

# Adaptation du réseau neuronal

```
net = googlenet
```

```
net =  
  DAGNetwork with properties:  
  
    Layers: [144x1 nnet.cnn.layer.Layer]  
    Connections: [170x2 table]  
    InputNames: {'data'}  
    OutputNames: {'output'}
```

```
lay = net.Layers
```

```
lay =
```

```
144x1 Layer array with layers:
```

1	'data'	Image Input	224x224x3 images with 'zerocenter' normalization
2	'conv1-7x7_s2'	Convolution	64 7x7x3 convolutions with stride [2 2] and padding [1 1]
3	'conv1-relu_7x7'	ReLU	ReLU
4	'pool1-3x3_s2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1]
5	'pool1-norm1'	Cross Channel Normalization	cross channel normalization with 5 channels
6	'conv2-3x3_reduce'	Convolution	64 1x1x64 convolutions with stride [1 1] and padding [1 1]
7	'conv2-relu_3x3_reduce'	ReLU	ReLU
8	'conv2-3x3'	Convolution	192 3x3x64 convolutions with stride [1 1] and padding [1 1]
9	'conv2-relu_3x3'	ReLU	ReLU
10	'conv2-norm2'	Cross Channel Normalization	cross channel normalization with 5 channels
11	'pool2-3x3_s2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1]
12	'inception_3a-1x1'	Convolution	64 1x1x192 convolutions with stride [1 1] and padding [1 1]
13	'inception_3a-relu_1x1'	ReLU	ReLU
14	'inception_3a-3x3_reduce'	Convolution	96 1x1x192 convolutions with stride [1 1] and padding [1 1]
15	'inception_3a-relu_3x3_reduce'	ReLU	ReLU
16	'inception_3a-3x3'	Convolution	128 3x3x96 convolutions with stride [1 1] and padding [1 1]
17	'inception_3a-relu_3x3'	ReLU	ReLU
18	'inception_3a-5x5_reduce'	Convolution	16 1x1x192 convolutions with stride [1 1] and padding [1 1]
19	'inception_3a-relu_5x5_reduce'	ReLU	ReLU
20	'inception_3a-5x5'	Convolution	32 5x5x16 convolutions with stride [1 1] and padding [1 1]
21	'inception_3a-relu_5x5'	ReLU	ReLU
22	'inception_3a-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1]
23	'inception_3a-pool_proj'	Convolution	32 1x1x192 convolutions with stride [1 1] and padding [1 1]
24	'inception_3a-relu_pool_proj'	ReLU	ReLU
25	'inception_3a-output'	Depth concatenation	Depth concatenation of 4 inputs
26	'inception_3b-1x1'	Convolution	128 1x1x256 convolutions with stride [1 1] and padding [1 1]
27	'inception_3b-relu_1x1'	ReLU	ReLU
28	'inception_3b-3x3_reduce'	Convolution	128 1x1x256 convolutions with stride [1 1] and padding [1 1]
29	'inception_3b-relu_3x3_reduce'	ReLU	ReLU
30	'inception_3b-3x3'	Convolution	192 3x3x128 convolutions with stride [1 1] and padding [1 1]
31	'inception_3b-relu_3x3'	ReLU	ReLU
32	'inception_3b-5x5_reduce'	Convolution	32 1x1x256 convolutions with stride [1 1] and padding [1 1]
33	'inception_3b-relu_5x5_reduce'	ReLU	ReLU
34	'inception_3b-5x5'	Convolution	96 5x5x32 convolutions with stride [1 1] and padding [1 1]
35	'inception_3b-relu_5x5'	ReLU	ReLU
36	'inception_3b-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1]
37	'inception_3b-pool_proj'	Convolution	64 1x1x256 convolutions with stride [1 1] and padding [1 1]
38	'inception_3b-relu_pool_proj'	ReLU	ReLU
39	'inception_3b-output'	Depth concatenation	Depth concatenation of 4 inputs
40	'pool3-3x3_s2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1]
41	'inception_4a-1x1'	Convolution	192 1x1x480 convolutions with stride [1 1] and padding [1 1]
42	'inception_4a-relu_1x1'	ReLU	ReLU
43	'inception_4a-3x3_reduce'	Convolution	96 1x1x480 convolutions with stride [1 1] and padding [1 1]

