

# The pytorch ecosystem

Machine Learning Operations

Nicki Skafte Detlefsen,

Postdoc

DTU Compute

# The ecosystem



Collection of frameworks  
build to be used in  
combination with  
Pytorch

The screenshot shows the "PyTorch Ecosystem Tools" page. At the top, the PyTorch logo is on the left, and navigation links for "Get Started", "Ecosystem" (highlighted with a red dot), "Mobile", "Blog", "Tutorials", "Docs", "Resources", and "GitHub" are on the right. Below the navigation bar, the heading "ECOSYSTEM TOOLS" is displayed in large, bold letters. A subheading reads: "Tap into a rich ecosystem of tools, libraries, and more to support, accelerate, and explore AI development." Below this is a link "Join the Ecosystem" and a "Sort" dropdown menu. The main content area features four tool cards arranged in a 2x2 grid:

- 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		

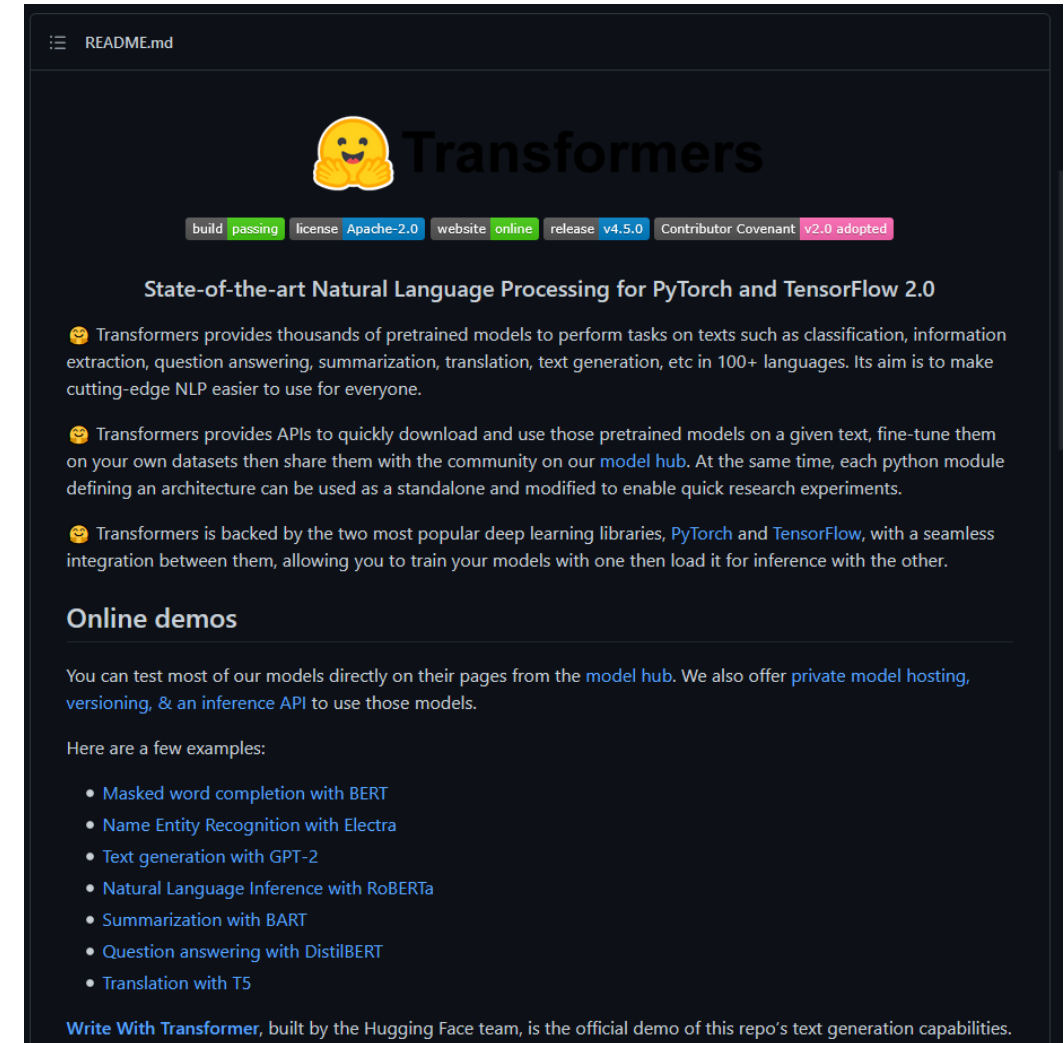


# Project 1: NLP



## Framework: Transformers (Huggingface)

- <https://github.com/huggingface/transformers>
- State-of-the-art NLP models
- Most starred framework in the ecosystem

