GoDeep!

A machine learning golang library, with data processing support.

Functions and classes

The godeep library works based on four modules, we start with a extract class, them that data goes to the process module, that processed data are used to obtain information at the data analysis module,them, finally, the learn module use machine learning algorithms to classify that information.

Extract

Example:

```
Var datasetextractor extract.ImageExtractor
origins := []string{"../data/ImagesData/danger",
                    "../data/ImagesData/asphalt",
                    "../data/ImagesData/grass"}

datasetextractor.SetOrigins(origins,&datasetextractor)

datasetextractor.Read(false,false,true)
```

Process

```
Var datatransformer process.ImageProcessing
Var normtype gocv.NormType = gocv.NormMinMax
Var glcm process.GLCM
Var normalize process.Normalize

datatransformer.GetImages(&datasetextractor)

glcm.SetParameters(1,0)
datatransformer.SetProcessStrategy(glcm)
datatransformer.ProcessGroup(true)

normalize.SetParameters(0.0, 255.0, normtype)
datatransformer.SetProcessStrategy(normalize)
datatransformer.ProcessGroup(true)
```

Data Analysis

```
Var datavision computervision.ComputerVison
datavision.GetBaseImages(&datatransformer)

datavision.GroupFeature(true,computervision.EnergyFeature,computervision.CorrelationFea
ture,computervision.ContrastFeature)
        datavision.PrintFeatures()
```

Learn

```
Var datalearner learnstrategy.DataLearner
datalearner.Build(&datavision.Information,datasetextractor.Readinfo,75)
datalearner.Printfeatures()

knn := &nonparametric.Knn{}
datalearner.SetLearnStrategy(knn)
datalearner.ProcessLearn()
datalearner.Printresults()
```