44	'inception_4a-relu_3x3_reduce'	ReLU	ReLU
45	'inception_4a-3x3'	Convolution	208 3x3x96 convolutions with stride [1 1] and padding [1 1]
46	'inception_4a-relu_3x3'	ReLU	ReLU
47	'inception_4a-5x5_reduce'	Convolution	16 1x1x480 convolutions with stride [1 1] and padding [1 1]
48	'inception_4a-relu_5x5_reduce'	ReLU	ReLU
49	'inception_4a-5x5'	Convolution	48 5x5x16 convolutions with stride [1 1] and padding [1 1]
50	'inception_4a-relu_5x5'	ReLU	ReLU
51	'inception_4a-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1]
52	'inception_4a-pool_proj'	Convolution	64 1x1x480 convolutions with stride [1 1] and padding [1 1]
53	'inception_4a-relu_pool_proj'	ReLU	ReLU
54	'inception_4a-output'	Depth concatenation	Depth concatenation of 4 inputs
55	'inception_4b-1x1'	Convolution	160 1x1x512 convolutions with stride [1 1] and padding [1 1]
56	'inception_4b-relu_1x1'	ReLU	ReLU
57	'inception_4b-3x3_reduce'	Convolution	112 1x1x512 convolutions with stride [1 1] and padding [1 1]
58	'inception_4b-relu_3x3_reduce'	ReLU	ReLU
59	'inception_4b-3x3'	Convolution	224 3x3x112 convolutions with stride [1 1] and padding [1 1]
60	'inception_4b-relu_3x3'	ReLU	ReLU
61	'inception_4b-5x5_reduce'	Convolution	24 1x1x512 convolutions with stride [1 1] and padding [1 1]
62	'inception_4b-relu_5x5_reduce'	ReLU	ReLU
63	'inception_4b-5x5'	Convolution	64 5x5x24 convolutions with stride [1 1] and padding [1 1]
64	'inception_4b-relu_5x5'	ReLU	ReLU
65	'inception_4b-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1]
66	'inception_4b-pool_proj'	Convolution	64 1x1x512 convolutions with stride [1 1] and padding [1 1]
67	'inception_4b-relu_pool_proj'	ReLU	ReLU
68	'inception_4b-output'	Depth concatenation	Depth concatenation of 4 inputs
69	'inception_4c-1x1'	Convolution	128 1x1x512 convolutions with stride [1 1] and padding [1 1]
70	'inception_4c-relu_1x1'	ReLU	ReLU
71	'inception_4c-3x3_reduce'	Convolution	128 1x1x512 convolutions with stride [1 1] and padding [1 1]
72	'inception_4c-relu_3x3_reduce'	ReLU	ReLU
73	'inception_4c-3x3'	Convolution	256 3x3x128 convolutions with stride [1 1] and padding [1 1]
74	'inception_4c-relu_3x3'	ReLU	ReLU
75	'inception_4c-5x5_reduce'	Convolution	24 1x1x512 convolutions with stride [1 1] and padding [1 1]
76	'inception_4c-relu_5x5_reduce'	ReLU	ReLU
77	'inception_4c-5x5'	Convolution	64 5x5x24 convolutions with stride [1 1] and padding [1 1]
78	'inception_4c-relu_5x5'	ReLU	ReLU
79	'inception_4c-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1]
80	'inception_4c-pool_proj'	Convolution	64 1x1x512 convolutions with stride [1 1] and padding [1 1]
81	'inception_4c-relu_pool_proj'	ReLU	ReLU
82	'inception_4c-output'	Depth concatenation	Depth concatenation of 4 inputs
83	'inception_4d-1x1'	Convolution	112 1x1x512 convolutions with stride [1 1] and padding [1 1]
84	'inception_4d-relu_1x1'	ReLU	ReLU
85	'inception_4d-3x3_reduce'	Convolution	144 1x1x512 convolutions with stride [1 1] and padding [1 1]
86	'inception_4d-relu_3x3_reduce'	ReLU	ReLU
87	'inception_4d-3x3'	Convolution	288 3x3x144 convolutions with stride [1 1] and padding [1 1]
88	'inception_4d-relu_3x3'	ReLU	ReLU
89	'inception_4d-5x5_reduce'	Convolution	32 1x1x512 convolutions with stride [1 1] and padding [1 1]
90	'inception_4d-relu_5x5_reduce'	ReLU	ReLU
91	'inception_4d-5x5'	Convolution	64 5x5x32 convolutions with stride [1 1] and padding [1 1]
92	'inception_4d-relu_5x5'	ReLU	ReLU
93	'inception_4d-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1]
94	'inception_4d-pool_proj'	Convolution	64 1x1x512 convolutions with stride [1 1] and padding [1 1]
95	'inception_4d-relu_pool_proj'	ReLU	ReLU
96	'inception_4d-output'	Depth concatenation	Depth concatenation of 4 inputs
97	'inception_4e-1x1'	Convolution	256 1x1x528 convolutions with stride [1 1] and padding [1 1]
98	'inception_4e-relu_1x1'	ReLU	ReLU
99	'inception_4e-3x3_reduce'	Convolution	160 1x1x528 convolutions with stride [1 1] and padding [1 1]
100	'inception_4e-relu_3x3_reduce'	ReLU	ReLU
101	'inception_4e-3x3'	Convolution	320 3x3x160 convolutions with stride [1 1] and padding [1 1]
102	'inception_4e-relu_3x3'	ReLU	ReLU
103	'inception_4e-5x5_reduce'	Convolution	32 1x1x528 convolutions with stride [1 1] and padding [1 1]
104	'inception_4e-relu_5x5_reduce'	ReLU	ReLU
105	'inception_4e-5x5'	Convolution	128 5x5x32 convolutions with stride [1 1] and padding [1 1]
106	'inception_4e-relu_5x5'	ReLU	ReLU
107	'inception_4e-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1]

