New CA Classifiers (random colours)

Wolfram Classes of ECAs

In[1]:=

```
CAclasses = {1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2,
   2, 2, 3, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 4, 2, 2,
   2, 2, 2, 3, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
   3, 2, 2, 3, 3, 2, 2, 2, 2, 2, 1, 3, 2, 2, 2, 3, 3, 2, 2, 3, 3, 3, 2, 2, 4, 2, 2, 2, 2,
   2, 2, 2, 2, 2, 3, 3, 3, 2, 4, 2, 3, 2, 1, 3, 2, 2, 2, 2, 2, 3, 1, 4, 2, 2, 2, 2, 2, 2,
   2, 2, 3, 4, 2, 3, 3, 3, 2, 3, 2, 2, 2, 2, 2, 2, 1, 3, 2, 2, 2, 3, 2, 2, 1, 3, 2, 2,
   2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 1, 4, 2, 3, 2, 2, 2, 2,
   2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 2,
   2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1};
```

Functions for creating net and random datasets (ECAs, all 4 classes)

In[2]:=

```
RandomRuleC[n_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[n, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules → {0 -> RandomColor[], 1 -> RandomColor[],
     3 -> RandomColor[], 4 → RandomColor[], 5 -> RandomColor[],
     6 -> RandomColor[], 7 -> RandomColor[]}, Frame → False]]
netC[W_Integer, H_Integer] := NetInitialize@NetChain[
   {ConvolutionLayer[16, {2, 3}], Ramp, PoolingLayer[{H, W} - {1, 2}], FlattenLayer[],
    LinearLayer[256], SoftmaxLayer[]}, "Input" → NetEncoder[{"Image", {W, H}}],
   "Output" → NetDecoder[{"Class", Range[0, 255]}]]
netTwoCC[W_Integer, H_Integer] := NetInitialize@
  NetChain[<|"conv1" → ConvolutionLayer[16, {2, 3}], "ramp1" → Ramp,
    "conv3" → ConvolutionLayer[16, {2, 3}], "ramp2" → Ramp,
    "pooling" → PoolingLayer[{H, W} - {2, 4}], "flatten" → FlattenLayer[],
    "linear" → 512, "linear2" → 4, "softmax" → SoftmaxLayer[]|>,
   "Input" → NetEncoder[{"Image", {W, H}}], "Output" → NetDecoder[{"Class", Range[1, 4]}]]
dataC[W_Integer, H_Integer, n_Integer] :=
 Table[RandomRuleC[i, W, H] → CAclasses[[i + 1]], {i, RandomChoice[Range[0, 255], n]}]
```

In[6]:=
```
netThreeCC[W_Integer, H_Integer] :=
 NetInitialize@NetChain[<|"conv1" → ConvolutionLayer[16, {2, 3}],
    "ramp1" → Ramp, "conv2" → ConvolutionLayer[16, {2, 3}],
    "ramp2" → Ramp, "conv3" → ConvolutionLayer[16, {2, 3}], "ramp3" → Ramp,
    "pooling" → PoolingLayer[{H, W} - {4, 8}], "flatten" → FlattenLayer[],
    "linear" → 512, "linear2" → 4, "softmax" → SoftmaxLayer[]|>,
   "Input" → NetEncoder[{"Image", {W, H}}], "Output" → NetDecoder[{"Class", Range[1, 4]}]]
```

In[7]:=
```
netThreeCC1024[W_Integer, H_Integer] :=
 NetInitialize@NetChain[<|"conv1" → ConvolutionLayer[16, {2, 3}],
    "ramp1" → Ramp, "conv2" → ConvolutionLayer[16, {2, 3}],
    "ramp2" → Ramp, "conv3" → ConvolutionLayer[16, {2, 3}], "ramp3" → Ramp,
    "pooling" → PoolingLayer[{H, W} - {4, 8}], "flatten" → FlattenLayer[],
    "linear" → 1024, "linear2" → 4, "softmax" → SoftmaxLayer[]|>,
   "Input" → NetEncoder[{"Image", {W, H}}], "Output" → NetDecoder[{"Class", Range[1, 4]}]]
```

In[8]:=
```
netFourCC512[W_Integer, H_Integer] :=
 NetInitialize@NetChain[<|"conv1" → ConvolutionLayer[32, {2, 3}],
    "ramp1" → Ramp, "conv3" → ConvolutionLayer[32, {2, 3}], "ramp2" → Ramp,
    "pooling" → PoolingLayer[{H, W} - {2, 4}], "flatten" → FlattenLayer[],
    "linear" → 512, "linear2" → 4, "softmax" → SoftmaxLayer[]|>,
   "Input" → NetEncoder[{"Image", {W, H}}], "Output" → NetDecoder[{"Class", Range[1, 4]}]]
```

In[9]:=
```
netFiveCC512[W_Integer, H_Integer] := NetInitialize@
  NetChain[<|"conv1" → ConvolutionLayer[32, {2, 3}], "bat1" → BatchNormalizationLayer[],
    "ramp1" → Ramp, "conv3" → ConvolutionLayer[32, {2, 3}],
    "bat2" → BatchNormalizationLayer[], "ramp2" → Ramp,
    "pooling" → PoolingLayer[{H, W} - {2, 4}], "flatten" → FlattenLayer[],
    "linear" → 512, "linear2" → 4, "softmax" → SoftmaxLayer[]|>,
   "Input" → NetEncoder[{"Image", {W, H}}], "Output" → NetDecoder[{"Class", Range[1, 4]}]]
```

In[10]:=
```
netSixCC512drop[W_Integer, H_Integer] := NetInitialize@
  NetChain[<|"drop1" → DropoutLayer[0.2], "conv1" → ConvolutionLayer[32, {3, 3}],
    "bat1" → BatchNormalizationLayer[], "ramp1" → Ramp,
    "conv3" → ConvolutionLayer[32, {3, 3}], "bat2" → BatchNormalizationLayer[],
    "ramp2" → Ramp, "pooling" → PoolingLayer[{H, W} - {4, 8}],
    "flatten" → FlattenLayer[], "linear" → 512, "drop2" → DropoutLayer[0.2],
    "linear2" → 4, "softmax" → SoftmaxLayer[]|>,
   "Input" → NetEncoder[{"Image", {W, H}}], "Output" → NetDecoder[{"Class", Range[1, 4]}]]
```

In[11]:=
```
netSevenCC512drop[W_Integer, H_Integer] :=
 NetInitialize@NetChain[<|"conv1" → ConvolutionLayer[24, {3, 3}],
    "bat1" → BatchNormalizationLayer[], "ramp1" → Ramp,
    "conv3" → ConvolutionLayer[24, {3, 3}], "bat2" → BatchNormalizationLayer[],
    "ramp2" → Ramp, "pooling" → PoolingLayer[{H, W} - {4, 8}],
    "flatten" → FlattenLayer[], "linear" → 512, "drop2" → DropoutLayer[0.2],
    "linear2" → 4, "softmax" → SoftmaxLayer[]|>,
   "Input" → NetEncoder[{"Image", {W, H}}], "Output" → NetDecoder[{"Class", Range[1, 4]}]]
```

In[12]:=
```
netEightCC512drop[W_Integer, H_Integer] :=
 NetInitialize@NetChain[<|"conv1" → ConvolutionLayer[24, {3, 3}],
    "bat1" → BatchNormalizationLayer[], "ramp1" → Ramp,
    "conv2" → ConvolutionLayer[16, {2, 3}], "bat2" → BatchNormalizationLayer[],
    "ramp2" → Ramp, "conv3" → ConvolutionLayer[24, {3, 3}],
    "bat3" → BatchNormalizationLayer[], "ramp3" → Ramp, "pooling" →
     PoolingLayer[{H, W} - {8, 16}], "flatten" → FlattenLayer[], "linear" → 1024,
    "drop2" → DropoutLayer[0.2], "linear2" → 4, "softmax" → SoftmaxLayer[]|>,
   "Input" → NetEncoder[{"Image", {W, H}}], "Output" → NetDecoder[{"Class", Range[1, 4]}]]
```

In[13]:=
```
netNineCC512drop[W_Integer, H_Integer] :=
 NetInitialize@NetChain[<|"conv1" → ConvolutionLayer[24, {3, 3}],
    "bat1" → BatchNormalizationLayer[], "ramp1" → Ramp,
    "conv2" → ConvolutionLayer[24, {3, 3}], "bat2" → BatchNormalizationLayer[],
    "ramp2" → Ramp, "conv3" → ConvolutionLayer[24, {3, 3}],
    "bat3" → BatchNormalizationLayer[], "ramp3" → Ramp, "pooling" →
     PoolingLayer[{H, W} - {12, 12}], "flatten" → FlattenLayer[], "linear" → 512,
    "drop2" → DropoutLayer[0.2], "linear2" → 4, "softmax" → SoftmaxLayer[]|>,
   "Input" → NetEncoder[{"Image", {W, H}}], "Output" → NetDecoder[{"Class", Range[1, 4]}]]
```

In[180]:=
```
netTenCC1024drop[W_Integer, H_Integer] :=
 NetInitialize@NetChain[<|"conv1" → ConvolutionLayer[24, {3, 3}],
    "bat1" → BatchNormalizationLayer[], "ramp1" → Ramp,
    "conv2" → ConvolutionLayer[24, {3, 3}], "bat2" → BatchNormalizationLayer[],
    "ramp2" → Ramp, "conv3" → ConvolutionLayer[24, {3, 3}],
    "bat3" → BatchNormalizationLayer[], "ramp3" → Ramp,
    "conv4" → ConvolutionLayer[24, {3, 3}], "bat4" → BatchNormalizationLayer[],
    "ramp4" → Ramp, "pooling" → PoolingLayer[{H, W} - {12, 12}],
    "flatten" → FlattenLayer[], "linear" → 1024,
    "drop2" → DropoutLayer[0.3], "linear2" → 4, "softmax" → SoftmaxLayer[]|>,
   "Input" → NetEncoder[{"Image", {W, H}}], "Output" → NetDecoder[{"Class", Range[1, 4]}]]
```

Functions for creating datasets (1D totalistic CAs)

k=3, r=1 totalistic (class 4 only)

In[14]:=
```
gen3TC[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {3, 1}}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
data3T2C[W_Integer, H_Integer, n_Integer] := Table[gen3TC[i, W, H] → 4,
  {i, RandomChoice[{1635, 1815, 2007, 2043, 2049, 1388, 1041}, n]}]
```

k=4, r=1 totalistic (class 4 only, 1 example)

In[16]:=
```
gen4TC[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {4, 1}}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
data4TC[W_Integer, H_Integer, n_Integer] := Table[gen4TC[1 004 600, W, H] → 4, n]
```

k=2, r=2 totalistic (all 4 classes)

In[18]:=
```
gen2r2C[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {2, 1}, 2}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
data2r2c4C[W_Integer, H_Integer, n_Integer] :=
 Table[gen2r2C[i, W, H] → 4, {i, RandomChoice[{20, 52}, n]}]
data2r2c3C[W_Integer, H_Integer, n_Integer] := Table[gen2r2C[i, W, H] → 3,
  {i, RandomChoice[{2, 6, 10, 12, 14, 18, 22, 26, 28, 30, 34, 38, 42, 44, 46, 50}, n]}]
data2r2c2C[W_Integer, H_Integer, n_Integer] :=
 Table[gen2r2C[i, W, H] → 2, {i, RandomChoice[{8, 24, 56}, n]}]
data2r2c1C[W_Integer, H_Integer, n_Integer] := Table[gen2r2C[i, W, H] → 1,
  {i, RandomChoice[{0, 4, 16, 32, 36, 40, 48, 54, 58, 60, 62}, n]}]
genData2r2C[W_Integer, H_Integer, n_Integer] :=
 Join[data2r2c4C[W, H, n], data2r2c3C[W, H, n], data2r2c2C[W, H, n], data2r2c1C[W, H, n]]
```

k=5, r=1 totalistic (class 4 only)

In[24]:=
```
gen5T4C[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {5, 1}}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
data5T4C[n_Integer, W_Integer, H_Integer] := Table[gen5T4C[i, W, H] → 4,
  {i, RandomChoice[{781 130 654, 772 514 435, 1 151 319 452, 309 095 787, 880 862 046,
     973 835 714, 779 446 817, 345 466 505, 535 500 975, 793 363 571, 1 052 373 865,
     455 984 785, 339 227 109, 1 050 973 846, 513 368 817, 91 315 820, 113 925 357}, n]}]
```

k=5, r=1 totalistic (classes 2/3/4)

In[26]:=
```
gen5TC[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {5, 1}, 1}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
data5T4CC[W_Integer, H_Integer, n_Integer] := Table[gen5TC[i, W, H] → 4,
  {i, RandomChoice[{644 218 533, 491 739 943, 6 889 640, 986 144 962, 1 099 816 682,
     988 971 204, 300 829 994, 272 622 024, 304 100 638, 626 595 633}, n]}]
data5T3CC[W_Integer, H_Integer, n_Integer] := Table[gen5TC[i, W, H] → 3,
  {i, RandomChoice[{889 082 395, 541 068 260, 807 907 479, 816 180 062, 650 485 139,
     643 827 745, 753 940 864, 871 525 323, 351 440 311, 83 501 460}, n]}]
data5T2CC[W_Integer, H_Integer, n_Integer] := Table[gen5TC[i, W, H] → 2,
  {i, RandomChoice[{525 735 659, 1 022 330 944, 1 007 796 739, 495 633 437, 1 036 827 943}, n]}]
genData5TCC[W_Integer, H_Integer, n_Integer] :=
 Join[data5T4CC[W, H, n], data5T3CC[W, H, n], data5T2CC[W, H, n]]
```

Generate test datasets

k=2, r=2 non-totalistic

In[31]:=
```
genk2r2C[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, 2, 2}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
datak2r2C[W_Integer, H_Integer, n_Integer] := Table[genk2r2C[i, W, H] → i,
  {i, RandomChoice[Range[0, 4 294 967 295], n]}]
```

k=2, r=3 non-totalistic

In[33]:=
```
genk2r3NT[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, 2, 3}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
datak2r3NT[W_Integer, H_Integer, n_Integer] :=
 Table[genk2r3NT[i, W, H] → i, {i, RandomInteger[2 ^ 2 ^ 7 - 1, n]}]
```

k=3, r=1 non-totalistic

In[35]:=
```
genk3r1NT[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, 3}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules → {0 -> RandomColor[], 1 -> RandomColor[],
     3 -> RandomColor[], 4 → RandomColor[], 5 -> RandomColor[],
     6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
datak3r1NT[W_Integer, H_Integer, n_Integer] :=
 Table[genk3r1NT[i, W, H] → i, {i, RandomInteger[3 ^ 3 ^ 3 - 1, n]}]
```

k=3, r=2 totalistic

In[37]:=
```
genk3r2C[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {3, 1}, 2}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
datak3r2C[W_Integer, H_Integer, n_Integer] := Table[genk3r2C[i, W, H] → i,
  {i, RandomChoice[Range[0, 177 146], n]}]
```

k=3, r=3 totalistic

In[39]:=
```
genk3r3C[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {3, 1}, 3}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
datak3r3C[W_Integer, H_Integer, n_Integer] := Table[genk3r3C[i, W, H] → i,
  {i, RandomChoice[Range[0, 14 348 906], n]}]
```

k=4, r=1 non-totalistic

In[41]:=
```
genk4r1NT[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, 4}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules → {0 -> RandomColor[], 1 -> RandomColor[],
     3 -> RandomColor[], 4 → RandomColor[], 5 -> RandomColor[],
     6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
datak4r1NT[W_Integer, H_Integer, n_Integer] :=
 Table[genk4r1NT[i, W, H] → i, {i, RandomInteger[4^4^3 - 1, n]}]
```

k=4, r=1 totalistic

In[43]:=
```
genk4r1C[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {4, 1}}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
datak4r1C[W_Integer, H_Integer, n_Integer] := Table[genk4r1C[i, W, H] → i,
  {i, RandomChoice[Range[0, 1 048 575], n]}]
```

k=4, r=2 totalistic

In[45]:=
```
genk4r2C[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {4, 1}, 2}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
datak4r2C[W_Integer, H_Integer, n_Integer] := Table[genk4r2C[i, W, H] → i,
  {i, RandomChoice[Range[0, 4 294 967 295], n]}]
```

k=5, r=1 totalistic

In[47]:=
```
gen5T2C[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {5, 1}, 1}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
data5T2C[n_Integer, W_Integer, H_Integer] := Table[gen5T2C[i, W, H] → i,
  {i, RandomChoice[Range[0, 1 220 703 125], n]}]
```

k=6, r=1 totalistic

In[49]:=
```
gen6TC[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {6, 1}, 1}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
data6TC[n_Integer, W_Integer, H_Integer] := Table[gen6TC[i, W, H] → i,
  {i, RandomInteger[2 821 109 907 455, n]}]
```

k=6, r=2 totalistic

In[51]:=
```
gen6T2C[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {6, 1}, 2}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame -> False]]
data6T2C[n_Integer, W_Integer, H_Integer] := Table[gen6T2C[i, W, H] → i,
  {i, RandomInteger[170 581 728 179 578 208 255, n]}]
```

k=7, r=1 totalistic

In[53]:=
```
gen7TC[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {7, 1}, 1}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame → False]]
data7TC[n_Integer, W_Integer, H_Integer] := Table[gen7TC[i, W, H] → i,
  {i, RandomInteger[11 398 895 185 373 142, n]}]
```

k=8, r=1 totalistic

In[55]:=
```
gen8TC[p_Integer, W_Integer, H_Integer] :=
 Image[ArrayPlot[CellularAutomaton[{p, {8, 1}, 1}, RandomInteger[1, W], H - 1],
   ImageSize → {W, H}, ColorRules →
    {0 -> RandomColor[], 1 -> RandomColor[], 3 -> RandomColor[], 4 → RandomColor[],
     5 -> RandomColor[], 6 -> RandomColor[], 7 -> RandomColor[]}, Frame → False]]
data8TC[n_Integer, W_Integer, H_Integer] := Table[gen8TC[i, W, H] → i,
  {i, RandomInteger[73 786 976 294 838 206 463, n]}]
```

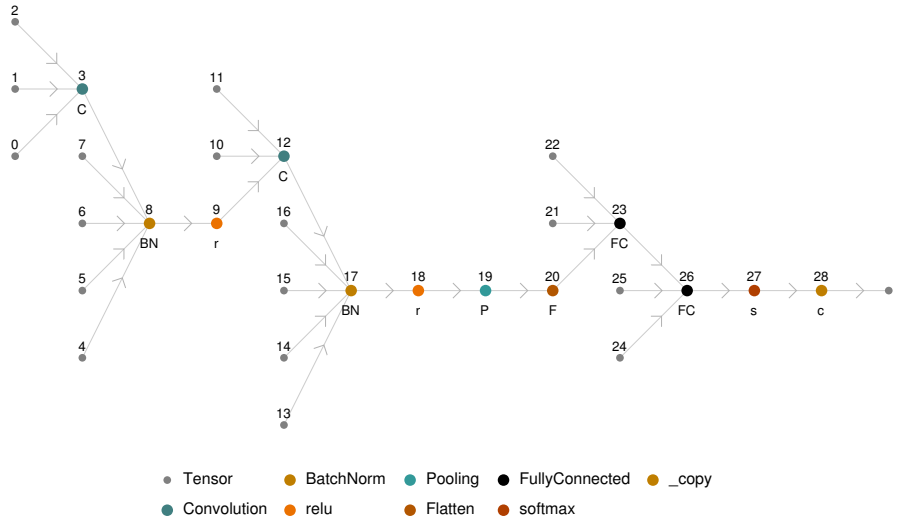
Network XIII - Two convolutions, dropout on linear only, BatchNorm

In[ ]:=
```
netECA13 = netSevenCC512drop[128, 128]
```

Out[ ]=
```
NetChain[ Input port:  image
          Output port:  class
          Number of layers:  12 ]
```

In[ ]:=
```
NetInformation[netECA13, "MXNetNodeGraphPlot"]
```

Out[ ]=



In[ ]:=
```
NetInformation[netECA13, "SummaryGraphic"]
```

Out[ ]=

In[ ]:=
```
dataECA13 = dataC[128, 128, 8192];
```

In[ ]:=
```
dataTotalistic2BigC13 = genData2r2C[128, 128, 1024];
```

In[ ]:=
```
dataTotalistic3BigC13 = data3T2C[128, 128, 1024];
```

In[ ]:=
```
dataTotalistic4BigC13 = data4TC[128, 128, 1024];
```

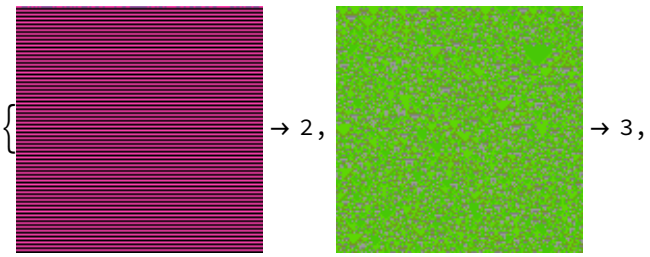
In[ ]:=
```
dataTotalistic5BigC13 = genData5TCC[128, 128, 4096];
```

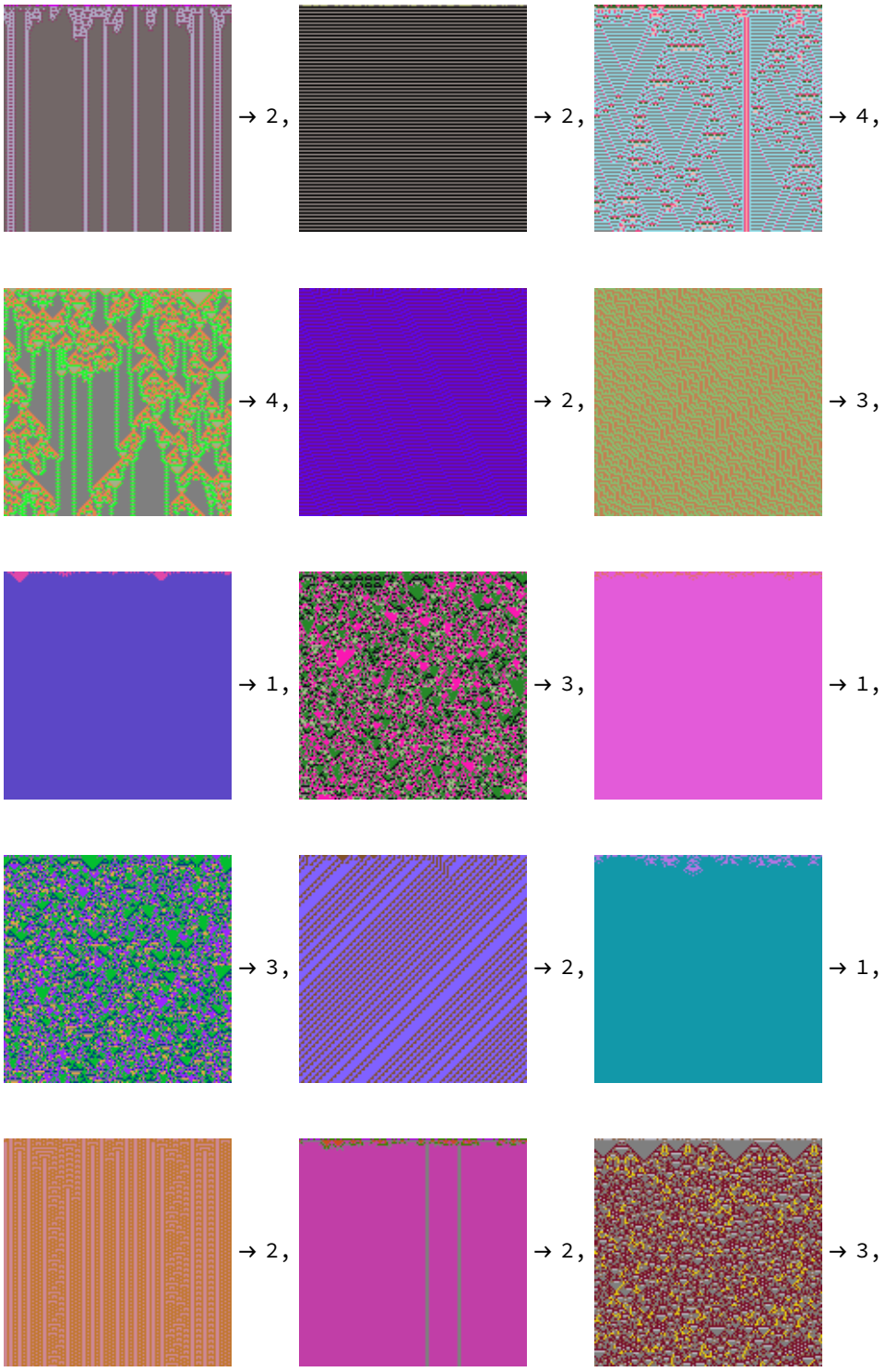
In[ ]:=
```
fullTrainingBigC13 = Join[dataECA13, dataTotalistic2BigC13,
    dataTotalistic3BigC13, dataTotalistic4BigC13, dataTotalistic5BigC13];
Length[fullTrainingBigC13]
```

