

Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ  
ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»  
(САМАРСКИЙ УНИВЕРСИТЕТ)

Институт информатики и кибернетики

Кафедра технической кибернетики

ОТЧЁТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2  
«Применение n8n для реализации пайплайна обработки видео с использованием  
LLM моделей.»

по дисциплине  
«Инженерия данных»

Студент \_\_\_\_\_ А.Е. Федякин

Преподаватель \_\_\_\_\_ Р. А. Парингер

Самара 2025

Целью работы является реализация пайплайна, способного:

- Принимать видеоконтент через Telegram-бота в виде файла или ссылки (YouTube и др.);
- Извлекать аудиодорожку из видео;
- Генерировать англоязычные субтитры с использованием современных STT-моделей;
- Выполнять машинный перевод субтитров с английского на русский язык;
- Инкрустовать полученные субтитры в исходное видео;
- Возвращать пользователю готовое видео с субтитрами.

Решение реализовано в виде распределённой системы на основе Docker Compose, включающей следующие ключевые компоненты:

n8n – Workflow-оркестратор: обрабатывает Telegram-вебхуки, управляет последовательностью операций, вызывает внешние сервисы и скрипты.

Telegram Bot API – Интерфейс взаимодействия с пользователем.

yt-dlp + Deno – Скачивание видео по публичной ссылке с поддержкой YouTube и других платформ.

faster-whisper-server – Выделенный сервис STT на основе модели Whisper (архитектура base.en). Запущен в отдельном контейнере с HTTP API.

Hugging Face Inference API - Перевод субтитров с английского на русский с использованием модели Helsinki-NLP/opus-mt-en-ru.

ffmpeg – Извлечение аудио, конвертация форматов, наложение субтитров.

Вспомогательные Python-скрипты – Перевод .srt-файлов, обработка ошибок, взаимодействие с API.

Пайплайн обработки запроса выглядит следующим образом:

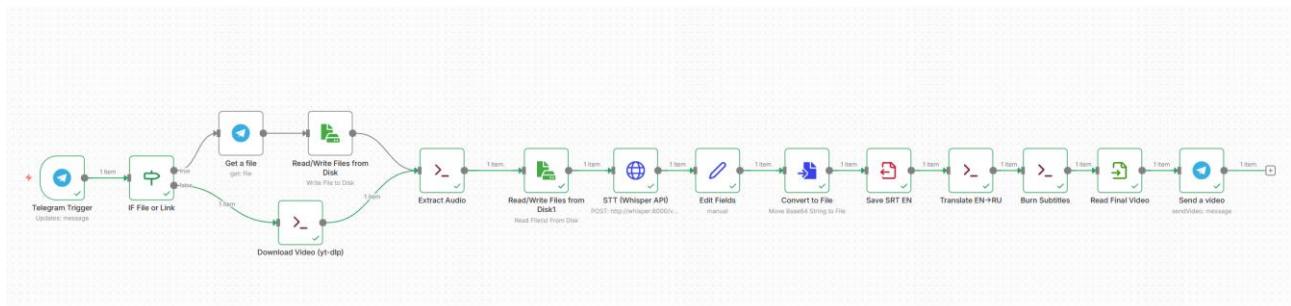


Рисунок 1 – Демонстрация workflow

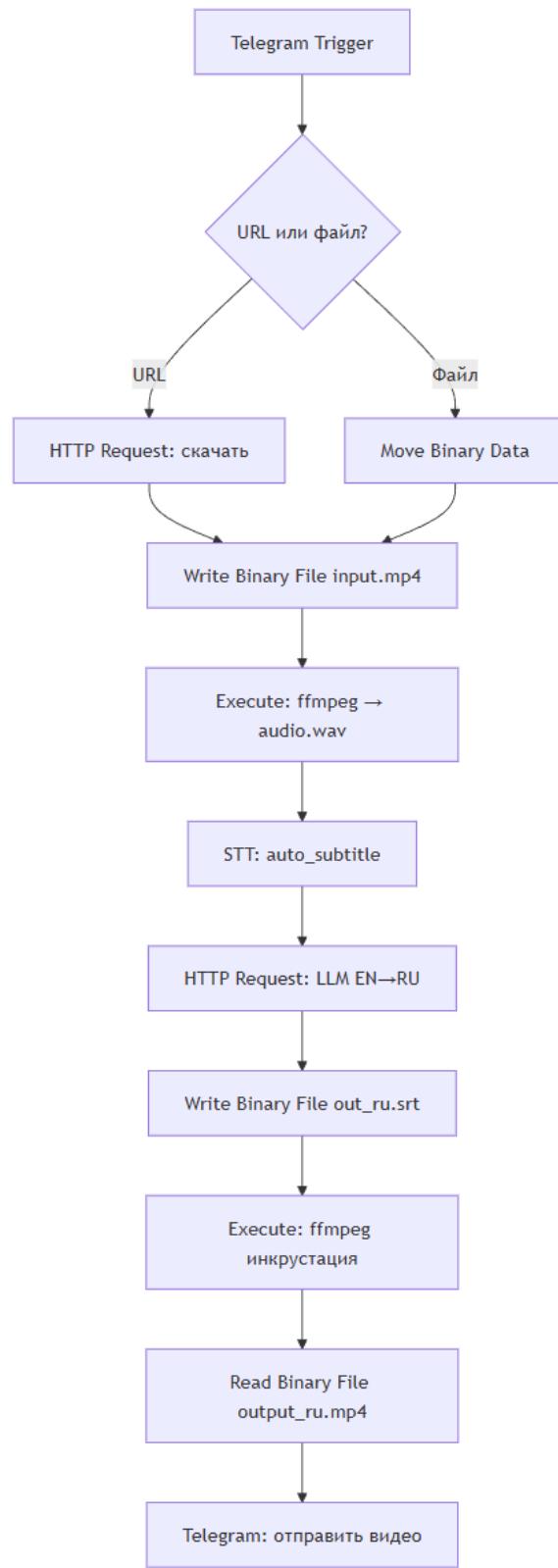


Рисунок 2 – Схема пайплайна

Он состоит из следующих этапов:

1. Приём входного видео через Telegram: Если прислан файл — сохраняется напрямую.

