

Санкт–Петербургский государственный университет

Плешканов Никита Вячеславович

Выпускная квалификационная работа
*Распознавание внедренных субтитров в видео
потоке*

Уровень образования: бакалавриат

Направление 02.03.02 «Фундаментальная информатика и
информационные технологии»

Основная образовательная программа «Программирование и
информационные технологии»

Научный руководитель:

Кандидат технических наук, доцент кафедры
компьютерного моделирования и многопроцессорных систем

Гришкин Валерий Михайлович

Рецензент:

Доктор физико-математических наук,
профессор кафедры информационных систем, заведующий
кафедрой механики управляемого движения

Матросов Александр Васильевич

Санкт-Петербург

2020 г.

Содержание

Введение	3
Постановка задачи	5
Глава 1. Обзор литературы	7
1.1. Обзор существующих методов	7
Глава 2. Методы сегментации текста на изображениях	8
2.1. Работа с видеопотоком	8
2.2. Метод MSER	8
2.3. Метод Otsu	9
2.4. Метод связанных компонент	10
2.5. Метод семантической сегментации с использованием нейронных сетей	10
Глава 3. Система распознавания внедренных субтитров	13
3.1. Выбор метода сегментации	13
3.2. Локализация текста на изображении	14
3.3. Распознавание текста изображения с помощью метода OCR	15
Глава 4. Реализация предложенного алгоритма	17
4.1. Структура системы	17
4.2. Работа программы	19
4.3. Результаты	22
Заключение	25
Список литературы	26

Введение

Человек может черпать информацию из различных источников, таких как интернет, книги, общение с другими людьми, телевидение и т.д. Значительную часть человек может получать из различных видео файлов, начиная от фильмов (исторических, документальных, художественных) и заканчивая видеозаписями лекций. Последний из названных способов один из наиболее удобных, так как позволяет получать информацию удаленно. Однако не каждый пользователь способен воспринимать информацию из видеопотока ввиду различных причин. В таких случаях субтитры играют важную роль. Что такое субтитры и насколько они важны для человека?

Субтитры - это сопровождение видеоряда в виде текста на языке оригинала или в переводе. Важно различать субтитры и интертитры, ведь последние выполнялись в виде отдельных монтажных кадров и использовались чаще всего в немом кино, а субтитры накладываются поверх основного изображения, что усложняет их распознавание. В настоящее время субтитры играют огромную роль для людей, которые не владеют языком, используемым в видеопотоке и не могут воспринимать полноценно передаваемую информацию. Также субтитры являются необходимостью для людей с частичным или полным нарушением слуха. Субтитры содержат в себе важную информацию, переводя в текстовый формат происходящее в видеопотоке, например реплики персонажей, надписи в кадре, какие-либо звуковые эффекты, а иногда и дополняя видео комментариями или формулами. Также субтитры используются в случаях, когда приходится расшифровывать видеозапись с звуковыми дефектами или неразборчивой речью говорящего. Значительная часть научных докладов не всегда нуждается в сопровождении видеоряда и достаточно лишь изучить прилагаемые к видеопотоку субтитры. Однако субтитры не всегда удобно воспринимать ввиду их большого количества в кадре или их быстрой смене. Таким образом процесс извлечения информации из видеопотока усложняется и становится более долгим и трудоемким.

Учитывая, что в последние годы значимость автоматизации процессов выросла и имеет достаточно удачные результаты, решение данной про-

блемы становится более актуальным.

Распознавая внедренные в видеопоток субтитры, появляется возможность предоставить зрителю отдельно от видеопотока работать с текстом, проводить его анализ. Несмотря на то, что существуют эффективные методы отделения текста от различных документов и статей, сегментация и распознавание текста в видеопотоке проблема актуальная и нерешенная. Достижение цели усложняется тем, что в видеопотоке присутствует постоянно сменяющаяся неоднородная фоновая структура, надписи, не являющиеся частью субтитров. Также фактором, осложняющим решение задачи, является непосредственно работа с видеопотоком, а именно определение, есть ли в кадре субтитры, не совпадают ли полученные субтитры с ранее обработанными, дублируя извлеченную информацию. Задача становится актуальнее ввиду огромного количества информации, архивированной человеком.