108	'inception_4e-pool_proj'	Convolution	128 1x1x528 convolutions with stride [1 1 1]
109	'inception_4e-relu_pool_proj'	ReLU	ReLU
110	'inception_4e-output'	Depth concatenation	Depth concatenation of 4 inputs
111	'pool4-3x3_s2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1]
112	'inception_5a-1x1'	Convolution	256 1x1x832 convolutions with stride [1 1 1]
113	'inception_5a-relu_1x1'	ReLU	ReLU
114	'inception_5a-3x3_reduce'	Convolution	160 1x1x832 convolutions with stride [1 1 1]
115	'inception_5a-relu_3x3_reduce'	ReLU	ReLU
116	'inception_5a-3x3'	Convolution	320 3x3x160 convolutions with stride [1 1 1]
117	'inception_5a-relu_3x3'	ReLU	ReLU
118	'inception_5a-5x5_reduce'	Convolution	32 1x1x832 convolutions with stride [1 1 1]
119	'inception_5a-relu_5x5_reduce'	ReLU	ReLU
120	'inception_5a-5x5'	Convolution	128 5x5x32 convolutions with stride [1 1 1]
121	'inception_5a-relu_5x5'	ReLU	ReLU
122	'inception_5a-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1]
123	'inception_5a-pool_proj'	Convolution	128 1x1x832 convolutions with stride [1 1 1]
124	'inception_5a-relu_pool_proj'	ReLU	ReLU
125	'inception_5a-output'	Depth concatenation	Depth concatenation of 4 inputs
126	'inception_5b-1x1'	Convolution	384 1x1x832 convolutions with stride [1 1 1]
127	'inception_5b-relu_1x1'	ReLU	ReLU
128	'inception_5b-3x3_reduce'	Convolution	192 1x1x832 convolutions with stride [1 1 1]
129	'inception_5b-relu_3x3_reduce'	ReLU	ReLU
130	'inception_5b-3x3'	Convolution	384 3x3x192 convolutions with stride [1 1 1]
131	'inception_5b-relu_3x3'	ReLU	ReLU
132	'inception_5b-5x5_reduce'	Convolution	48 1x1x832 convolutions with stride [1 1 1]
133	'inception_5b-relu_5x5_reduce'	ReLU	ReLU
134	'inception_5b-5x5'	Convolution	128 5x5x48 convolutions with stride [1 1 1]
135	'inception_5b-relu_5x5'	ReLU	ReLU
136	'inception_5b-pool'	Max Pooling	3x3 max pooling with stride [1 1] and padding [1 1]
137	'inception_5b-pool_proj'	Convolution	128 1x1x832 convolutions with stride [1 1 1]
138	'inception_5b-relu_pool_proj'	ReLU	ReLU
139	'inception_5b-output'	Depth concatenation	Depth concatenation of 4 inputs
140	'pool5-7x7_s1'	Global Average Pooling	Global average pooling
141	'pool5-drop_7x7_s1'	Dropout	40% dropout
142	'loss3-classifier'	Fully Connected	1000 fully connected layer
143	'prob'	Softmax	softmax
144	'output'	Classification Output	crossentropyex with 'tench' and 999 other classes

```
inLayer = lay(1)
```

```
inLayer =
  ImageInputLayer with properties:
    Name: 'data'
    InputSize: [224 224 3]

  Hyperparameters
    DataAugmentation: 'none'
    Normalization: 'zerocenter'
    NormalizationDimension: 'auto'
    Mean: [224x224x3 single]
```

```
inSize = inLayer.InputSize
```

```
inSize = 1x3
    224    224    3
```

```
lgraph = layerGraph(net);
newFc = fullyConnectedLayer(38,"Name","new_fc");
lgraph = replaceLayer(lgraph,"loss3-classifier",newFc)
```

```
lgraph =
```

