Hardware Accelerated Scientific Computing with Python

Recep Erol

COSMOS Friday Meeting

April 16, 2021
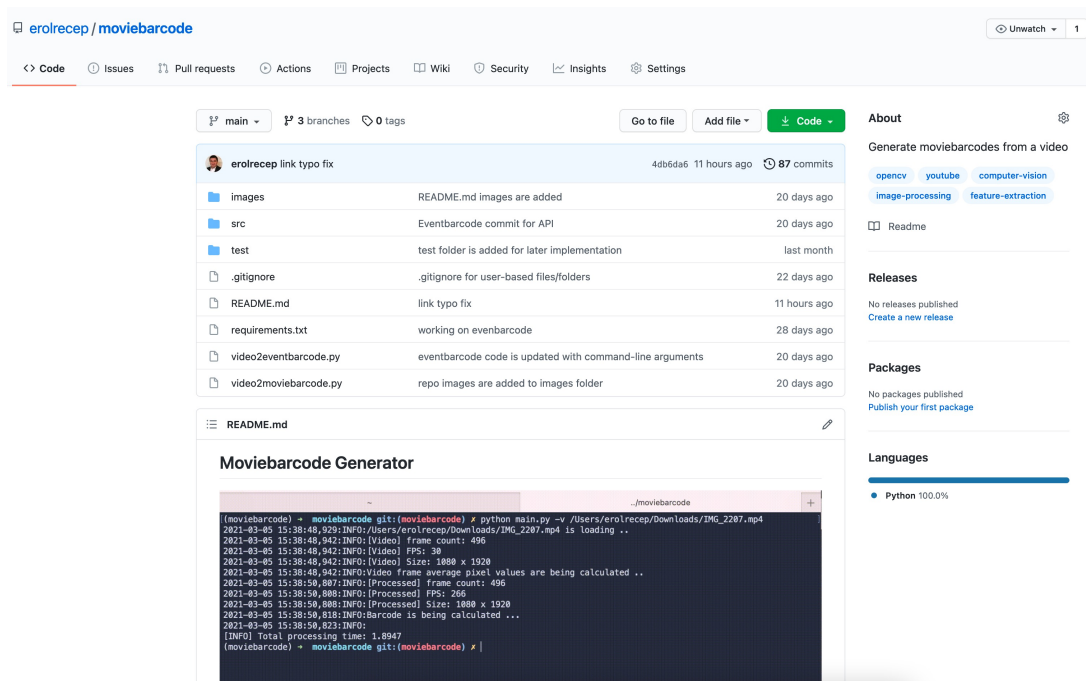
Agenda

- Research Progress
- Tech Stack and Tools
- Future Needs

Research Progress

- YouTube Video Categorization with
    - Moviebarcode
    - Topic Modeling
    - Data Fusion
    - t-SNE
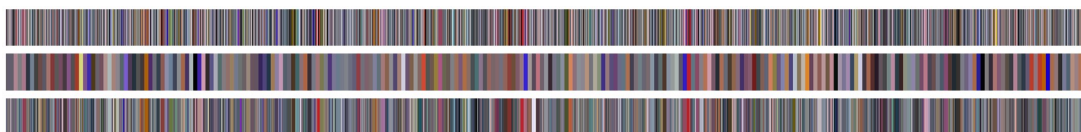   - Opinion-Mining

Research Progress - Moviebarcode

- Single moviebarcode
- Multi-thread moviebarcode
- Multi-process moviebarcode
- GPU-based moviebarcode (in progress)

