The pytorch ecosystem

02457 Machine Learning Operations

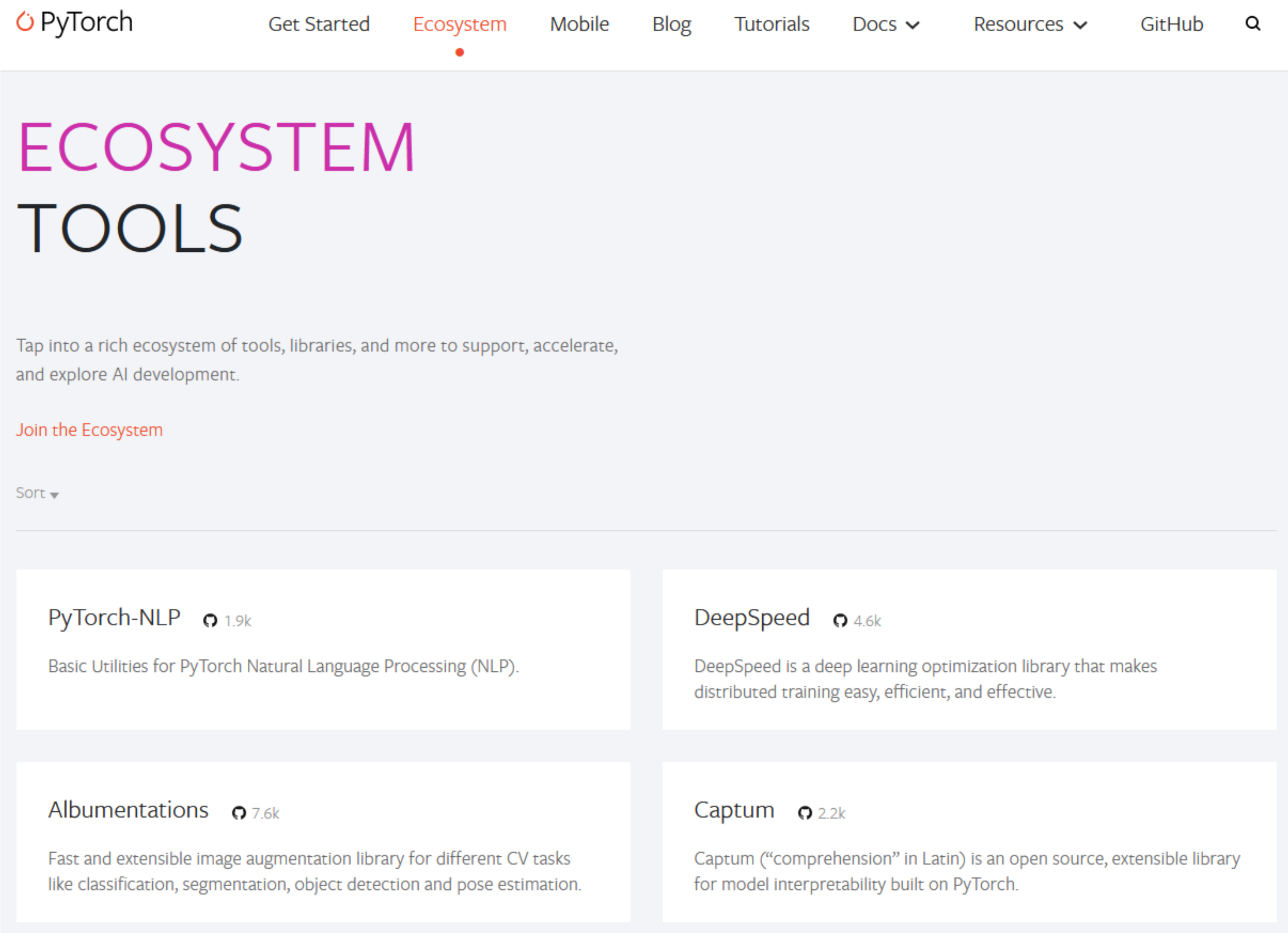
Nicki Skafte Detlefsen,

Postdoc

DTU Compute

The ecosystem

Collection of frameworks build to be used in combination with Pytorch



ECOSYSTEM
TOOLS

Tap into a rich ecosystem of tools, libraries, and more to support, accelerate, and explore AI development.

