

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ» (НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет **ФИЗИЧЕСКИЙ**

Кафедра **автоматизации физико-технических исследований**

Направление подготовки **03.03.02 ФИЗИКА**

Образовательная программа: **БАКАЛАВРИАТ**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Ванданов Сергей Александрович

Тема работы **Исследование применимости и оптимизация распространенных алгоритмов выделения актера для бытовых систем видеоконференций**

«К защите допущена»

И. о. зав. кафедрой

Научный руководитель

к.т.н.

вед. инженер, ИАиЭ СО РАН

Лысаков К.Ф./.....
(фамилия И., О.) / (подпись, МП)

Таранцев И.Г./.....
(фамилия И., О.) / (подпись, МП)

«.....».....20...г.

«.....».....20...г.

Дата защиты: «.....».....20...г.

Новосибирск, 2022

Оглавление

Оглавление	1
Введение	3
1. Обзор существующих распространенных методов и алгоритмов выделения актера	5
1.1. Алгоритмы, основанные на методах вычитания исходного фона.	7
1.2. Алгоритмы, основанные на использовании сверточной нейронной сети.	12
2. Детальное тестирование выбранных алгоритмов	17
2.1 Изучение условий, возникающих при проведении онлайн-трансляций.	17
2.1.1 Изучение шумовой составляющей веб-камер.	17
2.1.1.1 Временное распределение шумов в зависимости от их пространственного расположения.	18
2.1.1.2 Пространственное распределение шумов.	20
2.1.2 Изучение других особенностей веб-камер.	24
2.1.3 Изучение характерных фоновых изображений.	25
2.2 Разработка системы тестирования алгоритмов выделения актера с эмуляцией бытовых условий.	26
2.2.1 Получение образцовой маски актера.	27
2.2.2 Получение набора тестовых данных.	27
2.2.3 Эмуляция бытовых условий.	29
2.3. Тестирование алгоритмов.	31
2.3.1 Проведение сравнения алгоритмов.	31
2.3.1.1 Выбор метрик.	31
2.3.1.2 Модификация метрик.	32
2.3.1.3 Визуализация результатов.	33
2.3.2 Анализ результатов.	35
2.3.2.1 Исследование применимости алгоритмов.	37
2.3.3 Анализ двухпроцентных дефектов-выбросов.	41
3. Разработка собственного алгоритма	44
3.1. Попытки улучшить существующие решения.	44
3.1.1 Улучшение классических алгоритмов.	44
3.1.1.1 Комбинирование дифференциального кеинга с оптическим потоком.	44

3.1.2 Улучшение алгоритмов, основанных на методах нейронной сети.	46
3.1.2.1 Дообучение одних сетей на наборе данных других.	46
3.2. Разработанный оптимизированный алгоритм.	48
3.2.1 Описание разработанного алгоритма.	48
3.2.2 Тестирование разработанного алгоритма.	49
Заключение	53
Список используемой литературы	55
Приложение 1.	58

Введение

По мере развития сети Интернет все больше людей работает дома, участвуя во множестве видеоконференций и онлайн-трансляций. При этом очень востребованы решения, обеспечивающие замещение фона, на котором снимается актер, статическим или динамическим изображением. На сегодняшний день существует множество различных вариантов решения задачи по замене фона. Каждый алгоритм имеет свои достоинства и недостатки и, следовательно, различную область применимости.

Активно используемая на сегодняшний день технология выделения фона – “хромакей” позволяет получить отличный результат. Однако, эта технология требует наличия как специального одноцветного фона, так и стабильного равномерного освещения фона, а также, цвета выделяемого объекта/актера должны быть отличны от цвета фона. Данное требование является трудновыполнимым в обычных условиях, ввиду чего необходимо исследовать другие методы, которые позволят получить схожие результаты без наличия специального оборудования.

Несмотря на то, что на рынке представлено множество различных решений, до сих пор отсутствует их сравнение на наборе данных, приближенных к условиям, возникающим при проведении обычных видеоконференций и онлайн-трансляций. Кроме того, существующие решения обладают низким качеством разделения актер-фон, либо требуют специализированных, дорогостоящих условий съемки. В связи с этим, задача по исследованию применимости (с последующей оптимизацией) распространенных алгоритмов выделения актера для бытовых систем видеоконференций является актуальной и востребованной.

Целью работы является исследование применимости и оптимизация распространенных алгоритмов выделения актера для бытовых систем видеоконференций.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) Произвести обзор существующих распространенных методов и алгоритмов выделения актера.
- 2) Исследовать особенности съемки актера при проведении интернет-трансляций. Произвести оценку качества изображения, уровня шумов, стабильности фона, освещения. Создать набор тестовых данных с эмуляцией бытовых условий.
- 3) Разработать систему тестирования и провести сравнительный анализ выбранных решений.
- 4) Исследовать влияние различных факторов на эффективность и корректность работы рассматриваемых алгоритмов. Дать рекомендации по использованию этих алгоритмов.
- 5) Разработать алгоритм, позволяющий существенно улучшить существующие решения при проведении интернет трансляций.
- 6) Реализовать разработанный алгоритм и протестировать его.