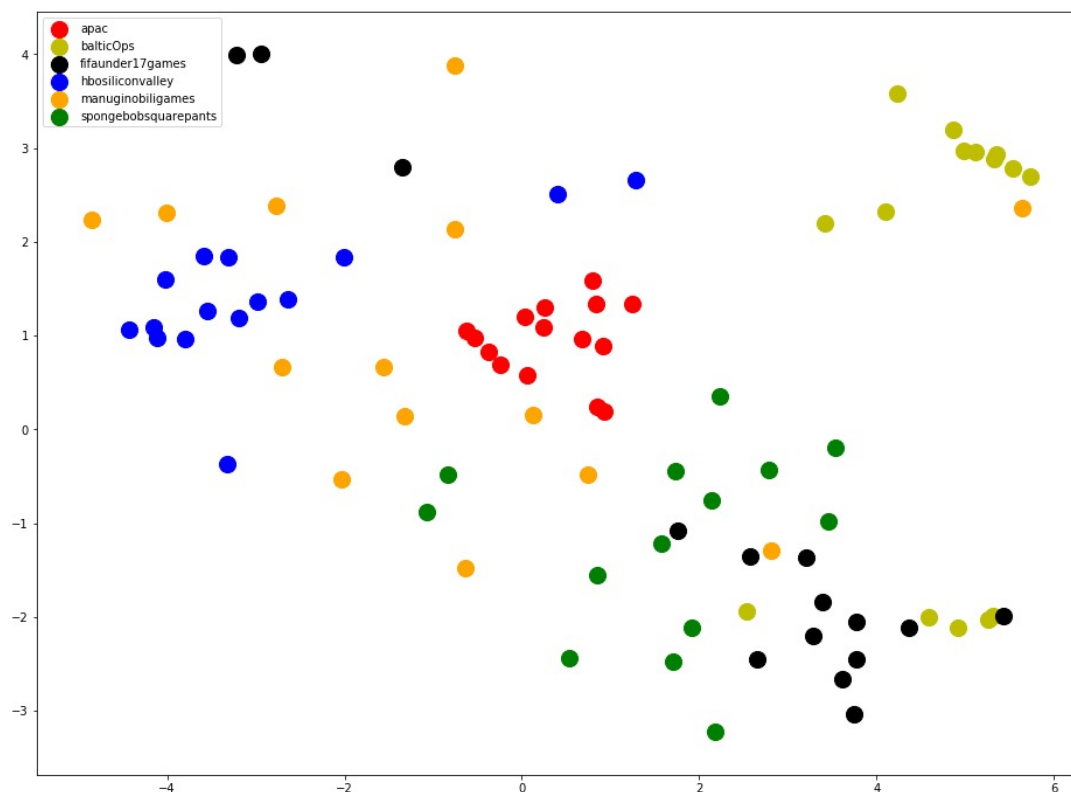# Research Progress - Eventbarcode

- Types of eventbarcode generations
  - First-frame (0.3 – 0.45 seconds)
  - Dominant Colors (1.2 – 2 seconds)
  - Short-clip (Future Plan)
- Single-thread eventbarcode
- Multi-process eventbarcode (in progress)
- GPU-based eventbarcode (future plan)