Join the Ecosystem

Sort

PyTorch-NLP 1.9k

Basic Utilities for PyTorch Natural Language Processing (NLP).

DeepSpeed 4.6k

DeepSpeed is a deep learning optimization library that makes distributed training easy, efficient, and effective.

Albumentations 7.6k

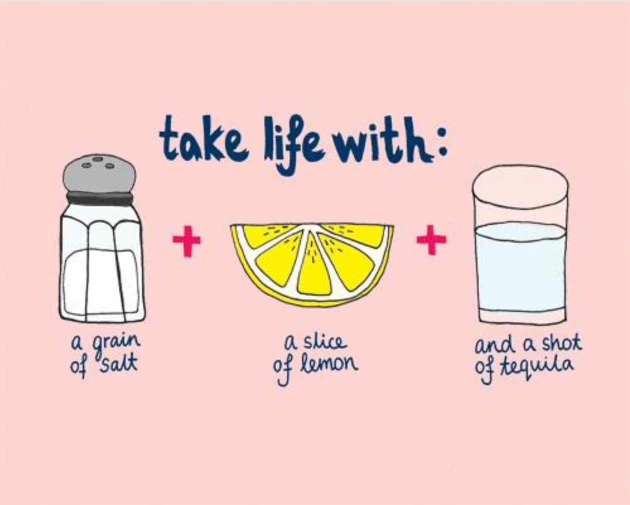
Fast and extensible image augmentation library for different CV tasks like classification, segmentation, object detection and pose estimation.

Captum 2.2k

Captum ("comprehension" in Latin) is an open source, extensible library for model interpretability built on PyTorch.

Fremwork categorising

Data specific frameworks	Training frameworks	Utility fremeworks
Transformers	fastai	Albumentations
Detectron2	Ray	PySyft
Pytorch geometric	Pytorch Lightning	Pyro
Flair	Horovod	Optuna
AllenNLP	DeepSpeed	Hydra
ParlAI	ONNX Runtime	Pytorch Metric Learning
DGL	skorch	Einops
PyTorch3D	Ignite	
MMF	Polyaxon	
Kornia		



Project work

- Organise, plan and execute an MLOps project
- What you will be evaluated on
- Done in groups
- You will be asked to implement what is showcased in the exercises in your project work
- Each Thursday afternoon you will be asked to upload you current progress to github
- Monday after will be given small feedback
- Final day of the course you will be presenting your project work

