In [7]:

```
import torch
import torch.nn as nn
from torchvision import datasets, models, transforms
import os
import pretrainedmodels
```

In [8]:

```
# Load the test data.
data_transforms = {
    'test': transforms.Compose([
        transforms.Resize([224, 224]),
        transforms.ToTensor()
    ])
}

data_dir = 'D:\data (augmented, 4 classes, tif)'
image_datasets = {x: datasets.ImageFolder(os.path.join(data_dir, x),
                                          data_transforms[x])
                  for x in ['test']}

batch_size = 32
dataloaders = {x: torch.utils.data.DataLoader(image_datasets[x], batch_size=batch_size,
                                             shuffle=True, num_workers=4)
              for x in ['test']}
dataset_sizes = {x: len(image_datasets[x]) for x in ['test']}
class_names = image_datasets['test'].classes

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

In [9]:

```
# Load the saved model state_dict for inference (done later to keep len(class_names) after the datal
oader dynamic).
model_ft = models.alexnet(pretrained=True)
model_ft.classifier[6] = nn.Linear(4096, len(class_names))
model_ft.classifier.add_module("7", nn.Dropout())

PATH = "D:\Models\model_20190411-093358.pth"
model_ft.load_state_dict(torch.load(PATH))

model_ft = model_ft.to(device)
```

In [10]:

```
was_training = model_ft.training

model_ft.eval()

all_labels = []
all_preds = []

with torch.no_grad():
    for inputs, labels in dataloaders['test']:    # The labels will correspond to the alphabetical o
rder of the class names (https://discuss.pytorch.org/t/how-to-get-the-class-names-to-class-label-map
ping/470).
        inputs = inputs.to(device)
        labels = labels.to(device)
        labels_list = labels.tolist()
        all_labels.extend(labels_list)

        outputs = model_ft(inputs)
        _, preds = torch.max(outputs, 1)
        preds_list = preds.tolist()
        all_preds.extend(preds_list)

    model_ft.train(mode=was_training)
```

In [11]:

```
from pandas_ml import ConfusionMatrix
cm = ConfusionMatrix(all_labels, all_preds)
cm.print_stats()
```

```
C:\Users\Apoorva Srivastava\Anaconda3\lib\site-packages\pandas_ml\confusion_matrix\stats.py:60: F
utureWarning: supplying multiple axes to axis is deprecated and will be removed in a future versi
on.
  num = df[df > 1].dropna(axis=[0, 1], thresh=1).applymap(lambda n: choose(n, 2)).sum().sum() - n
p.float64(nis2 * njs2) / n2
```

```
Confusion Matrix:

Predicted    0    1    2    3  __all__
Actual
0          120   10   43    7      180
1           25   95    9   78      207
2           43    3   64    8      118
3            5   23    6  105      139
__all__    193  131  122  198      644


Overall Statistics:

Accuracy: 0.5962732919254659
95% CI: (0.5572326376856002, 0.6344273292523064)
No Information Rate: ToDo
P-Value [Acc > NIR]: 1.3490732369560114e-51
Kappa: 0.4615401931431916
Mcnemar's Test P-Value: ToDo


Class Statistics:

Classes                                         0          1         2  \
Population                                    644        644       644
P: Condition positive                         180        207       118
N: Condition negative                         464        437       526
Test outcome positive                         193        131       122
Test outcome negative                         451        513       522
TP: True Positive                             120         95        64
TN: True Negative                             391        401       468
FP: False Positive                             73         36        58
FN: False Negative                             60        112        54
TPR: (Sensitivity, hit rate, recall)     0.666667   0.458937  0.542373
TNR=SPC: (Specificity)                   0.842672    0.91762  0.889734
PPV: Pos Pred Value (Precision)          0.621762   0.725191   0.52459
NPV: Neg Pred Value                      0.866962   0.781676  0.896552
FPR: False-out                           0.157328  0.0823799  0.110266
FDR: False Discovery Rate                0.378238   0.274809   0.47541
FNR: Miss Rate                           0.333333   0.541063  0.457627
ACC: Accuracy                            0.793478   0.770186  0.826087
F1 score                                 0.643432    0.56213  0.533333
MCC: Matthews correlation coefficient    0.498925   0.436881  0.426589
Informedness                             0.509339   0.376557  0.432107
Markedness                               0.488724   0.506867  0.421142
Prevalence                               0.279503   0.321429   0.18323
LR+: Positive likelihood ratio            4.23744    5.57099   4.91876
LR-: Negative likelihood ratio           0.395567   0.589637  0.514342
DOR: Diagnostic odds ratio                10.7123    9.44816   9.56322
FOR: False omission rate                 0.133038   0.218324  0.103448

Classes                                         3
Population                                    644
P: Condition positive                         139
N: Condition negative                         505
Test outcome positive                         198
Test outcome negative                         446
TP: True Positive                             105
TN: True Negative                             412
FP: False Positive                             93
FN: False Negative                             34
TPR: (Sensitivity, hit rate, recall)     0.755396
TNR=SPC: (Specificity)                   0.815842
PPV: Pos Pred Value (Precision)          0.530303
NPV: Neg Pred Value                      0.923767
FPR: False-out                           0.184158
FDR: False Discovery Rate                0.469697
FNR: Miss Rate                           0.244604
ACC: Accuracy                            0.802795
F1 score                                 0.623145
MCC: Matthews correlation coefficient    0.509295
Informedness                             0.571237
Markedness                                0.45407
Prevalence                               0.215839
LR+: Positive likelihood ratio            4.10188
LR-: Negative likelihood ratio           0.299818
```