Research Progress – Text Processing

- Text2Vec for feature extraction
- Topic modeling
- Language models (such as BERT)
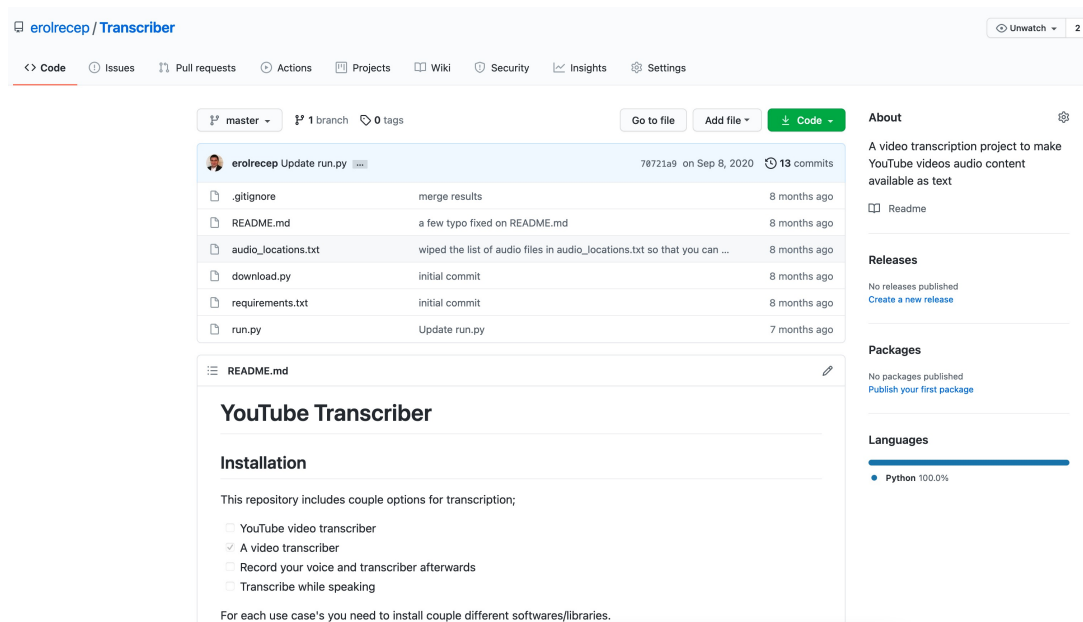- Opinion scores
  - Emotion
  - Toxicity
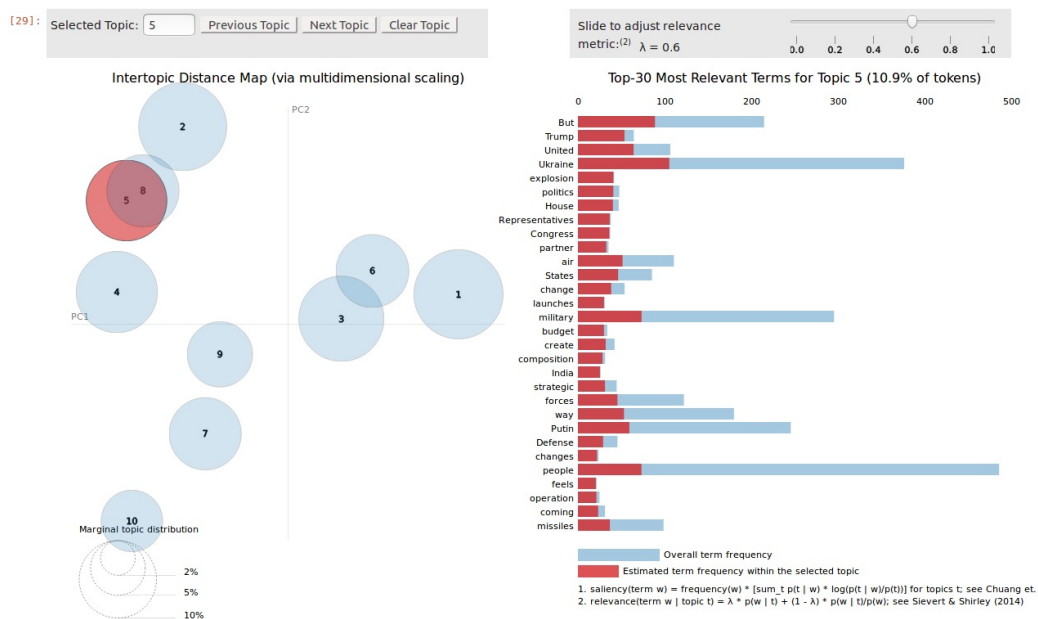
Research Progress – Unsupervised Visualization
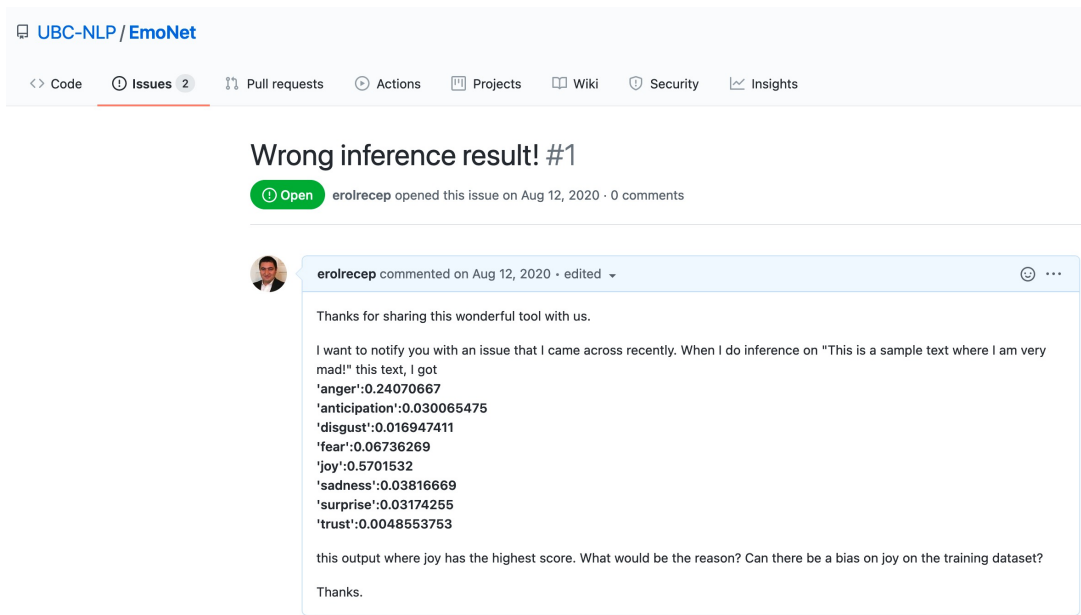

