Научная визуализация, 2025, том 17, номер 5, страницы 29 - 37, DOI: 10.26583/sv.17.5.04

Автоматизированный поиск в видеопотоке для исследований и контент-анализа видеофрагментов

Н.А. Бондарева[1], А.Е. Бондарев[2], С.В. Андреев[3], И.Г. Рыжова[4]

Институт прикладной математики им. М.В. Келдыша Российской академии наук

[1] ORCID: 0000-0002-7586-903X, nicibond9991@gmail.com
[2] ORCID: 0000-0003-3681-5212, bond@keldysh.ru
[3] ORCID: 0000-0001-8029-1124, esa@keldysh.ru
[4] ORCID: 0000-0003-1613-3038, ryzhova@gin.keldysh.ru

Аннотация

В данной работе рассматривается задача целенаправленного поиска в видеопотоке определенных объектов по запросу и фиксирование временных меток их появления. Поскольку представленные на современном рынке решения не соответствовали поставленным задачам, то было принято решение реализовать такой инструмент самостоятельно в рамках текущего проекта исследований, проводимых в ИПМ им. М.В. Келдыша РАН.

Ключевые слова: видеопоток, видеоконтент, поиск по запросу, временные метки.

1. Введение

Нарастающая роль видеоконтента во всех областях исследований, от социальных до научных, все чаще вызывает необходимость обращаться к количественным методам анализа видеоконтента. Исследователи в различных областях, например, в области социологии медиа и цифровых гуманитарных наук сталкиваются с необходимостью анализировать огромные объемы видеоматериалов для выявления паттернов репрезентации того или иного явления, изучения экранного времени персонажей, анализа гендерного баланса и других аспектов медиаконтента [1, 2].

Однако традиционные методы поиска и анализа появлений конкретных объектов в видеоматериалах остаются крайне трудозатратными. Исследователь вынужден просматривать часы видеоматериала, вручную фиксируя временные интервалы появления интересующих персонажей. Такой подход не только требует значительных временных ресурсов, но и подвержен субъективным ошибкам, особенно при работе с длительными видеоматериалами или большими медиа коллекциями [3].

Существующие коммерческие решения для видеоанализа либо ориентированы на профессиональное видеопроизводство (Adobe Premiere Pro [4], Final Cut Pro [5]) и не нацелены на задачи видеоанализа, либо требуют значительных финансовых вложений и технической экспертизы (облачные сервисы AWS Rekognition [5], Google Video Intelligence API [6]). На сегодняшний день существует очень ограниченный диапазон инструментов, которые способны предоставить исследователю возможность анализировать значительный объем материала согласно его потребностям. Часто задачи исследования имеют столь узконаправленную специфику, что широкодоступные программные комплексы функционально не соответствуют требованиям поставленных задач. Это ставит перед исследователем дополнительную задачу непосредственно перед научным исследованием сначала подобрать подходящий инструмент для поиска или анализа, а в случае отсутствия такового создать его самому.

В данной работе рассматривается задача целенаправленного поиска в видеопотоке определенных объектов по запросу и фиксирование временных меток их появления. Поскольку представленные на современном рынке решения не соответствовали поставленным задачам, то было принято решение реализовать такой инструмент самостоятельно в рамках текущего проекта исследований, проводимых в ИПМ им. М.В. Келдыша РАН.

Целью исследования является разработка и описание практического инструмента для автоматизированного поиска объектов в видеоконтенте, доступного для использования исследователями без специальных технических навыков.

В число поставленных задач входит:

1. Анализ существующих решений для поиска персонажей в видео и выявление их ограничений для исследовательского применения;
2. Описание архитектуры разработанного инструмента;
3. Разработка программной реализации
4. Выявление перспектив развития и возможности интеграции инструмента в различные области исследования.

2. Обзор существующих решений

Современный рынок программного обеспечения предлагает ряд решений для работы с видеоконтентом, однако их применимость для исследовательских задач ограничена рядом нюансов, имеющих узконаправленную специфику применения.