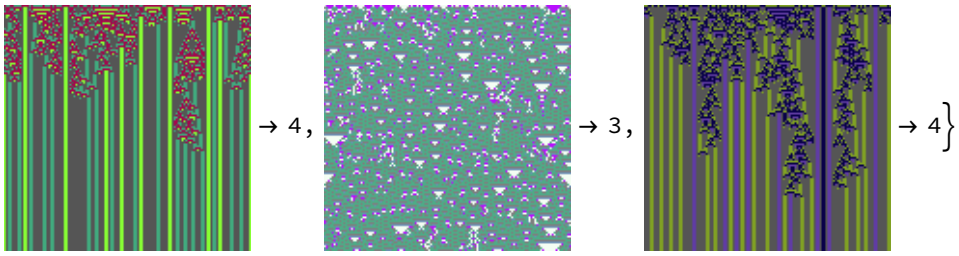
Out[ ]=
```
26 624
```

In[ ]:=
```
RandomSample[fullTrainingBigC13, 20]
```

Out[ ]=

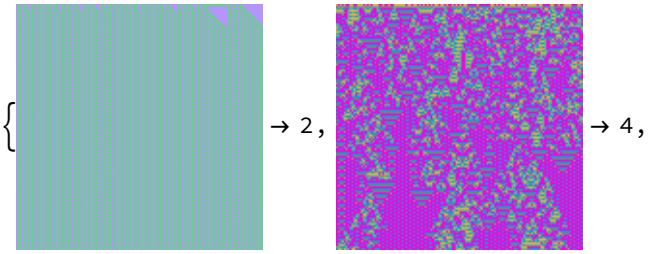


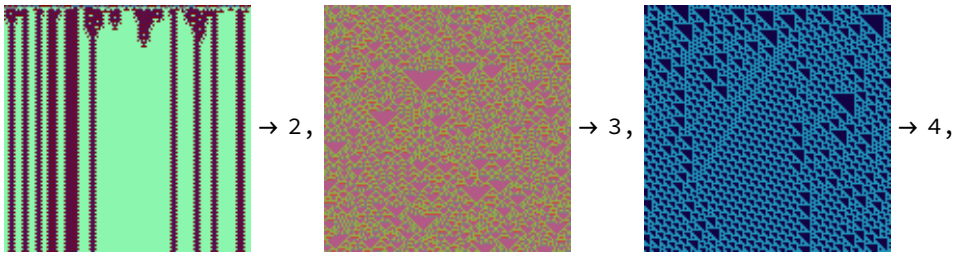

→ 2,
→ 2,
→ 4,
→ 4,
→ 2,
→ 3,
→ 1,
→ 3,
→ 1,
→ 3,
→ 2,
→ 1,
→ 2,
→ 2,
→ 3,

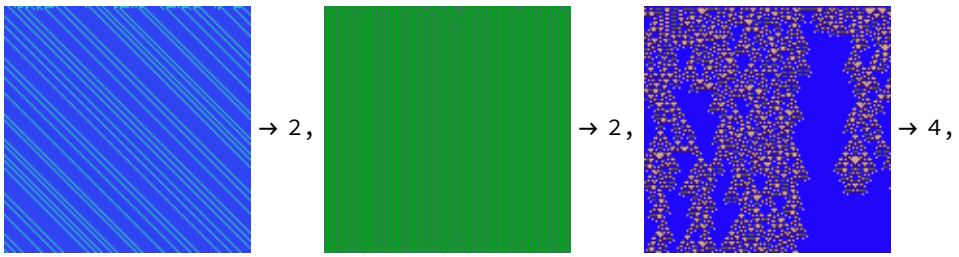


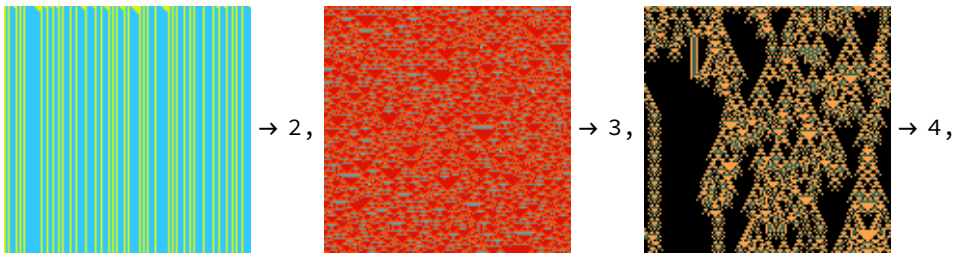
In[ ]:= **RandomSample[fullTrainingBigC13, 20]**

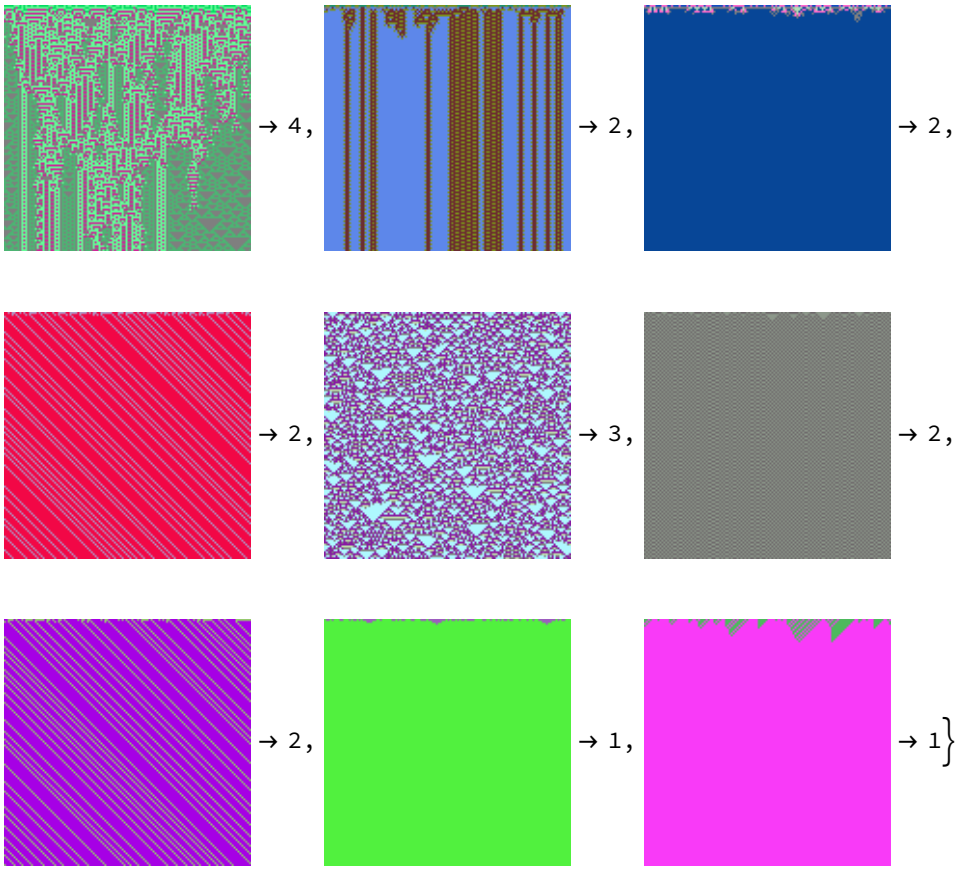
Out[ ]=











In[ ]:= **dir = SetDirectory[NotebookDirectory[]]**

Out[ ]= /Users/thorsilver/Downloads/Wolfram notebooks

In[ ]:= **netECA12 = Import["netECA12-r12.wlnet"]**

Out[ ]= NetChain[

		image
	Input	array (size: 3 × 120 × 200)
conv1	ConvolutionLayer	array (size: 24 × 118 × 198)
bat1	BatchNormalizationLayer	array (size: 24 × 118 × 198)
ramp1	Ramp	array (size: 24 × 118 × 198)
conv3	ConvolutionLayer	array (size: 24 × 116 × 196)
bat2	BatchNormalizationLayer	array (size: 24 × 116 × 196)
ramp2	Ramp	array (size: 24 × 116 × 196)
pooling	PoolingLayer	array (size: 24 × 1 × 5)
flatten	FlattenLayer	vector (size: 120)
linear	LinearLayer	vector (size: 512)
drop2	DropoutLayer	vector (size: 512)
linear2	LinearLayer	vector (size: 4)
softmax	SoftmaxLayer	vector (size: 4)
	Output	class

]

In[ ]:=
```
netECA13 =
 NetTrain[netECA13, fullTrainingBigC13, MaxTrainingRounds → 20, BatchSize → 256 * 4,
  TargetDevice → "CPU", TrainingProgressCheckpointing → {"Directory", dir}]
```

Out[ ]=
```
NetChain[ Input port:  image
          Output port:  class
          Number of layers:  12 ]
```

In[ ]:=
```
netECA13 = Import["netECA13-r20.wlnet"]
```

Out[ ]=
```
NetChain[ Input port:  image
          Output port:  class
          Number of layers:  12 ]
```

In[ ]:=
```
netECA13 =
 NetTrain[netECA13, fullTrainingBigC13, MaxTrainingRounds → 20, BatchSize → 256 * 4,
  TargetDevice → "CPU", TrainingProgressCheckpointing → {"Directory", dir}]
```

Out[ ]=
```
NetChain[ Input port:  image
          Output port:  class
          Number of layers:  12 ]
```

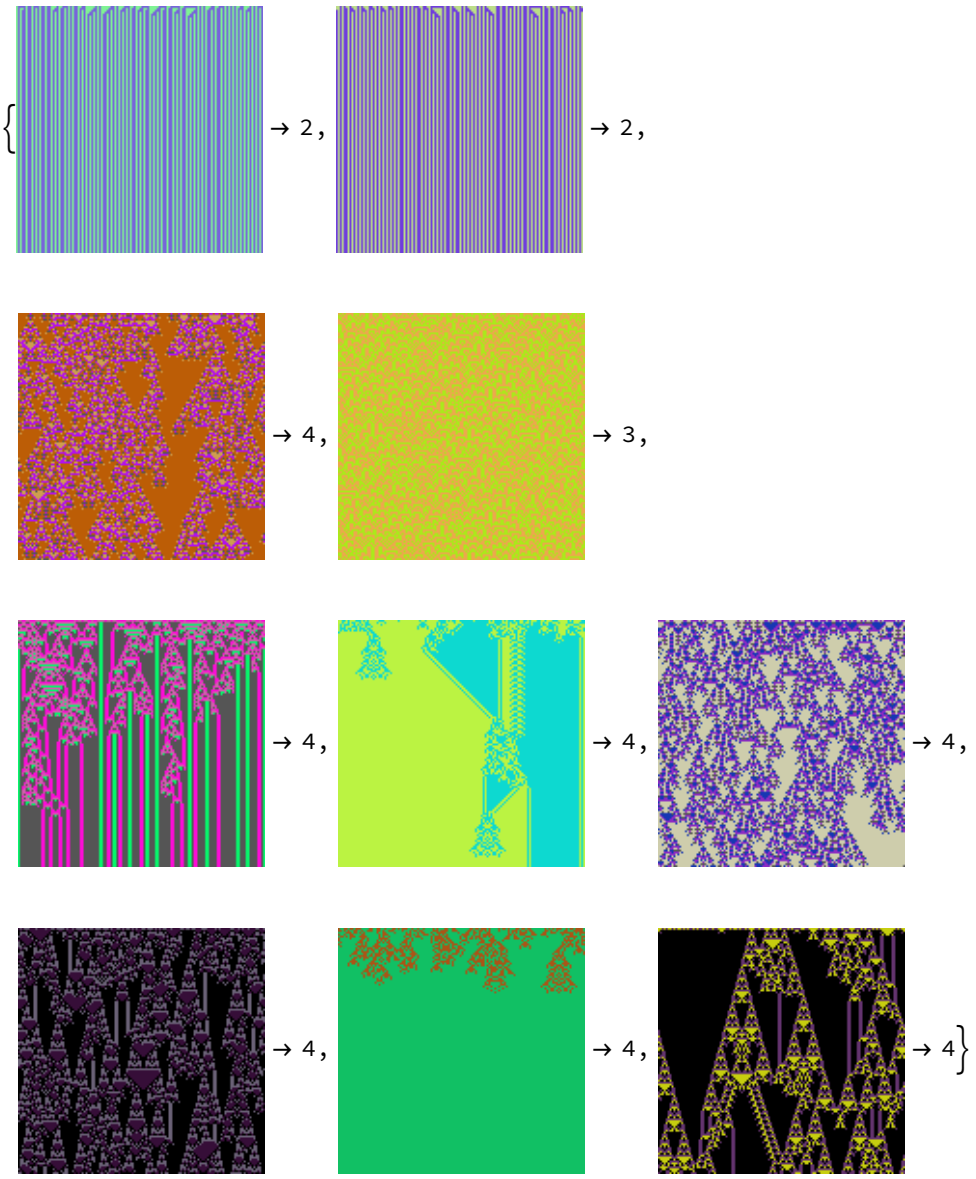
Generate test data for Network XIII

In[ ]:=
```
testDataECABigC = dataC[128, 128, 1024];
testData2TBigC = genData2r2C[128, 128, 1024];
testData3TBigC = data3T2C[128, 128, 1024];
testData4TBigC = data4TC[128, 128, 1024];
testData5TBigC = genData5TCC[128, 128, 1024];
fullTestSetBigC = Join[testDataECABigC,
   testData2TBigC, testData3TBigC, testData4TBigC, testData5TBigC];
Length[fullTestSetBigC]
```

Out[ ]=
```
10 240
```

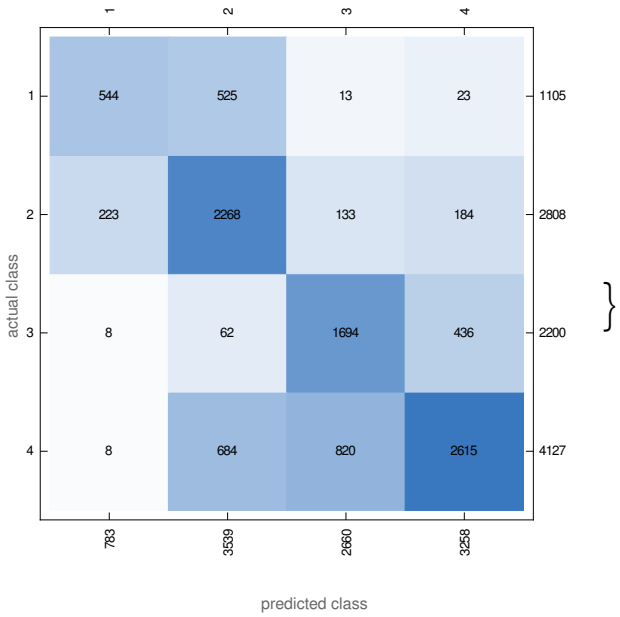
In[ ]:= **RandomSample[fullTestSetBigC, 10]**

Out[ ]=



In[ ]:=
```
NetMeasurements[netECA13, fullTestSetBigC,
 {"Accuracy", "Precision", "ConfusionMatrixPlot"}]
```

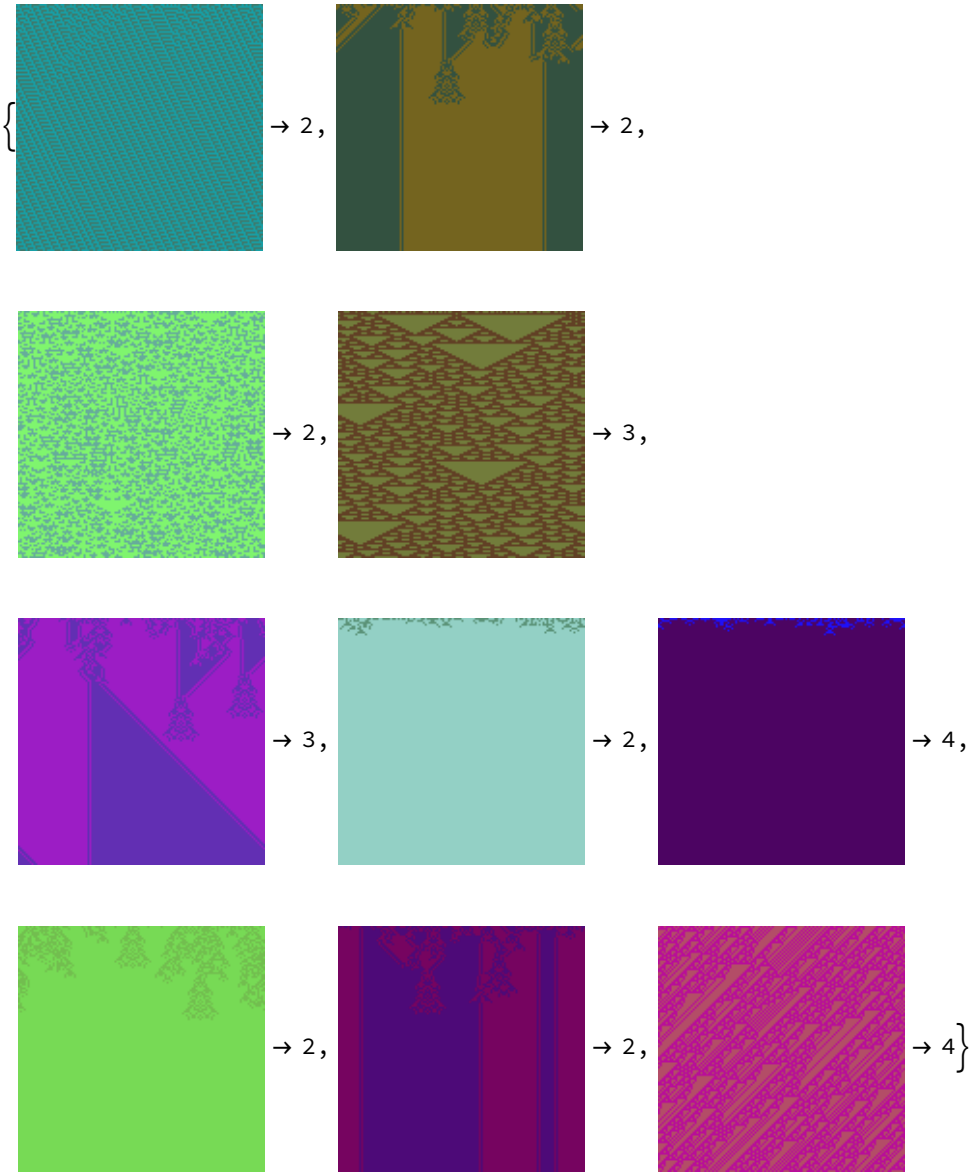
Out[ ]=
```
{0.69541, <|1 → 0.694764, 2 → 0.640859, 3 → 0.636842, 4 → 0.80264|>,
```



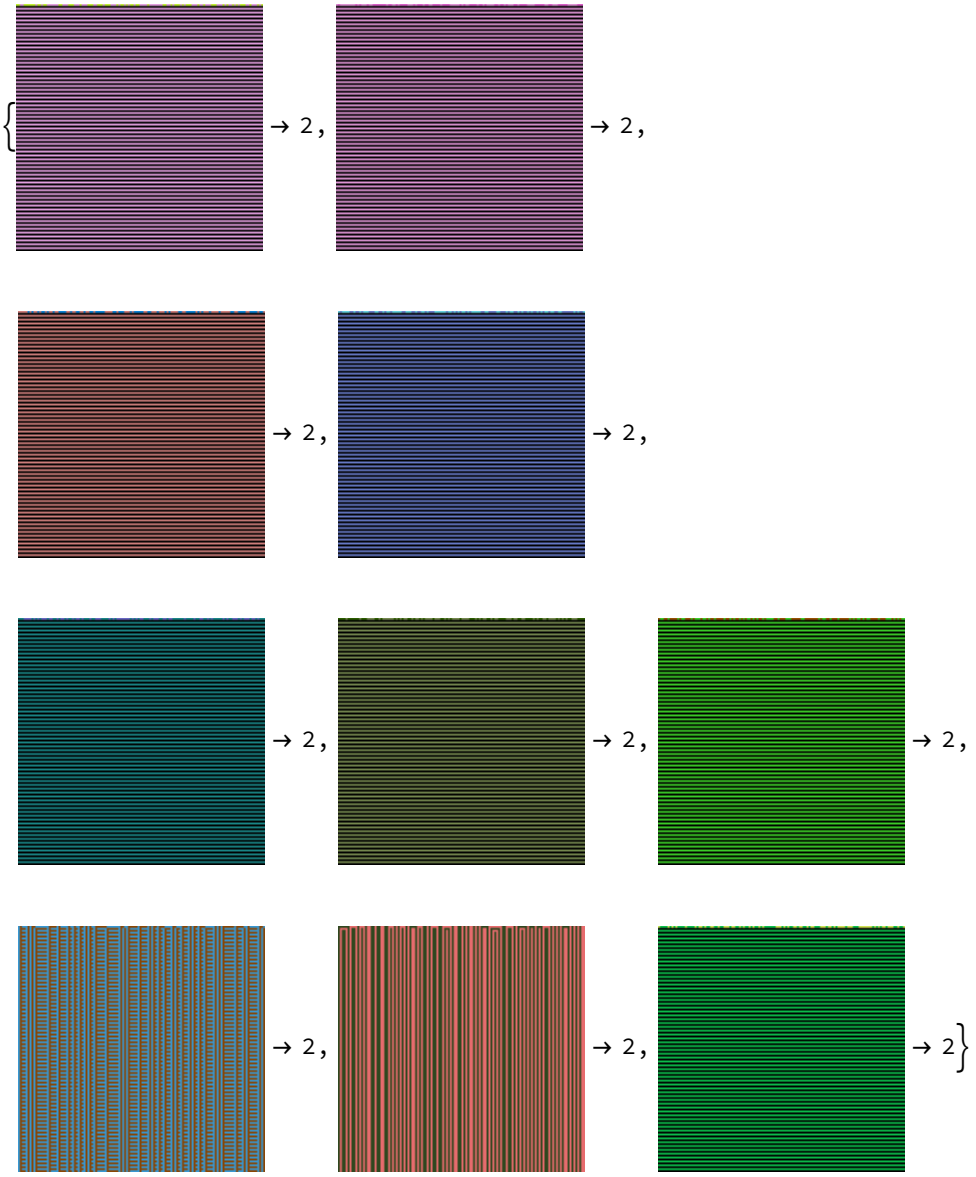
}

In[ ]:=
```
entropyImagesBigC = RandomSample[Keys[fullTestSetBigC], 500];
entropiesBigC = netECA13[entropyImagesBigC, "Entropy"];
highEntBigC = entropyImagesBigC[[Ordering[entropiesBigC, -10]]];
lowEntBigC = entropyImagesBigC[[Ordering[entropiesBigC, 10]]];
Thread[highEntBigC → netECA13[highEntBigC]]
Thread[lowEntBigC → netECA13[lowEntBigC]]
```

Out[ ]=



Out[ ]=



Network XIV - BatchNorm, 1024 linear, dropout

In[ ]:=
```
netECA14 = netEightCC512drop[128, 128]
```

Out[ ]=
```
NetChain[
```

	Input	image array (size: 3 × 128 × 128)
conv1	ConvolutionLayer	array (size: 24 × 126 × 126)
bat1	BatchNormalizationLayer	array (size: 24 × 126 × 126)
ramp1	Ramp	array (size: 24 × 126 × 126)
conv2	ConvolutionLayer	array (size: 16 × 125 × 124)
bat2	BatchNormalizationLayer	array (size: 16 × 125 × 124)
ramp2	Ramp	array (size: 16 × 125 × 124)
conv3	ConvolutionLayer	array (size: 24 × 123 × 122)
bat3	BatchNormalizationLayer	array (size: 24 × 123 × 122)
ramp3	Ramp	array (size: 24 × 123 × 122)
pooling	PoolingLayer	array (size: 24 × 4 × 11)
flatten	FlattenLayer	vector (size: 1056)
linear	LinearLayer	vector (size: 1024)
drop2	DropoutLayer	vector (size: 1024)
linear2	LinearLayer	vector (size: 4)
softmax	SoftmaxLayer	vector (size: 4)
	Output	class

```
]
```

```
netECA14 =
 NetTrain[netECA14, fullTrainingBigC13, MaxTrainingRounds → 20, BatchSize → 256 * 4,
  TargetDevice → "CPU", TrainingProgressCheckpointing → {"Directory", dir}]
```

In[ ]:=
```
dir = SetDirectory[NotebookDirectory[]]
```

Out[ ]=
```
/Users/thorsilver/Downloads/Wolfram notebooks
```

In[ ]:=
```
netECA14 = Import["netECA14-r20.wlnet"]
```

Out[ ]=
```
NetChain[
```

	Input	image array (size: 3 × 128 × 128)
conv1	ConvolutionLayer	array (size: 24 × 126 × 126)
bat1	BatchNormalizationLayer	array (size: 24 × 126 × 126)
ramp1	Ramp	array (size: 24 × 126 × 126)
conv2	ConvolutionLayer	array (size: 16 × 125 × 124)
bat2	BatchNormalizationLayer	array (size: 16 × 125 × 124)
ramp2	Ramp	array (size: 16 × 125 × 124)
conv3	ConvolutionLayer	array (size: 24 × 123 × 122)
bat3	BatchNormalizationLayer	array (size: 24 × 123 × 122)
ramp3	Ramp	array (size: 24 × 123 × 122)
pooling	PoolingLayer	array (size: 24 × 4 × 11)
flatten	FlattenLayer	vector (size: 1056)
linear	LinearLayer	vector (size: 1024)
drop2	DropoutLayer	vector (size: 1024)
linear2	LinearLayer	vector (size: 4)
softmax	SoftmaxLayer	vector (size: 4)
	Output	class

```
]
```

