The pytorch ecosystem

Machine Learning Operations

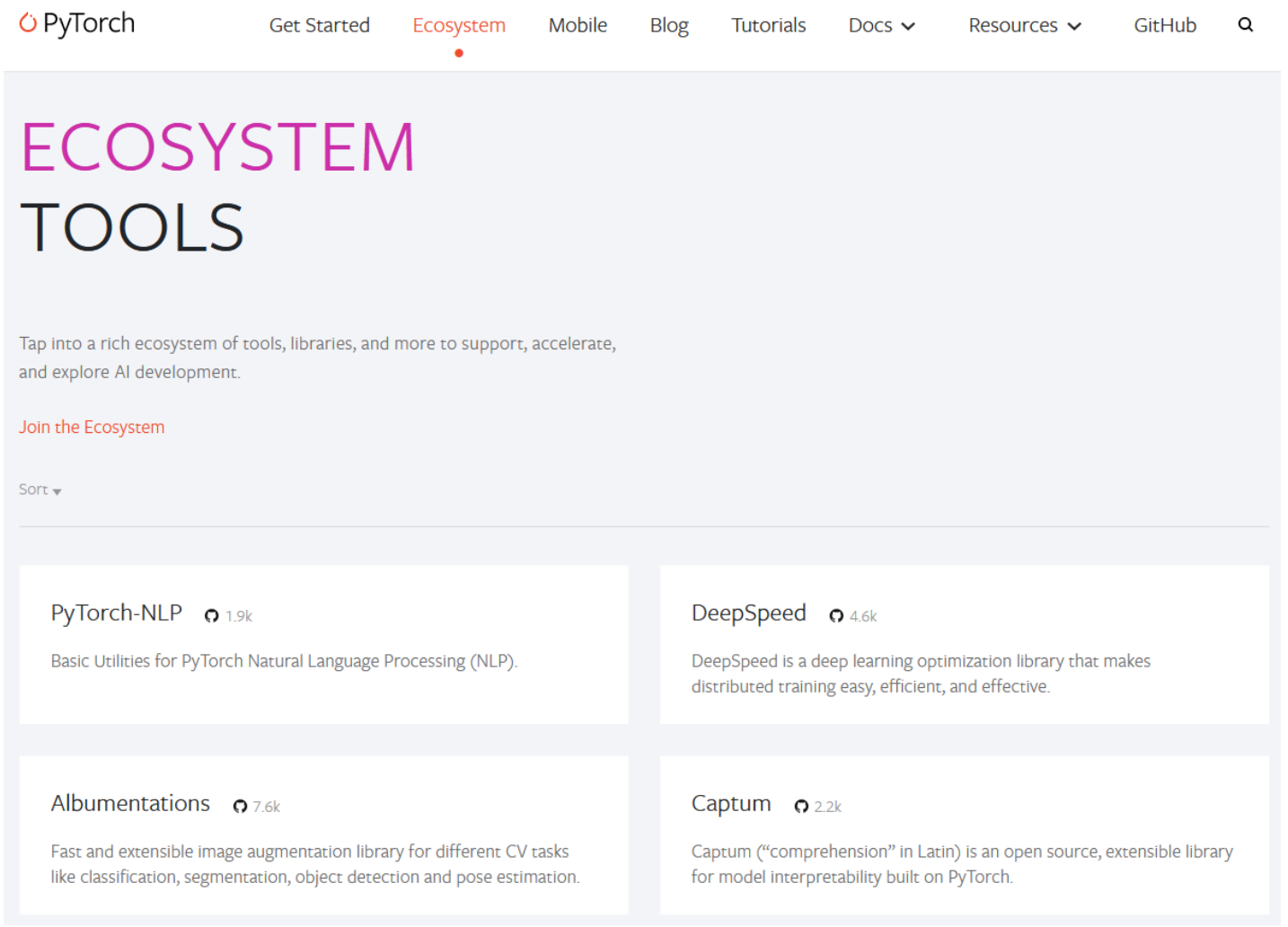
Nicki Skafte Detlefsen,

Postdoc

DTU Compute

The ecosystem

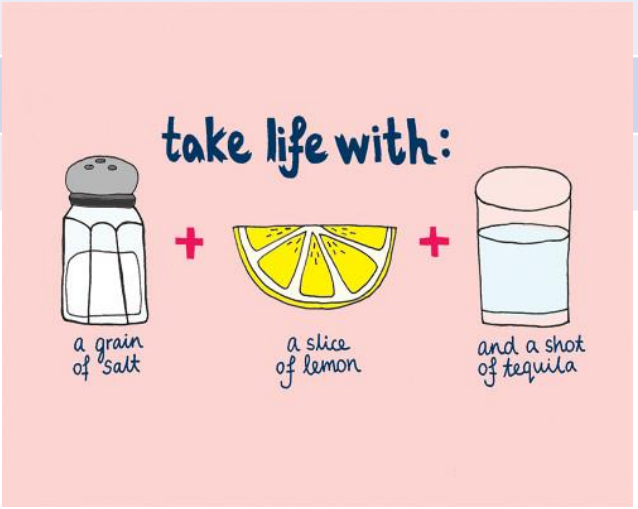
Collection of frameworks build to be used in combination with Pytorch



Fremwork categorising

MLOps

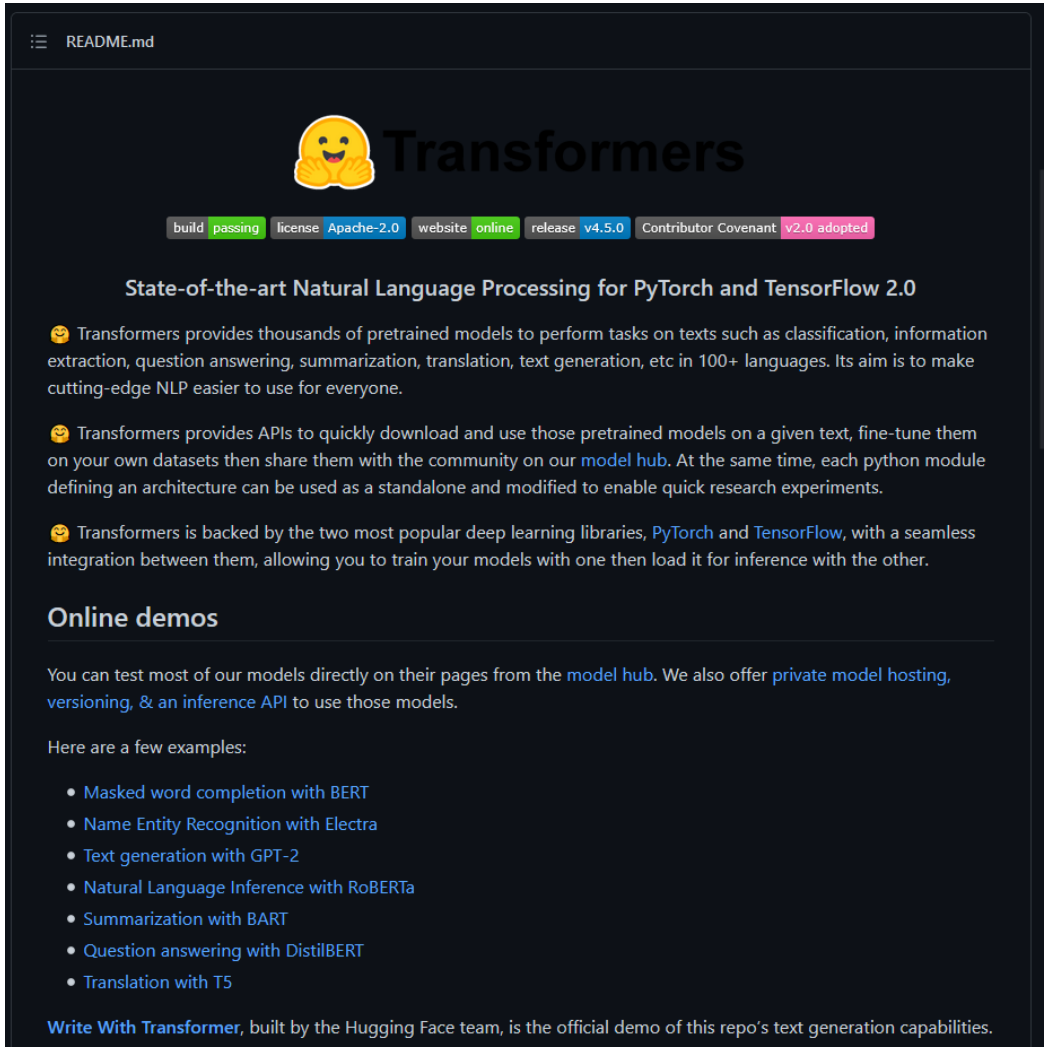
Data specific frameworks	Training frameworks	Utility fremeworks
Transformers	fastai	Albumentations
Detectron2	Ray	PySyft
Pytorch geometric	Pytorch Lightning	Pyro
Flair	Horovod	Optuna
AllenNLP	DeepSpeed	Hydra
ParlAI	ONNX Runtime	Pytorch Metric Learning
DGL	skorch	Einops
PyTorch3D	Ignite	
MMF	Polyaxon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