Профессиональные видеоредакторы такие как Adobe Premiere Pro и Final Cut Pro включают функции автоматического распознавания лиц, однако они ориентированы преимущественно на задачи видеопроизводства. Основные ограничения этих решений для исследовательского применения включают: высокую стоимость лицензий, сложность освоения для пользователей без технического бэкграунда, отсутствие возможности экспорта структурированных данных для последующего статистического анализа [8].

Облачные сервисы (AWS Rekognition Video, Google Video Intelligence API, Microsoft Azure Video Indexer) предоставляют мощные возможности для анализа видеоконтента, включая распознавание и отслеживание лиц. Однако их применение в академических исследованиях может быть затруднено по ряду причин: начиная от высокой стоимости обработки больших объемов видеоматериалов и заканчивая сложностью политической ситуации, продуцирующей ряд ограничений для российских ученых. Также следует учитывать тот факт, что облачные технологии подходят не для всех научных задач, так как некоторые данные требуют исключительно локальной обработки в силу своей специфики и требований к конфиденциальности при загрузке исследовательских материалов на внешние серверы [9, 10].

В академической среде существует ряд специализированных инструментов для анализа видеоконтента. К примеру, ELAN (EUDICO Linguistic Annotator) широко используется лингвистами для аннотации видеоматериалов, однако требует ручной разметки и не включает автоматического распознавания лиц или объектов [11]. OpenCV-based решения предоставляют мощную техническую базу для создания систем видеоанализа, однако их использование требует значительных программистских навыков. Большинство готовых реализаций ориентированы на технических специалистов и не предоставляют простого интерфейса для исследователей-гуманитариев [12].

Проект VideoANT (Video Annotation Tool) от Университета Миннесоты предлагает веб-интерфейс для аннотации видео, но также требует ручной работы и не включает функций автоматического распознавания [13]. В области цифровых гуманитарных наук разработаны отдельные инструменты для анализа медиаконтента. Cinemetrics проект фокусируется на анализе монтажных характеристик фильмов, но не включает функций поиска объектов или определенных персонажей [14]. CLARIAH Media Suite предостав-

ляет исследователям доступ к большим коллекциям медиаконтента с возможностями поиска и анализа, однако функции автоматического распознавания объектов ограничены и у него [15].

Среди существующих программных решений, представленных в открытых интернет-источниках, можно выявить несколько ключевых пробелов, в частности:

1. Недостаток простых решений для решения задачи поиска конкретных объектов/персонажей в видеопотоке;
2. Отсутствие технических средств, позволяющих запустить поиск по образцу по определенному видеоматериалу, предоставленного пользователем;
3. Недоступность инструментов для исследователей без технического бэкграунда - большинство решений требуют программистских навыков или значительных финансовых ресурсов;
4. Ограниченные возможности экспорта структурированных данных для последующего статистического анализа и визуализации;

Эти аспекты подтверждают необходимость и актуальность разработки простого, доступного инструмента для автоматизированного поиска определенных объектов в видеоконтенте, ориентированного на потребности исследователей в различных областях от экспериментальной физики до медиаисследований и цифровых гуманитарных наук.

3. Техническая реализация

Разработанная система представляет собой инструмент для автоматизированного поиска персонажей в видеоконтенте, состоящий из пяти основных этапов: предобработка эталонного изображения, декомпозиция видеоматериала, обнаружение объекта (в нашем случае лица) в кадрах, сопоставление с эталоном и формирование выходных данных в следующем виде:

- директория с вырезанными видеофрагментами
- файл с временными отметками появления искомого объекта в кадре.

Для работы программы создается организованная структура директорий, где группируются необходимые материалы.

