

Dlib C++ Library

The Library

[Algorithms](#)
[API Wrappers](#)
[Bayesian Nets](#)
[Compression](#)
[Containers](#)
[Graph Tools](#)
[Image Processing](#)
[Linear Algebra](#)
[Machine Learning](#)
[Metaprogramming](#)
[Miscellaneous](#)
[Networking](#)
[Optimization](#)
[Parsing](#)

Help/Info

[Dlib Blog](#)
[Examples: C++](#)
[Examples: Python](#)
[FAQ](#)
[Home](#)
[How to compile](#)
[How to contribute](#)
[Index](#)
[Introduction](#)
[License](#)
[Python API](#)
[Suggested Books](#)
[Who uses dlib?](#)

Current Release

[Change Log](#)

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib's [open source licensing](#) allows you to use it in any application, free of charge.

To follow or participate in the development of dlib subscribe to [dlib on github](#). Also be sure to read the [how to contribute](#) page if you intend to submit code to the project.

Major Features

- **Documentation**

- Unlike a lot of open source projects, this one provides complete and precise documentation for every class and function. There are also debugging modes that check the documented preconditions for functions. When this is enabled it will catch the vast majority of bugs caused by calling functions incorrectly or using objects in an incorrect manner.
- Lots of example programs are provided
- *I consider the documentation to be the most important part of the library.* So if you find anything that isn't documented, isn't clear, or has out of date documentation, tell me and I will fix it.

- **High Quality Portable Code**

- Good unit test coverage. The ratio of unit test lines of code to library lines of code is about 1 to 4.
- The library is tested regularly on MS Windows, Linux, and Mac OS X systems. However, it should work on any POSIX system and has been used on Solaris, HP/UX, and the BSDs.
- No other packages are required to use the library. Only APIs that are provided by an out of the box OS are needed.
- There is no installation or configure step needed before you can use the library. See the [How to compile](#) page for details.
- All operating system specific code is isolated inside the OS abstraction layers which are kept as small as possible. The rest of the library is either layered on top of the OS abstraction layers or is pure ISO standard C++.

[Release Notes](#)

Download dlib
ver.19.7

- **Machine Learning Algorithms**

- [Deep Learning](#)
- Conventional SMO based Support Vector Machines for [classification](#) and [regression](#)
- Reduced-rank methods for large-scale [classification](#) and [regression](#)
- Relevance vector machines for [classification](#) and [regression](#)
- General purpose [multiclass classification](#) tools
- A [Multiclass SVM](#)
- A tool for solving the optimization problem associated with [structural support vector machines](#).
- Structural SVM tools for [sequence labeling](#)
- Structural SVM tools for solving [assignment problems](#)
- Structural SVM tools for [object detection](#) in images as well as more powerful (but slower) [deep learning tools for object detection](#).
- Structural SVM tools for [labeling nodes](#) in graphs
- A large-scale [SVM-Rank](#) implementation
- An online [kernel RLS regression](#) algorithm
- An online [SVM classification](#) algorithm
- [Semidefinite Metric Learning](#)
- An online kernelized [centroid estimator](#)/novelty detector and offline support vector [one-class classification](#)
- Clustering algorithms: [linear](#) or [kernel k-means](#), [Chinese Whispers](#), and [Newman clustering](#).
- [Radial Basis Function Networks](#)
- [Multi layer perceptrons](#)

- **Numerical Algorithms**

- A fast [matrix](#) object implemented using the expression templates technique and capable of using BLAS and LAPACK libraries when available.
- Numerous linear algebra and mathematical operations are defined for the matrix object such as the [singular value decomposition](#), [transpose](#), [trig functions](#), etc.
- General purpose unconstrained non-linear optimization algorithms using the [conjugate gradient](#), [BFGS](#), and [L-BFGS](#) techniques
- [Levenberg-Marquardt](#) for solving non-linear least squares problems
- Box-constrained derivative-free optimization via the [BOBYQA](#) algorithm
- An implementation of the [Optimized Cutting Plane Algorithm](#)
- [Several quadratic program solvers](#)

- Combinatorial optimization tools for solving [optimal assignment](#) and [min cut/max flow](#) problems as well as the [CKY algorithm](#) for finding the most probable parse tree
- A [big integer](#) object
- A [random number](#) object
- **Graphical Model Inference Algorithms**
 - [Join tree](#) algorithm for exact inference in a Bayesian network.
 - [Gibbs sampler](#) markov chain monte carlo algorithm for approximate inference in a Bayesian network.
 - Routines for performing MAP inference in [chain-structured](#), [Potts](#), or [general](#) factor graphs.
- **Image Processing**
 - Routines for [reading](#) and [writing](#) common image formats.
 - Automatic color space conversion between various pixel types
 - Common image operations such as edge finding and morphological operations
 - Implementations of the [SURF](#), [HOG](#), and [FHOG](#) feature extraction algorithms.
 - Tools for [detecting objects](#) in images including [frontal face detection](#) and [object pose estimation](#).
 - High quality [face recognition](#)
- **Threading**
 - The library provides a portable and simple [threading API](#)
 - A message passing [pipe](#) for inter-thread and [inter-process](#) communication
 - A [timer](#) object capable of generating events that are regularly spaced in time
 - [Threaded objects](#)
 - [Threaded functions](#)
 - [Parallel for loops](#)
 - A [thread_pool](#) with support for futures
- **Networking**
 - The library provides a portable and simple [TCP sockets API](#)
 - An object to help you make TCP based [servers](#)
 - [iostream](#) and [streambuf](#) objects that enables TCP sockets to interoperate with the C++ iostreams library
 - A simple [HTTP server](#) object you can use to embed a web server into your applications
 - A message passing [pipe](#) for inter-thread and [inter-process](#) communication
 - A tool used to implement algorithms using the [Bulk Synchronous Parallel \(BSP\)](#) computing model
- **Graphical User Interfaces**
 - The library provides a portable and simple core [GUI API](#)

- Implemented on top of the core GUI API are numerous [widgets](#)
- Unlike many other GUI toolkits, the entire dlib GUI toolkit is threadsafe
- **Data Compression and Integrity Algorithms**
 - A [CRC 32](#) object
 - [MD5](#) functions
 - Various abstracted objects representing parts of [data compression](#) algorithms. Many forms of the PPM algorithm are included.
- **Testing**
 - A thread safe [logger](#) object styled after the popular Java logger log4j
 - A modular [unit testing framework](#)
 - Various [assert](#) macros useful for testing preconditions
- **General Utilities**
 - A type-safe [object](#) to convert between big and little endian byte orderings
 - A [command line parser](#) with the ability to parse and validate command lines with various types of arguments and options
 - An [XML parser](#)
 - An object that can perform [base64](#) conversions
 - Many [container classes](#)
 - [Serialization support](#)
 - Many [memory manager](#) objects that implement different memory pooling strategies
 - A tool that lets you easily [call C++ from MATLAB](#)

Last Modified:
Sep 17, 2017