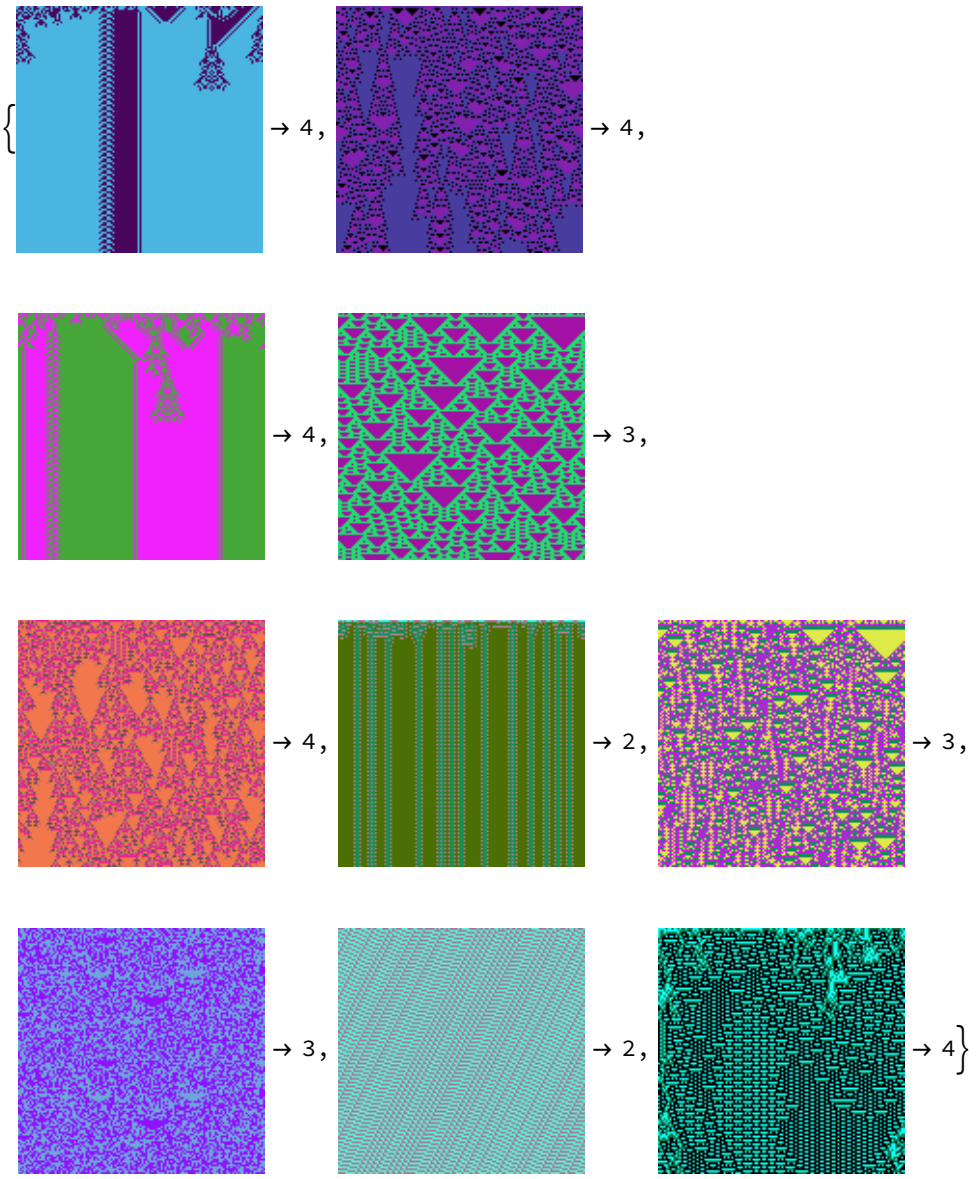
Generating test data for Network XIV

In[ ]:=
```
testDataECABigC = dataC[128, 128, 1024];
testData2TBigC = genData2r2C[128, 128, 1024];
testData3TBigC = data3T2C[128, 128, 1024];
testData4TBigC = data4TC[128, 128, 1024];
testData5TBigC = genData5TCC[128, 128, 1024];
fullTestSetBigC = Join[testDataECABigC,
   testData2TBigC, testData3TBigC, testData4TBigC, testData5TBigC];
Length[fullTestSetBigC]
```

Out[ ]=
```
10 240
```

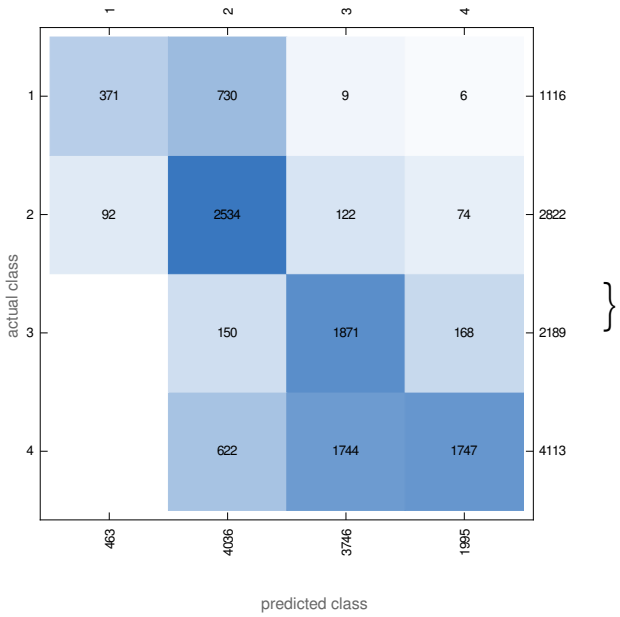
In[ ]:= `RandomSample[fullTestSetBigC, 10]`

Out[ ]=



In[ ]:=
```
NetMeasurements[netECA14, fullTestSetBigC,
 {"Accuracy", "Precision", "ConfusionMatrixPlot"}]
```

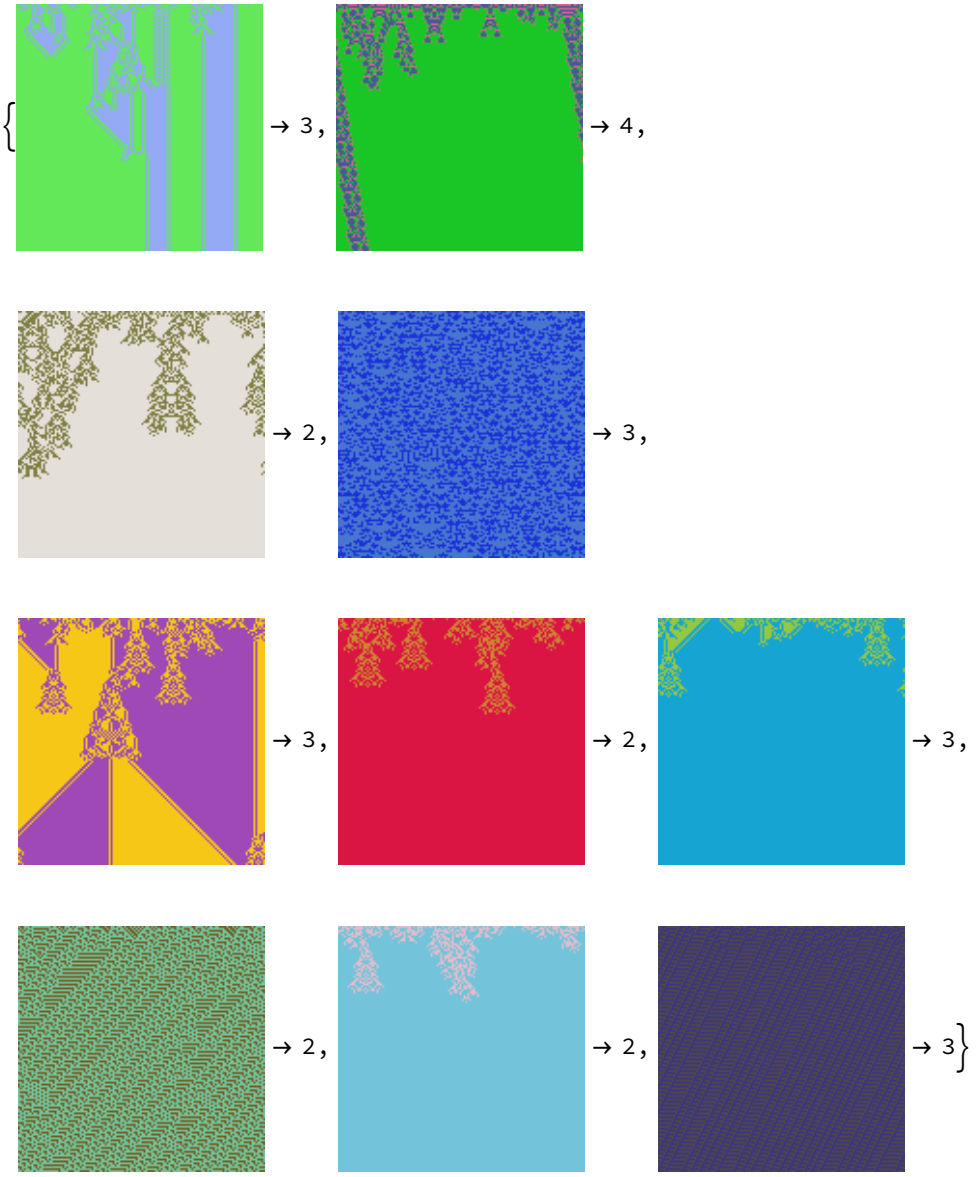
Out[ ]=
```
{0.637012, <|1 → 0.801296, 2 → 0.627849, 3 → 0.499466, 4 → 0.875689|>,
```



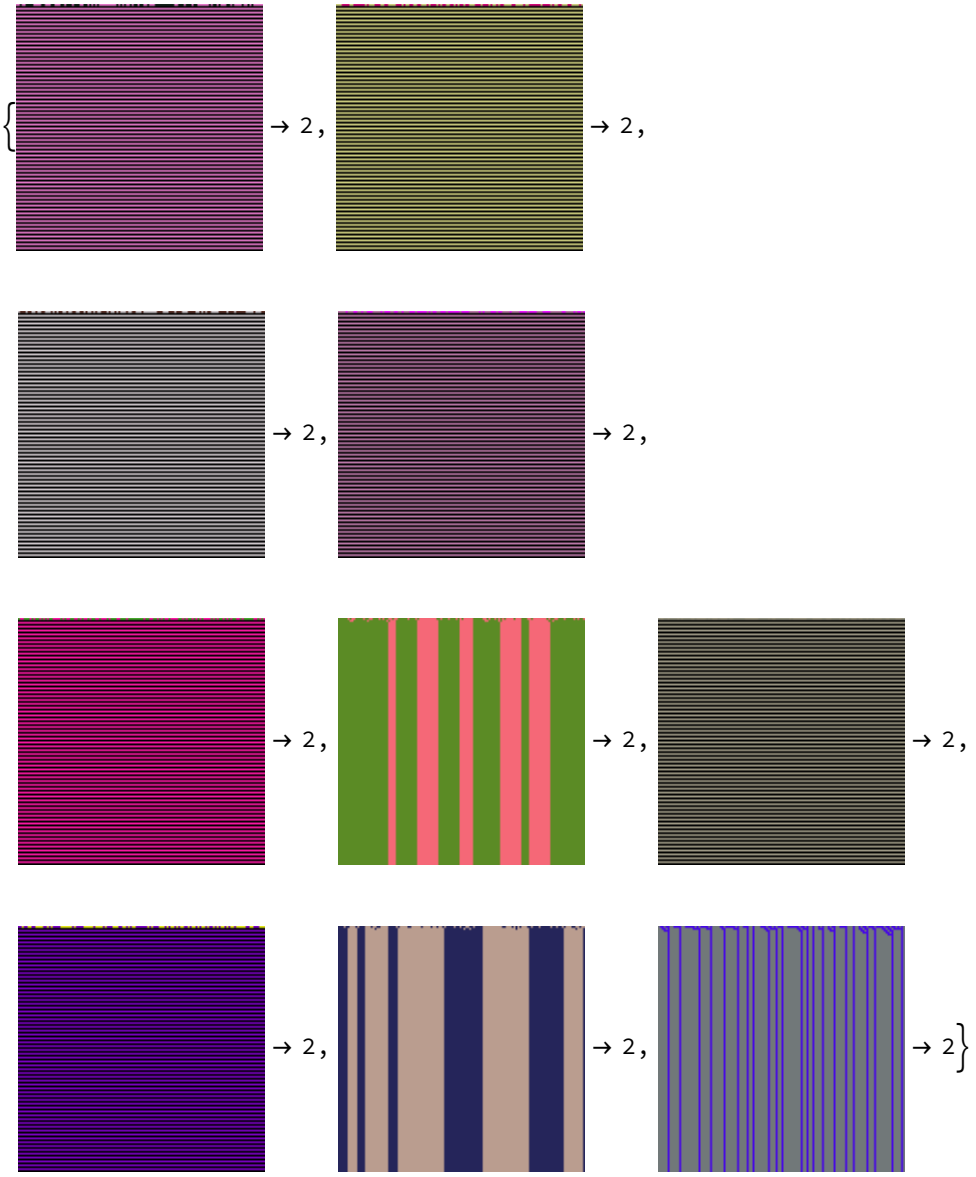

```
}
```

In[ ]:=
```
entropyImagesBigC = RandomSample[Keys[fullTestSetBigC], 500];
entropiesBigC = netECA14[entropyImagesBigC, "Entropy"];
highEntBigC = entropyImagesBigC[[Ordering[entropiesBigC, -10]]];
lowEntBigC = entropyImagesBigC[[Ordering[entropiesBigC, 10]]];
Thread[highEntBigC → netECA14[highEntBigC]]
Thread[lowEntBigC → netECA14[lowEntBigC]]
```

Out[ ]=



Out[ ]=



Network XV - Transfer learning with pre-trained image recognition net (VGG-16)

```
In[ ]:= netECA15 = NetModel["VGG-16 Trained on ImageNet Competition Data"]
```

Out[ ]= NetChain

	Input	image array (size: 3 × 224 × 224)
conv1_1	ConvolutionLayer	array (size: 64 × 224 × 224)
relu1_1	Ramp	array (size: 64 × 224 × 224)
conv1_2	ConvolutionLayer	array (size: 64 × 224 × 224)
relu1_2	Ramp	array (size: 64 × 224 × 224)
pool1	PoolingLayer	array (size: 64 × 112 × 112)
conv2_1	ConvolutionLayer	array (size: 128 × 112 × 112)
relu2_1	Ramp	array (size: 128 × 112 × 112)
conv2_2	ConvolutionLayer	array (size: 128 × 112 × 112)
relu2_2	Ramp	array (size: 128 × 112 × 112)
pool2	PoolingLayer	array (size: 128 × 56 × 56)
conv3_1	ConvolutionLayer	array (size: 256 × 56 × 56)
relu3_1	Ramp	array (size: 256 × 56 × 56)
conv3_2	ConvolutionLayer	array (size: 256 × 56 × 56)
relu3_2	Ramp	array (size: 256 × 56 × 56)
conv3_3	ConvolutionLayer	array (size: 256 × 56 × 56)
relu3_3	Ramp	array (size: 256 × 56 × 56)
pool3	PoolingLayer	array (size: 256 × 28 × 28)
conv4_1	ConvolutionLayer	array (size: 512 × 28 × 28)
relu4_1	Ramp	array (size: 512 × 28 × 28)
conv4_2	ConvolutionLayer	array (size: 512 × 28 × 28)
relu4_2	Ramp	array (size: 512 × 28 × 28)
conv4_3	ConvolutionLayer	array (size: 512 × 28 × 28)
relu4_3	Ramp	array (size: 512 × 28 × 28)
pool4	PoolingLayer	array (size: 512 × 14 × 14)
conv5_1	ConvolutionLayer	array (size: 512 × 14 × 14)
relu5_1	Ramp	array (size: 512 × 14 × 14)
conv5_2	ConvolutionLayer	array (size: 512 × 14 × 14)
relu5_2	Ramp	array (size: 512 × 14 × 14)
conv5_3	ConvolutionLayer	array (size: 512 × 14 × 14)
relu5_3	Ramp	array (size: 512 × 14 × 14)
pool5	PoolingLayer	array (size: 512 × 7 × 7)
flatten_0	FlattenLayer	vector (size: 25088)
fc6	LinearLayer	vector (size: 4096)
relu6	Ramp	vector (size: 4096)
drop6	DropoutLayer	vector (size: 4096)
fc7	LinearLayer	vector (size: 4096)
relu7	Ramp	vector (size: 4096)
drop7	DropoutLayer	vector (size: 4096)
fc8	LinearLayer	vector (size: 1000)
prob	SoftmaxLayer	vector (size: 1000)
	Output	class

```
In[ ]:= subNet = NetTake[netECA15, {"conv1_1", "flatten_0"}]
```

Out[ ]= NetChain

Input port:	image
Output port:	vector (size: 25088)
Number of layers:	32

```
In[ ]:= joinedNet = NetJoin[subNet,
          NetChain@<|"linear_new" → LinearLayer[1024], "linear_out" → LinearLayer[4],
            "prob" → SoftmaxLayer[]|>, "Output" → NetDecoder[{"Class", Range[1, 4]}]]
```

Out[ ]= NetChain

uninitialized

Input port:	image
Output port:	class
Number of layers:	35

In[ ]:=
```
netECA15final = NetPrepend[joinedNet, {"augment" → ImageAugmentationLayer[{224, 224}]},
   "Input" → NetExtract[joinedNet, "Input"]]
```

Out[ ]=
```
NetChain[ uninitialized  Input port: image
                         Output port: class
                         Number of layers: 36 ]
```

In[ ]:=
```
dataECA15 = dataC[224, 224, 8192];
```

In[ ]:=
```
dataTotalistic2BigC15 = genData2r2C[224, 224, 1024];
```

In[ ]:=
```
dataTotalistic3BigC15 = data3T2C[224, 224, 512];
```

In[ ]:=
```
dataTotalistic4BigC15 = data4TC[224, 224, 512];
```

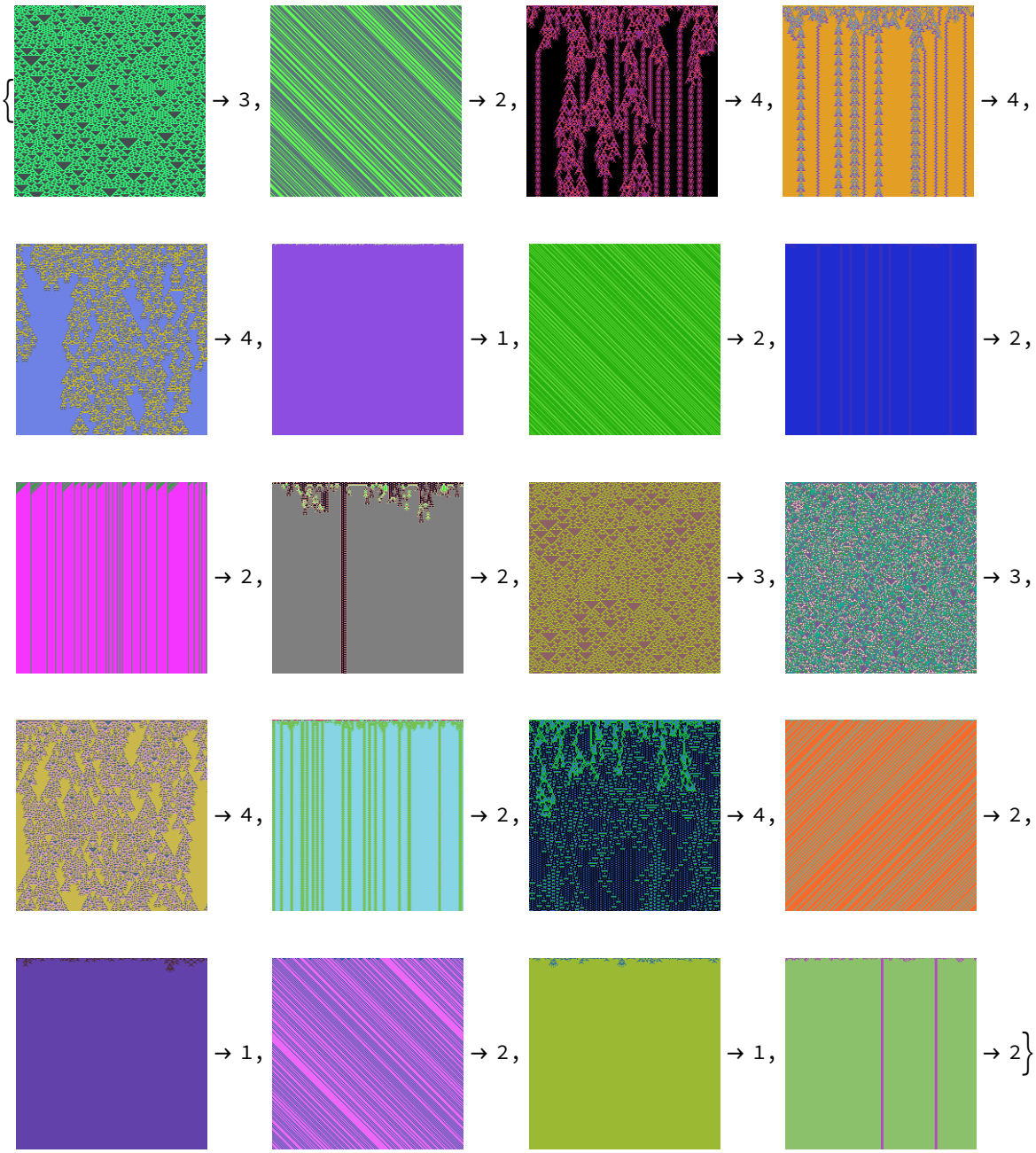
In[ ]:=
```
dataTotalistic5BigC15 = genData5TCC[224, 224, 1024];
```

In[ ]:=
```
fullTrainingBigC15 = Join[dataECA15, dataTotalistic2BigC15,
    dataTotalistic3BigC15, dataTotalistic4BigC15, dataTotalistic5BigC15];
Length[fullTrainingBigC15]
```

Out[ ]=
```
16 384
```

In[ ]:=
```
RandomSample[fullTrainingBigC15, 20]
```

Out[ ]=



In[ ]:=
```
netECA15final =
 NetTrain[netECA15final, fullTrainingBigC15, MaxTrainingRounds → 5, BatchSize → 256 * 4,
  TargetDevice → "CPU", TrainingProgressCheckpointing → {"Directory", dir},
  LearningRateMultipliers → {"linear_new" → 1, "linear_out" → 1, _ → 0}]
```

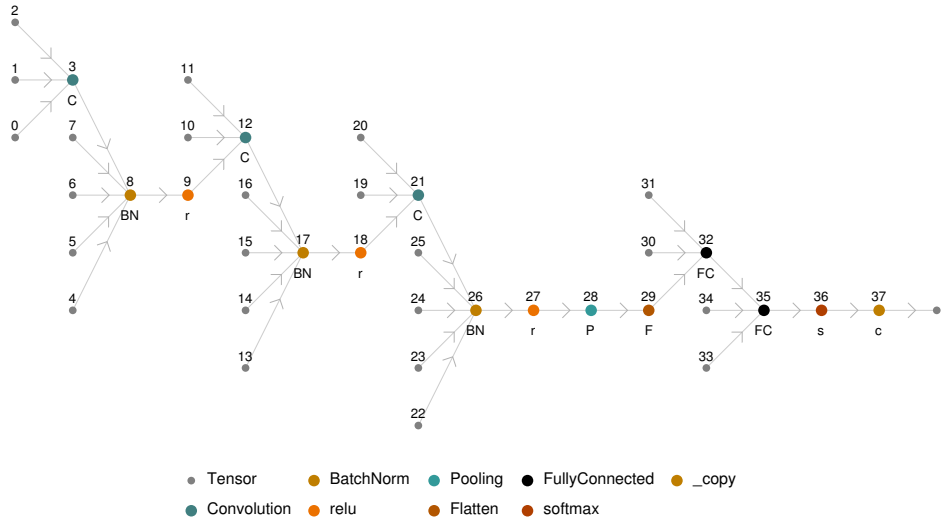
Network XVI - Three convolutions, dropout on linear only, BatchNorm

In[57]:=
```
netECA16 = netNineCC512drop[128, 128]
```

Out[57]=
```
NetChain[ Input port:  image
          Output port:  class
          Number of layers:  15 ]
```