После запуска исполняющего файла программа автоматически загружает эталонные лица, добавленные пользователем, затем проводит поиск по загруженным пользователем видеофайлам, анализируя кадры на наличие искомого объекта и сравнивая обнаруженные лица с эталонным образцом, а затем формирует директорию с результатами работы. В ней находятся:

- видеофайлы с найденными фрагментами, содержащими заданные лица;
- JSON-файл с таймкодами сцен, описывающий время начала и окончания каждого фрагмента;
- лог-файл с информацией о процессе обработки;
- файл настроек (для возможной корректировки параметров).

Работа пользователя сводится к подготовке данных (фото и видео) и запуску программы. После этого программа автоматически выполняет поиск лиц и создает результаты в указанной папке.

3.1. Архитектура системы

Система представляет собой модульное приложение, разработанное для автоматического извлечения фрагментов видео, содержащих определенные лица. Основной принцип работы заключается в сравнении лиц, обнаруженных в видеопотоке, с эталонными изображениями, представленными пользователем в виде фотографий.

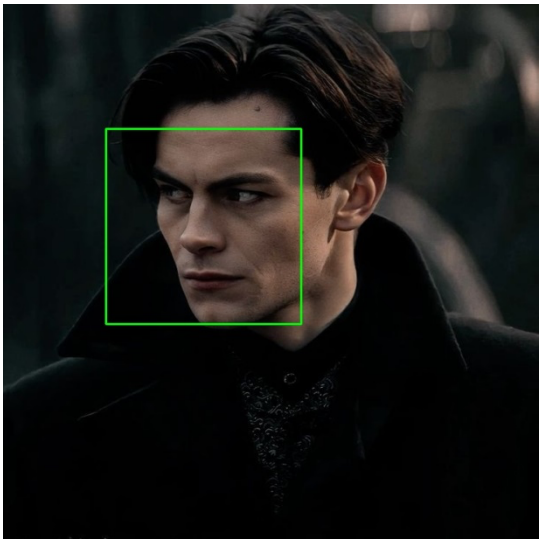

Рисунок 1. Пример распознанного лица

Алгоритм работы программы представлен на схеме (Рис. 2):

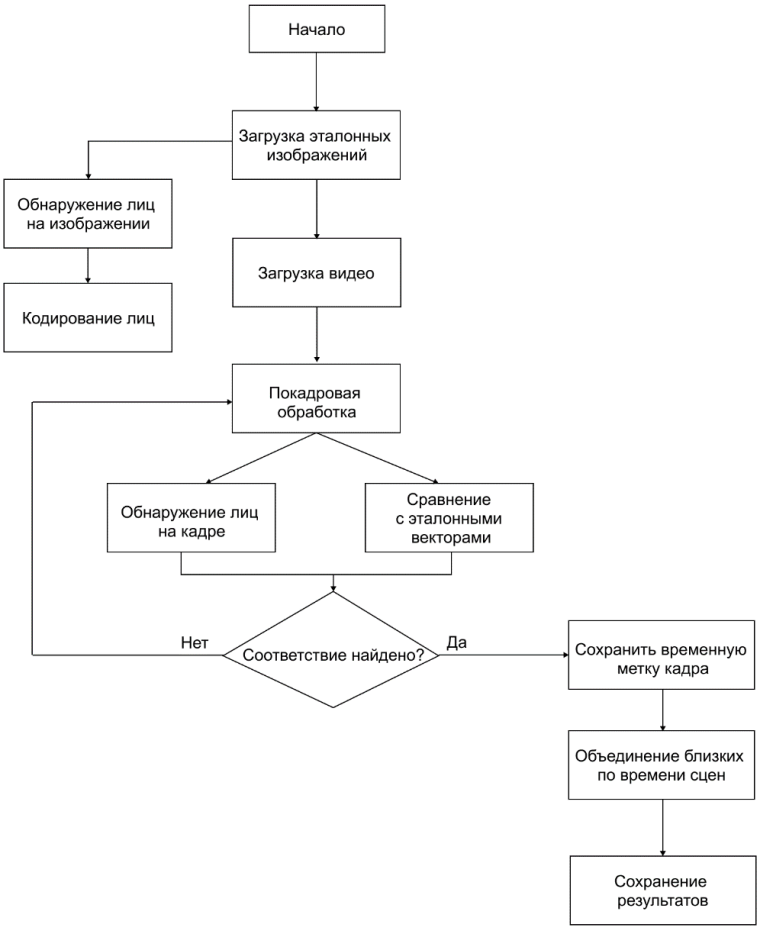

Рисунок 2. Схема алгоритма

Основным компонентом системы является класс FaceExtractor, который инкапсулирует всю логику обработки видео и распознавания лиц. Ключевые методы класса FaceExtractor:

- load_reference_faces(): Загружает эталонные изображения лиц из указанной директории и кодирует их в векторы признаков.

- process_video(): Обрабатывает видеофайл покадрово, обнаруживает лица на каждом кадре, кодирует их и сравнивает с эталонными векторами.
- _merge_scenes(): Объединяет близкие по времени сцены с обнаруженными лицами в единые фрагменты.
- _save_results(): Сохраняет видеофрагменты с обнаруженными лицами и метаданные в указанную директорию.

3.2. Технические параметры

Система разработана на языке программирования Python 3.9. Для распознавания и сопоставления лиц используется библиотека face_recognition [16], представляющая собой Python-обертку над библиотекой dlib [17]. В основе алгоритма лежит подход FaceNet [18], который создает 128-мерные векторы признаков (embeddings) для каждого обнаруженного лица с использованием глубокой нейронной сети на основе архитектуры ResNet.

Для реализации основных функций программы использовались следующие библиотеки:

- opencv-python (cv2) (версия 4.8.1.78) — для работы с видео и изображениями, используемая для загрузки, декодирования и покадровой обработки видео.
- moviepy (версия 1.0.3): — для редактирования видео, используемая для объединения сцен и сохранения видеофрагментов.
- numpy (версия 1.24.3) — для работы с массивами, используемая для хранения и обработки векторов признаков лиц.

Выбор данных программных средств обусловлен простотой использования, широкой доступностью библиотек для машинного обучения и обработки видео, а также хорошей производительностью.

3.3. Предобработка данных

Для обеспечения высокой точности и скорости распознавания лиц, к входным данным применяется ряд шагов предобработки.

Эталонные изображения лиц должны быть представлены в формате JPG, JPEG или PNG с разрешением, достаточным для четкого отображения лица. Процесс загрузки эталонных изображений включает в себя чтение файлов, обнаружение лиц на изображениях и кодирование лиц в 128-мерные векторы признаков с использованием библиотеки face_recognition.

Рисунок 3. Эталонные образцы лица

Система поддерживает видеофайлы в формате MP4, AVI и MKV. Процесс обработки видео включает в себя загрузку видеофайла, декодирование кадров и покадровую обработку. Для уменьшения вычислительной нагрузки, размер кадров изменяется до max_image_size пикселей (параметр настраивается в файле конфигурации). Для ускорения обработки, используется параметр frame_interval, который определяет, как часто будут обрабатываться кадры (например, frame_interval = 5 означает, что будет обрабатываться каждый пятый кадр).

3.4. Формат выходных данных

Результаты работы системы представлены в виде видеофрагментов, содержащих сцены с обнаруженными лицами, и метаданных, описывающих эти фрагменты.

Видеофрагменты сохраняются в формате MP4 с использованием кодека H.264. Метаданные сохраняются в формате JSON и содержат следующую информацию: video_name (имя исходного видеофайла); fragments (список фрагментов с обнаруженными лицами с временными отметками); settings (параметры обработки) и т.д. Также ведется запись лог-файлов (Рис 4).

```
2025-09-03 00:45:03,882 - INFO - Начало загрузки референсных изображений
2025-09-03 00:45:04,488 - INFO - Успешно загружено изображение: im1.jpg
2025-09-03 00:45:05,124 - INFO - Успешно загружено изображение: im2.jpg
2025-09-03 00:45:05,606 - INFO - Успешно загружено изображение: im3.jpg
2025-09-03 00:45:05,608 - INFO - Начало обработки видео: c:\Users\NiciBond\face_detection_project\input_videos\sara.mp4
2025-09-03 00:45:23,477 - INFO - Начало новой сцены: 3.30 сек
2025-09-03 00:45:34,524 - INFO - Конец сцены: 5.30 сек
2025-09-03 00:45:40,445 - INFO - Начало новой сцены: 6.30 сек
2025-09-03 00:45:45,913 - INFO - Конец сцены: 7.30 сек
```