Постановка задачи

Задача распознавания текста была уже решена с помощью множества методов, имеющих различные преимущества и недостатки в зависимости от входных данных, однако приоритетом является реализация алгоритма распознавания текста в видеопотоке с учетом, что текст является субтитрами, то есть имеет размер, меняющийся в течение всего видео в определенном диапазоне, определенный наклон и цвет шрифта. Для решения поставленной задачи распознавания внедренных в видеопоток субтитров необходимо решение следующих подзадач:

1. Определить метод работы с видеопотоком, содержащим внедренные субтитры.

Данный этап играет важную роль в работе программы, так как на этом этапе планируется отделять кадры от видеопотока и передавать их на обработку методам сегментации и распознавания текста. Для того, чтобы избежать обработки кадров, содержащих одинаковые субтитры, будет реализован подход, по которому лишь часть субтитров из видеопотока будет далее использоваться в работе программы. В данном походе будет учитываться такой параметр видео как fps (количество кадров в секунду).

2. Провести анализ существующих методов сегментации текста на изображениях.
3. Выбрать методы обработки изображений перед сегментацией.

На данном шаге производится фильтрация шума на изображении в предположении, что шум является гауссовым, с помощью алгоритма нелокальных средних [1]. Также будет применена бинаризация — преобразование изображения из цветного или полутонового в черно-белое изображение. Это является необходимым, так как упрощает процесс распознавания текста [2].

4. Реализовать наиболее подходящие для сегментации субтитров методы.

5. Провести распознавание локализованных субтитров с помощью OCR.
6. Провести экспериментальный анализ реализованной системы распознавания внедренных в видеопоток субтитров.

Глава 1. Обзор литературы

В процессе написания данной работы было использовано большое количество различных публикаций.

Первым делом хотелось бы отметить источники, позволившие детально рассмотреть методы сегментации. А именно [5] и [6] для метода Otsu, его основных преимуществ и недостатков. А также [4] и [3] для метода MSER. Для описания работы метода связных компонент были использованы источники [7], [8], [9].

Стоит отдельно упомянуть про статью [10], в которой оценивается применение сверточных нейронных сетей для сегментации изображения.

Также важный вклад внесли источники [2], [1], при помощи которых была получена информация об обработке изображений, для наиболее успешной работы алгоритмов сегментации и распознавания текста.

Для подробного рассмотрения двухпроходного подхода к распознаванию текста на изображении с помощью метода OCR была использована статья [12], детально раскрывающая принципы работы второго прохода.

1.1 Обзор существующих методов

В процессе решения поставленных задач было просмотрено большое количество различных источников.

Так, например, в статье [13] предложен метод, основанный на том свойстве, что цвет и границы текста остаются неизменными в течение нескольких кадров. Однако точность данного метода колеблется от 40% до 60%. Были найдены и другие методы с более высокой точностью, однако они ресурсозатратны и требуют большого количества времени для выполнения. Поэтому было решено реализовать собственный метод распознавания субтитров.

Глава 2. Методы сегментации текста на изображениях

В данной главе работы будут описаны рассмотренные методы сегментации изображений, их теоретическое обоснование, а также предобработка видеопотока изображений, для дальнейшей сегментации и локализации текста.

2.1 Работа с видеопотоком

На данном этапе происходит обработка полученного видеопотока. Так как субтитры предназначены для осмысленного восприятия их человеком, то целесообразно считать, что субтитры не меняются в видеопотоке как минимум одну секунду. Таким образом, для увеличения скорости работы программы и избавления от ненужных вычислений, из видеопотока для дальнейшей обработки будет передаваться каждый N -кадр,

$$N = fps,$$

где fps — количество кадров в секунду.

2.2 Метод MSER

Рассмотрим метод MSER (Maximally Stable Extremal Regions). Суть данного метода в следующем: к изображению применяется бинаризация по порогу со значением от 0 до 255. Таким образом, в процессе бинаризации на изображении будут появляться белые области. Белые области, не меняющиеся по крайней мере K раз при пороге равном K , считаются максимально устойчивыми областями экстремума. Преимущества алгоритма:

1. По сравнению с другими методами MSER находит большее количество областей. [3]
2. Результаты метода не зависят от языка текста.
3. Сложность данного алгоритма в худшем случае равна $O(n)$ [4].