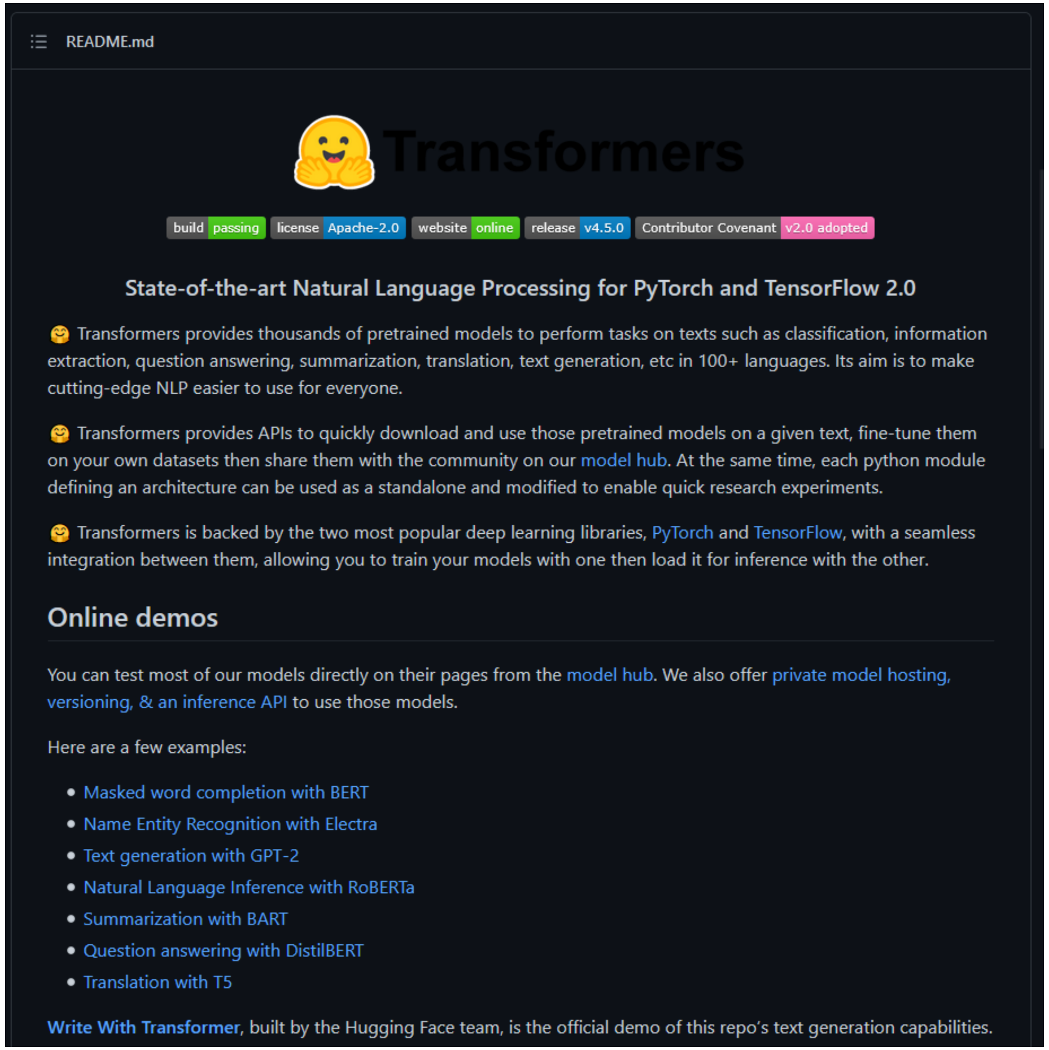
Project 1: NLP

Fremework: Transformers (Huggingface)

- https://github.com/huggingface/transformers

- State-of-the-art NLP models

- Most starred framework in the ecosystem



README.md

🤗 Transformers

build passing · license Apache-2.0 · website online · release v4.5.0 · Contributor Covenant v2.0 adopted

State-of-the-art Natural Language Processing for PyTorch and TensorFlow 2.0

🤗 Transformers provides thousands of pretrained models to perform tasks on texts such as classification, information extraction, question answering, summarization, translation, text generation, etc in 100+ languages. Its aim is to make cutting-edge NLP easier to use for everyone.

🤗 Transformers provides APIs to quickly download and use those pretrained models on a given text, fine-tune them on your own datasets then share them with the community on our model hub. At the same time, each python module defining an architecture can be used as a standalone and modified to enable quick research experiments.

🤗 Transformers is backed by the two most popular deep learning libraries, PyTorch and TensorFlow, with a seamless integration between them, allowing you to train your models with one then load it for inference with the other.

Online demos

You can test most of our models directly on their pages from the model hub. We also offer private model hosting, versioning, & an inference API to use those models.