Project 1: NLP

Framework: Transformers (Huggingface)

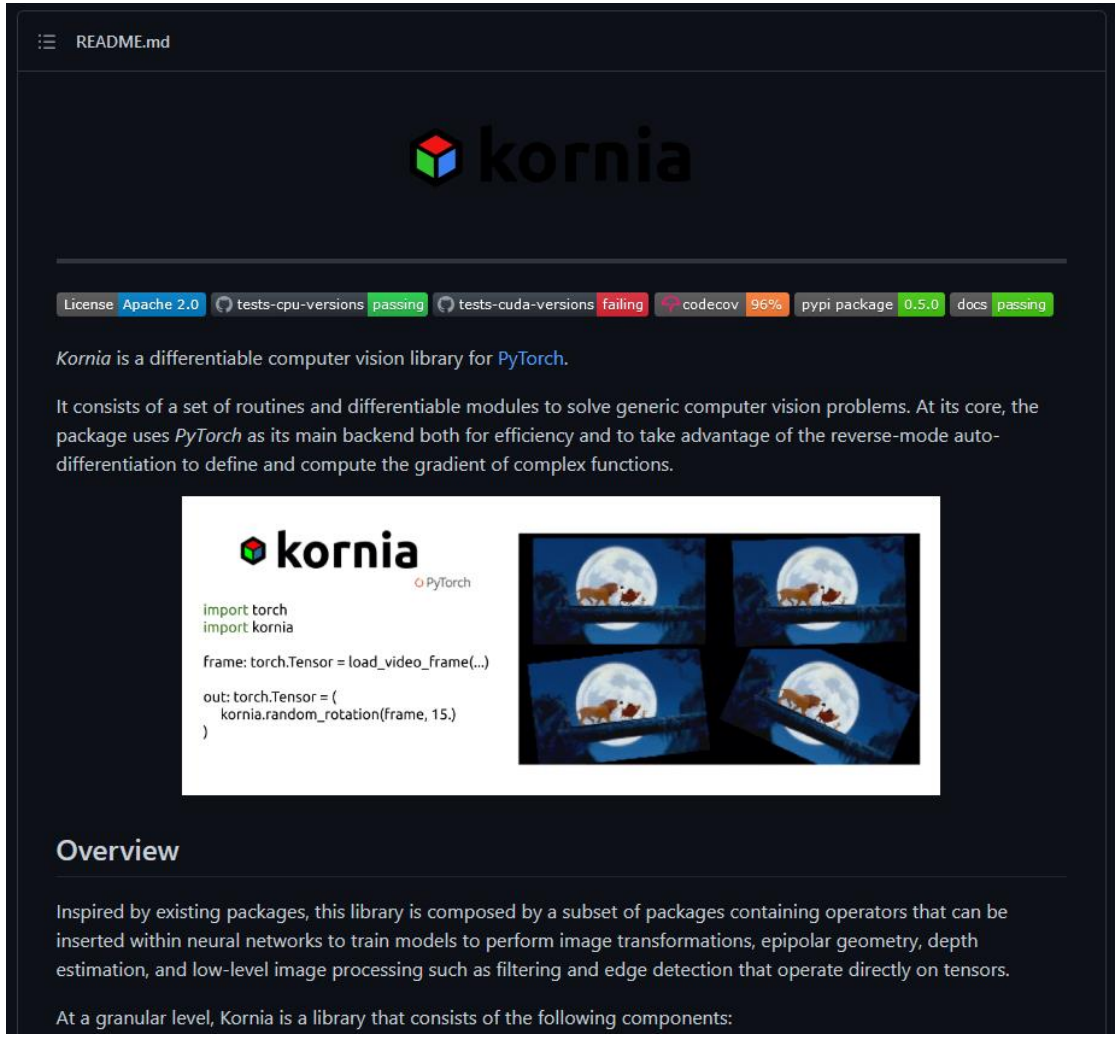
- https://github.com/huggingface/transformers
- State-of-the-art NLP models
- Most starred framework in the ecosystem