LayerGraph with properties:

```
Layers: [144x1 nnet.cnn.layer.Layer]
Connections: [170x2 table]
InputNames: {'data'}
OutputNames: {'output'}
```

```
newOut = classificationLayer("Name","new_out");
lgraph = replaceLayer(lgraph,"output",newOut)
```

lgraph =  
LayerGraph with properties:

```
Layers: [144x1 nnet.cnn.layer.Layer]
Connections: [170x2 table]
InputNames: {'data'}
OutputNames: {'new_out'}
```

## Image datastores

```
imds = imageDatastore("imgAll", "IncludeSubfolders",true, "LabelSource","foldernames")
```

imds =

ImageDatastore with properties:

```
Files: {
    ' ...\00075aa8-d81a-4184-8541-b692b78d398a__FREC_Scab 3335.JPG';
    ' ...\01a66316-0e98-4d3b-a56f-d78752cd043f__FREC_Scab 3003.JPG';
    ' ...\01a66316-0e98-4d3b-a56f-d78752cd043f__FREC_Scab 3003_270deg.JPG'
    ... and 70292 more
}
Folders: {
    'D:\cecile\Prof\Matlab\SI-IA-PlantDiseases\imgAll'
}
Labels: [Apple__Apple_scab; Apple__Apple_scab; Apple__Apple_scab ... and 70292 more categories]
AlternateFileSystemRoots: {}
ReadSize: 1
SupportedOutputFormats: ["png" "jpg" "jpeg" "tif" "tiff"]
DefaultOutputFormat: "png"
ReadFcn: @readDatastoreImage
```

imds.Labels

```
ans = 70295x1 categorical
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
...
```

```
[trainAllImgs, testAllImgs] = splitEachLabel(imds, 0.7, "randomized")
```

trainAllImgs =

ImageDatastore with properties:

```

Files: {
' ...\00075aa8-d81a-4184-8541-b692b78d398a__FREC_Scab 3335.JPG';
' ...\01a66316-0e98-4d3b-a56f-d78752cd043f__FREC_Scab 3003_90deg.JPG';
' ...\01f3deaa-6143-4b6c-9c22-620a46d8be04__FREC_Scab 3112.JPG'
... and 49204 more
}
Folders: {
'D:\cecile\Prof\Matlab\SI-IA-PlantDiseases\imgAll'
}
Labels: [Apple__Apple_scab; Apple__Apple_scab; Apple__Apple_scab ... and 49204 more categories]
AlternateFileSystemRoots: {}
ReadSize: 1
SupportedOutputFormats: ["png" "jpg" "jpeg" "tif" "tiff"]
DefaultOutputFormat: "png"
ReadFcn: @readDatastoreImage