Here are a few examples:

- Masked word completion with BERT
- Name Entity Recognition with Electra
- Text generation with GPT-2
- Natural Language Inference with RoBERTa
- Summarization with BART
- Question answering with DistilBERT
- Translation with T5

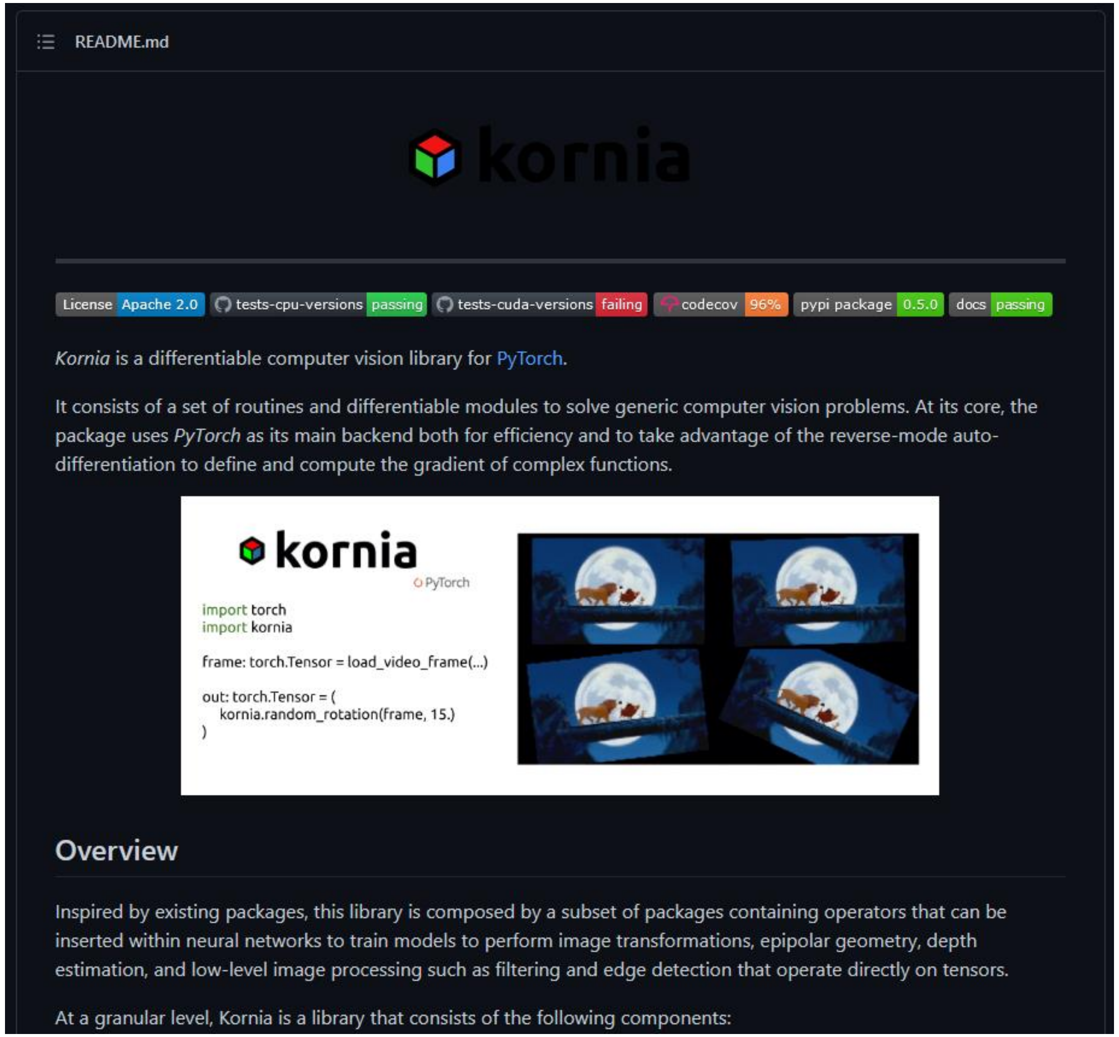
Write With Transformer, built by the Hugging Face team, is the official demo of this repo's text generation capabilities.

