. 11B

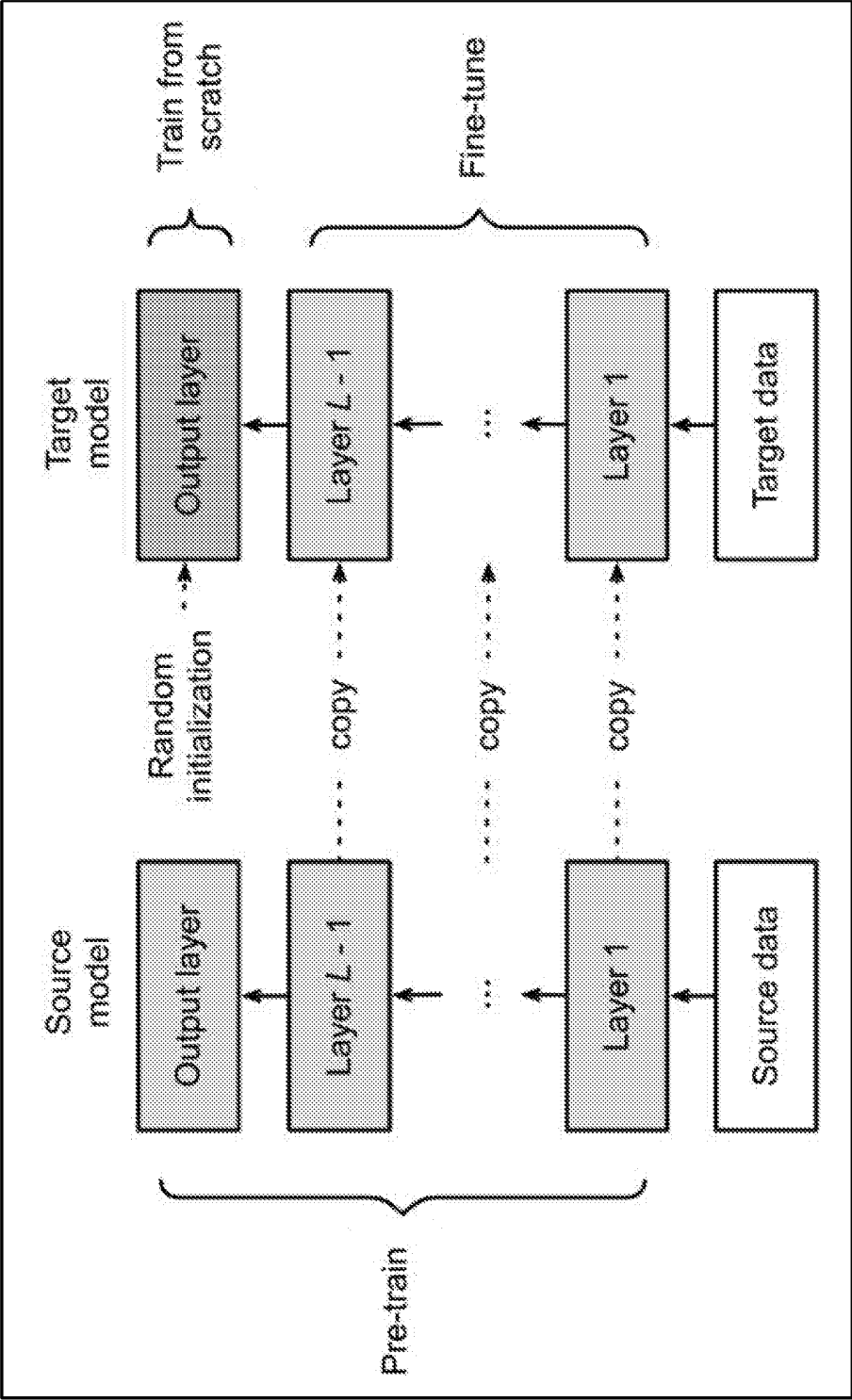


FIG. 12A

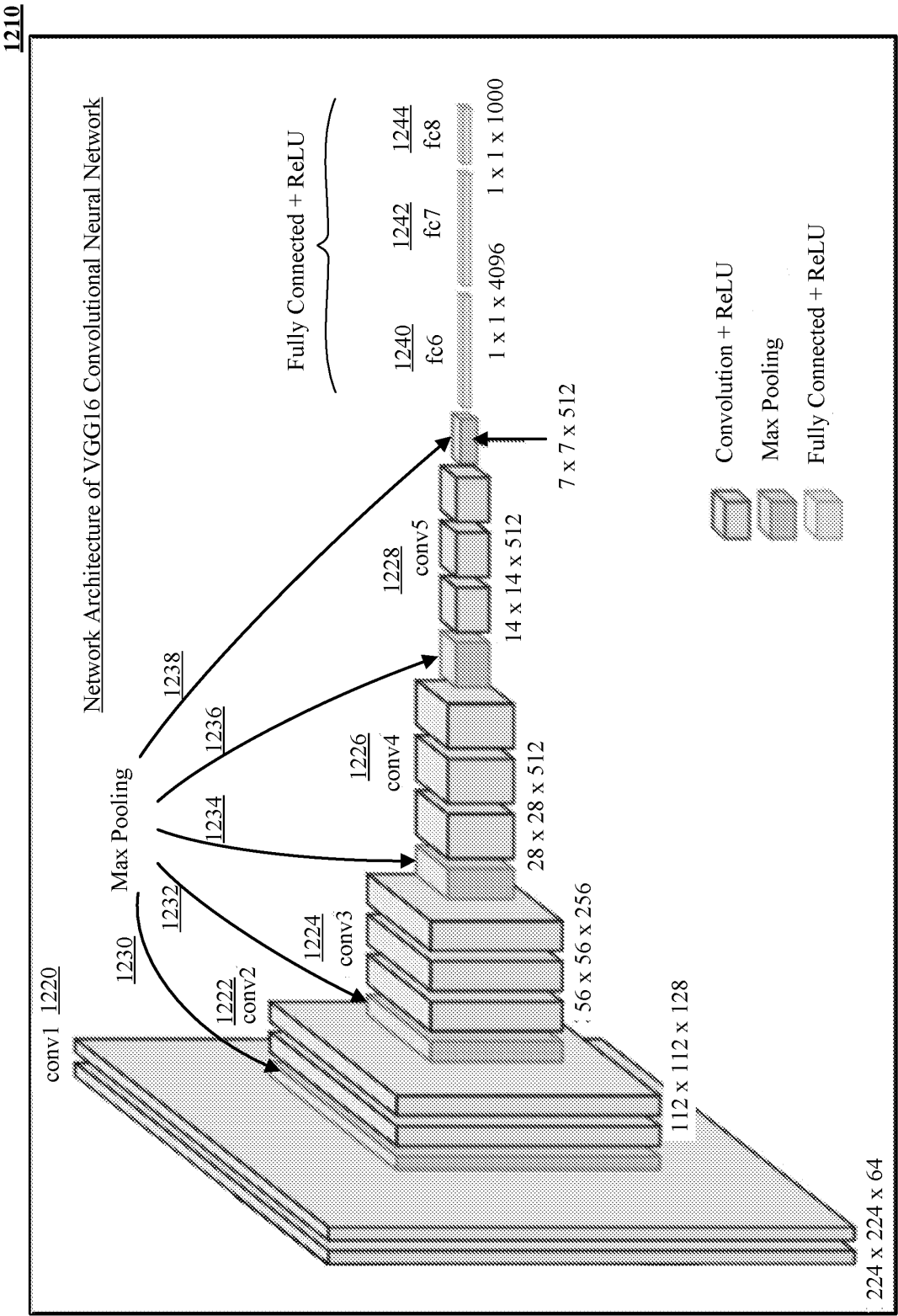


FIG. 12B

```

VGG(
  (features): Sequential(
conv 1 { (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
1220   (1): ReLU(inplace=True)
        (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (3): ReLU(inplace=True)
        (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
conv 2 { (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
1222   (6): ReLU(inplace=True)
        (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (8): ReLU(inplace=True)
        (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
conv 3 { (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
1224   (11): ReLU(inplace=True)
        (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (13): ReLU(inplace=True)
        (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (15): ReLU(inplace=True)
        (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
conv 4 { (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
1226   (18): ReLU(inplace=True)
        (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (20): ReLU(inplace=True)
        (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (22): ReLU(inplace=True)
        (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
conv 5 { (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
1228   (25): ReLU(inplace=True)
        (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (27): ReLU(inplace=True)
        (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (29): ReLU(inplace=True)
        (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
        )
    (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
    (classifier): Sequential(
fc 6 { (0): Linear(in_features=25088, out_features=4096, bias=True)
1240   (1): ReLU(inplace=True)
        (2): Dropout(p=0.5, inplace=False)
fc 7 { (3): Linear(in_features=4096, out_features=4096, bias=True)
1242   (4): ReLU(inplace=True)
        (5): Dropout(p=0.5, inplace=False)
fc 8 { (6): Linear(in_features=4096, out_features=2, bias=True)
1244   )
        )
  )