4. Так как в процессе MSER не используется сглаживание, то обнаруживаются структурные элементы разных масштабов [4].

Однако у данного метода есть и свои недостатки:

1. Высокая чувствительность алгоритма к размытым изображениям [3].
2. Алгоритм зачастую обнаруживает большое количество областей, не содержащих символов, из-за чего в последствие необходимы дополнительные действия по исключению ненужных регионов.

2.3 Метод Otsu

Рассмотрим альтернативный метод сегментации, а именно метод Otsu. Он находит порог, понижающий внутриклассовую дисперсию. Дисперсия в данном случае задается как сумма дисперсий двух классов [5]:

$$\sigma_w^2 = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t),$$

где веса ω_i — это вероятности двух классов, которые разделены порогом t , σ_w^2 — дисперсия этих классов. Также Оцу продемонстрировал, что максимизация дисперсии между классами равноценна минимизации дисперсии внутри класса:

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)],$$

выражающаяся в терминах вероятности ω_i и среднего арифметического класса μ_i , которое может обновляться итеративно.

Однако у метода Оцу так же есть свои недостатки, а именно:

1. Если область объекта мала по сравнению с областью фона, то в таком случае гистограмма не проявляет бимодальности и объекты могут быть утеряны [6].
2. Метод может приводить к «слипанию» символов, находящихся близко к друг другу.

Стоит отметить, что в субтитрах между элементами (буквами) соблюдается определенное расстояние для того, чтобы повысить читабельность текста. Также размер субтитров обычно достаточно велик для их комфортного восприятия. Таким образом недостатки метода Оцу нивелируются.

2.4 Метод связных компонент

Также считаю необходимым рассмотреть в этой главе метод связных компонент, который в последующем сыграет важную роль в работе предложенного алгоритма. Метод связных компонент представляет собой процесс сканирования изображений с целью выявления областей связных пикселей, которые имеют одинаковое свойство [7]. Существуют несколько разных подходов к определению связных компонент.

1. Одна компонента за раз.

Быстрый и простой для реализации метод, основанный на поиске в глубину из теории графов. Суть этого метода заключается в том, что как только найден первый пиксель компоненты связности, так все остальные пиксели данной компоненты связности помечаются перед переходом к другому пикселю данного изображения [8].

2. Двухпроходный подход [9].

Данный алгоритм выполняет два прохода по изображению. Первый проход необходим для проставления временных меток и эквивалентных записей, а на втором проходе каждая временная метка заменяется на наименьшую метку в своем классе эквивалентности.

2.5 Метод семантической сегментации с использованием нейронных сетей

В настоящее время нейронные сети демонстрируют внушительные результаты в различных областях, в том числе и в области обработки и анализа изображений. В рамках задачи сегментации изображения хорошие результаты показала полная сверточная сеть [10].

Полные сверточные сети позволяют работать с изображениями произвольного размера. Нейронная сеть на вход получает вектор значений, который в дальнейшем преобразуется с помощью слоев нейронной сети. Так как применение операции свертки к изображению большого размера достаточно затратная операция, то часть слоев нейронной сети уменьшает размерность изображения.

Сверточная сеть состоит из ряда основных слоев: слой свертки, слой активации, слой субдискретизации, полносвязный слой.

1. Слой свертки.

На каждом канале данного слоя присутствует фильтр, в рамках которого полученные из предыдущего слоя данные обрабатываются матрицей весов, которая также называется ядром свертки. Коэффициенты данной матрицы устанавливаются в процессе обучения.

2. Слой активации.

Данный слой обычно применяется после слоя свертки. Ранее часто использовались сигмоидальные или гиперболические функции активации, однако в последние годы наиболее эффективной считается функция ReLU, позволяющая ускорить и упростить процесс обучения [11].

3. Слой субдискретизации.

Данный слой представляет собой слой пуллинга (подвыборки). На этом шаге матрица пикселей уплотняется (уменьшается) в размерах. Наиболее частым подходом является выбор пикселя, значение которого максимальное, из группы пикселей размером 2×2 . Это позволяет значительно уменьшить размер изображения.

4. Слой полносвязной нейронной сети.