Project 2: CV

Framework: Kornia

- https://github.com/kornia/kornia

- Differentiable computer vision algoritms



README.md

kornia

License Apache 2.0 | tests-cpu-versions passing | tests-cuda-versions failing | codecov 96% | pypi package 0.5.0 | docs passing

Kornia is a differentiable computer vision library for PyTorch.

It consists of a set of routines and differentiable modules to solve generic computer vision problems. At its core, the package uses *PyTorch* as its main backend both for efficiency and to take advantage of the reverse-mode auto-differentiation to define and compute the gradient of complex functions.

kornia PyTorch

```
import torch
import kornia

frame: torch.Tensor = load_video_frame(...)

out: torch.Tensor = (
   kornia.random_rotation(frame, 15.)
)
```

Overview

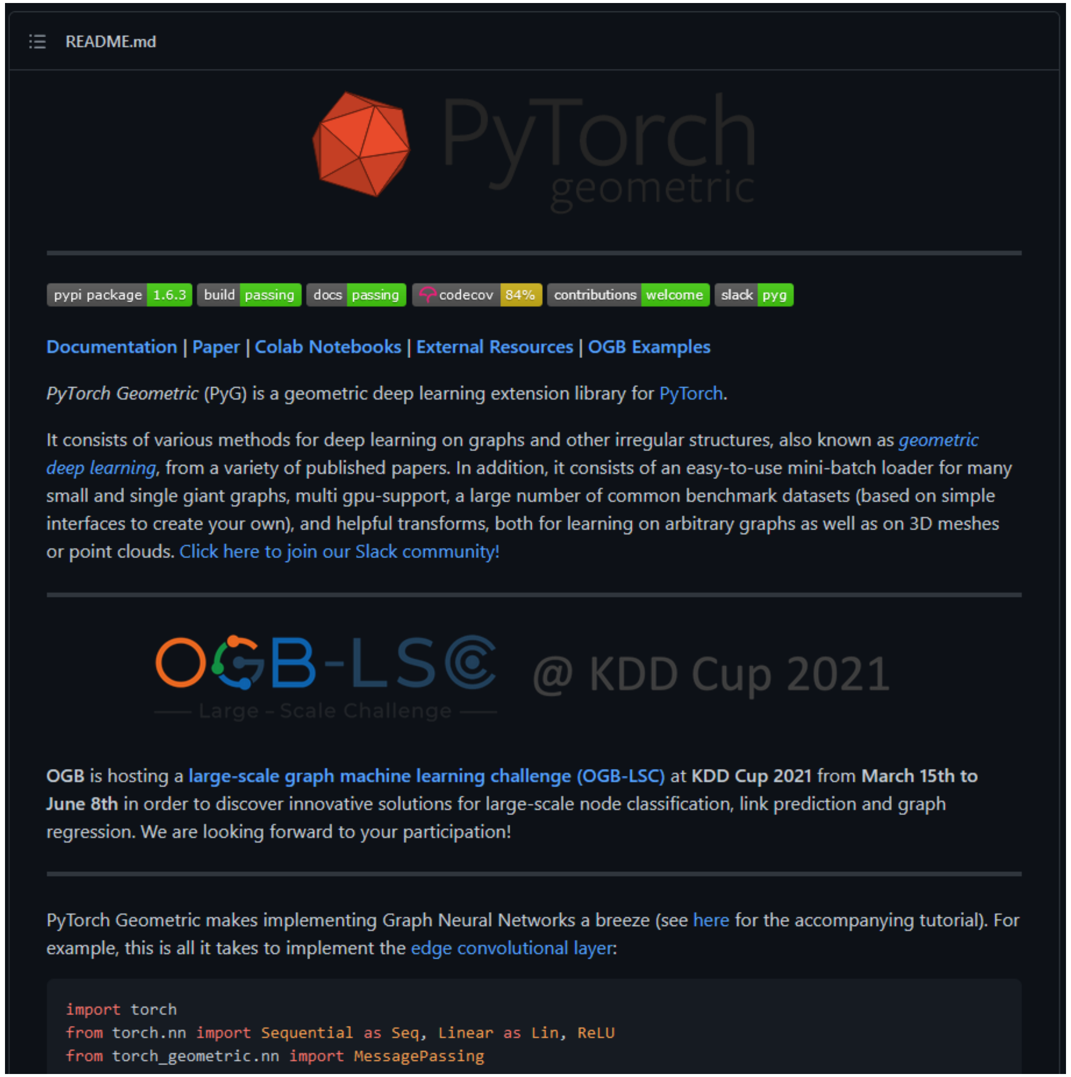
Inspired by existing packages, this library is composed by a subset of packages containing operators that can be inserted within neural networks to train models to perform image transformations, epipolar geometry, depth estimation, and low-level image processing such as filtering and edge detection that operate directly on tensors.