```

testAllImgs =

ImageDatastore with properties:

```

Files: {
' ...\01a66316-0e98-4d3b-a56f-d78752cd043f__FREC_Scab 3003.JPG';
' ...\01a66316-0e98-4d3b-a56f-d78752cd043f__FREC_Scab 3003_270deg.JPG';
' ...\01a66316-0e98-4d3b-a56f-d78752cd043f__FREC_Scab 3003_new30degFlipLR.JPG'
... and 21085 more
}
Folders: {
'D:\cecile\Prof\Matlab\SI-IA-PlantDiseases\imgAll'
}
Labels: [Apple__Apple_scab; Apple__Apple_scab; Apple__Apple_scab ... and 21085 more categories]
AlternateFileSystemRoots: {}
ReadSize: 1
SupportedOutputFormats: ["png" "jpg" "jpeg" "tif" "tiff"]
DefaultOutputFormat: "png"
ReadFcn: @readDatastoreImage

```

pour travailler sur une partie des images et pas sur la totalité

```

trainds = augmentedImageDatastore([224 224],trainAllImgs);
testds = augmentedImageDatastore([224 224],testAllImgs);

```

## Entraînement du réseau

```

options = trainingOptions("sgdm",...
'InitialLearnRate', 0.01, ...
'LearnRateDropFactor',0.2, ...
'LearnRateDropPeriod',5, ...
'MaxEpochs',10, ...
'MiniBatchSize',64, ...
'Plots','training-progress',...
'Shuffle','every-epoch');

```

```

plantDiseasesNet = trainNetwork(trainds,lgraph,options)

```

Training on single CPU.

Initializing input data normalization.