In[ ]:=
```
NetInformation[netECA16, "MXNetNodeGraphPlot"]
```

Out[ ]=



In[ ]:=
```
NetInformation[netECA16, "SummaryGraphic"]
```

Out[ ]=

In[58]:=
```
dataECA16 = dataC[128, 128, 8192];
```

In[59]:=
```
dataTotalistic2BigC16 = genData2r2C[128, 128, 1024];
```

In[60]:=
```
dataTotalistic3BigC16 = data3T2C[128, 128, 1024];
```

In[61]:=
```
dataTotalistic4BigC16 = data4TC[128, 128, 1024];
```

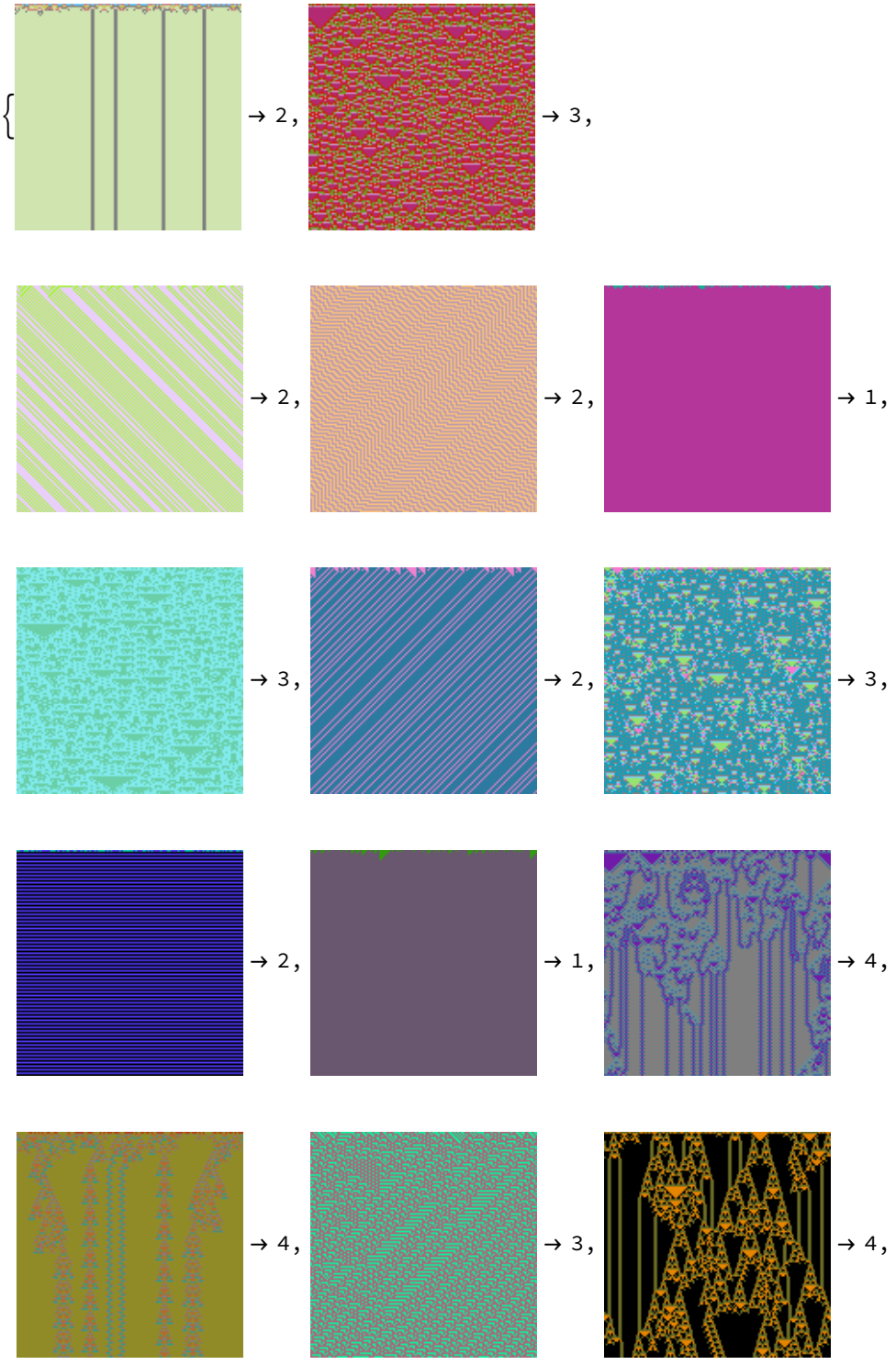
In[62]:=
```
dataTotalistic5BigC16 = genData5TCC[128, 128, 4096];
```

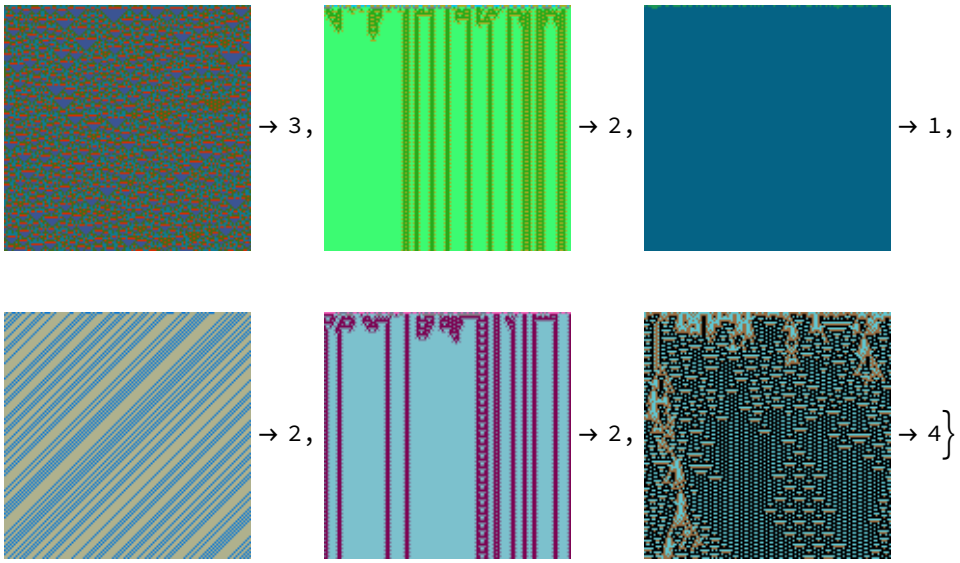
In[63]:=
```
fullTrainingBigC16 = Join[dataECA16, dataTotalistic2BigC16,
    dataTotalistic3BigC16, dataTotalistic4BigC16, dataTotalistic5BigC16];
Length[fullTrainingBigC16]
```

Out[64]=
```
26 624
```

In[65]:=
```
RandomSample[fullTrainingBigC16, 20]
```

Out[65]=





In[66]:=
```
dir = SetDirectory[NotebookDirectory[]]
```
Out[66]= /home/esilverman/Documents

In[67]:=
```
netECA16 =
 NetTrain[netECA16, fullTrainingBigC16, MaxTrainingRounds → 200, BatchSize → 256,
  TargetDevice → "GPU", TrainingProgressCheckpointing → {"Directory", dir}]
```
Out[67]= NetChain[Input port: image / Output port: class / Number of layers: 15]

```
netECA16 = Import["netECA16-r20.wlnet"]
```

```
netECA16 =
 NetTrain[netECA16, fullTrainingBigC16, MaxTrainingRounds → 20, BatchSize → 256 * 4,
  TargetDevice → "CPU", TrainingProgressCheckpointing → {"Directory", dir}]
```

Generate test data for Network XVI

In[ ]:=
```
dir = SetDirectory[NotebookDirectory[]]
```
Out[ ]= /Users/thorsilver/Downloads/Wolfram notebooks

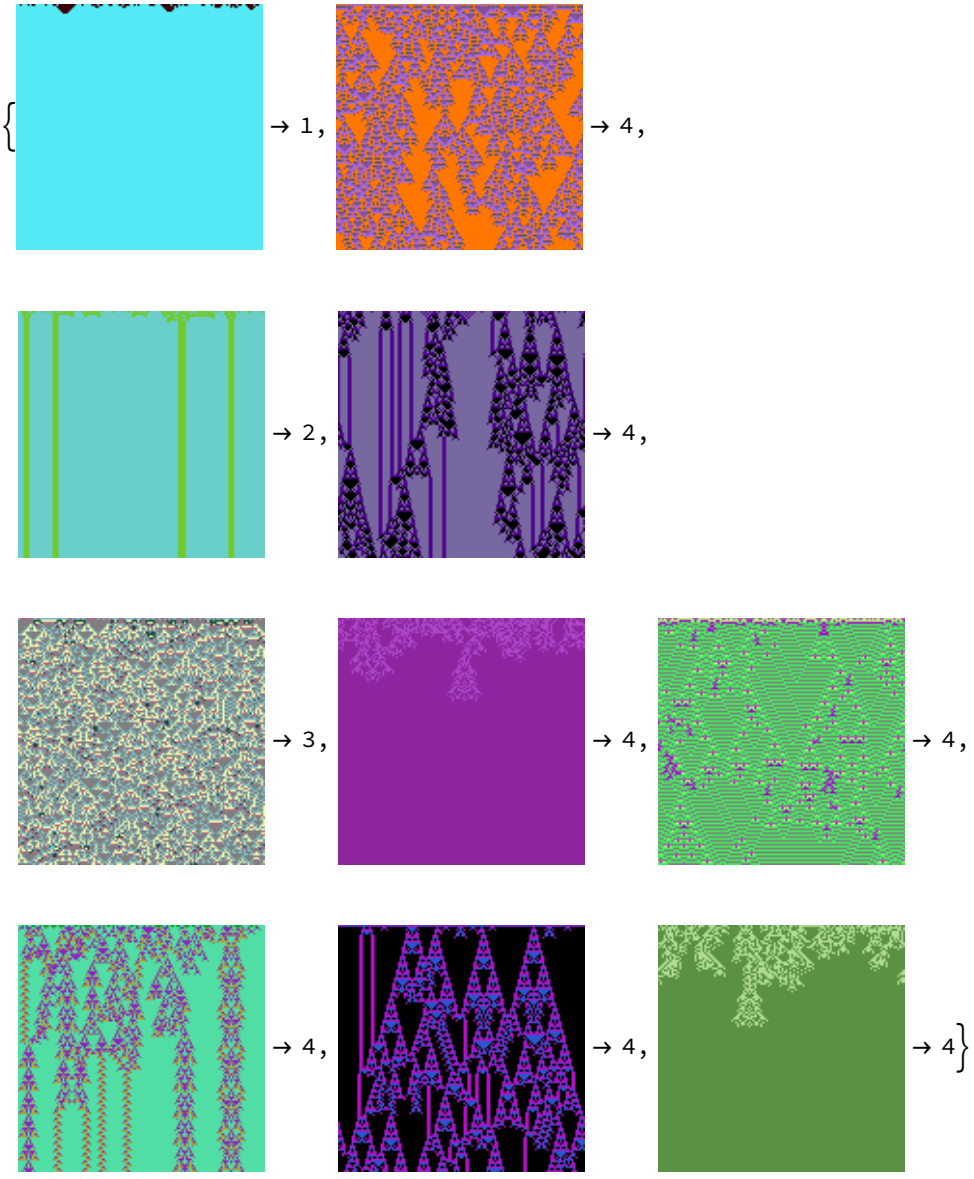
In[ ]:=
```
netECA16 = Import["netECA16-r20.wlnet"]
```
Out[ ]= NetChain[Input port: image / Output port: class / Number of layers: 15]

In[68]:=
```
testDataECABigC = dataC[128, 128, 1024];
testData2TBigC = genData2r2C[128, 128, 1024];
testData3TBigC = data3T2C[128, 128, 1024];
testData4TBigC = data4TC[128, 128, 1024];
testData5TBigC = genData5TCC[128, 128, 1024];
fullTestSetBigC = Join[testDataECABigC,
   testData2TBigC, testData3TBigC, testData4TBigC, testData5TBigC];
Length[fullTestSetBigC]
```

Out[74]= 10 240

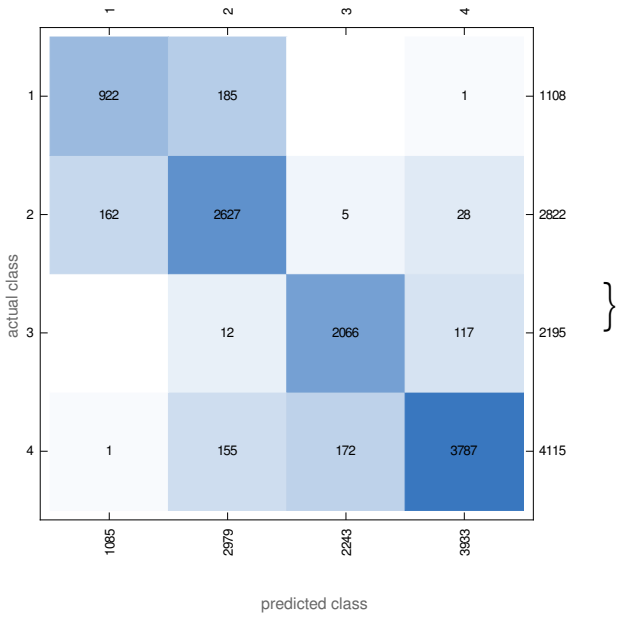
In[77]:=
```
RandomSample[fullTestSetBigC, 10]
```

Out[77]=



In[76]:=
```
NetMeasurements[netECA16, fullTestSetBigC,
 {"Accuracy", "Precision", "ConfusionMatrixPlot"}]
```

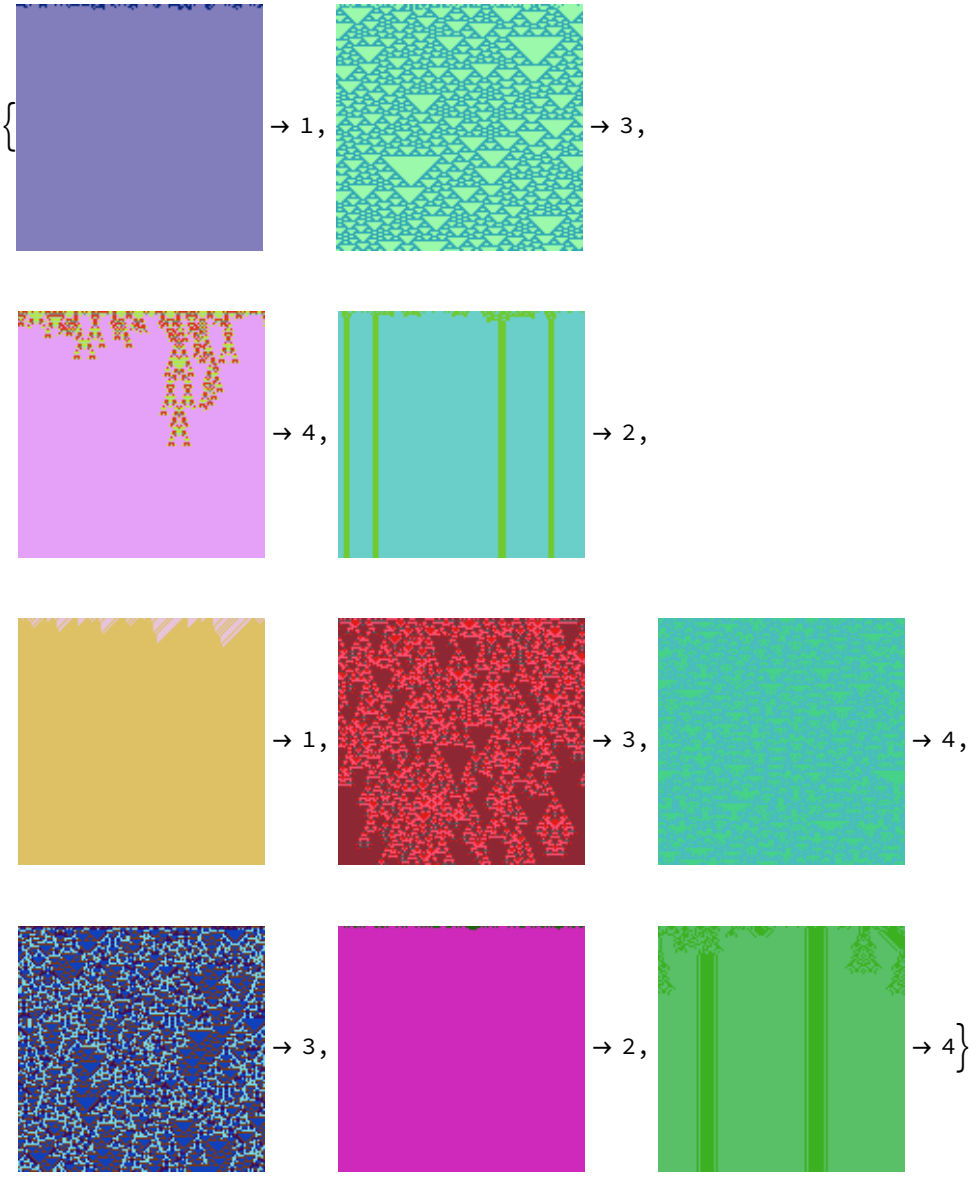
Out[76]=
```
{0.918164, <|1 → 0.84977, 2 → 0.88184, 3 → 0.921088, 4 → 0.962878|>,
```



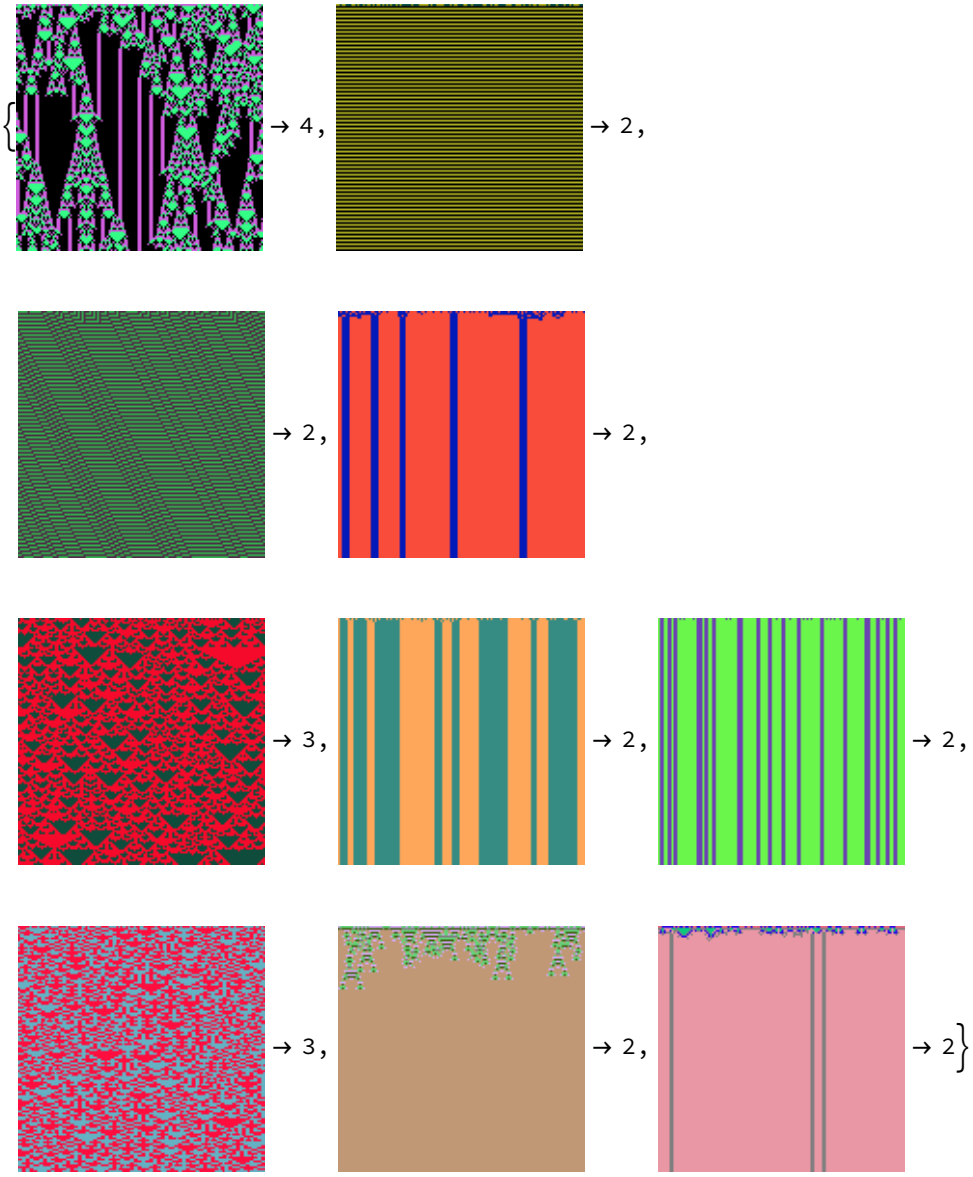

```
}
```

In[78]:=
```
entropyImagesBigC = RandomSample[Keys[fullTestSetBigC], 500];
entropiesBigC = netECA16[entropyImagesBigC, "Entropy"];
highEntBigC = entropyImagesBigC[[Ordering[entropiesBigC, -10]]];
lowEntBigC = entropyImagesBigC[[Ordering[entropiesBigC, 10]]];
Thread[highEntBigC → netECA16[highEntBigC]]
Thread[lowEntBigC → netECA16[lowEntBigC]]
```

Out[82]=



Out[83]=

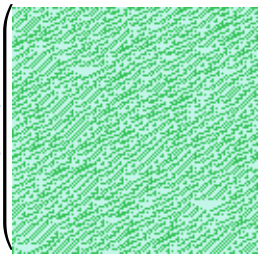


Testing Network XVI on unseen CA rule spaces

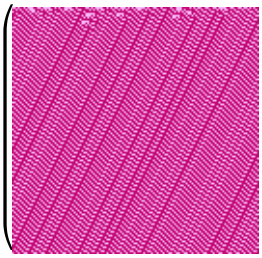
2-colour non-totalistic, range 2

In[84]:=
```
test4Data2kr2C16 = datak2r2C[128, 128, 8];
Thread[test4Data2kr2C16 → netECA16[Keys@test4Data2kr2C16, {"TopProbabilities", 2}]]
```

Out[85]=

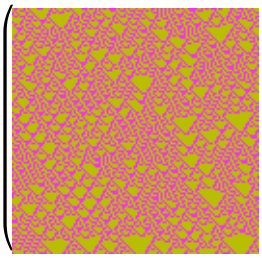


→ 142 978 078 → {4 → 0.0000385332, 3 → 0.999961},

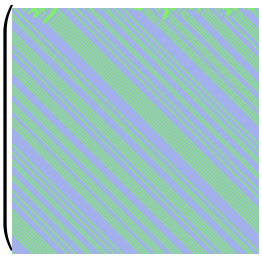


→ 2 651 048 833 → $\{4 \to 8.69455 \times 10^{-12}, 2 \to 1.\}$,

→ 2 132 867 963 → $\{4 \to 2.86202 \times 10^{-17}, 2 \to 1.\}$,

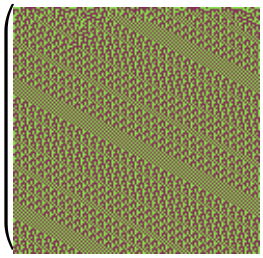


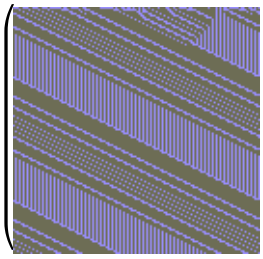
→ 3 644 758 968 → $\{4 \to 6.11899 \times 10^{-7}, 3 \to 0.999999\}$,



→ 1 762 420 096 → $\{1 \to 2.34707 \times 10^{-9}, 2 \to 1.\}$,


→ 1 983 429 391 → {4 → 0.0547227, 3 → 0.945277},

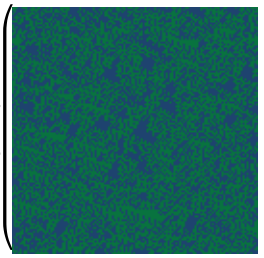

→ 3 013 553 323 → {3 → 0.0109364, 2 → 0.989064},


→ 4 222 841 103 → {4 → 0.406794, 3 → 0.593206}}

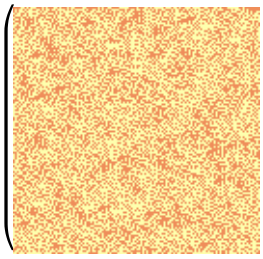
2-colour non-totalistic, range 3

In[88]:=
```
test4Data2kr3C16 = datak2r3NT[128, 128, 8];
Thread[test4Data2kr3C16 → netECA16[Keys@test4Data2kr3C16, {"TopProbabilities", 2}]]
```

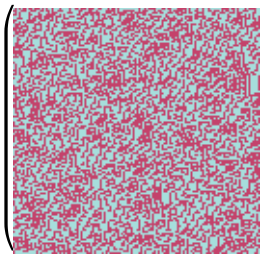
Out[89]=


→ 326 920 608 662 421 742 840 307 341 423 521 605 984 →

{4 → 0.250823, 3 → 0.749175},

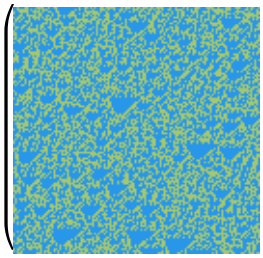

→ 207 378 363 536 986 723 396 466 725 374 349 770 886 →

$\{4 \to 3.99297 \times 10^{-14}, 3 \to 1.\},$



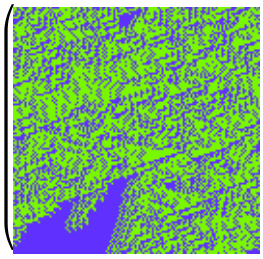
$\to 208\,879\,936\,260\,575\,914\,948\,311\,642\,807\,908\,540\,203 \to$

$\{4 \to 1.58015 \times 10^{-11}, 3 \to 1.\},$



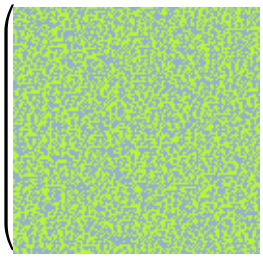
$\to 287\,306\,647\,832\,264\,924\,895\,832\,728\,648\,359\,695\,369 \to$

$\{4 \to 1.21845 \times 10^{-8}, 3 \to 1.\},$



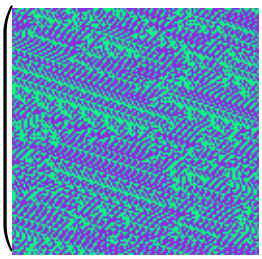
$\to 229\,589\,239\,336\,289\,801\,086\,527\,228\,211\,518\,173\,928 \to$

$\{3 \to 0.0173989, 4 \to 0.982601\},$



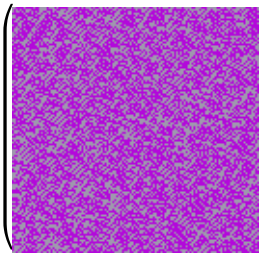
$\to 25\,263\,864\,036\,549\,856\,421\,518\,677\,057\,185\,002\,153 \to$

$\{4 \to 2.486 \times 10^{-11}, 3 \to 1.\},$



→ 122 573 209 377 430 367 268 072 050 742 932 826 218 →

$\{4 \to 1.46881 \times 10^{-9}, 3 \to 1.\}$,



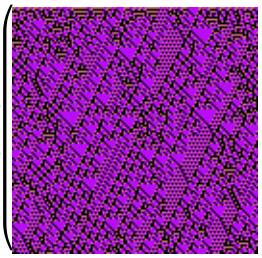
→ 120 291 014 848 530 508 056 036 077 228 147 332 137 →