Рисунок 4. Пример отчета программы

Тестирование системы проводилось на следующем оборудовании:

- CPU: Intel Core i7-8700K (6 ядер, 12 потоков).
- RAM: 16 GB DDR4.
- GPU: NVIDIA GeForce GTX 1070 (8 GB)
- Тип накопителя: SSD.
- Программное обеспечение:
- Операционная система: Windows 10 (64-bit).
- Python: 3.9.

3.5. Описание тестов

Для оценки производительности системы были проведены следующие тесты:

1. Тест размера изображения: Целью данного теста является определение влияния размера изображения на скорость обработки. В ходе теста изменялся размер изображения (от 500 до 2000 пикселей) и измерялось время обработки одного кадра.
2. Тест параллельной обработки: Целью данного теста является определение оптимального размера пакета для параллельной обработки. В ходе теста изменялся размер пакета (от 1 до 16) и измерялось время обработки одного кадра.
3. Тест обработки видео: Целью данного теста является определение влияния интервала между кадрами на скорость обработки. В ходе теста изменялся интервал между кадрами (от 1 до 30) и измерялась скорость обработки видео (кадров в секунду).

Для оценки результатов тестов использовались следующие метрики:

- Время обработки (в секундах).
- Скорость обработки (кадров в секунду).
- Использование памяти (в MB).
- Загрузка CPU (в процентах).

Результаты тестов показали, что система успешно обнаруживает заданных персонажей в видеопотоке, формирует видеофрагменты с их участием и сохраняет метаданные, описывающие эти фрагменты.

В частности, тесты подтвердили:

- Эффективность использования библиотеки face_recognition для распознавания лиц.
- Положительное влияние предобработки данных (изменение размера изображений, пропуск кадров) на скорость обработки видео.
- Возможность настройки параметров системы (например, max_image_size, frame_interval, recognition_threshold) для адаптации к различным требованиям к производительности и точности.

Таким образом, можно заключить, что первая версия программного обеспечения успешно реализована и выполняет поставленные задачи. Система демонстрирует стабильную работу и позволяет эффективно находить заданных персонажей в видеопотоке.

Полученные результаты открывают широкие перспективы для дальнейшего развития и применения системы в различных областях, таких как:

- Автоматическая разметка видеоархивов;
- Системы безопасности и видеонаблюдения.
- Анализ медиаконтента и выявление ключевых персонажей.
- Создание персонализированных видеоподборок.

4. Обсуждение

Разработанная система обладает рядом достоинств, которые делают ее эффективной для использования в различных задачах:

- простота использования: система основана на широко известных и доступных библиотеках Python, что упрощает ее настройку.
- эффективность: благодаря использованию предобученной нейронной сети и оптимизированным алгоритмам обработки видео, система обеспечивает высокую скорость и точность распознавания лиц.
- гибкость: Система позволяет настраивать параметры обработки видео (например, размер изображения, интервал между кадрами, порог распознавания) для адаптации к различным требованиям к производительности и точности.
- модульность: Модульная архитектура системы упрощает ее расширение и модификацию для решения новых задач.