```

ResNet-18 Architecture (1/2)1280

```

ResNet( 1259
(conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
(bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(rel): ReLU(inplace=True)
(maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
(layer1): Sequential( 1260
(0): BasicBlock(
(conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(rel): ReLU(inplace=True)
(conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
(1): BasicBlock(
(conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(rel): ReLU(inplace=True)
(conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
(layer2): Sequential( 1262
(0): BasicBlock(
(conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(rel): ReLU(inplace=True)
(conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(downsample): Sequential(
(0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
(1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
(1): BasicBlock(
(conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(rel): ReLU(inplace=True)
(conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
)
)
)

```

FIG. 12D

SUBSTITUTE SHEET (RULE 26)

1264ResNet-18 Architecture (2/2)

```

(layer3): Sequential(
  (0): BasicBlock(
    (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (downsample): Sequential(
      (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (1): BasicBlock(
    (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
(layer4): Sequential(
  (0): BasicBlock(
    (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (downsample): Sequential(
      (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
      (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (1): BasicBlock(
    (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
    (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  )
)
  (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
  (fc): Linear(in_features=512, out_features=2, bias=True)
)

```

1270**FIG. 12E**

SUBSTITUTE SHEET (RULE 26)

1300

Performance Comparison

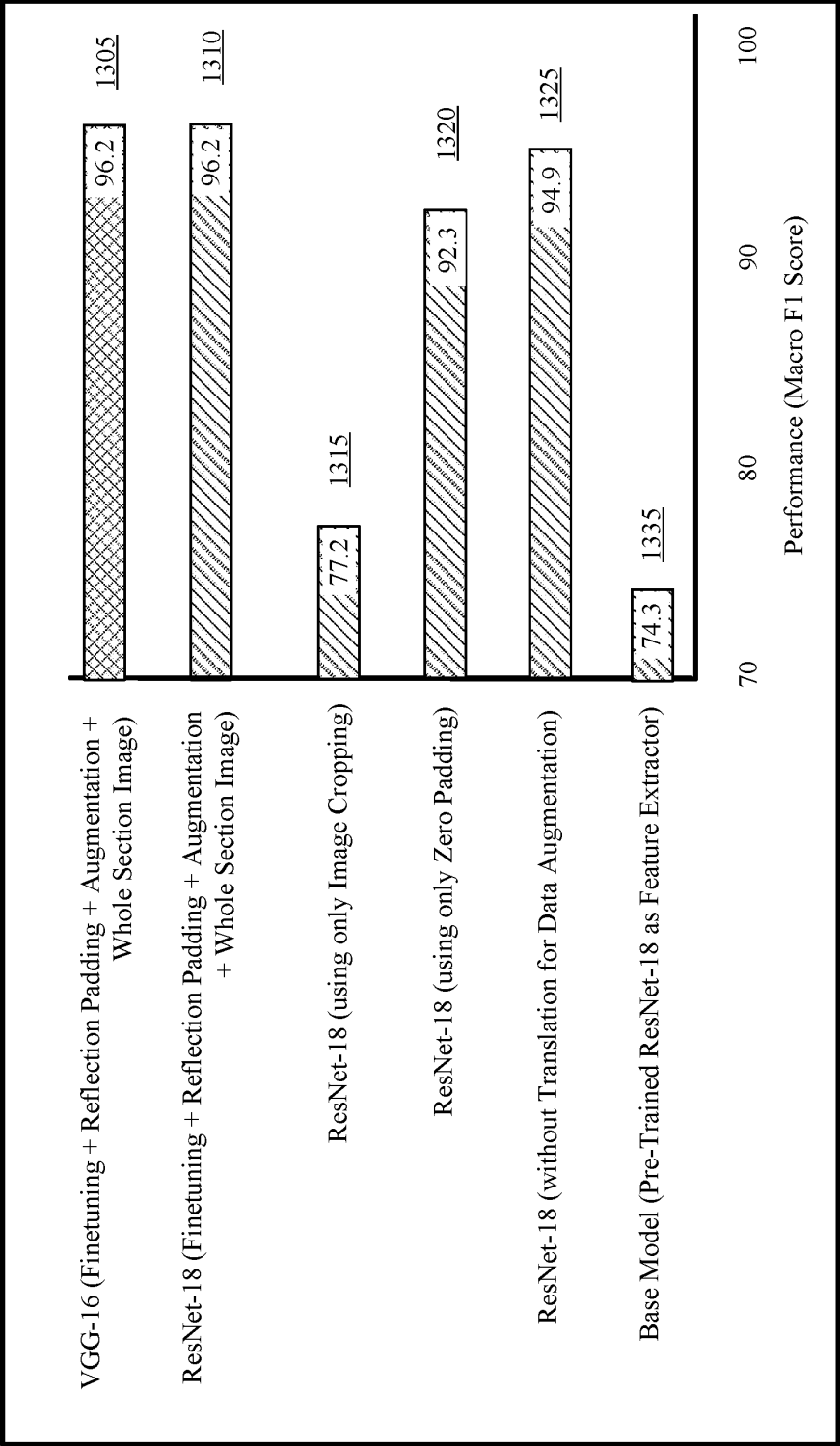


FIG. 13

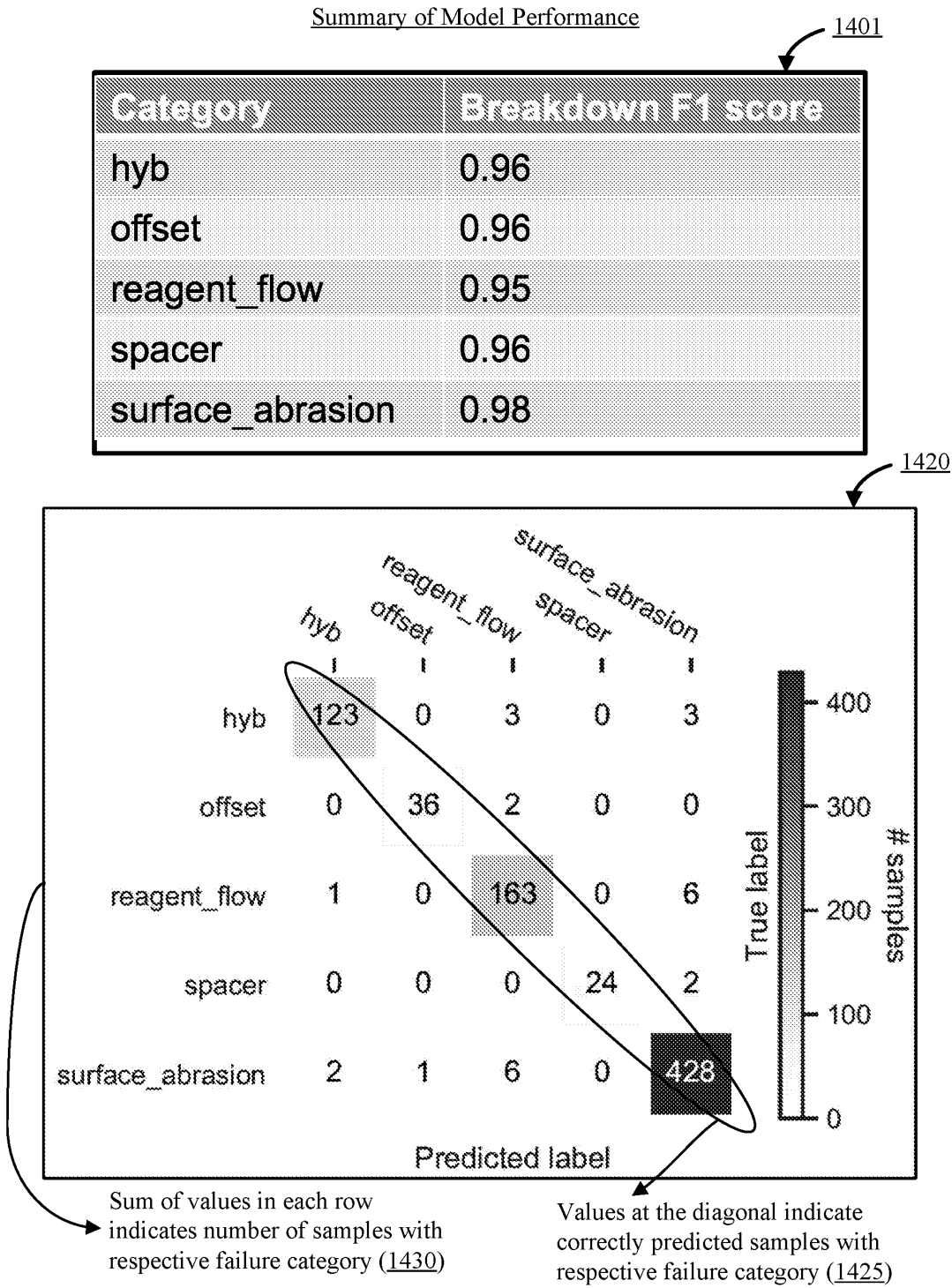


FIG. 14

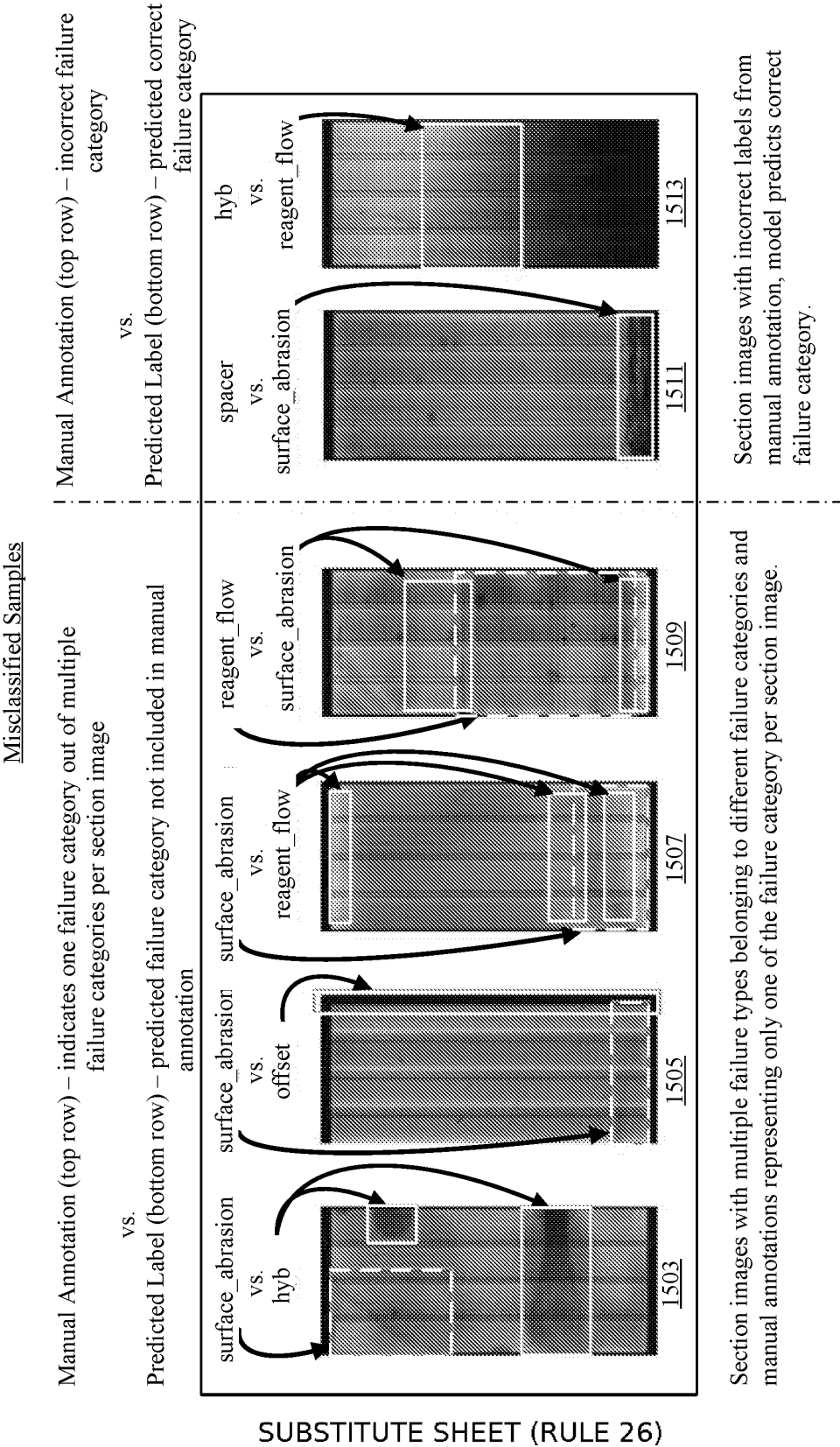


FIG. 15

1601

Process Flow For Training
Good vs. Bad Classifier and Applying Trained Model

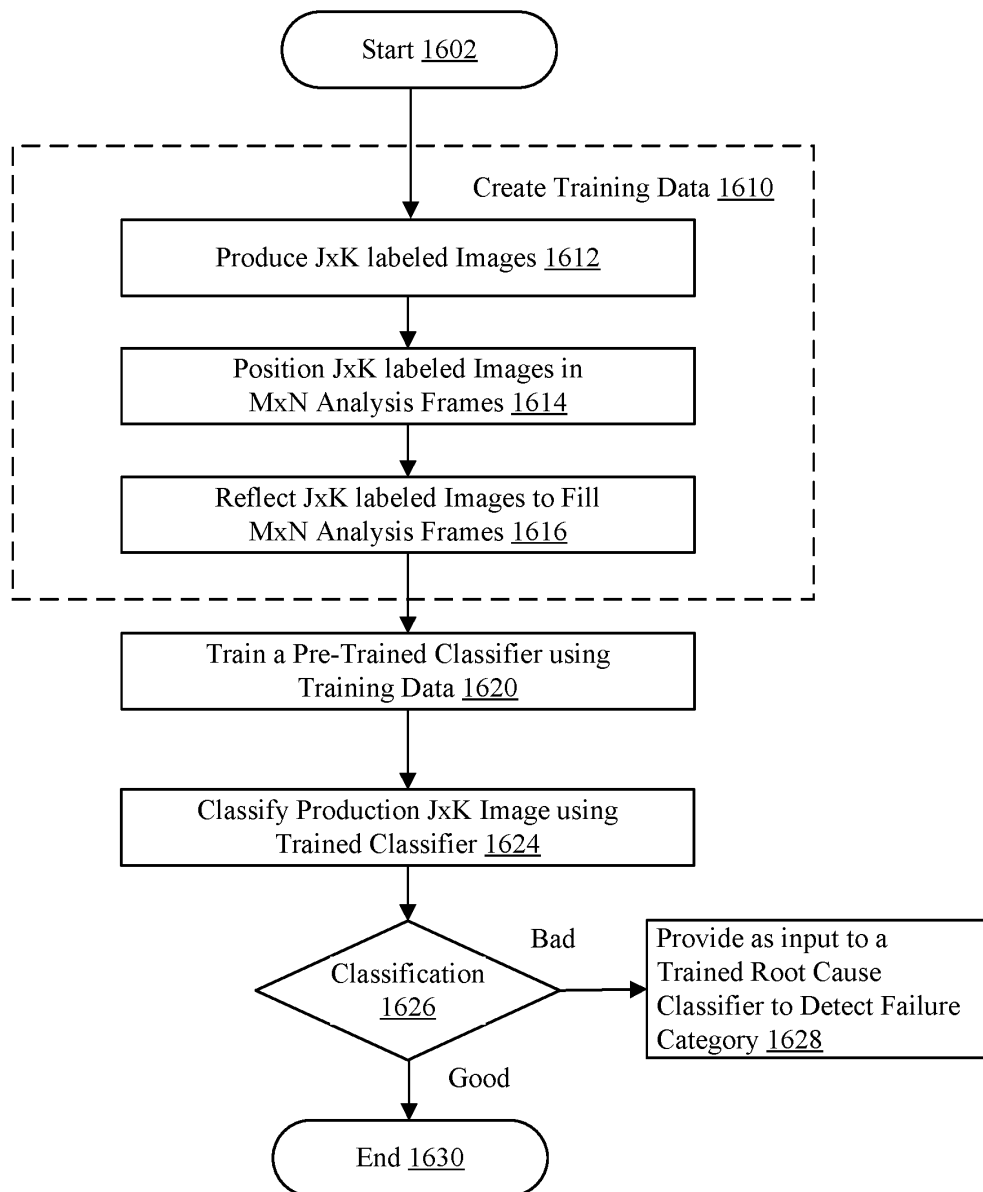


FIG. 16

1700

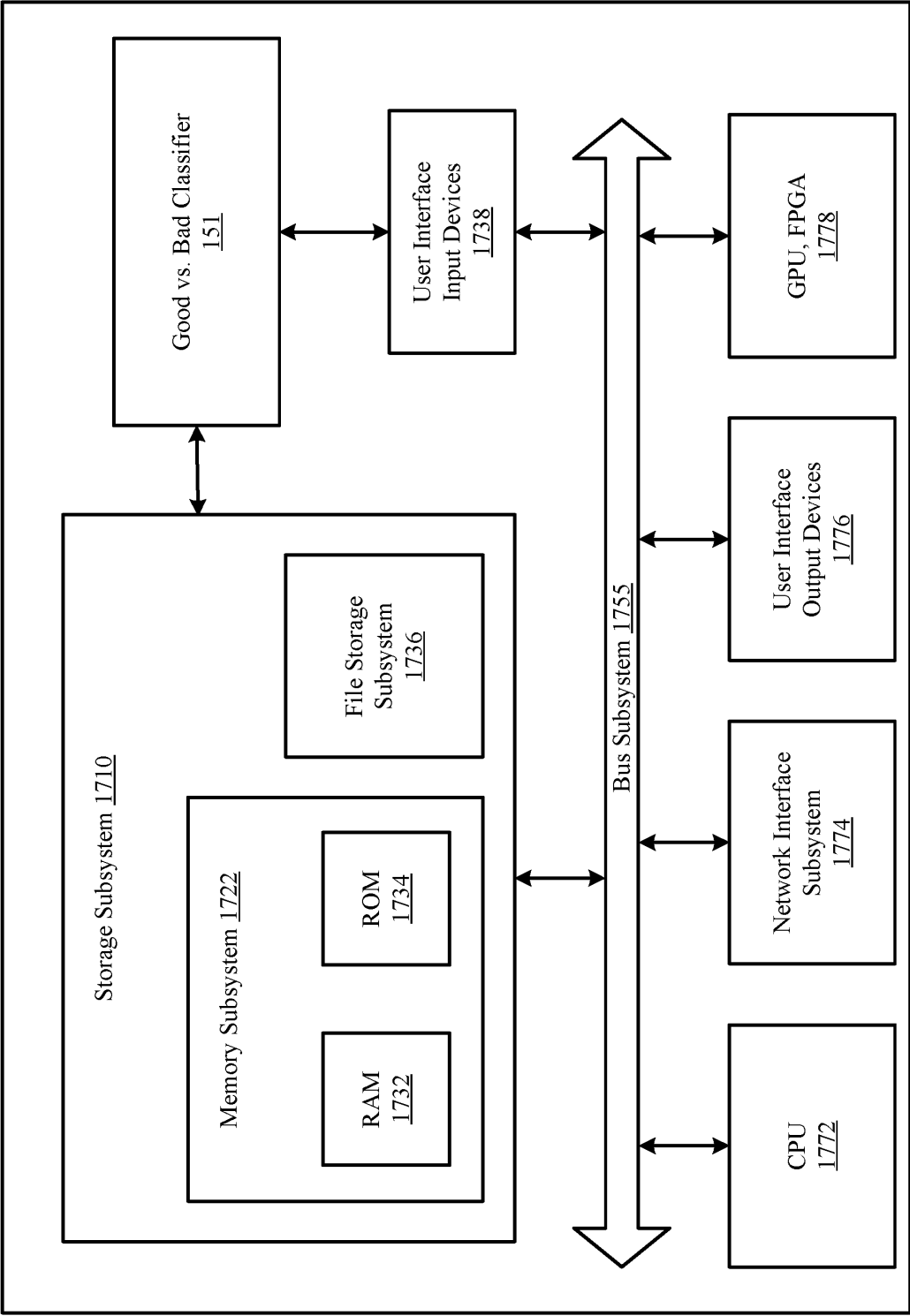


FIG. 17