At a granular level, Kornia is a library that consists of the following components:

Project 3: Graphs and points

Framework: Pytorch Geometric

- https://github.com/rusty1s/pytorch_geometric
- Neural networks on graphs and pointclouds



Documentation | Paper | Colab Notebooks | External Resources | OGB Examples

PyTorch Geometric (PyG) is a geometric deep learning extension library for PyTorch.

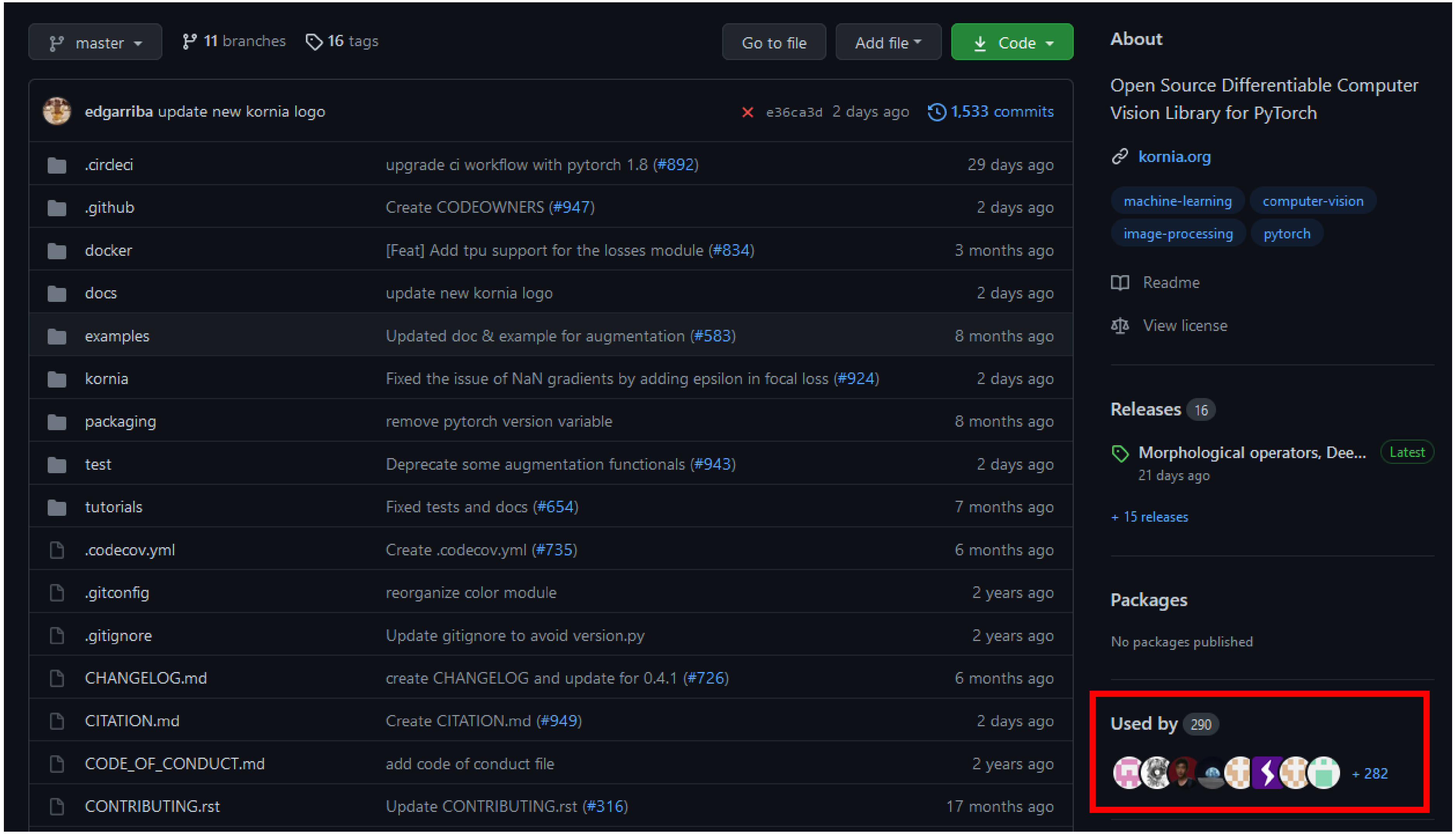
It consists of various methods for deep learning on graphs and other irregular structures, also known as *geometric deep learning*, from a variety of published papers. In addition, it consists of an easy-to-use mini-batch loader for many small and single giant graphs, multi gpu-support, a large number of common benchmark datasets (based on simple interfaces to create your own), and helpful transforms, both for learning on arbitrary graphs as well as on 3D meshes or point clouds. Click here to join our Slack community!

