

Usually this environment is used to display code, so here's a simple C program.

```
#include <stdio.h>
#define NUMYAYS 10

int main()
{
    int i;
    for (i = 0; i ≤ NUMYAYS; i++)
    {
        printf("Yay! Overleaf is Awesome!\n");
    }
    return 0;
}
```

1 ВВЕДЕНИЕ

Видеоконтент на текущий момент активно применяется во множестве сфер нашей жизни. Большая часть трафика в интернете - это видео. Ежедневно мы смотрим различные фильмы, сериалы, шоу и, что особенно актуально в наше время, ролики в интернете - это могут быть обзоры на товары, клипы, видеоблоги и множество других форматов, представленных на YouTube, в Instagram, TikTok и других площадках. В большей части случаев создатели видеоконтента зарабатывают на рекламе.

Видео отличается от других форматов тем, что предоставляет большую связь с происходящим на нем, так как сочетает как аудио, так и визуальную коммуникацию. Таким образом человек невольно строит более сильные ассоциации с героями роликов или фильмов, а в рекламе конкретный бренд может ассоциироваться с человеком, показываемым на экране.

Но использование людей для съёмок становится большим риском для компаний, постоянно производящих видеоконтент. Например, используя человека как лицо своего бренда компания имеет много рисков: постоянно возрастающая стоимость, опоздания, отказ от сотрудничества и другие возможные проявления человеческого фактора.

Этих проблем при производстве видео-контента можно было бы избежать, если не привязываться к конкретному человеку, а работать с неким виртуальным персонажем. Достичь этого позволяют, например, технологии замены лица.

2 МАРКЕТИНГОВОЕ ОПИСАНИЕ

Malivar - сервис для замены лица на видео для бизнеса. Наш сервис позволяет с помощью одного снимка сделать замену лица на видео. Пользователь выбирает лицо и загружает видео, а на выходе получает видео уже с выбранным лицом.

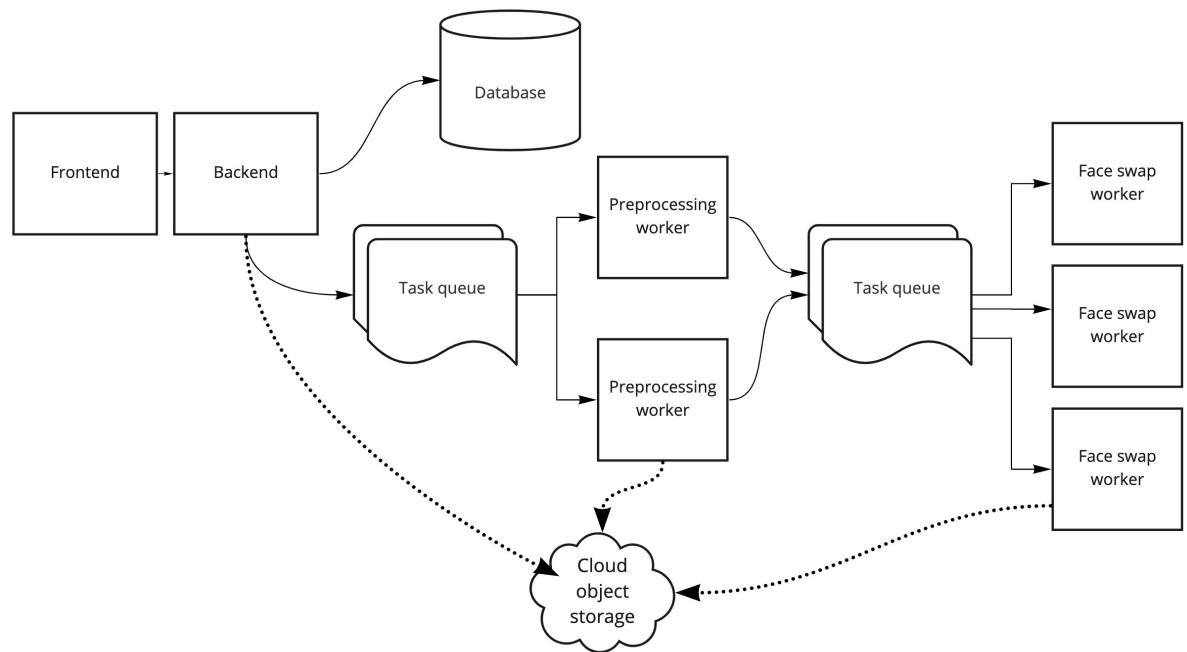
3 ТЕХНИЧЕСКОЕ ОПИСАНИЕ

Основа сервиса - ансамбль нейронных сетей, комбинирующий сети для сегментации лица, замены лица (identity injection) и сеть для повышения качества изображения (super resolution).

Очевидно, что такой публичный сервис должен иметь возможность масштабирования в будущем, обработки одновременно большого количества видео для разных клиентов.

Обработка происходит на GPU (видеокарте) и занимает продолжительное время, например обработка 1 минуты видео занимает в среднем около 10 минут.

Исходя из этих ограничений архитектура не может быть монолитной. Сервис состоит из множества компонентов, каждый из которых можно легко масштабировать в соответствии с нагрузкой и ожидаемым количеством пользователей:



miro

- Фронтенд - клиентская часть приложения со своим сервером для серверного рендеринга
- Бэкенд - основная серверная часть, ответственная за обработку запросов пользователя и соединение остальных компонентов
- База данных
- Брокер сообщений - для реализации очередей обработки и других асинхронных и параллельных процессов
- Сервис(ы) препроцессинга - берут задачу по обработки видео из очереди и осуществляют предварительную и пост-обработку, например перекодирование видео
- Сервис(ы) обработки - достают задачи из очереди и проделывают основную работу по обработке видео

Так, ключевая ценность проекта содержится в нейронной сети, работающей внутри сервиса обработки. Но для работы сервиса необходима разработка множества других компонентов и вспомогательных сервисов. Я занимался разработкой каждой части системы, кроме самого ядра обработки.