- t-SNE
- UMAP (Future Plan)

Research Progress – Speech Recognition

- SoX

- DeepSpeech

- Wav2letter++ (future plan)

- Nemo (future plan)

Research Progress – Opinion Mining

- EmoNet
  - Training needed
  - Labeled data
- Audio-based opinion-mining (future plan)
- Visual content-based opinion mining (future plan)

Research Progress – Video Characterization

- Plotly based treemap visualization

- Building web tool for YouTube Video Characterization

Tech Stack and Tools - Current

- Python & C++
- NumPy, Scikit-Learn, TensorFlow, and Keras
- OpenCV
- CUDA, CUDA-C
- SoX – Sound Exchange

- CPU Programming
- Parallel Computing
- Distributed Computing
- Shared Memory Programming (OpenMP & MPI)
- GPU Programming
- Computer Vision Programming

Tech Stack and Tools - Future

- CPython

- Numba (or PyCUDA)

- Faiss

- CUDA – cuDNN

- JAX

Tech Stack and Tools - Future

Numba

Build for Scientific Computing

Just in Time (JIT) Compiler

LLVM Compiler Support

CUDA and cuDNN Support

C and Fortran Codebase

NumPy code -> Hardware Code -> Machine Code

```python
from numba import jit
import random

@jit(nopython=True)
def monte_carlo_pi(nsamples):
    acc = 0
    for i in range(nsamples):
        x = random.random()
        y = random.random()
        if (x ** 2 + y ** 2) < 1.0:
            acc += 1
    return 4.0 * acc / nsamples
```

Tech Stack and Tools - Future

- AutoGrad (Deprecated)
- JIT
- XLA
- JAX
- Hardware Acceleration on GPU and TPU

Tech Stack and Tools - Future

- Python optimized AVX instruction Sets
- Built-in CPU acceleration for NumPy, Scikit-Learn, and visualization libraries

Intel® Distribution for Python*
Develop fast, performant Python code with this set of essential computational packages including NumPy, SciPy, scikit-learn*, and more.

Future Needs

- New coders are needed for computer vision & GPU programming
- Tool development engineers are needed:
  - Computer vision programming (OpenCV & Python)
  - Machine learning algorithm programmer (TensorFlow & CUDA)

Hardware Accelerated Scientific Computing with Python

Recep Erol

COSMOS Friday Meeting

April 16, 2021