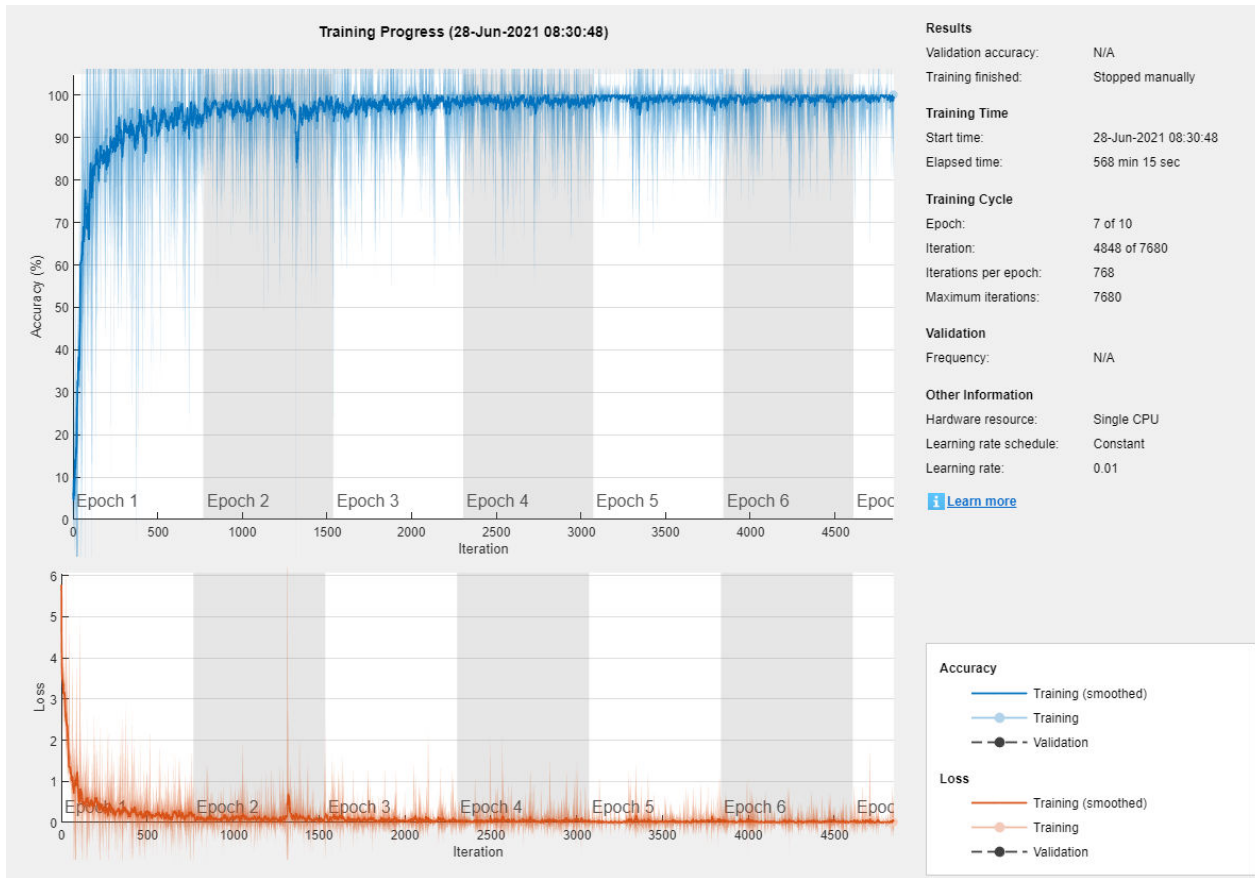
```

=====
| Epoch | Iteration | Time Elapsed | Mini-batch | Mini-batch | Base Learning |
|       |           | (hh:mm:ss)  | Accuracy  | Loss       | Rate         |
=====

