FOSSIA

EXPLORING FRAMESTORY

Architecture and extending support to modern Python tooling for improved accessibility

PRESENTED BY:

Keerthana Rajesh Kumar
Founder, FOSSIA

Access the slides

https://fossia.org/blog/extending-framestory/

About Me

- Yet another free software and DEIA advocate, known as @grittypuffy
- Used to do security stuff as an undergraduate
- Transitioned to accessibility development
- Founder at FOSSIA and InLibre
- Started building multimedia accessibility utilities after facing personal struggles with navigating visual content while working on sign language data

Agenda

01	Overview
02	Challenges in multimedia accessibility
03	Building framestoryx
04	Working
05	Design Considerations and Challenges
06	What next?

Overview

framestoryx is an extended fork of FrameStory, a Python library developed for generating video descriptions

Features

01 Frame-wise video description generation

02 Improved compatibility with modern Python tooling

03 De-duplication of video descriptions

Challenges in multimedia accessibility

Lack of video descriptions

Creates challenges for photosensitive people and persons with visual impairment

Lack of open-source video description libraries

Creates reliance on proprietary and cloud-based services, posing vendor lock-in and privacy concerns

Poor integration with other accessibility libraries

Creates friction in delivering robust multimedia accessibility experience

Building framestoryx: Philosophy

Robust tooling support

The library must be well maintained and support modern Python tooling

Good Integration

The library must be usable by other Python applications for development of multimedia accessibility services

TranscribeIt already uses framestoryx for video processing

Extensible

The library must address accessibility and UX barriers with video description generation

Working

1. Video frame capture: Create frame captures using OpenCV Video Capture (in-memory)
2. Convert frames to grayscale to compute structural difference
3. Check if threshold has exceeded in difference in pixels between 2 consecutive frames and add the frame to significant frames
4. Generate description for frames using Salesforce BLIP model
5. De-duplicate descriptions and capture timestamps

Design Considerations and Challenges

BLIP over CLIP or YOLO

BLIP provides detailed descriptions for real-time videos over CLIP (which is useful for semantic indexing) or YOLO (which is used for fixed object detection)

Duplicate descriptions despite de-duplication

Despite de-duplication of descriptions for significant frames, there are considerable amount of duplicates due to significant frame intervention which impacts UX of accessible video description generation.

A viable approach will be to generate a summary for the description, which may reduce accuracy, but provides overall context.

Processing efficiency

The frame and description generation takes an average of 7 times per second of video which hinders efficiency. Parallel description generation would mean hindrance on de-duplication at cost of optimization.

What Next?

Integrate translation for video descriptions

Integrate timestamp and duration skips for frame processing

Generation of summary for video description

Improve efficiency of processing frames using thread-safe async processing

Thank you

REACH OUT

E-mail	fossia@riseup.net
Website	https://fossia.org
Codeberg	https://codeberg.org/fossiaorg
GitHub	https://github.com/fossiaorg
Matrix (Chat)	https://matrix.to/#/#fossiaorg:matrix.org

VISIT OUR WEBSITE