$\{4 \to 0.00683298, 3 \to 0.993167\}\}$

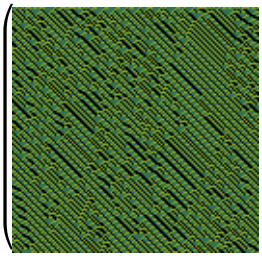
3-colour non-totalistic, range 1

In[92]:=
```
test4Data3kr1C16 = datak3r1NT[128, 128, 8];
Thread[test4Data3kr1C16 → netECA16[Keys@test4Data3kr1C16, {"TopProbabilities", 2}]]
```

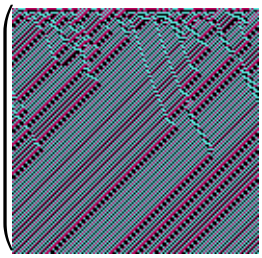
Out[93]=



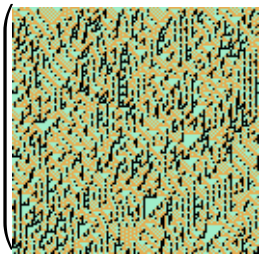
→ 4 431 477 695 805 → {4 → 0.000746188, 3 → 0.999254},



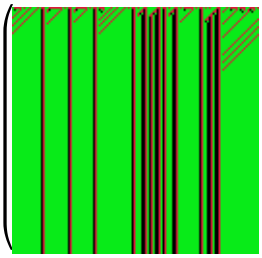
→ 1 627 958 441 874 → {4 → 0.00025369, 3 → 0.999746},



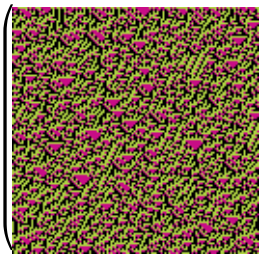
→ 4 241 674 451 024 → {3 → 0.194892, 2 → 0.805108},



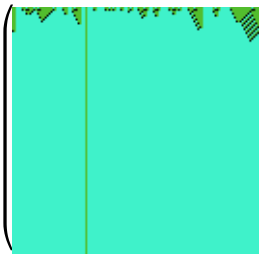
→ 4 177 916 755 057 → $\{3 \to 9.07174 \times 10^{-18}, 4 \to 1.\}$,



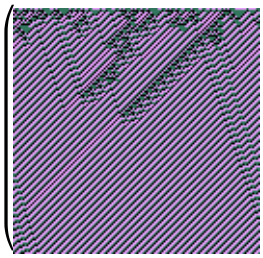
→ 2 504 235 138 103 → $\{4 \to 1.3375 \times 10^{-21}, 2 \to 1.\}$,



→ 2 281 646 033 785 → {4 → 0.164883, 3 → 0.835117},



→ 886 946 817 375 → {2 → 0.000352038, 1 → 0.999648},



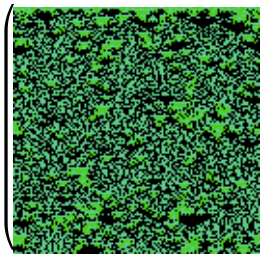
→ 1 468 391 136 669 → {4 → 1.85451 × 10^-11, 2 → 1.}

3-colour totalistic, range 2

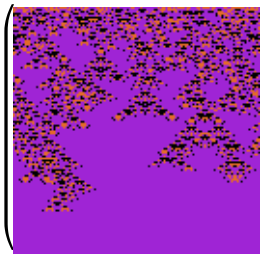
In[96]:=
```
test4Data3kr2C16 = datak3r2C[128, 128, 8];
Thread[test4Data3kr2C16 → netECA16[Keys@test4Data3kr2C16, {"TopProbabilities", 2}]]
```

Out[97]=

→ 31 794 → {2 → 0.0000274564, 1 → 0.999973},

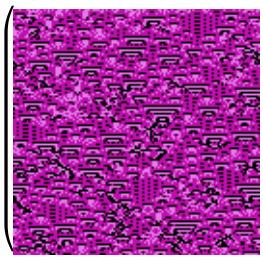


→ 154 722 → {4 → 3.53893 × 10^-6, 3 → 0.999996},

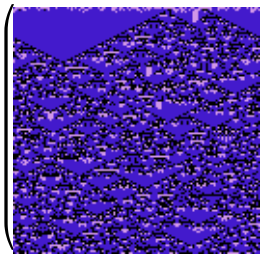


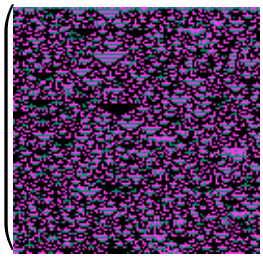
→ 47 093 → {3 → 0.0000226662, 4 → 0.999977},

→ 40 833 → {2 → 7.38958 × 10^-10, 1 → 1.},


→ 70 499 → {4 → 9.14168 × 10^−7, 3 → 0.999999},


→ 104 031 → {2 → 0.00469642, 1 → 0.995304},

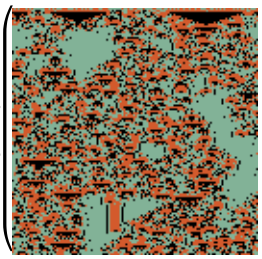

→ 102 416 → {4 → 0.00127827, 3 → 0.998722},

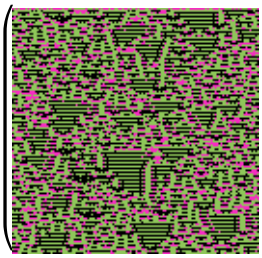

→ 143 812 → {3 → 0.00152229, 4 → 0.998478}}

3-colour totalistic, range 3

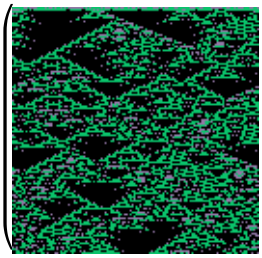
In[102]:=
```
test4Data3kr3C16 = datak3r3C[128, 128, 8];
Thread[test4Data3kr3C16 → netECA16[Keys@test4Data3kr3C16, {"TopProbabilities", 2}]]
```

Out[103]=

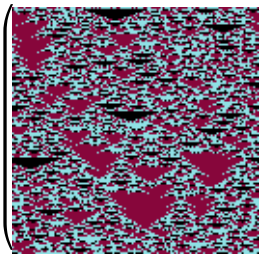

→ 9 694 493 → {3 → 0.480724, 4 → 0.519276},



→ 1 266 350 → {3 → 2.07073 × 10^-17, 4 → 1.},

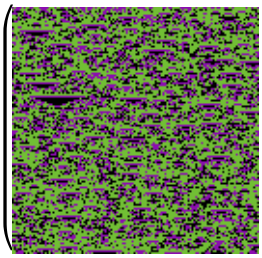


→ 10 922 251 → {4 → 0.0000302967, 3 → 0.99997},

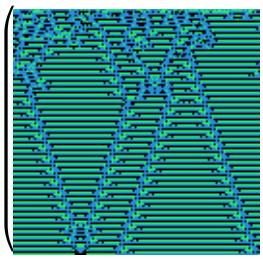


→ 10 284 081 → {4 → 0.0000121386, 3 → 0.999988},

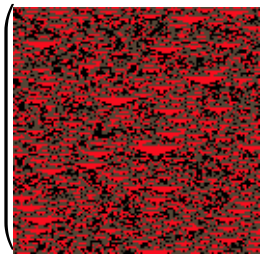
→ 3 664 255 → {1 → 0.0137727, 2 → 0.986227},



→ 10 298 881 → {4 → 0.000133186, 3 → 0.999867},



→ 8 621 297 → {3 → 5.17911 × 10^-10, 4 → 1.},

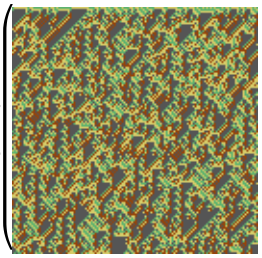


→ 11 392 150 → {4 → 8.14877 × 10^-14, 3 → 1.}}

4-colour non-totalistic, range 1

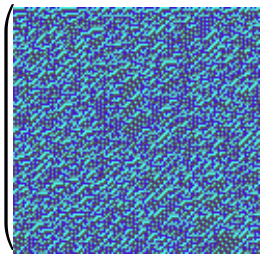
In[108]:=
```
test4Data4kr1C16 = datak4r1NT[128, 128, 8];
Thread[test4Data4kr1C16 → netECA16[Keys@test4Data4kr1C16, {"TopProbabilities", 2}]]
```

Out[109]=



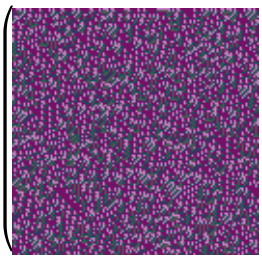
→ 154 080 220 988 097 676 666 997 866 552 654 645 473 →

{3 → 2.29402 × 10^-6, 4 → 0.999998},



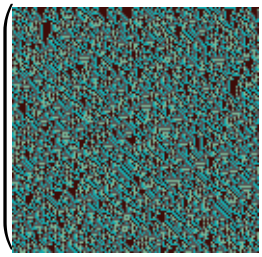
→ 23 699 455 383 307 433 676 732 305 926 546 946 154 →

{4 → 9.18698 × 10^-10, 3 → 1.},



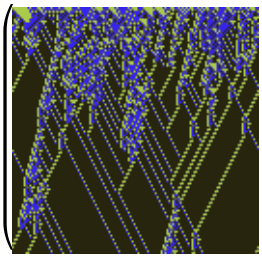
→ 128 473 503 291 830 537 978 738 932 215 015 934 320 →

{4 → 0.016884, 3 → 0.983116},



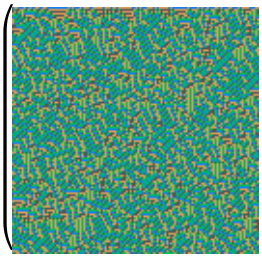
→ 229 632 577 374 973 873 063 824 352 556 638 470 668 →

$\{3 \to 3.66751 \times 10^{-6}, 4 \to 0.999996\}$,



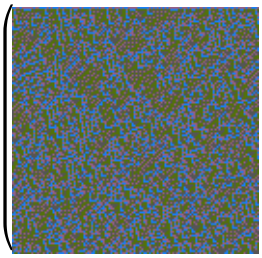
→ 296 751 325 089 499 848 049 286 011 129 052 334 280 →

{2 → 0.0356663, 4 → 0.964334},



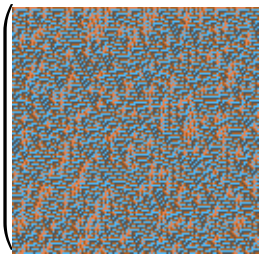
→ 58 389 931 388 288 805 968 545 394 973 250 265 077 →

{4 → 0.392533, 3 → 0.607467},



→ 190 838 234 746 017 730 793 117 442 629 783 928 461 →

{3 → 0.0000369307, 4 → 0.999963},



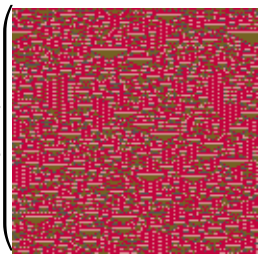
→ 203 920 243 365 735 713 208 234 812 082 196 839 807 →

{4 → 0.00577653, 3 → 0.994223}}

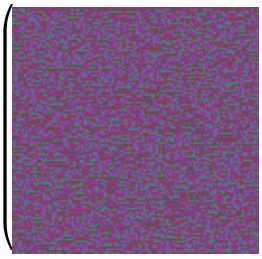
4-colour totalistic, range 2

In[112]:=
```
test4Data4kr2C16 = datak4r2C[128, 128, 8];
Thread[test4Data4kr2C16 → netECA16[Keys@test4Data4kr2C16, {"TopProbabilities", 2}]]
```

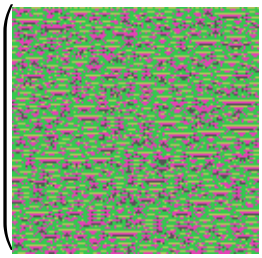
Out[113]=



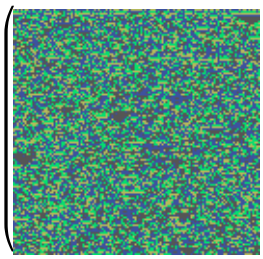
→ 25 517 204 → {4 → 1.29127 × 10^-13, 3 → 1.},



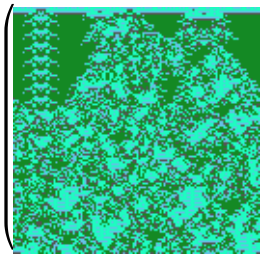
→ 3 053 925 273 → {4 → 0.00215091, 3 → 0.997849},



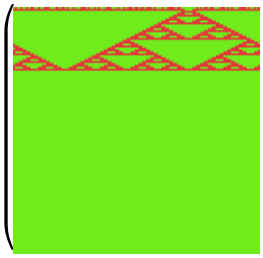
→ 2 735 868 989 → {4 → 0.000282149, 3 → 0.999718},



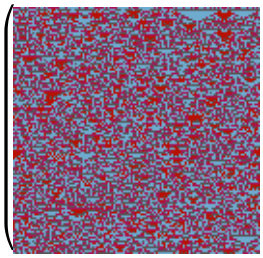
→ 1 440 927 950 → {4 → 0.0889018, 3 → 0.911098},



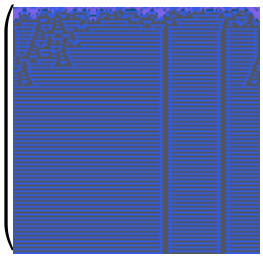
→ 3 727 816 705 → $\{3 \to 2.78599 \times 10^{-7}, 4 \to 1.\}$,



→ 818 963 457 → $\{4 \to 4.45422 \times 10^{-19}, 3 \to 1.\}$,



→ 3 948 742 311 → $\{4 \to 2.06369 \times 10^{-10}, 3 \to 1.\}$,

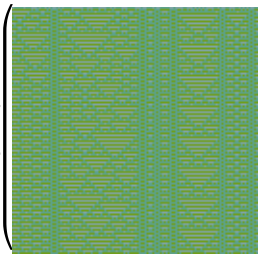


→ 1 159 108 912 → {4 → 0.431018, 2 → 0.568975}}

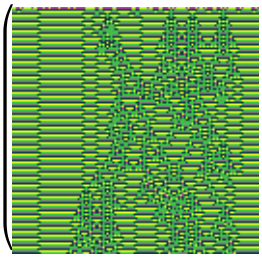
5-colour totalistic, range 1

In[116]:=
```
test4Data5kr1C16 = data5T2C[8, 128, 128];
Thread[test4Data5kr1C16 → netECA16[Keys@test4Data5kr1C16, {"TopProbabilities", 2}]]
```

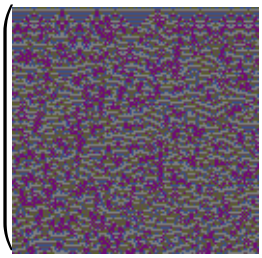
Out[117]=



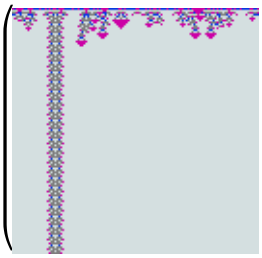
→ 81 353 109 → {4 → 4.44391 × 10^-7, 2 → 1.},



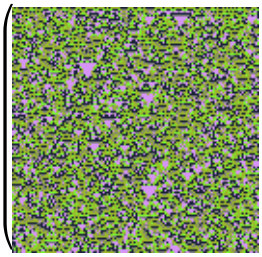
→ 626 536 724 → {3 → 3.46291 × 10^-8, 4 → 1.},



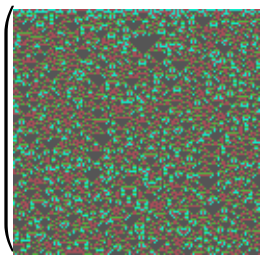
→ 129 595 314 → {4 → 0.00257287, 3 → 0.997427},

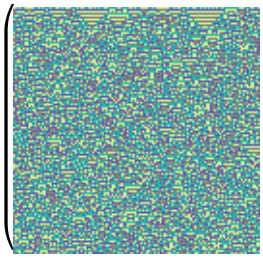


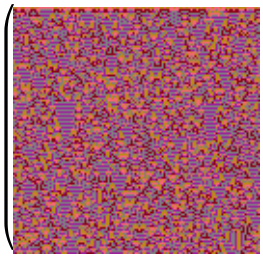
→ 513 885 470 → {1 → 1.41572 × 10^-11, 2 → 1.},



→ 494 894 021 → {3 → 0.0136503, 4 → 0.98635},


→ 459 414 632 → {4 → 0.0000813289, 3 → 0.999919},

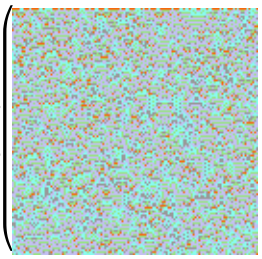

→ 327 281 621 → {4 → 2.29489 × 10^-8, 3 → 1.},

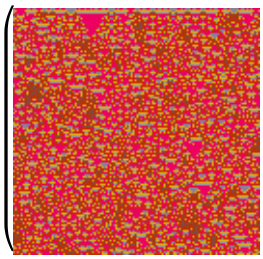

→ 503 044 041 → {4 → 0.191949, 3 → 0.808051}

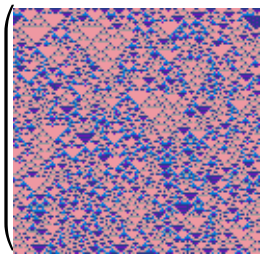
6-colour totalistic, range 1

In[118]:=
```
test4Data6kr1C16 = data6TC[8, 128, 128];
Thread[test4Data6kr1C16 → netECA16[Keys@test4Data6kr1C16, {"TopProbabilities", 2}]]
```

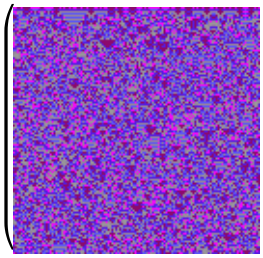
Out[119]=


→ 1 522 715 109 251 → {4 → 2.02852 × 10^-8, 3 → 1.},

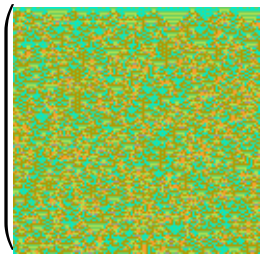

→ 1 026 953 898 330 → {4 → 2.88279 × 10^-8, 3 → 1.},



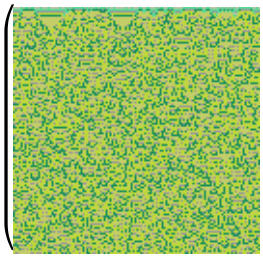
→ 1 583 652 682 → {3 → 0.429972, 4 → 0.570028},



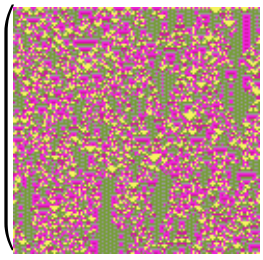
→ 2 123 073 201 165 → {4 → 6.23239 × 10^{-10}, 3 → 1.},



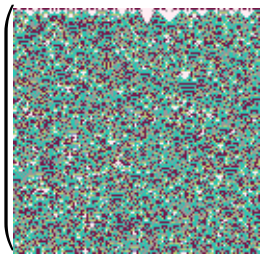
→ 341 591 565 791 → {4 → 0.00212154, 3 → 0.997878},



→ 2 568 539 246 083 → {4 → 0.00197237, 3 → 0.998028},



→ 1 213 730 554 155 → {4 → 0.0020281, 3 → 0.997972},

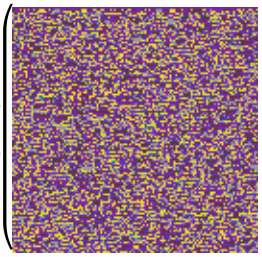


$\to 389\,841\,312\,036 \to \{4 \to 2.91693 \times 10^{-11},\ 3 \to 1.\}\Big\}$

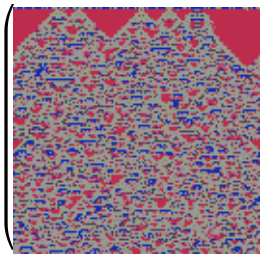
6-colour totalistic, range 2

In[138]:=
```
test4Data6kr2C16 = data6T2C[8, 128, 128];
Thread[test4Data6kr2C16 → netECA16[Keys@test4Data6kr2C16, {"TopProbabilities", 2}]]
```

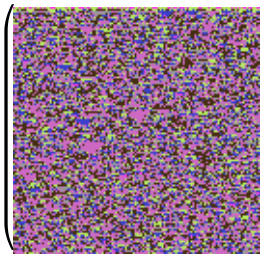
Out[139]=



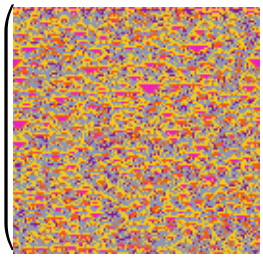
$\to 46\,177\,535\,535\,728\,053\,148 \to \{4 \to 6.75757 \times 10^{-8},\ 3 \to 1.\},$



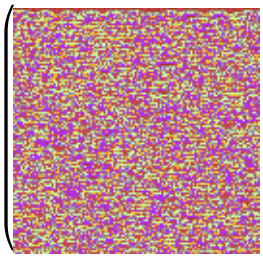
$\to 35\,643\,164\,656\,729\,746\,413 \to \{4 \to 4.68349 \times 10^{-15},\ 3 \to 1.\},$

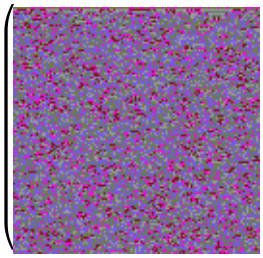


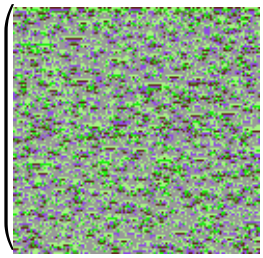
$\to 151\,294\,335\,263\,255\,298\,785 \to \{4 \to 0.0673459,\ 3 \to 0.932654\},$

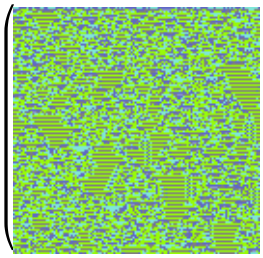


$\to 8\,803\,703\,818\,914\,948\,546 \to \{4 \to 0.00560205,\ 3 \to 0.994398\},$


→ 46 723 275 025 483 150 950 → {4 → 0.00307226, 3 → 0.996928},


→ 72 312 079 279 485 910 528 → {4 → 0.00153324, 3 → 0.998467},

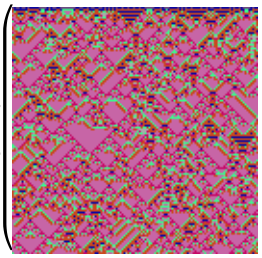

→ 22 158 237 683 799 083 047 → $\{4 \to 3.51784 \times 10^{-13}, 3 \to 1.\}$,

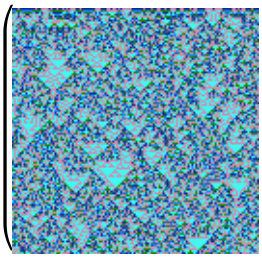

→ 142 446 781 366 136 429 283 → $\{4 \to 3.01302 \times 10^{-11}, 3 \to 1.\}$

7-colour totalistic, range 1

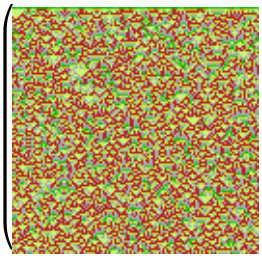
In[154]:=
```
test4Data7kr1C16 = data7TC[8, 128, 128];
Thread[test4Data7kr1C16 → netECA16[Keys@test4Data7kr1C16, {"TopProbabilities", 2}]]
```

Out[155]=

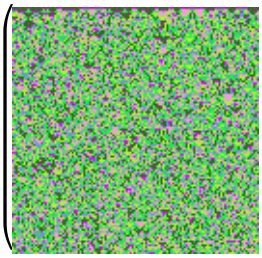

→ 3 109 608 593 887 262 → {2 → 0.0267983, 4 → 0.973202},



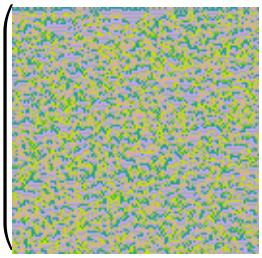
→ 10 516 337 788 191 339 → {4 → 0.202783, 3 → 0.797217},



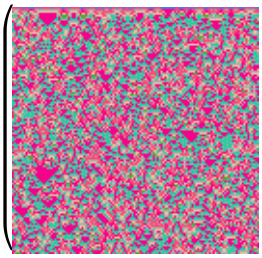
→ 10 218 434 972 470 056 → {4 → 2.59313 × 10^{-9}, 3 → 1.},



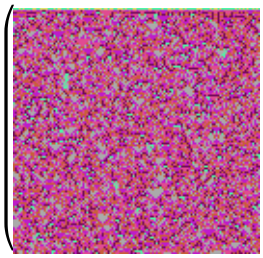
→ 11 301 098 979 433 534 → {4 → 5.31247 × 10^{-20}, 3 → 1.},