OGB is hosting a **large-scale graph machine learning challenge (OGB-LSC)** at **KDD Cup 2021** from **March 15th to June 8th** in order to discover innovative solutions for large-scale node classification, link prediction and graph regression. We are looking forward to your participation!

PyTorch Geometric makes implementing Graph Neural Networks a breeze (see here for the accompanying tutorial). For example, this is all it takes to implement the edge convolutional layer:

```
import torch
from torch.nn import Sequential as Seq, Linear as Lin, ReLU
from torch_geometric.nn import MessagePassing
```

Getting a good idea



edgarriba update new kornia logo ✕ e36ca3d 2 days ago 1,533 commits

.circleci	upgrade ci workflow with pytorch 1.8 (#892)	29 days ago
.github	Create CODEOWNERS (#947)	2 days ago
docker	[Feat] Add tpu support for the losses module (#834)	3 months ago
docs	update new kornia logo	2 days ago
examples	Updated doc & example for augmentation (#583)	8 months ago
kornia	Fixed the issue of NaN gradients by adding epsilon in focal loss (#924)	2 days ago
packaging	remove pytorch version variable	8 months ago
test	Deprecate some augmentation functionals (#943)	2 days ago
tutorials	Fixed tests and docs (#654)	7 months ago
.codecov.yml	Create .codecov.yml (#735)	6 months ago
.gitconfig	reorganize color module	2 years ago
.gitignore	Update gitignore to avoid version.py	2 years ago
CHANGELOG.md	create CHANGELOG and update for 0.4.1 (#726)	6 months ago
CITATION.md	Create CITATION.md (#949)	2 days ago
CODE_OF_CONDUCT.md	add code of conduct file	2 years ago
CONTRIBUTING.rst	Update CONTRIBUTING.rst (#316)	17 months ago

About

Open Source Differentiable Computer Vision Library for PyTorch

kornia.org

machine-learning computer-vision image-processing pytorch

Readme

View license

Releases 16

Morphological operators, Dee... Latest
21 days ago

+ 15 releases

Packages

No packages published

Used by 290

+ 282

Summary

- Pick a framework (try running their notebooks/examples!):
  - Project 1: NLP
  - Project 2: CV
  - Project 3: Graphs and points
- Brainstorm a project. It does not have to be particular big as you only have 4 full days for working on it
- Write a small (max 1 page) project description including:
  - What model do intent to implement
  - What data are you going to use
  - How you think the chosen framework can be incorporated