```

1	1	00:00:11	6.25%	5.7812	0.0100
1	50	00:05:56	59.38%	1.4397	0.0100
1	100	00:11:41	76.56%	0.7109	0.0100
1	150	00:17:24	78.12%	0.8188	0.0100
1	200	00:23:07	93.75%	0.3224	0.0100
1	250	00:28:50	92.19%	0.3133	0.0100
1	300	00:34:33	90.62%	0.2258	0.0100
1	350	00:40:15	92.19%	0.2211	0.0100
1	400	00:45:57	95.31%	0.1818	0.0100
1	450	00:51:39	93.75%	0.1709	0.0100
1	500	00:57:22	90.62%	0.3845	0.0100
1	550	01:03:04	93.75%	0.2067	0.0100
1	600	01:08:47	92.19%	0.1714	0.0100
1	650	01:14:30	96.88%	0.1061	0.0100
1	700	01:20:12	90.62%	0.2579	0.0100
1	750	01:25:55	92.19%	0.3408	0.0100
2	800	01:31:41	93.75%	0.1857	0.0100
2	850	01:37:26	98.44%	0.0331	0.0100
2	900	01:43:11	96.88%	0.0736	0.0100
2	950	01:48:51	93.75%	0.2519	0.0100
2	1000	01:54:32	98.44%	0.0500	0.0100
2	1050	02:00:13	98.44%	0.0685	0.0100
2	1100	02:05:57	98.44%	0.0626	0.0100
2	1150	02:11:40	100.00%	0.0380	0.0100
2	1200	02:17:23	93.75%	0.1335	0.0100
2	1250	02:23:06	96.88%	0.1045	0.0100
2	1300	02:28:49	96.88%	0.0516	0.0100
2	1350	02:34:40	95.31%	0.1546	0.0100
2	1400	02:40:45	93.75%	0.1138	0.0100
2	1450	02:46:48	98.44%	0.1392	0.0100
2	1500	02:52:43	96.88%	0.1109	0.0100
3	1550	02:58:33	92.19%	0.1185	0.0100
3	1600	03:04:21	92.19%	0.1855	0.0100
3	1650	03:10:10	98.44%	0.0693	0.0100
3	1700	03:15:58	100.00%	0.0092	0.0100
3	1750	03:22:05	98.44%	0.0782	0.0100
3	1800	03:28:08	96.88%	0.1889	0.0100
3	1850	03:34:09	96.88%	0.1276	0.0100
3	1900	03:40:12	95.31%	0.1527	0.0100
3	1950	03:46:12	100.00%	0.0074	0.0100
3	2000	03:52:15	100.00%	0.0170	0.0100
3	2050	03:58:06	98.44%	0.0312	0.0100
3	2100	04:03:56	98.44%	0.1042	0.0100
3	2150	04:09:46	98.44%	0.0592	0.0100
3	2200	04:15:36	98.44%	0.0456	0.0100
3	2250	04:21:26	100.00%	0.0140	0.0100
3	2300	04:27:16	98.44%	0.0195	0.0100
4	2350	04:33:06	100.00%	0.0030	0.0100
4	2400	04:38:55	98.44%	0.0395	0.0100
4	2450	04:44:45	98.44%	0.0272	0.0100
4	2500	04:50:39	98.44%	0.0175	0.0100
4	2550	04:56:36	98.44%	0.0246	0.0100
4	2600	05:02:31	93.75%	0.1035	0.0100
4	2650	05:08:29	100.00%	0.0040	0.0100
4	2700	05:14:24	100.00%	0.0167	0.0100
4	2750	05:20:20	96.88%	0.0683	0.0100
4	2800	05:26:15	98.44%	0.0884	0.0100
4	2850	05:32:10	100.00%	0.0009	0.0100
4	2900	05:38:06	100.00%	0.0041	0.0100
4	2950	05:44:03	98.44%	0.0817	0.0100
4	3000	05:49:54	100.00%	0.0131	0.0100
4	3050	05:55:43	100.00%	0.0039	0.0100
5	3100	06:01:32	100.00%	0.0014	0.0100
5	3150	06:07:32	100.00%	0.0026	0.0100

5	3200	06:13:33	100.00%	0.0113	0.0100
5	3250	06:19:25	100.00%	0.0025	0.0100
5	3300	06:25:15	100.00%	0.0053	0.0100
5	3350	06:31:05	96.88%	0.0717	0.0100
5	3400	06:37:00	100.00%	0.0021	0.0100
5	3450	06:43:00	100.00%	0.0010	0.0100
5	3500	06:48:59	100.00%	0.0050	0.0100
5	3550	06:54:59	100.00%	0.0020	0.0100
5	3600	07:00:55	98.44%	0.0538	0.0100
5	3650	07:06:49	100.00%	0.0045	0.0100
5	3700	07:12:41	96.88%	0.0762	0.0100
5	3750	07:18:31	98.44%	0.0200	0.0100
5	3800	07:24:22	100.00%	0.0068	0.0100
6	3850	07:30:14	96.88%	0.0530	0.0100
6	3900	07:36:03	96.88%	0.0644	0.0100
6	3950	07:41:55	98.44%	0.0784	0.0100
6	4000	07:47:49	100.00%	0.0324	0.0100
6	4050	07:53:45	100.00%	0.0115	0.0100
6	4100	07:59:36	98.44%	0.1423	0.0100
6	4150	08:05:26	98.44%	0.0754	0.0100
6	4200	08:11:21	100.00%	0.0015	0.0100
6	4250	08:17:31	98.44%	0.0312	0.0100
6	4300	08:23:29	100.00%	0.0079	0.0100
6	4350	08:29:20	100.00%	0.0006	0.0100
6	4400	08:35:10	100.00%	0.0050	0.0100
6	4450	08:41:03	98.44%	0.0320	0.0100
6	4500	08:47:00	100.00%	0.0115	0.0100
6	4550	08:52:57	100.00%	0.0016	0.0100
6	4600	08:58:52	100.00%	0.0082	0.0100
7	4650	09:04:45	98.44%	0.0388	0.0100
7	4700	09:10:35	100.00%	0.0241	0.0100
7	4750	09:16:30	100.00%	0.0089	0.0100
7	4800	09:22:21	100.00%	0.0014	0.0100
7	4848	09:28:15	100.00%	0.0007	0.0100



```
plantDiseasesNet =
  DAGNetwork with properties:

    Layers: [144x1 nnet.cnn.layer.Layer]
    Connections: [170x2 table]
    InputNames: {'data'}
    OutputNames: {'new_out'}
```

```
save ("plantDiseasesNet")
```