Если прислана ссылка — скачивается с помощью yt-dlp в формате 480p для ускорения и

уменьшения объёма. Данное событие было реализовано через Telegram Trigger в n8n. А для связи с Telegram использовался ngrok

2. Извлечение аудио с помощью ffmpeg.
3. Генерация субтитров: HTTP-запрос к локальному сервису faster-whisper-server с полученным аудиофайлом.
4. Перевод строкового формата в бинарный файл.
5. Перевод субтитров с английского на русский с помощью написанного python скрипта, где построчно происходят запросы к модели из Hugging Face Interface.
6. Инкрустация субтитров в видео с помощью ffmpeg.
7. Отправка полученного видео в Telegram.



Рисунок 3 – Демонстрация создания и инкрустации субтитров по файлу

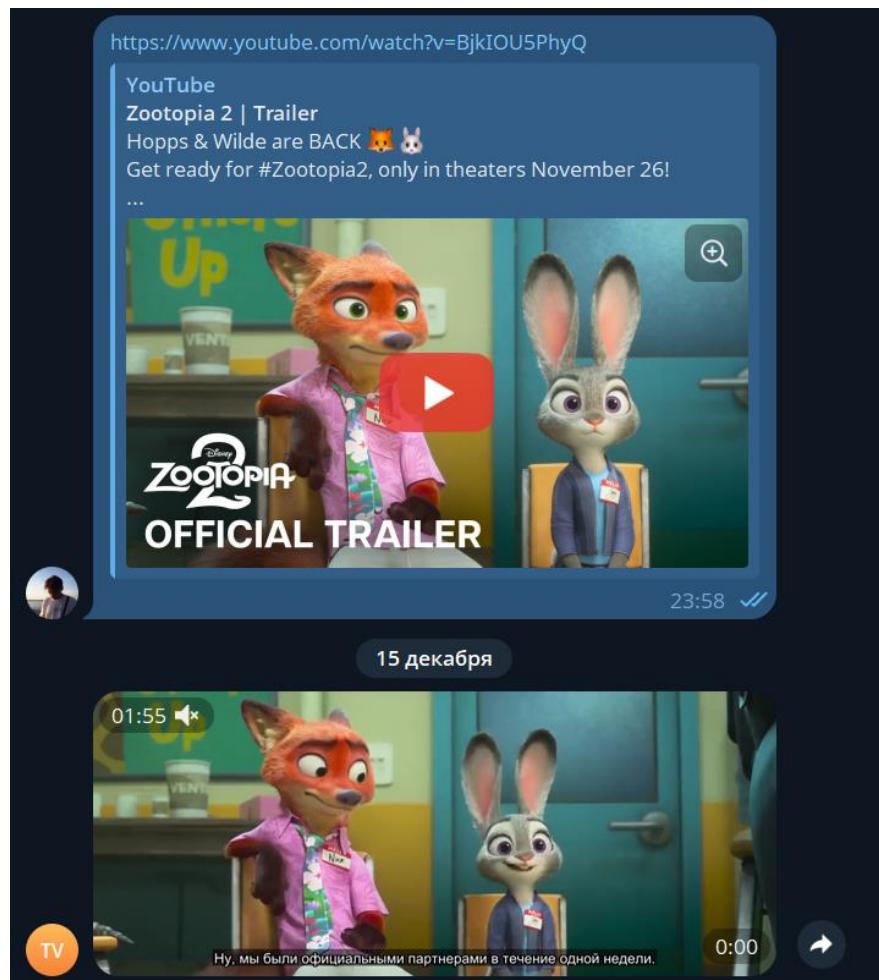


Рисунок 4 – Демонстрация создания и инкрустации субтитров по ссылке

Таким образом в лабораторной работе были освоены оркестратор n8n, библиотеки ffmpeg, yt-dlp, модель Whisper и улитка ngrok. Реализованный пайплайн полностью удовлетворяет поставленной задаче и демонстрирует корректное применение современных инструментов для обработки мультимедиа.

В данной лабораторной работе также были и сложности:

1. Мой webhook цеплялся к localhost, а не к ссылке ngrok. На решение было потрачено довольно много времени, а ошибка была в названии переменной окружения...

2. Настройка корректной работы yt-dlp.

3. Проблема со скриптом, где ошибка заключалась в отсутствии модуля requests.

Изначально я думал, что описание ошибки говорит о невозможности подключения к модели. После того как я снова потратил немало времени, всё перепроверил и обнаружил реальную ошибку, данный пакет очень долго не мог поставитьсь в докере. А затем еще было потрачено время из-за того, что я запускал свой скрипт не из виртуальной среды...