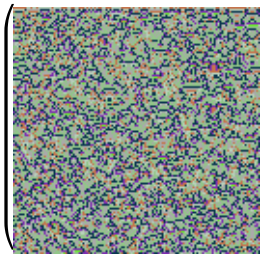


→ 4 222 218 586 098 008 → {4 → 2.3505 × 10^{-8}, 3 → 1.},



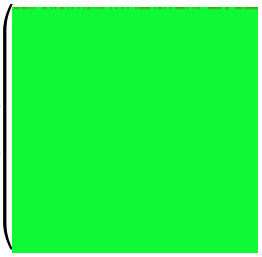
→ 6 272 918 110 265 620 → {4 → 2.82334 × 10^{-12}, 3 → 1.},

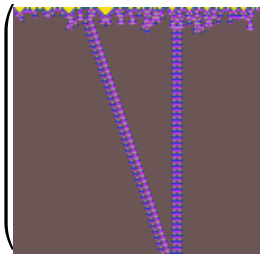

→ 7 237 960 144 206 103 → {4 → 1.82804 × 10^-11, 3 → 1.},

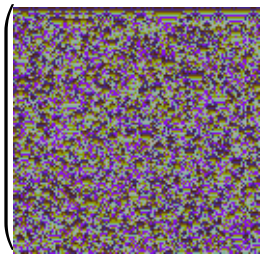

→ 8 157 411 943 012 914 → {4 → 7.30163 × 10^-8, 3 → 1.}}

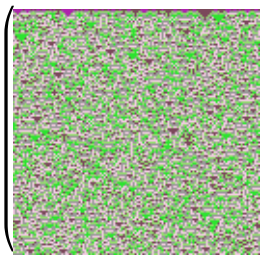
In[162]:=
```
test4Data7kr1C16 = data7TC[8, 128, 128];
Thread[test4Data7kr1C16 → netECA16[Keys@test4Data7kr1C16, {"TopProbabilities", 2}]]
```

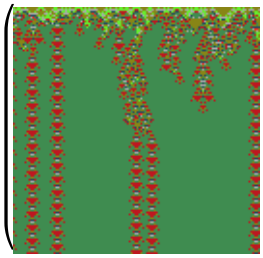
Out[163]=

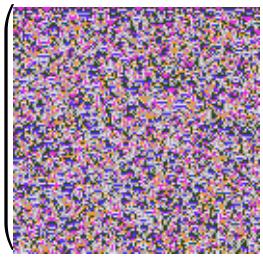

→ 2 054 187 704 193 738 → {2 → 1.00393 × 10^-7, 1 → 1.},

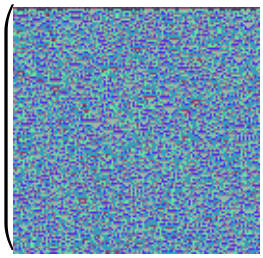

→ 4 502 314 670 347 259 → {1 → 0.000272502, 2 → 0.999727},

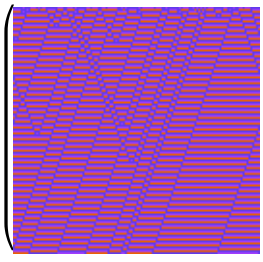

→ 6 433 286 718 439 853 → {4 → 3.57308 × 10^-13, 3 → 1.},


→ 10 115 271 094 201 812 → {4 → 1.83956 × 10^-14, 3 → 1.},


→ 2 056 629 839 849 700 → {4 → 7.03567 × 10^-6, 2 → 0.999993},

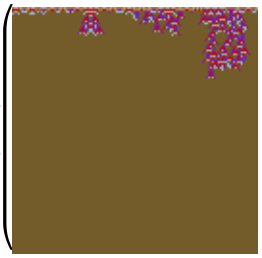

→ 6 016 684 767 156 829 → {4 → 0.0021258, 3 → 0.997874},


→ 1 150 898 749 617 983 → {4 → 5.05985 × 10^-9, 3 → 1.},

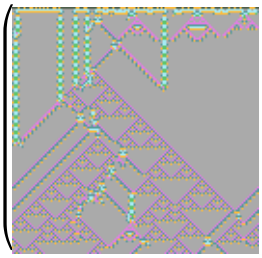

→ 3 441 885 208 643 463 → {3 → 1.57168 × 10^-8, 2 → 1.}}

In[170]:=
```
test4Data7kr1C16 = data7TC[8, 128, 128];
Thread[test4Data7kr1C16 → netECA16[Keys@test4Data7kr1C16, {"TopProbabilities", 2}]]
```

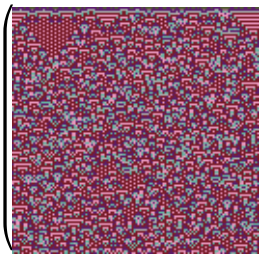
Out[171]=



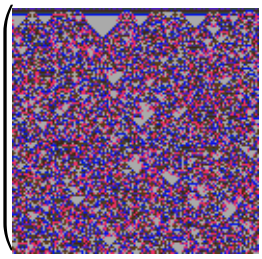
→ 8 718 538 805 570 808 → {4 → 0.0199047, 2 → 0.980095},



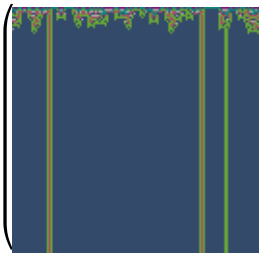
→ 5 687 458 247 703 346 → $\{3 \to 3.931 \times 10^{-6}, 4 \to 0.999995\}$,



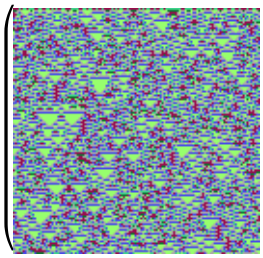
→ 2 004 300 484 518 722 → {3 → 0.0438658, 4 → 0.956134},



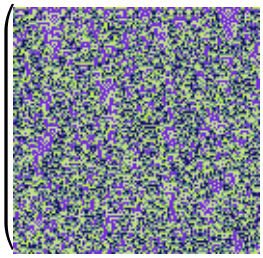
→ 2 106 485 862 858 275 → $\{4 \to 3.36807 \times 10^{-10}, 3 \to 1.\}$,



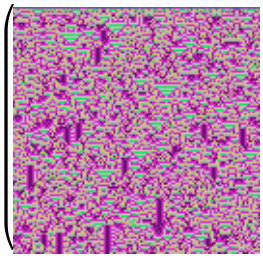
→ 10 335 102 717 390 268 → $\{4 \to 1.40275 \times 10^{-9}, 2 \to 1.\}$,



→ 1 948 304 137 097 478 → {4 → 3.0757 × 10^−6, 3 → 0.999997},



→ 742 855 573 082 935 → {3 → 0.00423487, 4 → 0.995765},

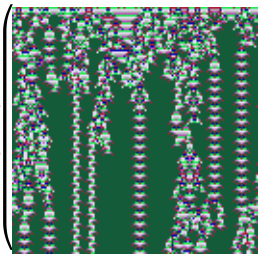


→ 550 991 988 954 034 → {3 → 0.0000158322, 4 → 0.999984}}

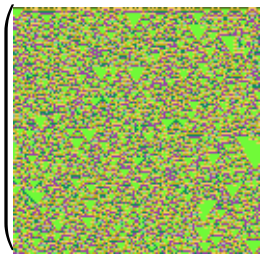
8-colour totalistic, range 1

In[174]:=
```
test4Data8kr1C16 = data8TC[8, 128, 128];
Thread[test4Data8kr1C16 → netECA16[Keys@test4Data8kr1C16, {"TopProbabilities", 2}]]
```

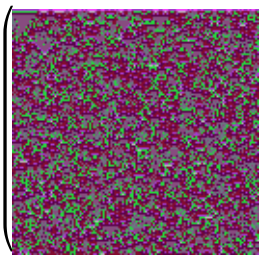
Out[175]=



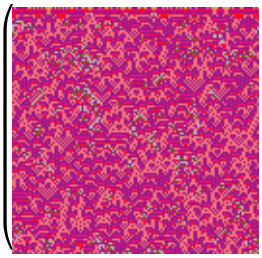
→ 26 660 220 729 300 317 631 → {3 → 4.53753 × 10^−16, 4 → 1.},



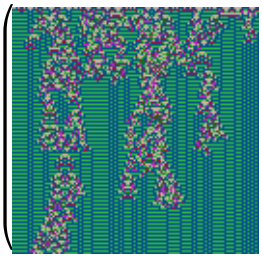
→ 57 706 619 077 455 169 987 → {4 → 0.00012086, 3 → 0.999879},



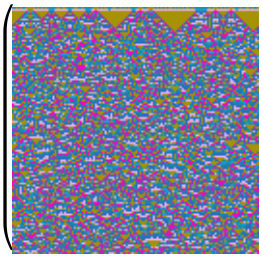
→ 64 248 301 738 433 598 883 → {4 → 8.62498 × 10^-7, 3 → 0.999999},



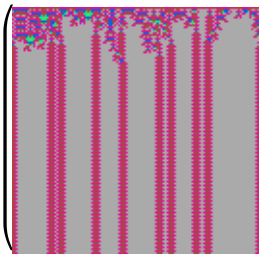
→ 38 309 191 234 358 472 181 → {3 → 0.0920227, 4 → 0.907977},



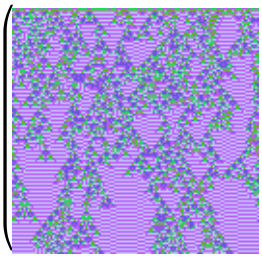
→ 10 057 418 236 647 939 786 → {3 → 0.00153869, 4 → 0.998461},



→ 55 038 816 396 722 824 044 → {4 → 7.93818 × 10^-11, 3 → 1.},

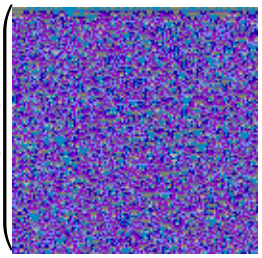


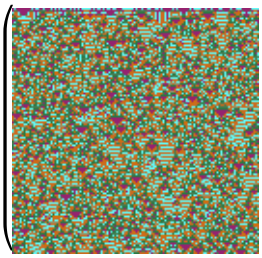
→ 13 857 790 822 319 662 750 → {4 → 1.6375 × 10^-9, 2 → 1.},

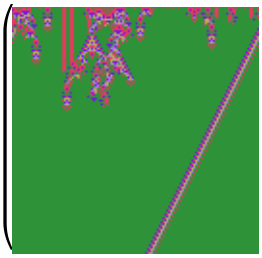

→ 35 001 739 471 058 241 746 → {3 → 0.146189, 4 → 0.853811}

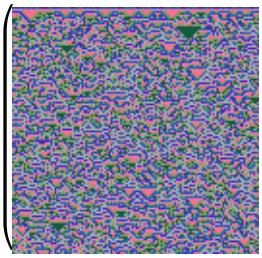
In[178]:=
```
test4Data8kr1C16 = data8TC[8, 128, 128];
Thread[test4Data8kr1C16 → netECA16[Keys@test4Data8kr1C16, {"TopProbabilities", 2}]]
```

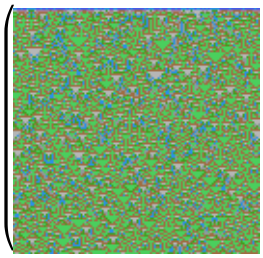
Out[179]=


→ 8 889 571 206 431 822 669 → {4 → 7.71651 × 10^-13, 3 → 1.},

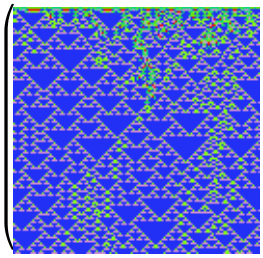

→ 12 932 107 158 159 577 869 → {4 → 0.000487127, 3 → 0.999513},


→ 38 300 014 541 797 689 408 → {2 → 4.40131 × 10^-9, 4 → 1.},

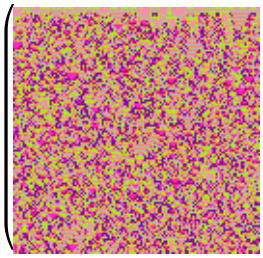

→ 73 619 662 786 582 031 542 → {4 → 2.6954 × 10^-22, 3 → 1.},



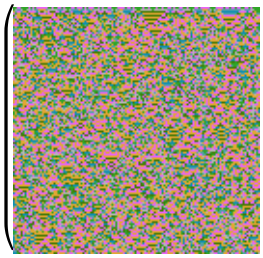
→ 25 075 664 454 379 326 631 → {4 → 2.7484 × 10^-6, 3 → 0.999997},



→ 66 251 301 754 800 867 134 → {4 → 0.045885, 3 → 0.954115},



→ 45 939 929 171 629 412 480 → {3 → 0.0015837, 4 → 0.998416},



→ 48 918 067 825 593 479 863 → {4 → 0.000601606, 3 → 0.999398}}

Network XVII - Four convolutions, dropout on linear only, BatchNorm

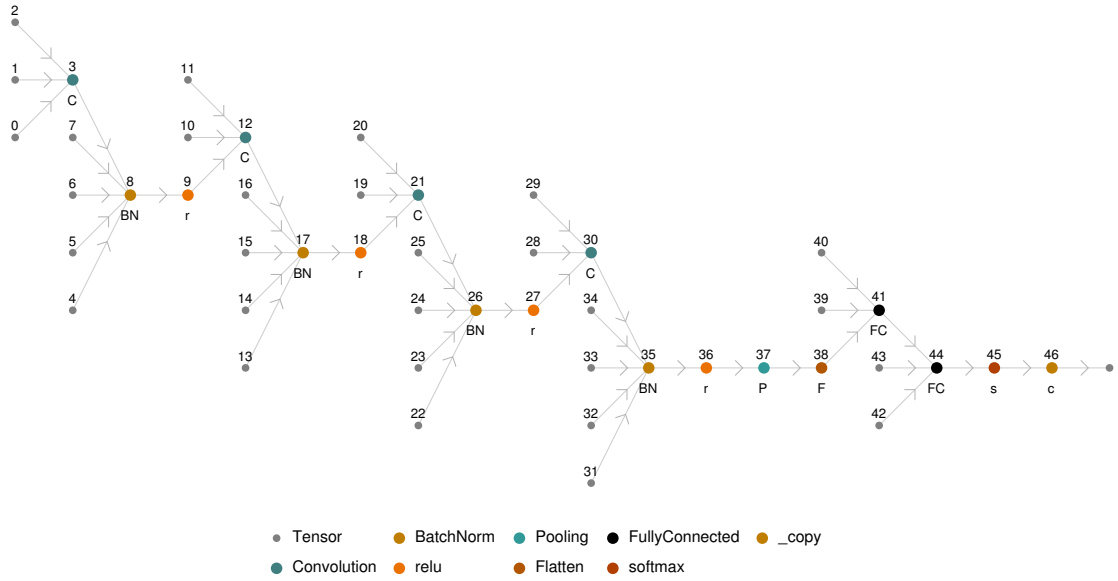
In[181]:=
```
netECA17 = netTenCC1024drop[128, 128]
```

Out[181]= NetChain

	Input	image array (size: 3 × 128 × 128)
conv1	ConvolutionLayer	array (size: 24 × 126 × 126)
bat1	BatchNormalizationLayer	array (size: 24 × 126 × 126)
ramp1	Ramp	array (size: 24 × 126 × 126)
conv2	ConvolutionLayer	array (size: 24 × 124 × 124)
bat2	BatchNormalizationLayer	array (size: 24 × 124 × 124)
ramp2	Ramp	array (size: 24 × 124 × 124)
conv3	ConvolutionLayer	array (size: 24 × 122 × 122)
bat3	BatchNormalizationLayer	array (size: 24 × 122 × 122)
ramp3	Ramp	array (size: 24 × 122 × 122)
conv4	ConvolutionLayer	array (size: 24 × 120 × 120)
bat4	BatchNormalizationLayer	array (size: 24 × 120 × 120)
ramp4	Ramp	array (size: 24 × 120 × 120)
pooling	PoolingLayer	array (size: 24 × 5 × 5)
flatten	FlattenLayer	vector (size: 600)
linear	LinearLayer	vector (size: 1024)
drop2	DropoutLayer	vector (size: 1024)
linear2	LinearLayer	vector (size: 4)
softmax	SoftmaxLayer	vector (size: 4)
	Output	class

In[182]:=
```
NetInformation[netECA17, "MXNetNodeGraphPlot"]
```

Out[182]=



In[183]:=
```
NetInformation[netECA17, "SummaryGraphic"]
```

Out[183]=

In[184]:=
```
dataECA17 = dataC[128, 128, 16 384];
```

In[185]:=
```
dataTotalistic2BigC17 = genData2r2C[128, 128, 2048];
```

In[186]:=
```
dataTotalistic3BigC17 = data3T2C[128, 128, 2048];
```

In[187]:=
```
dataTotalistic4BigC17 = data4TC[128, 128, 2048];
```

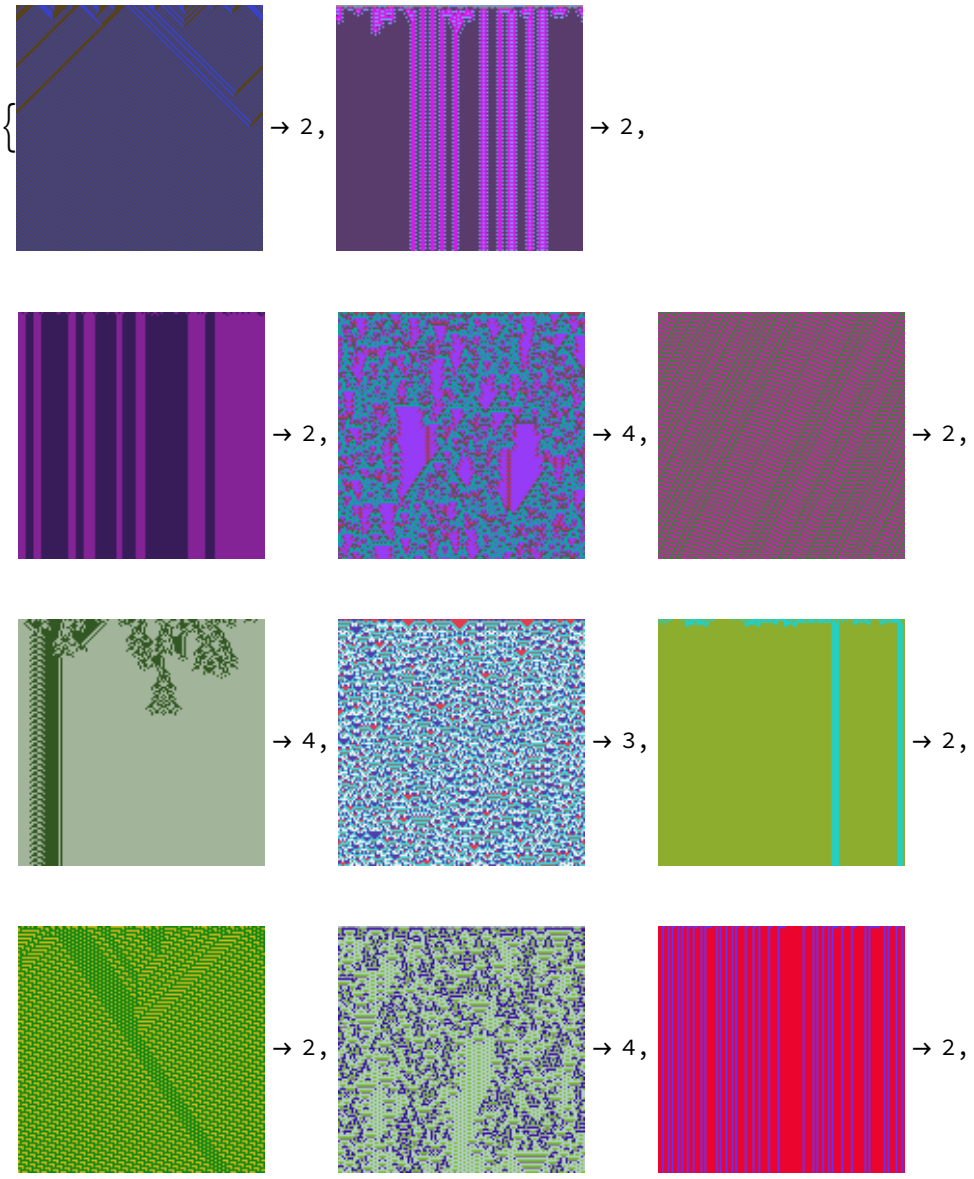
In[188]:=
```
dataTotalistic5BigC17 = genData5TCC[128, 128, 8192];
```

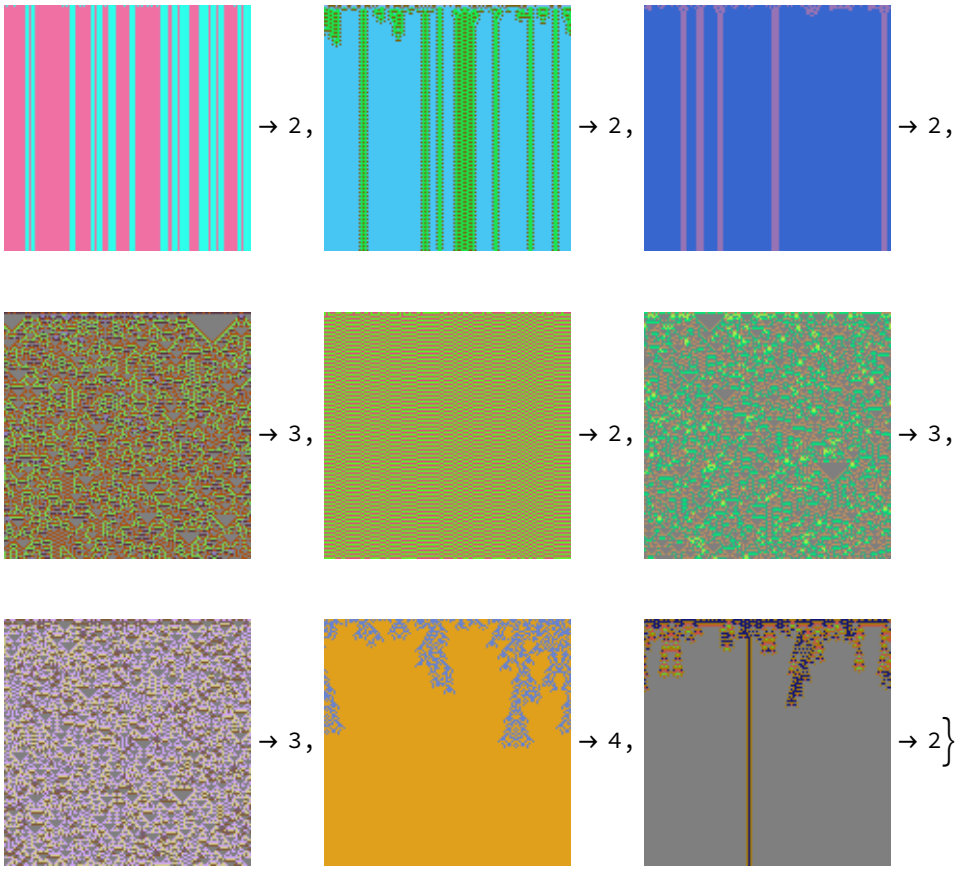
In[189]:=
```
fullTrainingBigC17 = Join[dataECA17, dataTotalistic2BigC17,
   dataTotalistic3BigC17, dataTotalistic4BigC17, dataTotalistic5BigC17];
Length[fullTrainingBigC17]
```

Out[190]= 53 248

In[191]:=
```
RandomSample[fullTrainingBigC17, 20]
```

Out[191]=





In[ ]:=
```
dir = SetDirectory[NotebookDirectory[]]
```

In[192]:=
```
"/home/esilverman/Documents"
```

Out[192]=
```
/home/esilverman/Documents
```

In[193]:=
```
netECA17 =
 NetTrain[netECA17, fullTrainingBigC17, MaxTrainingRounds → 200, BatchSize → 256,
  TargetDevice → "GPU", TrainingProgressCheckpointing → {"Directory", dir}]
```

Out[193]=
```
NetChain[ Input port: image
          Output port: class
          Number of layers: 18 ]
```

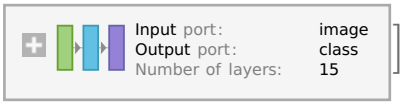
```
netECA17 = Import["netECA17-r200.wlnet"]
```

Generate test data for Network XVII (200 epochs)

In[ ]:=
```
dir = SetDirectory[NotebookDirectory[]]
```

Out[ ]=
```
/Users/thorsilver/Downloads/Wolfram notebooks
```

```
netECA17 = Import["netECA17-r200.wlnet"]
```