Project 2: CV

Framework: Kornia

- https://github.com/kornia/kornia
- Differentiable computer vision algorithms



Project 3: Graphs and points

Framework: Pytorch Geometric

- https://github.com/rusty1s/pytorch_geometric

- Neural networks on graphs and point clouds

README.md

PyTorch geometric

pypi package 1.6.3 | build passing | docs passing | codecov 84% | contributions welcome | slack pyg

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-LSC @ KDD Cup 2021

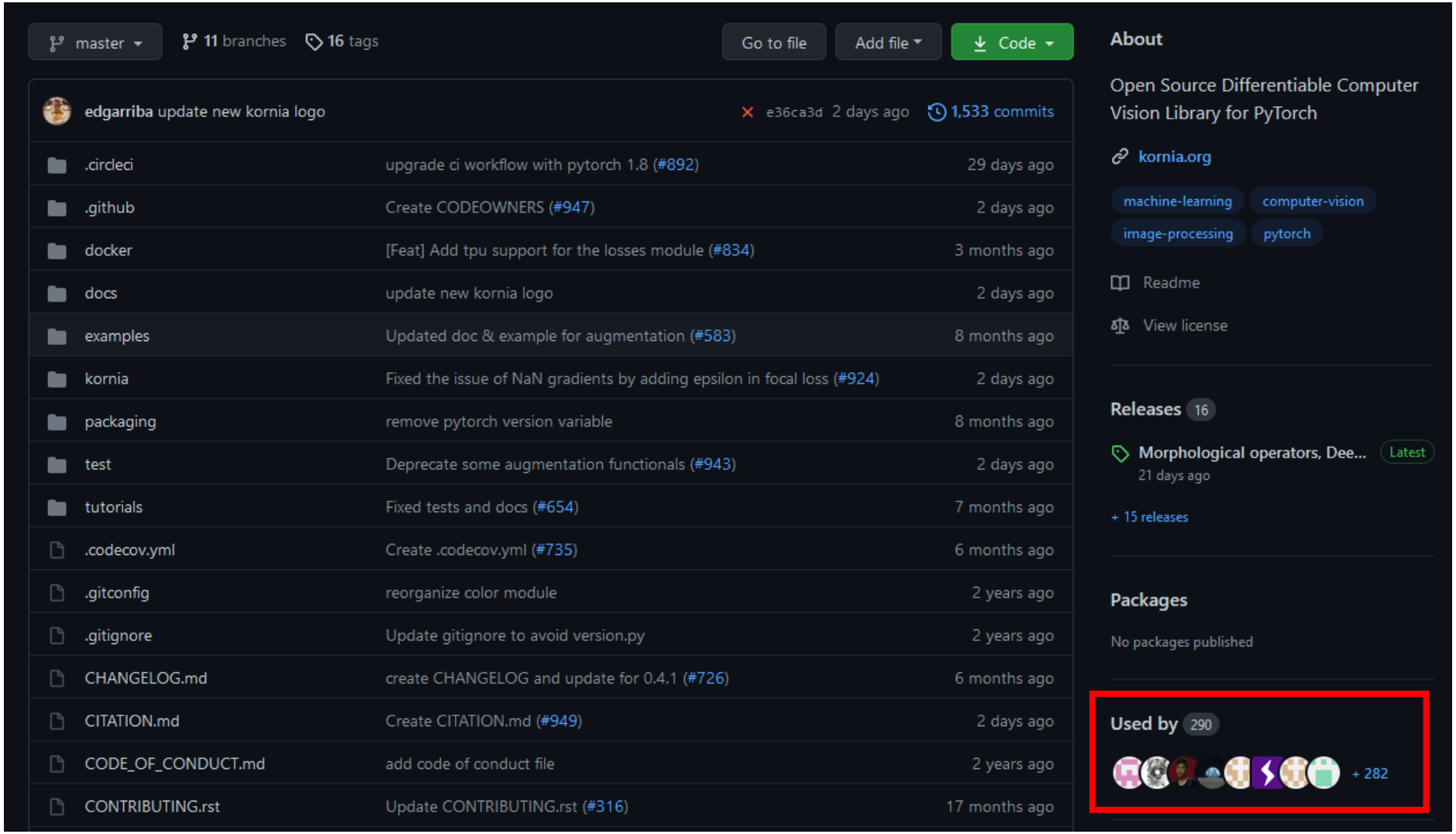
Large - Scale Challenge

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



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

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 to some profiling and see if you can optimize your code
- Remember to fill out the `requirement.py` file with whatever dependencies that you are using
- Write unit tests for some part of the codebase and calculate the
- Get some continues 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) with the following info
  - Link to github reposatory
  - 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