Данный слой содержит полноценную нейронную сеть, которая состоит из нескольких слоев. После сокращения на предыдущих слоях количества входных данных, данный слой определяет вероятность принадлежности полученных данных к определенному классу.

Сегментация изображений с помощью полных сверточных сетей имеет достаточно точные результаты, однако структура сети может иметь слишком большое количество параметров, которые необходимо настраивать под конкретные задачи. Такими параметрами могут быть количество слоев, количество ядер у каждого слоя, степень уменьшения размерности изображения.

Несмотря на то, что нейронные сети показывают хорошие результаты в области сегментации изображения, требуется большое количество времени для их обучения. Также одним из основных недостатков нейронных сетей является их высокая ресурсоемкость.

Еще одним немаловажным недостатком нейронных сетей является необходимость в специфических обучающих датасетах. Датасеты необходимы для наиболее успешного обучения сети, чтобы учесть различные возможные ситуации. На данный момент датасеты, приспособленные для субтитров в видеопотоке, не известны, следовательно их необходимо формировать самостоятельно, учитывая различные параметры как видеопотока, так и самих субтитров. Также, если не использовать специальный графический процессор (GPU), который быстрее справляется с различными расчетам, значительно понижается быстродействие нейронных сетей.

Глава 3. Система распознавания внедренных субтитров

В данной главе описано какой из методов сегментации был выбран и почему. Также описан алгоритм локализации текста и его распознавания.

3.1 Выбор метода сегментации

Наилучшие результаты среди методов сегментации показали методы MSER и Otsu, принцип работы которых был описан в предыдущей главе. Пример успешной работы обоих алгоритмов:



Рис. 1: Исходное изображение.



Рис. 2: Изображение, полученное после сегментации методом MSER.

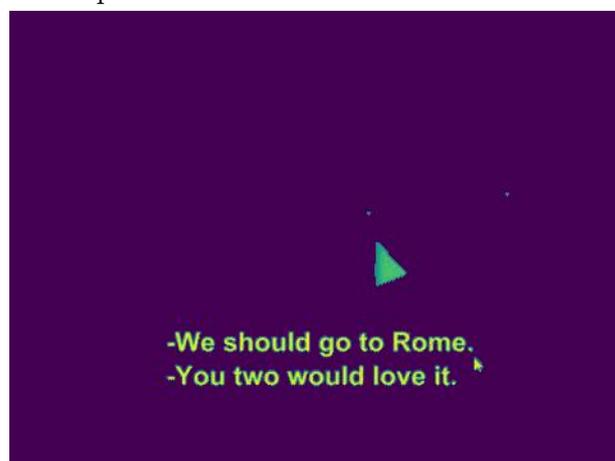


Рис. 3: Изображение, полученное после сегментации методом Otsu.

Однако высокая чувствительность алгоритма MSER к размытым изображениям, а также выделение большого количества областей, не принадлежащих к субтитрам, в случаях, когда метод Otsu показывал успешные

результаты, привела к тому, что метод Otsu был выбран в качестве основного метода сегментации изображения.

Ниже приведен один из вышеупомянутых случаев:



Рис. 4: Исходное изображение.



Рис. 5: Изображение, полученное после сегментации методом MSER.



Рис. 6: Изображение, полученное после сегментации методом Otsu.

3.2 Локализация текста на изображении

Так как методы сегментации зачастую обнаруживают области, не принадлежащие областям субтитров, то необходимо дополнительно обработать полученные после сегментации изображения. На этом этапе применяется метод связных компонент. Результатом применения этого метода является полутонное изображение, разделенное на области, у каждой из которых есть определенный список параметров, а именно:

1. Координаты прямоугольника, ограничивающего данную область.
2. Центр области.
3. Площадь области.

Далее происходит фильтрация областей по полученным параметрам. Первым параметром является площадь прямоугольника, ограничивающего полученную область. Субтитры обладают размером, который лежит в определенном диапазоне, предполагающий комфортное восприятие текста

зрителем. Таким образом можно отфильтровать области, лежащие вне этого диапазона.