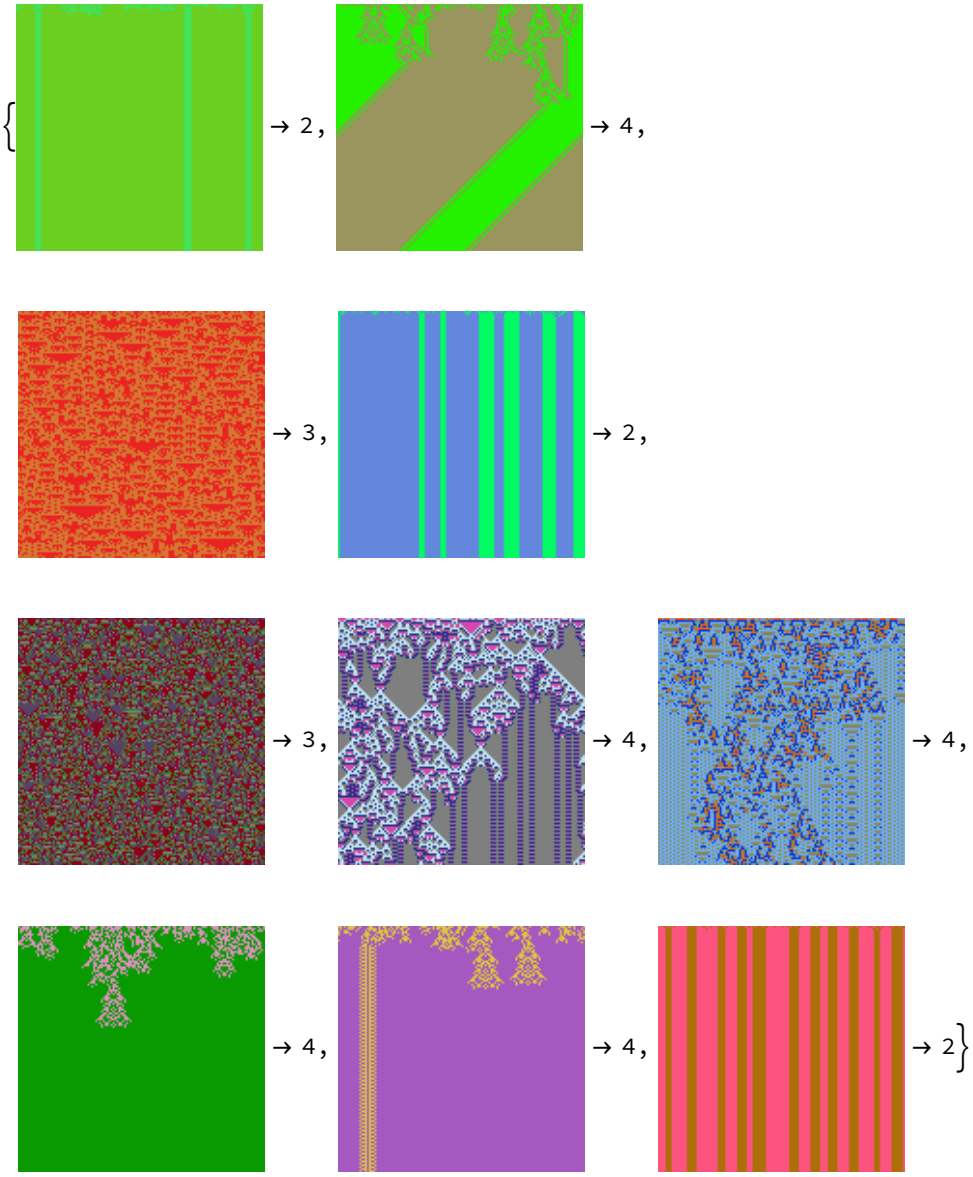
Out[ ]= NetChain[]

In[202]:=
```
testDataECABigC = dataC[128, 128, 1024];
testData2TBigC = genData2r2C[128, 128, 1024];
testData3TBigC = data3T2C[128, 128, 1024];
testData4TBigC = data4TC[128, 128, 1024];
testData5TBigC = genData5TCC[128, 128, 1024];
fullTestSetBigC = Join[testDataECABigC,
   testData2TBigC, testData3TBigC, testData4TBigC, testData5TBigC];
Length[fullTestSetBigC]
```

Out[208]= 10 240

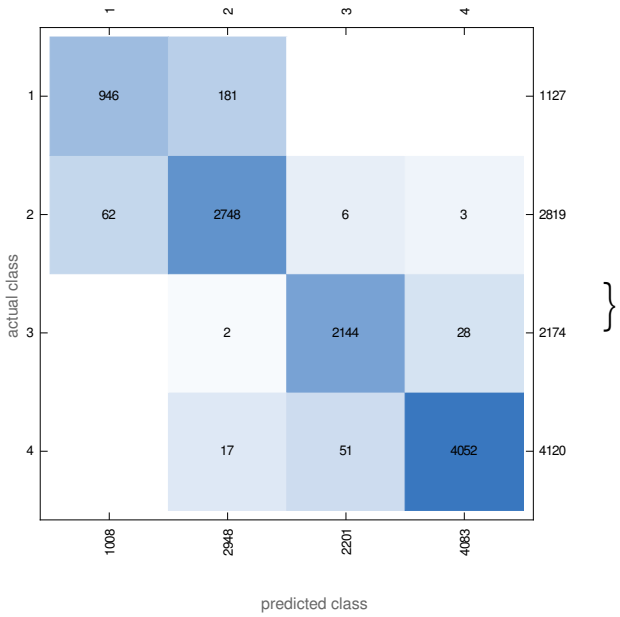
In[209]:= `RandomSample[fullTestSetBigC, 10]`

Out[209]=



In[217]:=
```
NetMeasurements[netECA17, fullTestSetBigC,
 {"Accuracy", "Precision", "ConfusionMatrixPlot"}]
```

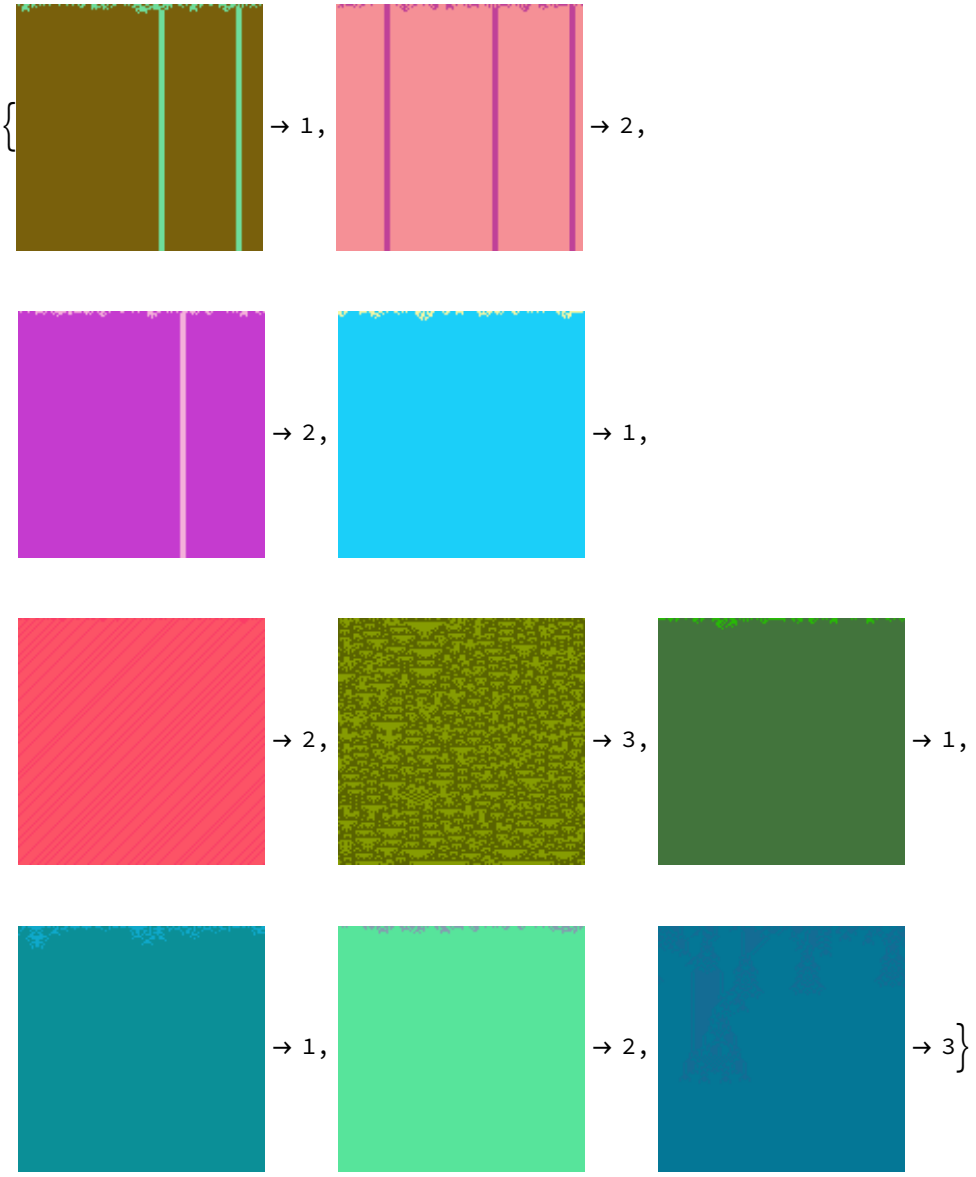
Out[217]=
```
{0.96582, <|1 → 0.938492, 2 → 0.932157, 3 → 0.974103, 4 → 0.992408|>,
```



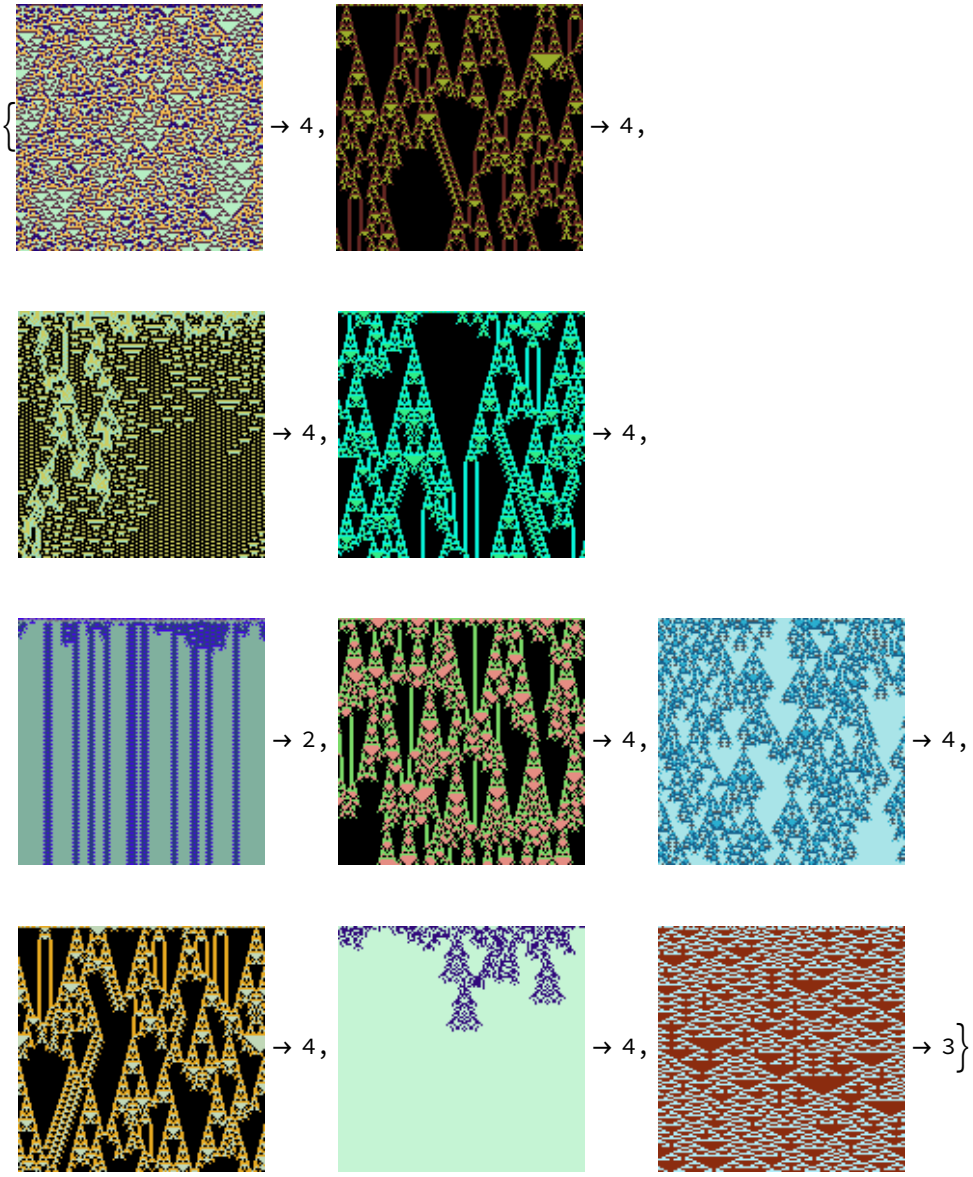

```
}
```

In[218]:=
```
entropyImagesBigC = RandomSample[Keys[fullTestSetBigC], 500];
entropiesBigC = netECA17[entropyImagesBigC, "Entropy"];
highEntBigC = entropyImagesBigC[[Ordering[entropiesBigC, -10]]];
lowEntBigC = entropyImagesBigC[[Ordering[entropiesBigC, 10]]];
Thread[highEntBigC → netECA17[highEntBigC]]
Thread[lowEntBigC → netECA17[lowEntBigC]]
```

Out[222]=



Out[223]=

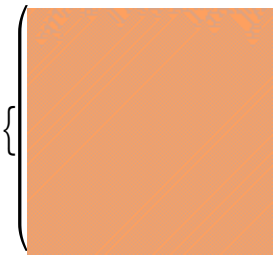


Testing Network XVII (200 epochs) on unseen CA rule spaces

2-colour non-totalistic, range 2

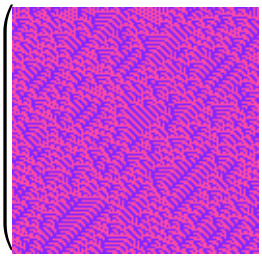
In[224]:=
```
test4Data2kr2C17 = datak2r2C[128, 128, 8];
Thread[test4Data2kr2C17 → netECA17[Keys@test4Data2kr2C17, {"TopProbabilities", 2}]]
```

Out[225]=

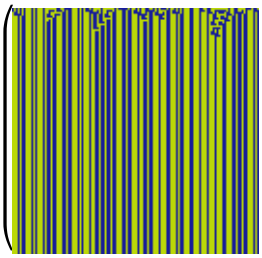


→ 3 594 886 935 → {3 → 1.19587 × 10^-7, 2 → 1.},

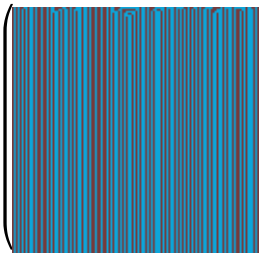
→ 4 012 014 789 → {4 → 0.00317589, 3 → 0.996824},



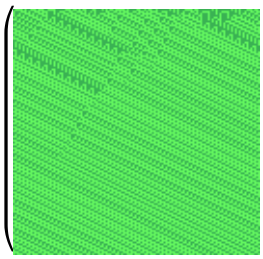
→ 736 342 145 → {4 → 0.000138652, 3 → 0.999861},



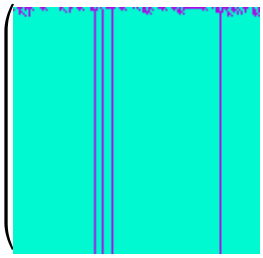
→ 3 597 938 931 → {4 → 5.42024 × 10^-16, 2 → 1.},



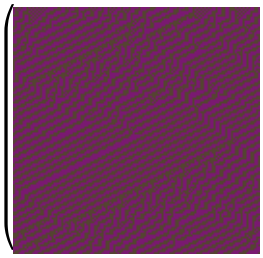
→ 49 406 137 → {1 → 4.03179 × 10^-30, 2 → 1.},



→ 669 500 034 → {4 → 0.0129747, 2 → 0.983657},



→ 4 122 605 661 → {1 → 6.18382 × 10^-9, 2 → 1.},

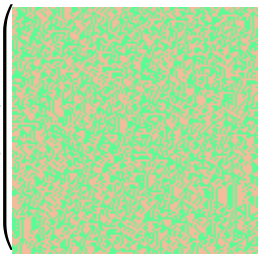


→ 2 751 618 302 → {4 → 5.91412 × 10^-9, 3 → 1.}

2-colour non-totalistic, range 3

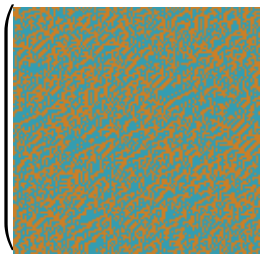
In[230]:=
```
test4Data2kr3C17 = datak2r3NT[128, 128, 8];
Thread[test4Data2kr3C17 → netECA17[Keys@test4Data2kr3C17, {"TopProbabilities", 2}]]
```

Out[231]=



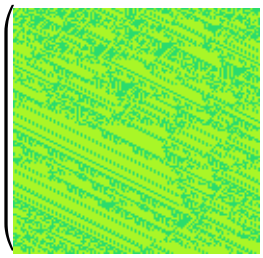
→ 34 540 309 799 423 528 179 267 302 306 223 835 543 →

{4 → 0.0000190167, 3 → 0.999981},



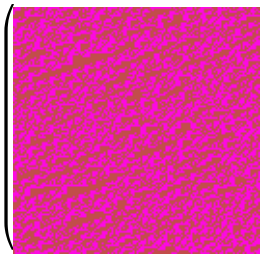
→ 271 913 994 962 168 934 532 188 611 494 783 284 369 →

$\{4 \to 8.79258 \times 10^{-15}, 3 \to 1.\},$



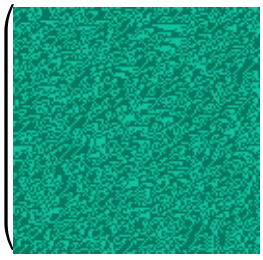
$\to 328\,899\,203\,265\,424\,040\,719\,063\,648\,351\,222\,904\,052 \to$

$\{3 \to 0.000609094, 4 \to 0.999391\},$



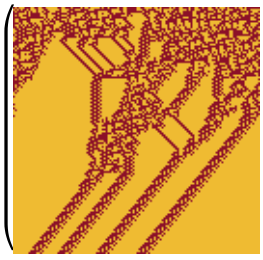
$\to 193\,952\,196\,319\,600\,964\,880\,345\,641\,761\,505\,849\,023 \to$

$\{4 \to 8.96571 \times 10^{-10}, 3 \to 1.\},$



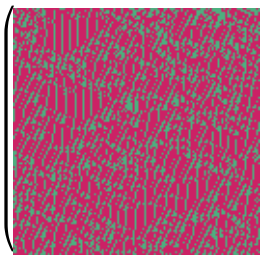
$\to 196\,656\,308\,173\,162\,097\,107\,061\,836\,715\,379\,270\,739 \to$

$\{4 \to 3.36397 \times 10^{-6}, 3 \to 0.999997\},$



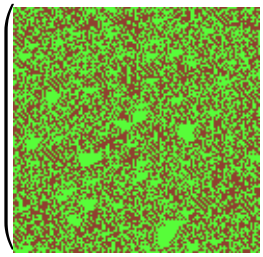
$\to 26\,038\,517\,878\,084\,050\,104\,120\,010\,197\,636\,533\,172 \to$

$\{3 \to 5.4757 \times 10^{-7}, 4 \to 0.999999\},$



→ 252 109 815 601 340 225 609 723 212 759 419 626 786 →

$\{4 \to 1.35911 \times 10^{-8}, 3 \to 1.\},$



→ 306 252 930 975 464 521 922 673 887 271 988 404 115 →

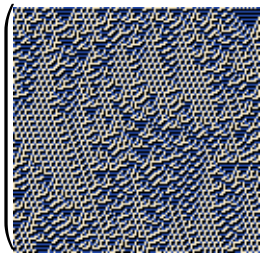
$\{4 \to 5.68649 \times 10^{-7}, 3 \to 0.999999\}\}$

3-colour non-totalistic, range 1

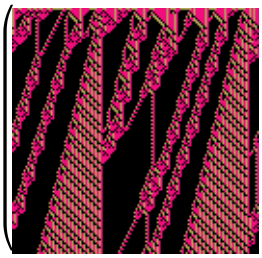
In[232]:=
```
test4Data3kr1C17 = datak3r1NT[128, 128, 8];
Thread[test4Data3kr1C17 → netECA17[Keys@test4Data3kr1C17, {"TopProbabilities", 2}]]
```

Out[233]=

→ 1 924 646 489 567 → $\{3 \to 1.76606 \times 10^{-30}, 2 \to 1.\},$

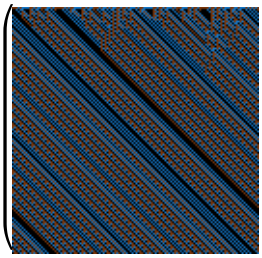


→ 3 672 534 501 071 → $\{2 \to 0.0000110699, 4 \to 0.999989\},$

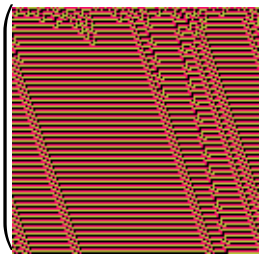


→ 5 833 330 297 781 → {2 → 0.000232935, 4 → 0.999767},

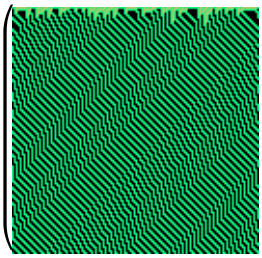
→ 7 606 192 973 798 → $\{2 \to 6.802 \times 10^{-10}, 1 \to 1.\}$,



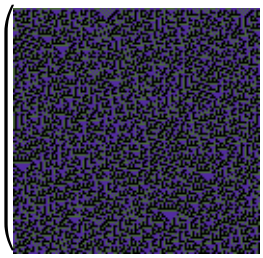
→ 7 622 301 560 954 → {3 → 0.0391643, 2 → 0.960836},



→ 3 685 910 174 297 → $\{3 \to 2.7602 \times 10^{-8}, 4 \to 1.\}$,



→ 5 743 838 876 456 → $\{1 \to 6.15406 \times 10^{-23}, 2 \to 1.\}$,

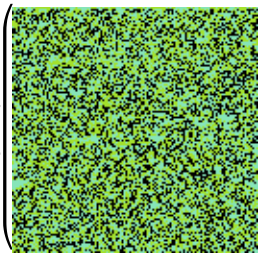


→ 3 655 040 152 191 → {4 → 0.0374305, 3 → 0.962569}

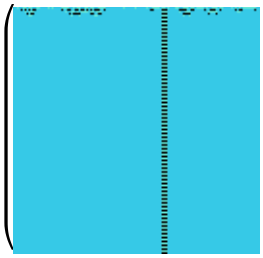
3-colour totalistic, range 2

In[234]:=
```
test4Data3kr2C17 = datak3r2C[128, 128, 8];
Thread[test4Data3kr2C17 → netECA17[Keys@test4Data3kr2C17, {"TopProbabilities", 2}]]
```

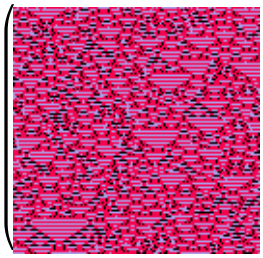
Out[235]=



→ 43 149 → {4 → 1.6989 × 10^−8, 3 → 1.},



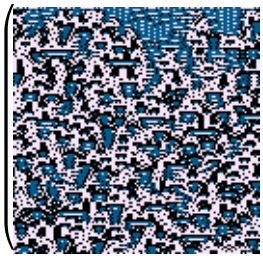
→ 160 434 → {1 → 2.05354 × 10^−18, 2 → 1.},

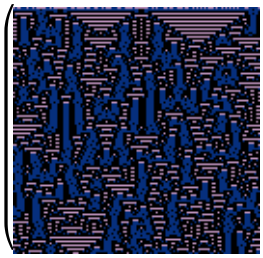


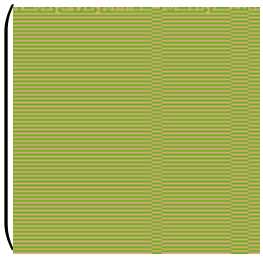
→ 9535 → {3 → 0.0000598239, 4 → 0.99994},

→ 172 239 → {1 → 0.00705153, 2 → 0.992948},


→ 174 680 → {2 → 5.824 × 10^-11, 1 → 1.},


→ 55 945 → {4 → 0.0138349, 3 → 0.986165},

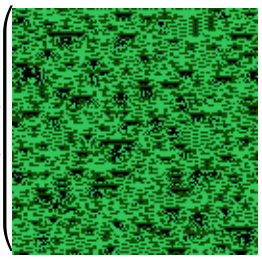

→ 113 483 → {4 → 3.72822 × 10^-6, 3 → 0.999996},

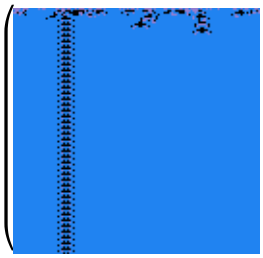

→ 67 810 → {1 → 6.91386 × 10^-17, 2 → 1.}

3-colour totalistic, range 3

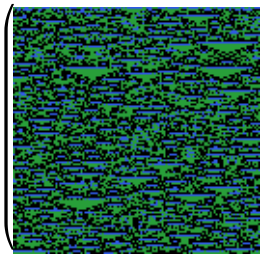
In[236]:=
```
test4Data3kr3C17 = datak3r3C[128, 128, 8];
Thread[test4Data3kr3C17 → netECA17[Keys@test4Data3kr3C17, {"TopProbabilities", 2}]]
```

Out[237]=

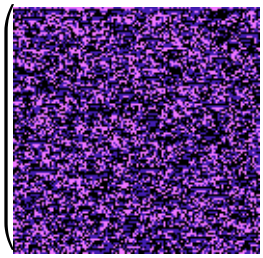

→ 3 046 610 → {4 → 7.58312 × 10^-7, 3 → 0.999999},



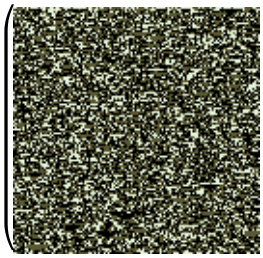
→ 7 801 434 → {1 → 1.19167 × 10^-14, 2 → 1.},