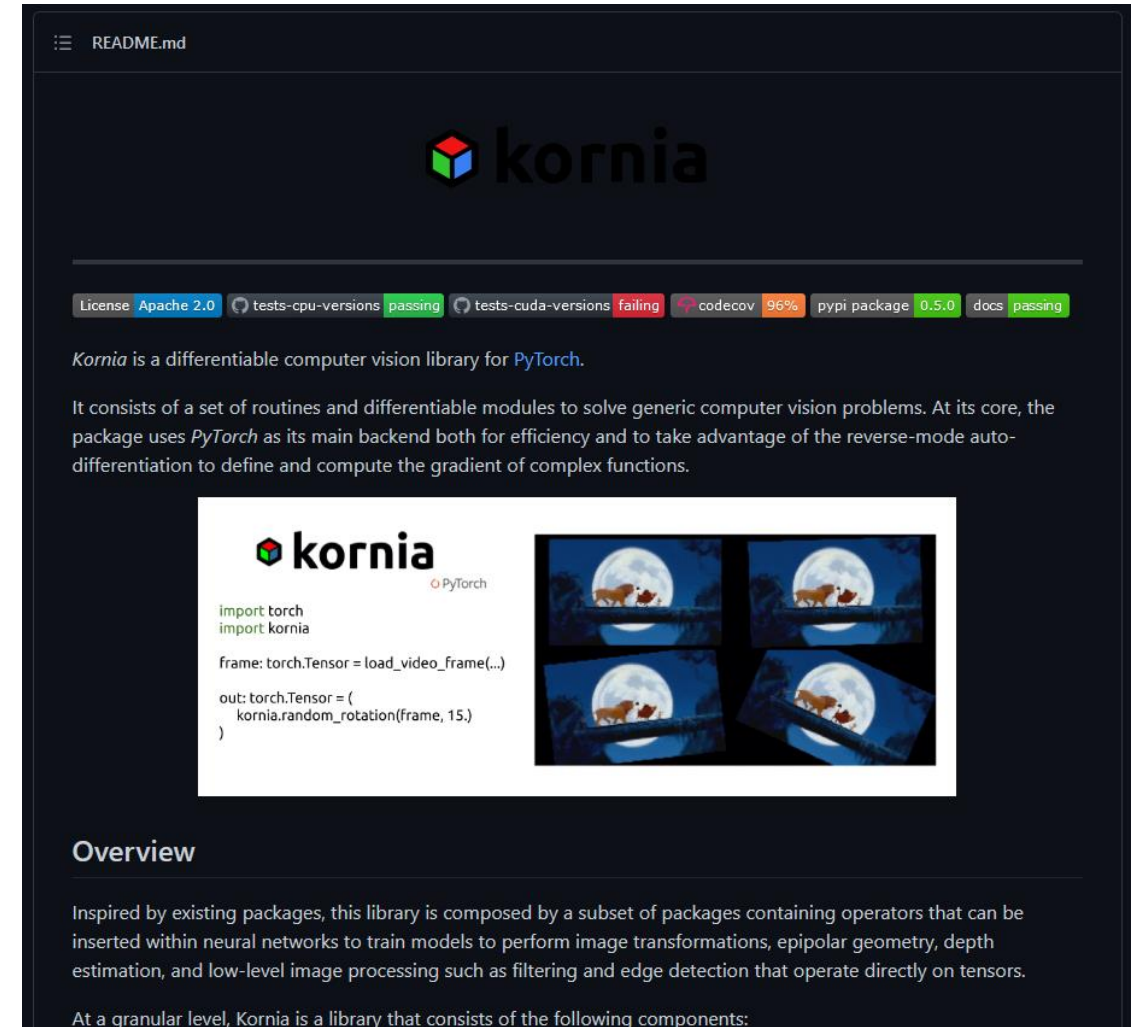


# Project 2: CV



## 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.

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

- [https://github.com/rusty1s/pytorch\\_geometric](https://github.com/rusty1s/pytorch_geometric)
- Neural networks on graphs and point clouds



# 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... 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 particularly big as you only have 4 full days for working on it
- Write a small (max 1 page) project description including:
  - What model do you intend to implement
  - What data are you going to use
  - How you think the chosen framework can be incorporated



# Checklist (also in todays readme)



- Create a git repository
- Make sure that all team members have write access to the github repository
- Create a dedicated environment for you project to keep track of your packages
- Create the initial file structure using cookiecutter
- Fill out the ``make_dataset.py`` file such that it downloads whatever data you need and
- Add a model file and a training script and get that running
- When you have something that works somewhat, remember at some point to do some profiling and see if you can optimize your code
- Remember to fill out the ``requirements.py`` file with whatever dependencies that you are using
- Write unit tests for some part of the codebase and calculate the
- Get some continuous integration running on the github repository
- Use either tensorboard or wandb to log training progress and other important metrics/artifacts in your code
- Remember to comply with good coding practices while doing the project

# Hand-in



- By 17:00 today ONE group member should send an email to me ([nsde@dtu.dk](mailto:nsde@dtu.dk)) with the following info
  - Link to github repository
  - Study number of all group members
  - Your project description

# Exam format



Thursday 24/6 – internal evaluation by Nicki and Søren

- Group presentation
  - 7-10 minutes of powerpoint/reposatory showcase
  - 7-10 minutes of discussion
- What you will be evaluated on:
  - How well you have included what is though in the course
- What you will NOT be evaluated on
  - How epic your deep learning model is

# Some good advise



1. Document everything
  - Take screenshots of your work
  
2. Parallize work
  - Many of the checkpoints are independent of each other