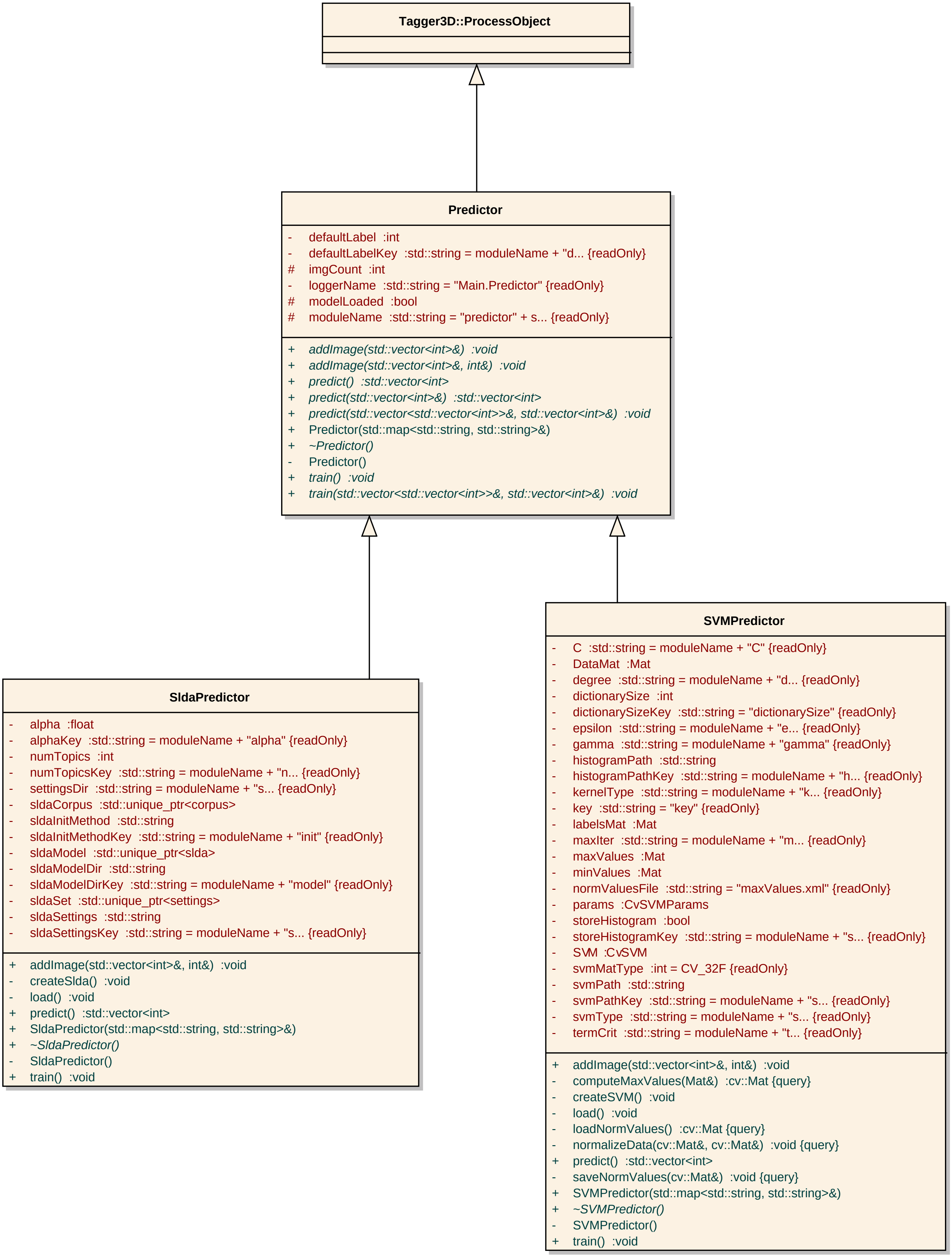
Tagger3D::ProcessObject
Predictor
- defaultLabel :int
- defaultLabelKey :std::string = moduleName + "d... {readOnly}
# imgCount :int
- loggerName :std::string = "Main.Predictor" {readOnly}
# modelLoaded :bool
# moduleName :std::string = "predictor" + s... {readOnly}
+ *addImage(std::vector<int>&) :void*
+ *addImage(std::vector<int>&, int&) :void*
+ *predict() :std::vector<int>*
+ *predict(std::vector<int>&) :std::vector<int>*
+ *predict(std::vector<std::vector<int>>&, std::vector<int>&) :void*
+ Predictor(std::map<std::string, std::string>&)
+ *~Predictor()*
- Predictor()
+ *train() :void*
+ *train(std::vector<std::vector<int>>&, std::vector<int>&) :void*
SldaPredictor
- alpha :float
- alphaKey :std::string = moduleName + "alpha" {readOnly}
- numTopics :int
- numTopicsKey :std::string = moduleName + "n... {readOnly}
- settingsDir :std::string = moduleName + "s... {readOnly}
- sldaCorpus :std::unique_ptr<corpus>
- sldaInitMethod :std::string
- sldaInitMethodKey :std::string = moduleName + "init" {readOnly}
- sldaModel :std::unique_ptr<slda>
- sldaModelDir :std::string
- sldaModelDirKey :std::string = moduleName + "model" {readOnly}
- sldaSet :std::unique_ptr<settings>
- sldaSettings :std::string
- sldaSettingsKey :std::string = moduleName + "s... {readOnly}
+ addImage(std::vector<int>&, int&) :void
- createSlda() :void
- load() :void
+ predict() :std::vector<int>
+ SldaPredictor(std::map<std::string, std::string>&)
+ *~SldaPredictor()*
- SldaPredictor()
+ train() :void
SVMPredictor
- C :std::string = moduleName + "C" {readOnly}
- DataMat :Mat
- degree :std::string = moduleName + "d... {readOnly}
- dictionarySize :int
- dictionarySizeKey :std::string = "dictionarySize" {readOnly}
- epsilon :std::string = moduleName + "e... {readOnly}
- gamma :std::string = moduleName + "gamma" {readOnly}
- histogramPath :std::string
- histogramPathKey :std::string = moduleName + "h... {readOnly}
- kernelType :std::string = moduleName + "k... {readOnly}
- key :std::string = "key" {readOnly}
- labelsMat :Mat
- maxIter :std::string = moduleName + "m... {readOnly}
- maxValues :Mat
- minValues :Mat
- normValuesFile :std::string = "maxValues.xml" {readOnly}
- params :CvSVMParams
- storeHistogram :bool
- storeHistogramKey :std::string = moduleName + "s... {readOnly}
- SVM :CvSVM
- svmMatType :int = CV_32F {readOnly}
- svmPath :std::string
- svmPathKey :std::string = moduleName + "s... {readOnly}
- svmType :std::string = moduleName + "s... {readOnly}
- termCrit :std::string = moduleName + "t... {readOnly}
+ addImage(std::vector<int>&, int&) :void
- computeMaxValues(Mat&) :cv::Mat {query}
- createSVM() :void
- load() :void
- loadNormValues() :cv::Mat {query}
- normalizeData(cv::Mat&, cv::Mat&) :void {query}
+ predict() :std::vector<int>
- saveNormValues(cv::Mat&) :void {query}
+ SVMPredictor(std::map<std::string, std::string>&)
+ *~SVMPredictor()*
- SVMPredictor()
+ train() :void