Принимая во внимание тот факт, что символы в строке субтитров находятся примерно на одной и той же высоте, необходимо учитывать центры полученных областей. Сделать это можно, разделив набор областей на классы. Данные классы содержат в себе области, разница центров которых относительно мала. Если количество элементов в классе менее некоторого заданного числа (например 3), то в таком случае данный класс считается не принадлежащим субтитрам и области далее не рассматриваются.

Далее учитывается горизонтальная координата оставшихся областей. Если для элементов, содержащихся в одном классе, существует определенная периодичность в расстояниях между ними, то элементы считаются частью субтитров. Если же разброс в расстояниях между элементами большой, то данные элементы далее не рассматриваются.

Оставшиеся после фильтрации области считаются частью субтитров и передаются на распознавание методу OCR.

3.3 Распознавание текста изображения с помощью метода OCR

Рассмотрим также алгоритм OCR. В рамках этого алгоритма машинописанный или печатный текст переводится в текстовые данные посредством электронного или механического распознавания. В данной работе реализация метода OCR будет взята из пакета PyTesseract языка python. Этот пакет использует двухпроходный подход для распознавания символов.

На первом проходе извлеченные из изображения объекты разбиваются на несколько «подобъектов», которые могут являться линиями, замкнутыми линиями, пересечениями линий, направлениями линий. Извлеченные черты сопоставляются характерным чертам для символа из словаря. На втором проходе данного метода используются символы, которые с высокой достоверностью были распознаны на первом проходе. Они учитываются для того, чтобы на втором проходе более точно распознать оставшиеся сим-

волы. Такой алгоритм полезен в случаях, когда используется необычный шрифт, изображение имеет низкое качество или текст искажен [12].

Предложенный библиотекой PyTesseract метод *image_to_string* мультиязычен. Необходимо лишь в параметрах данного метода указать желаемый для распознавания язык текста. По умолчанию стоит параметр «eng», который можно легко заменить на «rus» или другие языки при необходимости.

Глава 4. Реализация предложенного алгоритма

В данной главе будет представлена структура программы, пример ее работы, а так же таблица с результатами и точностью алгоритма.

4.1 Структура системы

В этом разделе будет кратко описана структура программы, а именно из каких блоков она состоит и для чего они нужны. Ссылка на код программы [14] представлена в списке литературы данной работы.

Программа на вход принимает строку, являющуюся расположением видеофайла. Данная строка передается на вход блока "Обработка видеопотока".

1. Обработка видеопотока.

Данный блок получает на вход строку с расположением видеофайла и обрабатывает ее с помощью библиотеки OpenCV и функции VideoCapture. Следующим шагом вычисляются важные для дальнейшей работы программы параметры видеопотока. Такими параметрами являются длительность видео и количество кадров в секунду.

2. Основной цикл.

В рамках данного блока полученный видеопоток последовательно, с шагом i равным fps видео, разбивается на кадры, которые в дальнейшем передаются на вход методам обработки изображений, сегментации, распознавания текста. Разбиение происходит пока выполняется условие:

$$i \leq N,$$

где N — количество кадров в видеопотке. В противном случае алгоритм завершает свою работу.

Сперва полученное изображение преобразуется в градации серого, посредством метода `cvtColor`. В дальнейшем к изображению применяется фильтр Гаусса, таким образом уменьшая количество шума.