Вместе с тем, стоит отметить, что система в первой версии имеет узконаправленный функционал и ориентирована только на поиск лиц в видеопотоке имеет и некоторые ограничения, которые необходимо учитывать при ее использовании:

- зависимость от качества эталонных изображений: Точность распознавания лиц напрямую зависит от качества эталонных фотографий. Нечеткие, плохо освещенные или повернутые фотографии могут привести к снижению точности распознавания.
- ограниченность предобученной модели: Система использует предобученную нейронную сеть, которая может быть неоптимальной для распознавания лиц в определенных условиях (например, при плохом освещении, сильных изменениях ракурса или наличии окклюзий).
- вычислительная сложность: Обработка видео требует значительных вычислительных ресурсов, что может быть проблемой при использовании системы на маломощных устройствах.

По мере работы над улучшениями характеристик системы и расширением ее функциональности можно выделить следующие направления развития:

- использование более современных архитектур нейронных сетей. Замена предобученной модели на более современную архитектуру (например, ResNet, EfficientNet)

может повысить точность распознавания и расширить области её применения и позволить распознавать не только лица, но и более специфические объекты, как к примеру, определенные явления в физических экспериментах.

- разработка адаптивных алгоритмов. Разработка алгоритмов, которые автоматически настраивают параметры системы (например, размер изображения, интервал между кадрами, порог распознавания) в зависимости от характеристик видеопотока, позволит повысить ее эффективность и гибкость.
- оптимизация для работы на GPU. Использование GPU для ускорения вычислений может значительно повысить скорость обработки видео, особенно при использовании сложных нейронных сетей.
- интеграция с другими системами. Интеграция системы с другими системами (например, системами видеонаблюдения, базами данных лиц) позволит расширить ее функциональность и область применения.
- разработка графического интерфейса пользователя (GUI). Создание удобного графического интерфейса упростит использование системы для пользователей, не имеющих опыта работы с командной строкой.

Разработанная система имеет потенциал для применения как для прикладных задач в сфере анализа медиа, так и в различных областях науки: от экспериментальных установок до социальных исследований.

5. Заключение

В данной работе представлена система для автоматического поиска лиц в видеопотоке, основанная на использовании библиотеки face_recognition и оптимизированная для достижения высокой производительности.

Разработанная система представляет собой инструмент для поиска лиц в видеопотоке, который может быть использован в различных областях науки и техники. Предложенные направления развития системы, такие как использование более современных архитектур нейронных сетей, разработка адаптивных алгоритмов и оптимизация для работы на GPU, позволят расширить ее функциональность и повысить ее эффективность, что откроет новые возможности для ее применения в решении широкого круга задач в различных областях науки и техники.

Вклад данной работы заключается в демонстрации практической применимости существующих библиотек и методов оптимизации для решения реальной задачи поиска лиц в видеопотоке. Разработанная система может быть использована в качестве основы для создания более сложных и функциональных приложений в области анализа видеоданных.

Список литературы

1. Moretti F. Distant Reading. - London: Verso, 2013. - 298 p.
2. Manovich L. Cultural Analytics: Analysing Cultural Patterns in the Era of "More Media" // Domus, 2009. - №923. - P. 1-7.
3. Salt B. Film Style and Technology: History and Analysis. - London: Starword, 2009. - 398 p.
4. Adobe Premiere Pro - URL: https://www.adobe.com/products/premiere.html
5. Final Cut Pro - URL: https://www.apple.com/final-cut-pro/
6. Amazon Rekognition - URL: https://aws.amazon.com/rekognition/
7. Video Intelligence API documentation - URL: https://cloud.google.com/video-intelligence/docs
8. Adobe Systems Inc. Adobe Premiere Pro User Guide // Adobe Documentation. - 2023. - URL: https://helpx.adobe.com/premiere-pro/user-guide.html
9. Amazon Web Services. Amazon Rekognition Video Developer Guide. - 2023. - URL: https://docs.aws.amazon.com/rekognition/

