

The pytorch ecosystem

02457 Machine Learning Operations

Nicki Skafte Detlefsen,

Postdoc

DTU Compute

The ecosystem



Collection of framework
build to be used in
combination with
Pytorch

PyTorch

Get Started Ecosystem Mobile Blog Tutorials Docs ▾ Resources ▾ GitHub 🔍

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 frameworks
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 Friday afternoon you will be asked to upload your 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



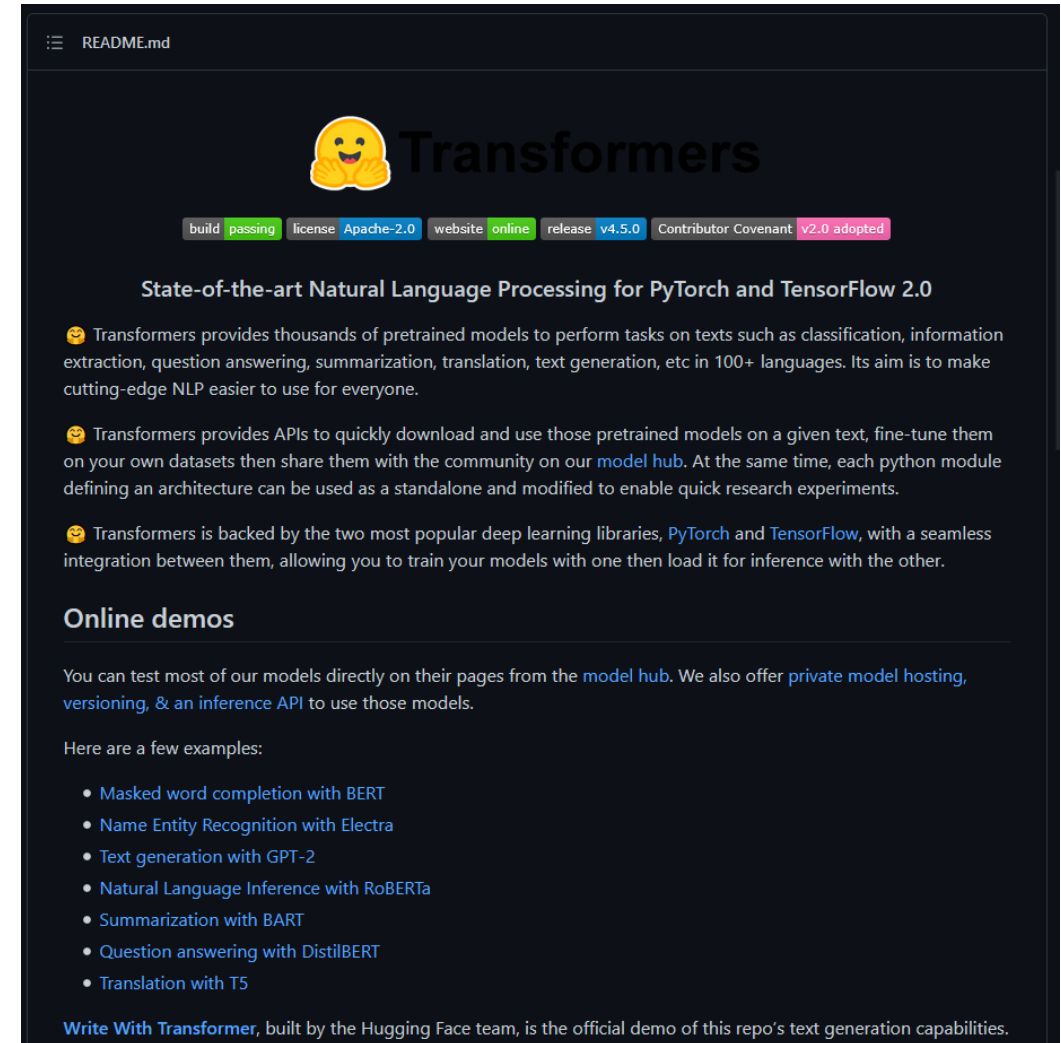
Framework: Transformers (Huggingface)

*

<https://github.com/huggingface/transformers>

Most starred framework in the ecosystem

Use to do Natural Language Processing



Project 2: CV



Framework: Kornia

<https://github.com/kornia/kornia>

Differentiable computer vision
algorithms

The screenshot shows the README for the Kornia library. At the top is the Kornia logo, a 3D cube with red, green, and blue faces. Below the logo is a horizontal bar with various status indicators: License (Apache 2.0), tests-cpu-versions (passing), tests-cuda-versions (failing), codecov (96%), pypi package (0.5.0), and docs (passing). The text describes Kornia as a differentiable computer vision library for PyTorch, consisting of routines and modules for solving generic computer vision problems. It uses PyTorch as its main backend. Below this is a code snippet showing how to load a video frame and apply random rotation. To the right of the code is a 2x2 grid of images showing a cow on a bridge with a full moon in the background, demonstrating the rotation effect. The bottom section is titled 'Overview' and describes the library's composition and its focus on tensor-based operations.

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.

```
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

The screenshot shows the README.md file for PyTorch Geometric. At the top, there's a navigation menu with 'README.md' selected. Below the menu is the PyTorch Geometric logo, which consists of an orange 3D polyhedron and the text 'PyTorch geometric'. Under the logo is a row of status badges: 'pypi package 1.6.3', 'build passing', 'docs passing', 'codecov 84%', 'contributions welcome', 'slack', and 'pyg'. Below the badges are links for 'Documentation', 'Paper', 'Colab Notebooks', 'External Resources', and 'OGB Examples'. The main text describes PyTorch Geometric (PyG) as a geometric deep learning extension library for PyTorch, mentioning its various methods for deep learning on graphs and other irregular structures, its mini-batch loader, benchmark datasets, and helpful transforms. Below this is a section for the 'OGB-LS@ KDD Cup 2021' challenge, which is a large-scale graph machine learning challenge. The text mentions that OGB is hosting this challenge from March 15th to June 8th for node classification, link prediction, and graph regression. At the bottom, there's a section about implementing Graph Neural Networks with PyTorch Geometric, followed by a code snippet showing how to import the necessary modules.

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

Getting a good idea



master 11 branches 16 tags

Go to file Add file Code

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... 21 days ago Latest

+ 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
- Come up with a project
- Write a small (max 1 page) project description including:
 - What model do intent to implement
 - What data are you going to use
 - How to you intent to include the framework