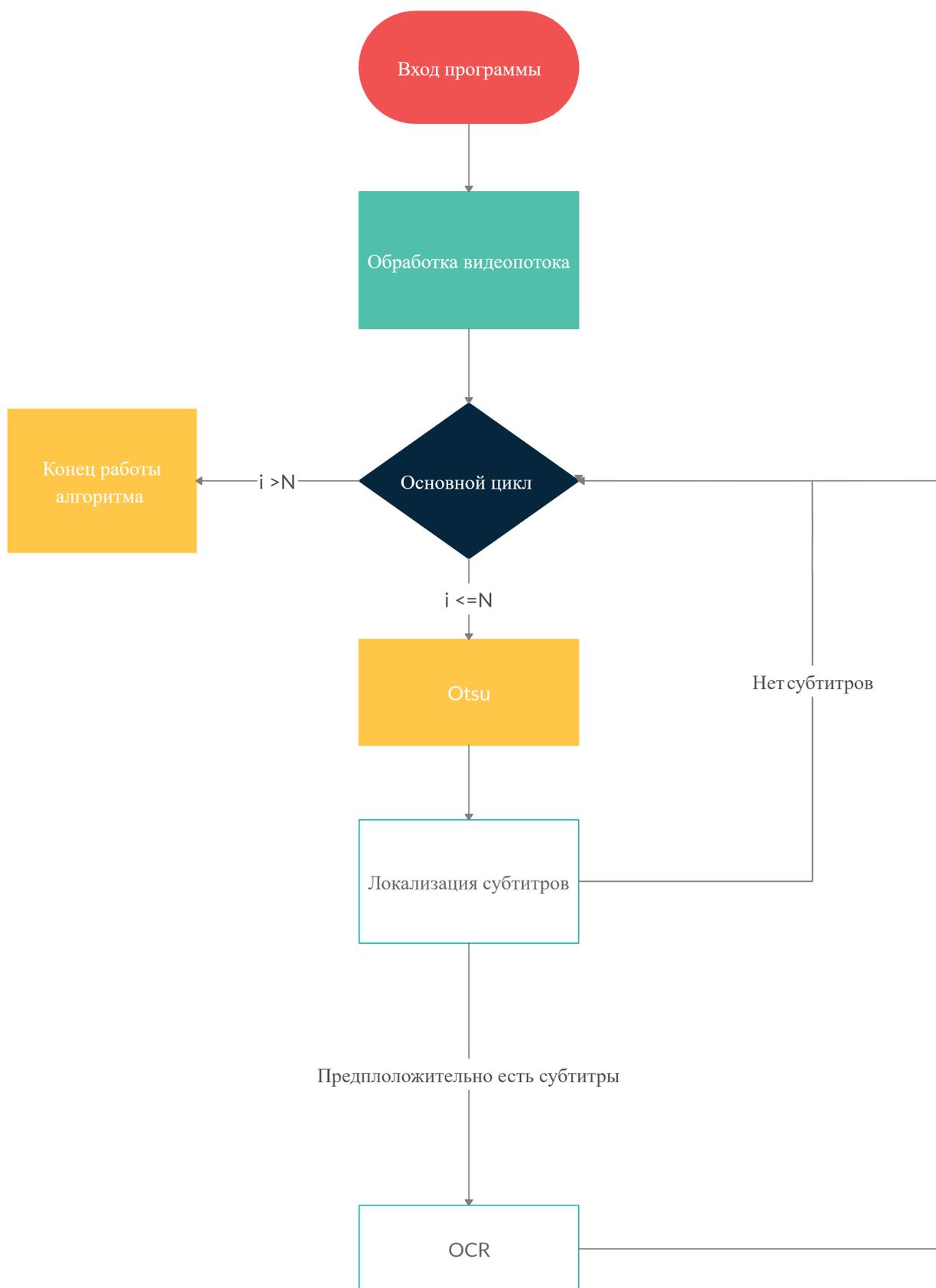


Рис. 7: Структура системы.

3. Блок Otsu.

Данный блок получает на вход изображение после его перевода в градации серого и уменьшении количества шума и применяет выбранный ранее метод сегментации, а именно методом Otsu.

4. Локализация субтитров.

На данном шаге происходит разделение изображения на связные компоненты и последующая локализация текста, принципы которой описаны в разделе 3.2. Если после локализации не осталось областей, возможно являющихся субтитрами, то в дальнейшем изображение не обрабатывается и происходит обработка следующего кадра.

5. Блок OCR.

В этом блоке был реализован метод OCR при помощи библиотеки PyTesseract. На данном этапе оставшиеся после сегментации и локализации текста области изображения подвергаются анализу методом OCR, который выводит в консоль результаты распознавания текста.

Выводом программы является совокупность распознанных методом OCR символов. После обработки методом OCR предоставленных областей происходит переход к следующему кадру.

4.2 Работа программы

В данном разделе будет приведен пример работы программы, отображающий результат после каждого шага, описанного в разделе 4.1.

Итак, после того, как видеопоток был подан на вход программы и были определены fps и длительность видеопотока, происходит разделение видеопотока с определенным шагом на отдельные кадры и передача этих кадров для последующей обработки и анализа.

На Рис. 8 представлено одно из таких изображений.