10. Google Cloud. Video Intelligence API Documentation. - 2023. - URL: https://cloud.google.com/video-intelligence/docs
11. Wittenburg P., Brugman H., Russel A. ELAN: a Professional Framework for Multimodality Research // Proceedings of the 5th International Conference on Language Resources and Evaluation. - 2006. - P. 1556-1559.
12. Bradski G., Kaehler A. Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library. - Sebastopol: O'Reilly Media, 2016. - 1024 p.
13. VideoANT Project. Video Annotation Tool // University of Minnesota. - URL: https://ant.umn.edu/
14. Cinemetrics. Film Measurement and Analysis // Cinemetrics Database. - URL: http://www.cinemetrics.lv/
15. CLARIAH Media Suite. Research Platform for Media Studies // CLARIAH Consortium. - URL: https://mediasuite.clariah.nl/
16. Geitgey A. Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning // Medium. - 2016. - URL: https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78
17. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). - 2016. - P. 770-778.
18. Schroff F., Kalenichenko D., Philbin J. FaceNet: A Unified Embedding for Face Recognition and Clustering // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). - 2015. - P. 815-823

Scientific Visualization, 2025, volume 17, number 5, pages 29 - 37, DOI: 10.26583/sv.17.5.04

Automated video stream search for research and content analysis of video fragments

N.A. Bondareva[1], A.E. Bondarev[2], S.V. Andreev[3], I.G. Ryzhova[4]

Keldysh Institute of Applied Mathematics RAS

[1] ORCID: 0000-0002-7586-903X, nicibond9991@gmail.com
[2] ORCID: 0000-0003-3681-5212, bond@keldysh.ru
[3] ORCID: 0000-0001-8029-1124, esa@keldysh.ru
[4] ORCID: 0000-0003-1613-3038, ryzhova@gin.keldysh.ru

Abstract

This paper examines the problem of targeted search for specific objects in a video stream on request and recording the timestamps of their appearance. Since the solutions currently available on the market were inadequate for the task, it was decided to implement such a tool independently as part of an ongoing research project conducted at the Keldysh Institute of Applied Mathematics of the Russian Academy of Sciences

Keywords: video stream, video content, search query, timestamps.

References

1. Moretti F. Distant Reading. - London: Verso, 2013. - 298 p.
2. Manovich L. Cultural Analytics: Analyzing Cultural Patterns in the Era of “More Media” // Domus, 2009. - No. 923. - P. 1-7.
3. Salt B. Film Style and Technology: History and Analysis. - London: Starword, 2009. - 398 p.
4. Adobe Premiere Pro - URL: https://www.adobe.com/products/premiere.html
5. Final Cut Pro - URL: https://www.apple.com/final-cut-pro/
6. Amazon Rekognition - URL: https://aws.amazon.com/rekognition/
7. Video Intelligence API documentation - URL: https://cloud.google.com/video-intelligence/docs
8. Adobe Systems Inc. Adobe Premiere Pro User Guide // Adobe Documentation. - 2023. - URL: https://helpx.adobe.com/premiere-pro/user-guide.html
9. Amazon Web Services. Amazon Rekognition Video Developer Guide. - 2023. - URL: https://docs.aws.amazon.com/rekognition/
10. Google Cloud. Video Intelligence API Documentation. - 2023. - URL: https://cloud.google.com/video-intelligence/docs
11. Wittenburg P., Brugman H., Russel A. ELAN: a Professional Framework for Multimodality Research // Proceedings of the 5th International Conference on Language Resources and Evaluation. - 2006. - P. 1556-1559.
12. Bradski G., Kaehler A. Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library. - Sebastopol: O'Reilly Media, 2016. - 1024 p.
13. VideoANT Project. Video Annotation Tool // University of Minnesota. - URL: https://ant.umn.edu/
14. Cinemetrics. Film Measurement and Analysis // Cinemetrics Database. - URL: http://www.cinemetrics.lv/
15. CLARIAH Media Suite. Research Platform for Media Studies // CLARIAH Consortium. - URL: https://mediasuite.clariah.nl/

16. Geitgey A. Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning // Medium. - 2016. - URL: https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

17. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). - 2016. - P. 770-778.

18. Schroff F., Kalenichenko D., Philbin J. FaceNet : A Unified Embedding for Face Recognition and Clustering // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). - 2015. - P. 815-823