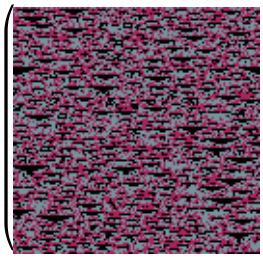
→ 5 445 843 → {4 → 1.60992 × 10^-19, 3 → 1.},



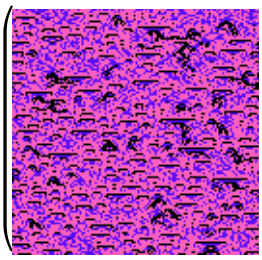
→ 1 451 413 → {4 → 0.144413, 3 → 0.855587},



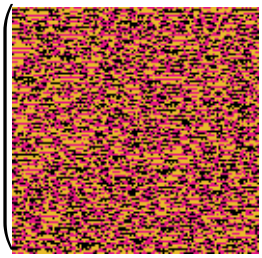
→ 10 676 790 → {3 → 0.0738921, 4 → 0.926108},



→ 10 375 449 → {4 → 1.04031 × 10^-17, 3 → 1.},



→ 5 181 761 → {4 → 1.75908 × 10^−8, 3 → 1.},

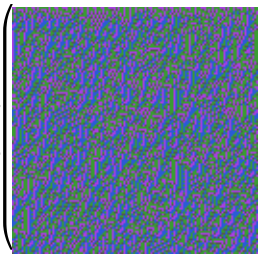


→ 8 884 285 → {4 → 7.91486 × 10^−6, 3 → 0.999992}}

4-colour non-totalistic, range 1

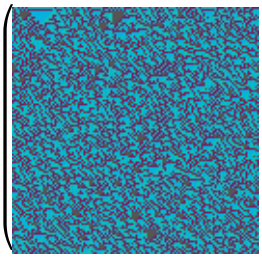
In[238]:=
```
test4Data4kr1C17 = datak4r1NT[128, 128, 8];
Thread[test4Data4kr1C17 → netECA17[Keys@test4Data4kr1C17, {"TopProbabilities", 2}]]
```

Out[239]=



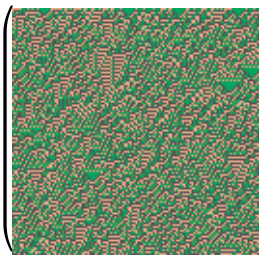
→ 255 219 118 556 246 495 764 448 982 135 818 252 673 →

{3 → 2.62807 × 10^−6, 4 → 0.999997},



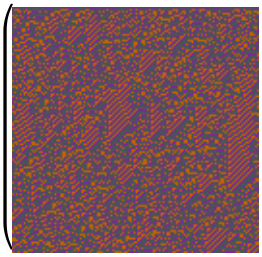
→ 256 372 744 774 750 994 462 308 116 064 689 670 029 →

{4 → 1.66442 × 10^−17, 3 → 1.},



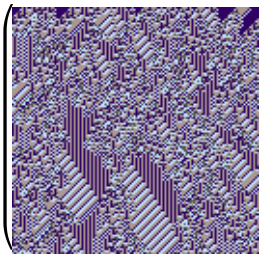
→ 287 015 413 736 003 568 798 718 608 141 279 906 438 →

$\{4 \to 2.56385 \times 10^{-7}, 3 \to 1.\}$,



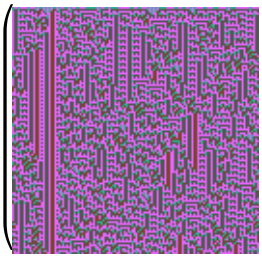
→ 102 047 809 483 228 794 436 919 245 364 544 665 433 →

{4 → 0.000696463, 3 → 0.999304},



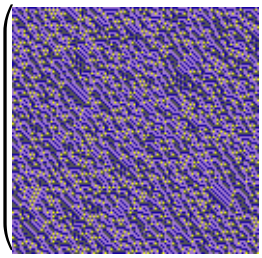
→ 240 555 046 861 732 528 519 285 194 385 835 154 772 →

$\{4 \to 3.62136 \times 10^{-16}, 3 \to 1.\}$,



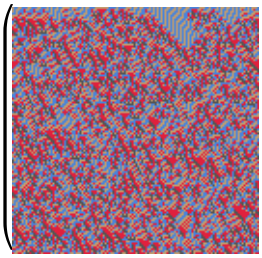
→ 16 370 468 900 338 540 956 022 703 184 589 781 262 →

$\{3 \to 1.45252 \times 10^{-15}, 4 \to 1.\}$,



→ 121 099 422 536 966 135 180 998 722 819 969 755 137 →

{4 → 0.0000270873, 3 → 0.999973},



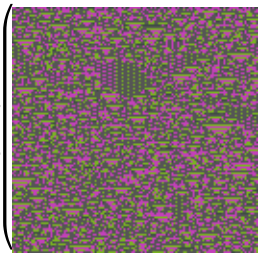
→ 264 867 193 286 423 906 380 401 526 070 744 380 919 →

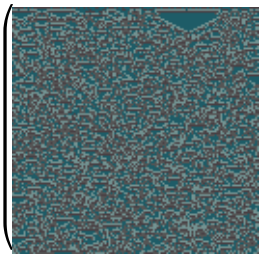
{4 → 0.105214, 3 → 0.894786}}

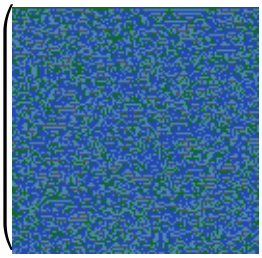
4-colour totalistic, range 2

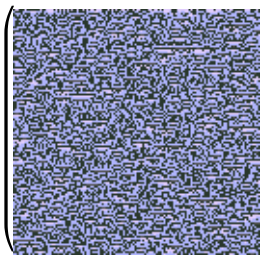
In[262]:=
```
test4Data4kr2C17 = datak4r2C[128, 128, 8];
Thread[test4Data4kr2C17 → netECA17[Keys@test4Data4kr2C17, {"TopProbabilities", 2}]]
```

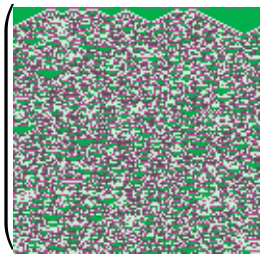
Out[263]=

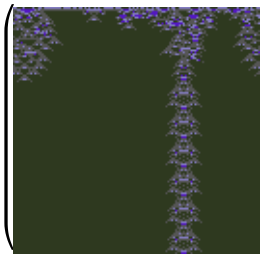
{  → 616 082 315 → {4 → 8.8653 × 10^-10, 3 → 1.},

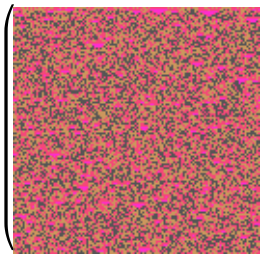
 → 1 568 191 428 → {4 → 5.21264 × 10^-11, 3 → 1.},

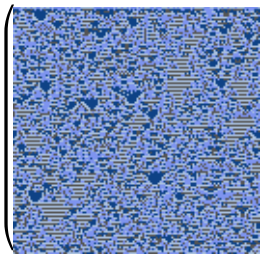
 → 1 216 110 065 → {4 → 0.000040346, 3 → 0.99996},


→ 2 419 903 949 → {4 → 3.69897 × 10^-10, 3 → 1.},


→ 453 961 055 → {4 → 3.89961 × 10^-8, 3 → 1.},


→ 3 969 029 930 → {2 → 0.0000237283, 4 → 0.999976},

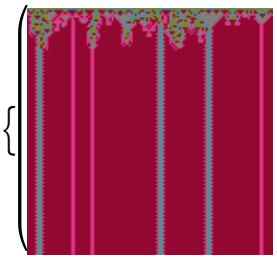

→ 3 970 208 335 → {4 → 4.03487 × 10^-9, 3 → 1.},


→ 3 418 384 250 → {4 → 9.0126 × 10^-15, 3 → 1.}}

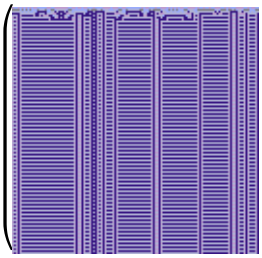
5-colour totalistic, range 1

In[242]:=
```
test4Data5kr1C17 = data5T2C[8, 128, 128];
Thread[test4Data5kr1C17 → netECA17[Keys@test4Data5kr1C17, {"TopProbabilities", 2}]]
```

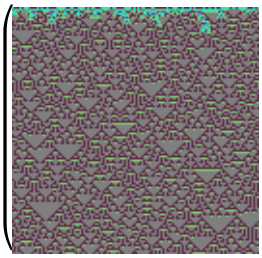
Out[243]=



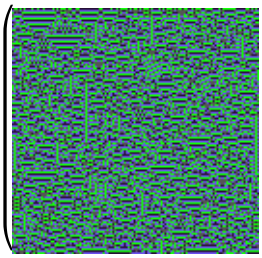
→ 720 503 516 → {2 → 0.000105936, 4 → 0.999894},



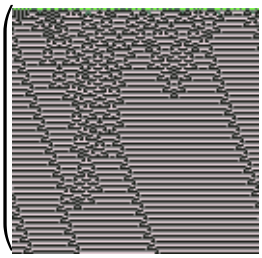
→ 771 013 684 → $\{4 \to 1.94282 \times 10^{-8}, 2 \to 1.\}$,



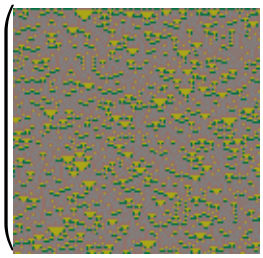
→ 543 872 434 → $\{4 \to 6.11423 \times 10^{-7}, 3 \to 0.999999\}$,



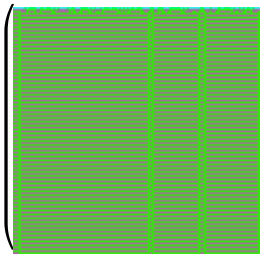
→ 341 908 586 → {4 → 0.310854, 3 → 0.689146},



→ 664 036 861 → {2 → 0.00511847, 4 → 0.994882},



→ 1 182 110 899 → {4 → 0.039023, 3 → 0.960977},



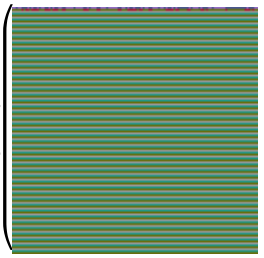
→ 976 082 949 → {4 → 9.09593 × 10^-19, 2 → 1.},

→ 1 019 517 181 → {1 → 6.47917 × 10^-10, 2 → 1.}}

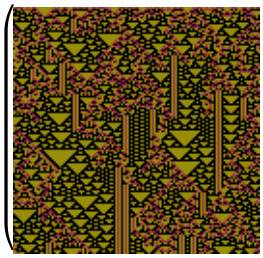
6-colour totalistic, range 1

In[244]:=
```
test4Data6kr1C17 = data6TC[8, 128, 128];
Thread[test4Data6kr1C17 → netECA17[Keys@test4Data6kr1C17, {"TopProbabilities", 2}]]
```

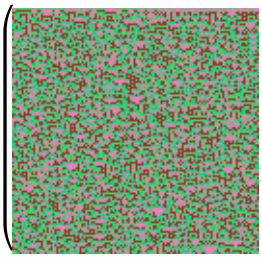
Out[245]=



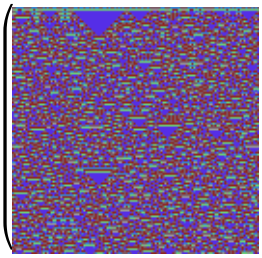
→ 1 128 957 409 115 → {2 → 0.387573, 1 → 0.612427},



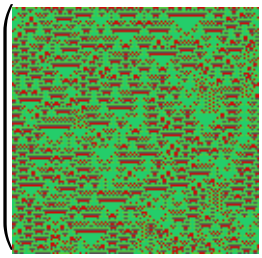
→ 744 151 919 694 → {3 → 1.28454 × 10^-11, 4 → 1.},



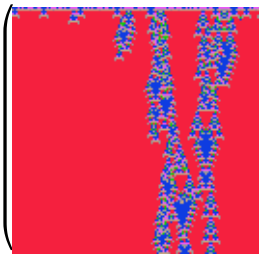
→ 411 482 269 593 → {4 → 9.50671 × 10^-6, 3 → 0.99999},



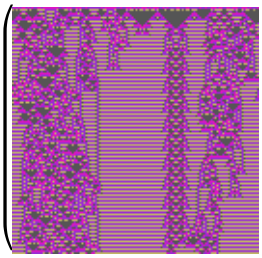
→ 2 122 826 252 429 → {4 → 4.58698 × 10^-10, 3 → 1.},



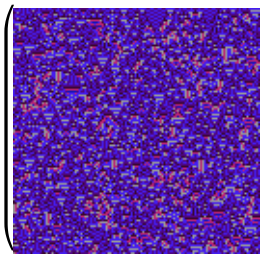
→ 2 443 710 325 124 → {4 → 5.97811 × 10^-9, 3 → 1.},



→ 2 519 595 515 832 → {2 → 0.396179, 4 → 0.603821},



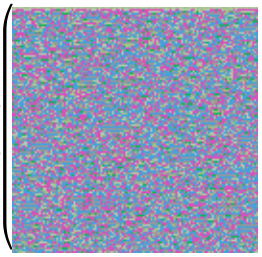
→ 572 558 234 379 → {4 → 1.63969 × 10^-11, 3 → 1.},

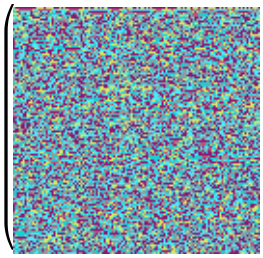

→ 2 103 617 145 248 → {3 → 8.41391 × 10^-9, 4 → 1.}

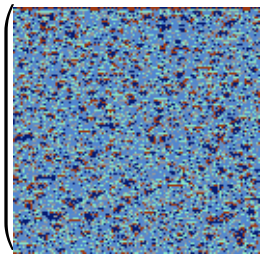
6-colour totalistic, range 2

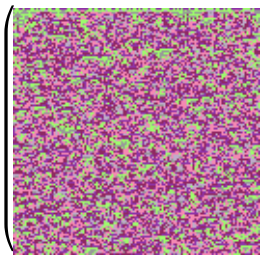
In[246]:=
```
test4Data6kr2C17 = data6T2C[8, 128, 128];
Thread[test4Data6kr2C17 → netECA17[Keys@test4Data6kr2C17, {"TopProbabilities", 2}]]
```

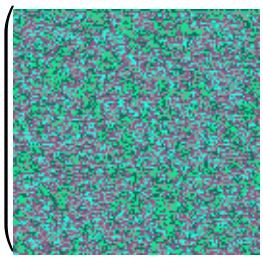
Out[247]=

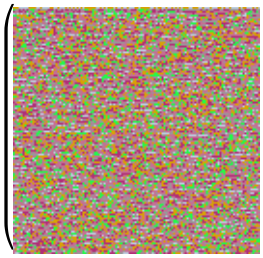

→ 56 518 611 906 036 513 420 → {4 → 0.00132297, 3 → 0.998677},

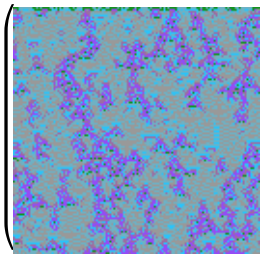

→ 149 111 145 742 470 978 404 → {4 → 0.00110016, 3 → 0.9989},

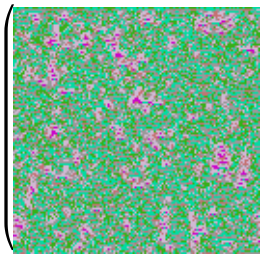

→ 60 075 298 400 874 491 559 → {4 → 0.0000816385, 3 → 0.999918},


→ 61 137 219 885 741 406 688 → {4 → 6.01526 × 10^-9, 3 → 1.},


→ 138 083 937 800 052 503 915 → {4 → 1.63338 × 10^-6, 3 → 0.999998},


→ 102 848 890 668 267 918 696 → {4 → 4.51684 × 10^-8, 3 → 1.},

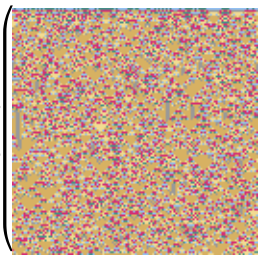

→ 52 002 759 529 482 240 344 → {4 → 0.0161382, 3 → 0.983862},

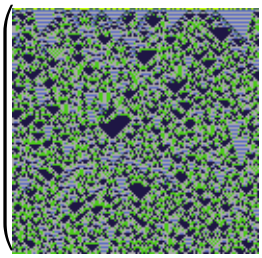

→ 3 771 326 190 903 203 597 → {4 → 1.57635 × 10^-10, 3 → 1.}}

7-colour totalistic, range 1

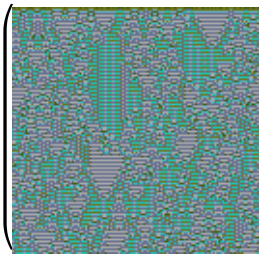
In[248]:=
```
test4Data7kr1C17 = data7TC[8, 128, 128];
Thread[test4Data7kr1C17 → netECA17[Keys@test4Data7kr1C17, {"TopProbabilities", 2}]]
```

Out[249]=

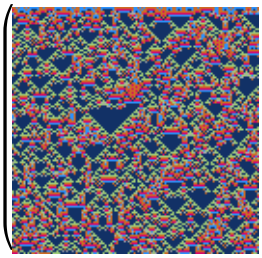

→ 249 739 897 876 317 → {4 → 7.58753 × 10^-13, 3 → 1.},



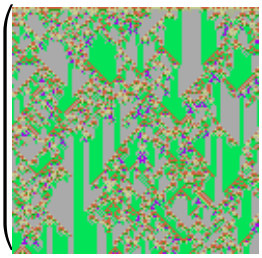
→ 6 589 873 174 284 234 → {4 → 3.70203 × 10^-21, 3 → 1.},



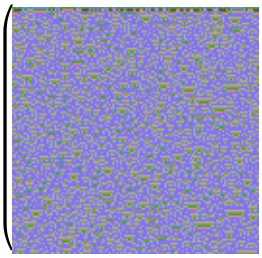
→ 2 838 251 451 633 386 → {3 → 0.0000362001, 4 → 0.999964},



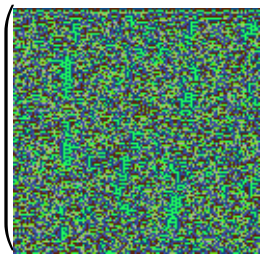
→ 3 069 021 856 393 877 → {4 → 4.6982 × 10^-6, 3 → 0.999995},



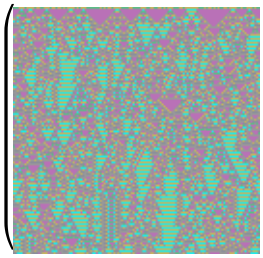
→ 10 282 712 720 317 214 → {3 → 4.14045 × 10^-19, 4 → 1.},



→ 203 015 413 423 084 → {4 → 2.87431 × 10^-9, 3 → 1.},



→ 9 746 864 148 555 591 → {3 → 1.53822 × 10^-7, 4 → 1.},

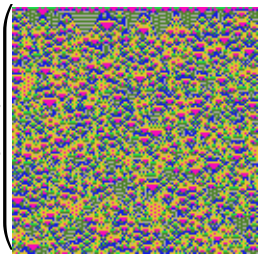


→ 8 003 062 538 664 128 → {3 → 2.05185 × 10^-8, 4 → 1.}}

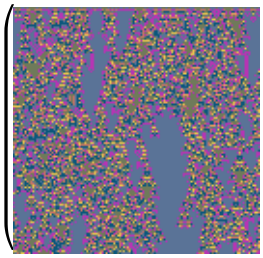
8-colour totalistic, range 1

In[250]:=
```
test4Data8kr1C17 = data8TC[8, 128, 128];
Thread[test4Data8kr1C17 → netECA17[Keys@test4Data8kr1C17, {"TopProbabilities", 2}]]
```

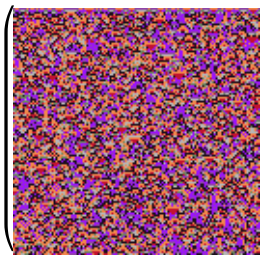
Out[251]=



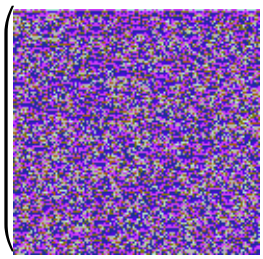
→ 10 266 196 594 935 096 075 → {4 → 8.5791 × 10^-18, 3 → 1.},



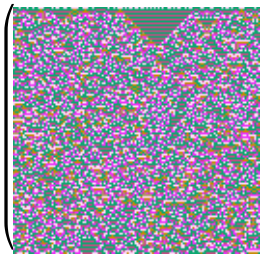
→ 731 338 973 990 773 560 → {2 → 7.53019 × 10^-12, 4 → 1.},



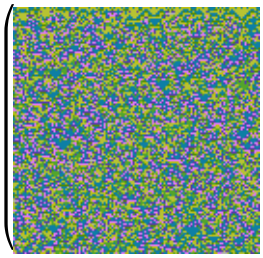
→ 11 247 173 012 174 218 620 → {4 → 0.0000380778, 3 → 0.999962},



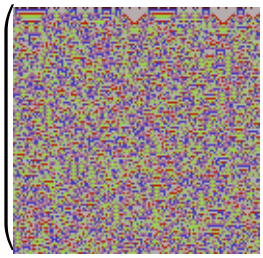
→ 63 742 472 032 617 219 918 → {4 → 0.0000371126, 3 → 0.999963},



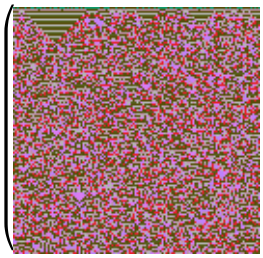
→ 7 382 455 380 800 363 015 → {4 → 8.07468 × 10^-15, 3 → 1.},



→ 59 100 651 667 569 734 000 → {4 → 1.27228 × 10^-11, 3 → 1.},



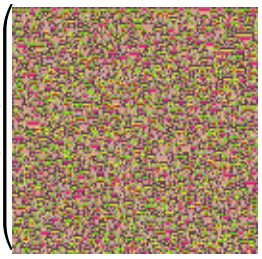
→ 24 971 306 247 396 766 335 → {4 → 0.0333734, 3 → 0.966627},

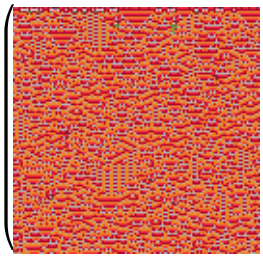


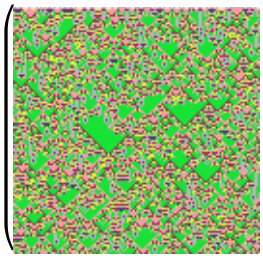
→ 45 946 581 080 593 555 746 → {4 → 1.08598 × 10^-15, 3 → 1.}}

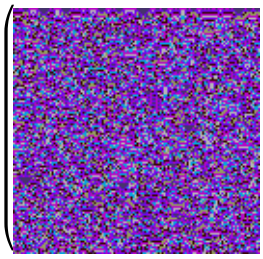
In[256]:=
```
test4Data8kr1C17 = data8TC[8, 128, 128];
Thread[test4Data8kr1C17 → netECA17[Keys@test4Data8kr1C17, {"TopProbabilities", 2}]]
```

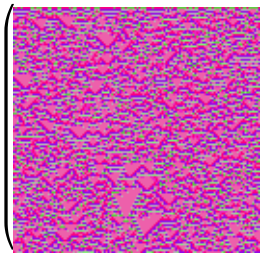
Out[257]=

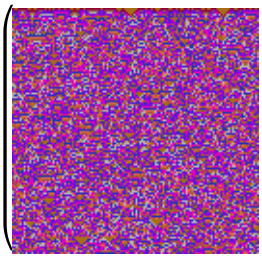

→ 14 955 350 598 586 141 683 → {4 → 1.70104 × 10^-7, 3 → 1.},


→ 30 727 455 169 449 395 964 → {4 → 4.31945 × 10^-11, 3 → 1.},

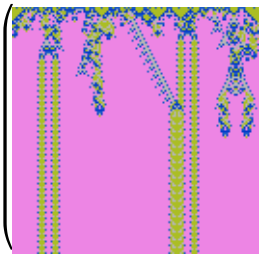

→ 42 490 152 676 883 207 115 → {3 → 0.0000793636, 4 → 0.999921},


→ 18 395 296 261 071 222 192 → {4 → 0.0142026, 3 → 0.985797},

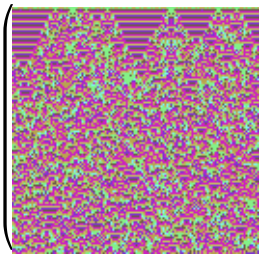

→ 22 317 090 484 634 250 431 → {3 → 0.00783339, 4 → 0.992167},



→ 33 329 414 465 629 594 174 → {4 → 0.00851294, 3 → 0.991487},



→ 68 439 232 681 205 604 962 → {2 → 1.97568 × 10^{-13}, 4 → 1.},



→ 53 049 830 479 062 864 751 → {4 → 3.6815 × 10^{-10}, 3 → 1.}}