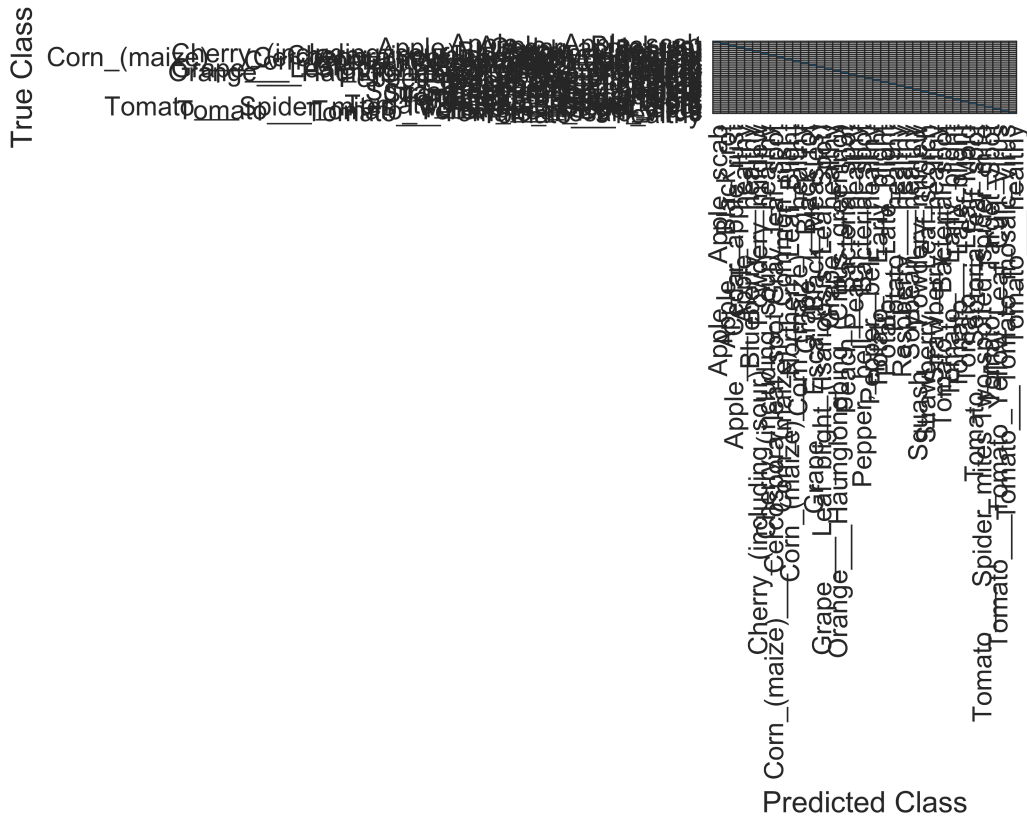
## Evaluation du réseau sur les données de test

```
predictions = classify(plantDiseasesNet, testds)
```

```
predictions = 21088x1 categorical
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
Apple__Apple_scab
⋮
```

```
confusionchart(testAllImgs.Labels, predictions)
```





## Temps d'exécution sur une image

```
img = imread("img/Corn__Cercospora_leaf_spot Gray_leaf_spot/0a01cc10-3892-4311-9c48-0ac6ab3c7c...");
img = imresize(img, [224,224]);
tic
label = classify (plantDiseasesNet,img)
```

```
label = categorical
    Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot
```

```
toc
```

Elapsed time is 1.378213 seconds.