Рис. 8: Исходное изображение.

Далее изображение переводится в оттенки серого (модель *grayscale*). На Рис. 9 представлен результат перевода исходного изображения.



Рис. 9: Изображение в модели *grayscale*.

Следом на изображении посредством Гауссовского фильтра уменьшается количество шумов.



Рис. 10: Изображение, после применения фильтра Гаусса.

В дальнейшем изображение подвергается обработке выбранным методом сегментации, а именно Otsu. Результаты его работы представлены на Рис. 11.



Рис. 11: Изображение, после сегментации методом Otsu.

Как и было отмечено ранее, на изображении выделяется большое количество областей, не принадлежащих субтитрам, поэтому следующим шагом является локализация текста с помощью метода связанных компонент. Результат его работы представлен на Рис. 12.

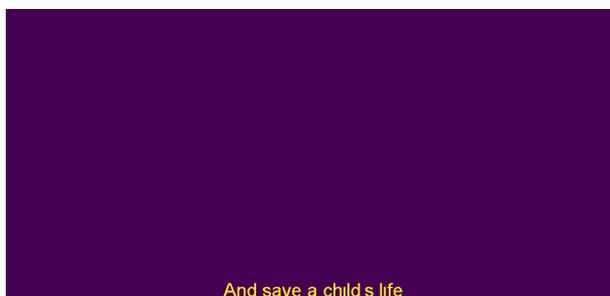


Рис. 12: Изображение, после алгоритма локализации.

Таким образом, из изображения были удалены некоторые области, не являющиеся субтитрами. Далее изображение снова переводится в оттенки серого (модель grayscale) и передается на вход методу OCR.

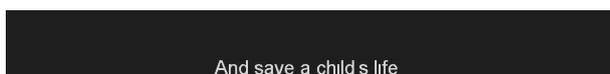


Рис. 13: Изображение, готовое для обработки методом OCR.

Результатом работы OCR в данном случае является «And save a childs life»

4.3 Результаты

Работа программы была протестирована на различных типах видеопотока:

1. Видеопоток с субтитрами, имеющими приблизительно одну и ту же область расположения и размер.
2. Видеопоток с субтитрами переменного размера.
3. Видеопоток с субтитрами переменного расположения.

Так как в процессе реализации предложенного алгоритма не было найдено готового датасета с перечисленными типами, видео для тестирования алгоритма было подобрано вручную.

Для оценки точности локализации были использованы следующие понятия и критерии:

1. Полная точность локализации.

В данном случае области, выделенные алгоритмом локализации, включают в себя только субтитры. Более того учитываются все субтитры находящиеся на изображении.

2. Хорошая точность.

Точность работы алгоритма локализации будет считаться хорошей в случае, если были обнаружены все субтитры, а также в области входят некоторые элементы, не являющиеся субтитрами.

3. Средняя точность.

После локализации на изображении выделены субтитры, а также большое количество элементов, не принадлежащих субтитрам (среди всех выявленных областей как минимум 50% не является субтитрами).

4. Плохая точность.

Локализована лишь часть субтитров. Оставшаяся часть субтитров не входит в области, которые будут переданы на вход методу OCR и таким образом будет утеряна часть информации.

Похожим образом введены критерии для оценки точности работы алгоритма распознавания текста. Точность распознавания текста в процентах будем считать как:

$$acc = \frac{\sum_{i=1}^N \frac{k_i}{C_i}}{N} \cdot 100\%,$$

где acc — точность распознавания метода OCR в процентах, i — номер изображения, поданного на обработку, N — общее количество изображений, поданных на обработку методу OCR, k_i — количество правильно распознанных текстовых символов методом OCR на i -м изображении, C_i — общее количество текстовых символов на i -м изображении.

1. Полная точность распознавания текста.

В данном случае области алгоритм правильно распознал большую часть предоставленного ему текст на изображении. Точность распознавания более 90% текста.

2. Хорошая точность распознавания текста.

Результат алгоритма распознавания текста будет считаться хорошим, если распознано правильно от 70% до 90% текста на переданном изображении.

3. Средняя точность распознавания текста.

Данная оценка будет применена, если точность распознавания находится в пределах от 50% до 70%.

4. Плохая точность распознавания текста.

Если точность распознавания текстовых символов на изображении менее 50%, то в таком случае результат работы алгоритма считается плохим.

Результат работы алгоритма с различными типами видеопотоков представлен в Таблице 1.

Таблица 1: Результаты применения алгоритма к различным видеопотокам

Тип видеозаписи	Точность локализации	Точность распознавания
С субтитрами неизменного рамера	хорошая	хорошая
С субтитрами с переменным положением	хорошая	полная
С субтитрами переменного рамера	средняя	хорошая

Для большинства типов видеопотока результаты работы алгоритма имеет хорошие показатели точности (более 70%). Это обусловлено тем, что построенный алгоритм не зависит от изменения размера и расположения субтитров. Точность работы алгоритма снижалась, когда на кадре присутствовало большое количество однообразных элементов, расположенных на одной высоте и имеющих одинаковый размер. В таких случаях алгоритмы сегментации и локализации ошибочно определяли данные области, как области возможно являющиеся субтитрами.

Заключение

В рамках данной работы были выполнены все поставленные задачи. Был определен метод работы с видеопотоком, позволяющий сократить количество обрабатываемых изображений. Также проведен анализ существующих методов сегментации, из которых впоследствии был выбран и реализован один. В процессе реализации алгоритма было обнаружено, что необходим дополнительный шаг с локализацией субтитров для наиболее точных результатов работы программы. Данный шаг так же был реализован с помощью метода связанных компонент.

Также в ходе работы экспериментальным путем было обнаружено, что используя различные алгоритмы предобработки изображения, можно увеличить точность методов сегментации и локализации. Поэтому были использованы методы уменьшения шума на изображении и бинаризация.

Завершающим шагом работы программы является распознавание текста на изображениях, которые ранее были обработаны вышеупомянутыми методами. Построенный алгоритм был протестирован на нескольких видеозаписях с различными условиями и была построена таблица точности. В дальнейшем планируется модифицировать построенный алгоритм для улучшения его результатов.

Список литературы

- [1] Buades A., Coll B., Morel J. M. «Non-Local Means Denoising». 2011 Image Processing On Line, P. 208–212.
- [2] Gupta M., Jacobson N., Garcia E. «OCR binarization and image pre-processing for searching historical documents». 2007 Pattern Recognition. 40 (2): P. 389.
- [3] Forssen P.E., Lowe D.G. «Shape Descriptors for Maximally Stable Extremal Regions». 2011
- [4] Nister D., Stewenius H. «Linear Time Maximally Stable Extremal Regions». 2008
- [5] Otsu N. «A threshold selection method from gray-level histograms». 1979 IEEE Trans. Sys. Man. Cyber. 9 (1): PP. 62–66.
- [6] Kittler J., Illingworth J. «On threshold selection using clustering criteria». 1985 IEEE Transactions on Systems, Man and Cybernetics. SMC-15 (5): PP. 652–655
- [7] Paralic M. «Fast connected component labeling in binary images». 2012 35th International Conference on Telecommunications and Signal Processing (TSP) PP. 706-707
- [8] Vincent L., Soille P. «Watersheds in digital spaces: an efficient algorithm based on immersion simulations». 1991 IEEE Transactions on Pattern Analysis and Machine Intelligence P. 583
- [9] Shapiro L., Stockman G. «Computer Vision». 2002. Prentice Hall. PP. 69–73.
- [10] Long J., Shelhamer E., Darrell T. «Fully Convolutional Networks for Semantic Segmentation». 2016. IEEE Transactions on Pattern Analysis and Machine Intelligence , Volume 39 , Issue 4, PP. 640 - 651

- [11] Nair V., Hinton G.E. «Rectified linear units improve restricted boltzmann machines». 2010 ICML'10: Proceedings of the 27th International Conference on International Conference on Machine Learning, PP. 807–814
- [12] Smith R. «An Overview of the Tesseract OCR Engine». 2013. Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), PP. 629-633
- [13] Kim D., Sohn K. «Static text region detection in video sequences using color and orientation consistencies». 2008. 19th International Conference on Pattern Recognition, PP. 1-4
- [14] Ссылка на исходный код разработанной программы URL:<https://github.com/nikitapleshkanov/diploma>.