1. Обзор существующих распространенных методов и алгоритмов выделения актера

Задача по определению принадлежности каждого пикселя изображения к некоторому классу называют задачей по выделению (сегментации). В применении к рассматриваемой задаче это сегментация изображения на один класс (“актер”) и все остальное (“фон”).

При проведении видеоконференций (интернет-трансляций), как правило, используется бытовая статическая камера разрешения 1920x1080 или 1280x720. Соответственно, необходимо сформулировать особенности видеосъемки, которые впоследствии будут накладывать ограничения, к которым рассматриваемые алгоритмические решения должны быть устойчивы. Используемая камера может немного изменять свою позицию в течении времени, например, вследствие микроколебаний стола. Освещение при этом может изменяться во времени в результате локальных или глобальных возмущений (движение солнца за окном, автобаланс цвета, включение света в комнате). Из-за отсутствия профессионального освещения на месте съемки все входные изображения обладают высоким уровнем шумов. В кадр, как правило, попадает только верхняя часть актера, включая руки и пальцы. Фон, на котором происходит съемка актера, может быть произвольным, как со статическими, так и с динамическими объектами позади.

На сегодняшний день существует шесть принципиально различных подходов к решению поставленной задачи:

1. Разностная замена фона - вычитание предварительно сфотографированного изображения без актера из последующих изображений с актером.
2. Использование метода “хромакей”.

3. Использование классических алгоритмов по сегментации изображения.
4. Распознавание движения, с последующим выделением актера на основе анализа оптического потока.
5. Выделение актера на основе технологии использования данных от z-камер (с данными о глубине в каждом пикселе).
6. Использование сверточных нейронных сетей, обученных на сегментацию человека.

Для каждого из этих подходов существует несколько различных методов, рассмотрим наиболее эффективные и популярные из них. Предварительно проанализировав каждый подход на возможность решения поставленной задачи, принимая во внимание ограничения, накладываемые условием бытовой видеосъемки.

- 1) Вариант с вычитанием исходного фона можно использовать при использовании цветов одежды актера, отличных от цвета фона. Данный метод позволяет достичь режима реального времени, а алгоритмическая постобработка теоретически позволит преодолеть введенные ограничения. При этом необходимо предварительно оценить уровень шумов, для определения пороговых значений разности цветов и на основании этого сделать вывод об итоговой применимости данного алгоритма. Кроме того, стоит отметить, что недостатком этого метода является необходимость предварительного выхода актера из кадра.
- 2) Использование метода “хромакей” не представляется возможным, ввиду накладывания им сильных ограничений на однотонность цвета фона и на качество освещения, недостижимых в бытовых условиях.
- 3) Использование классических алгоритмов для сегментации объектов на изображении (Graph Based Segmentation [1], Normalized Graph Cuts [2]) теоретически представляется возможным, но из-за низкого

быстродействия ввиду необходимости трансляции видео в режиме реального времени на практике не используется.

- 4) Анализ оптического потока позволяет выделить актера только во время его движения, ввиду чего, при остановке (замирании) актера его выделение будет невозможно. Модификация этого метода, основанная на его комбинации с дифференциальной заменой фона дает неприемлемые результаты на одноцветных участках изображения - на которых, ввиду алгоритмической особенности алгоритма, невозможно определить движение.
- 5) Вариант с использованием z-камер не представляется возможным, ввиду их отсутствия у большинства пользователей. Общедоступные недорогие камеры имеют плохое качество и низкое разрешение z-сенсоров, а также сильную зашумленность.
- 6) Вариант с использованием сверточной нейронной сети представляется наиболее перспективным, ввиду отсутствия необходимости во внешнем оборудовании и предварительного выхода актера из кадра. Кроме того, этот вариант обладает устойчивостью ко всем описанным выше ограничениям, может работать в режиме реального времени и позволяет актеру обладать любым цветовым сочетанием по сравнению с фоном.

Таким образом, целесообразно провести поиск и выбор алгоритмов выделения актера с использованием методов 1 и 6.

1.1. Алгоритмы, основанные на методах вычитания исходного фона.

В настоящее время существует большое количество алгоритмических решений, основанных на вычитании исходного фона. В результате поиска было найдено 12 различных алгоритмов. Поиск происходил с преимущественной фильтрацией по новизне, цитируемости и анализа

представленных результатов авторов, удовлетворяющих нашим ограничениям. Алгоритмы выбирались с различными идейными подходами к решению поставленной задачи.

В дальнейшем было проведено сравнение алгоритмов на наборе данных, полученных с камеры ноутбука MSI GE76 (Рис. 1). При этом автобаланс у камеры был выключен, а опорный кадр брался как результат усреднения видеопоследовательности из 100 кадров.

В первой строке представлен набор из 4 изображений с различными фонами и положениями актера, характерными при проведении видеотрансляций, на котором проходило тестирование. В последней строке представлена образцовая маска, полученная вручную с использованием программы Adobe Photoshop.

Параметры для каждого алгоритма подбирались так, чтобы было достигнуто максимальное значение метрики F_1 , которая определяется как:

$$F_1 = \frac{TP}{TP + 0.5 \times (FP + FN)} \quad (1)$$

где **TP** (true positive) - количество пикселей предсказанных как актер, и действительно являющихся актером; **FP** (false positive) - количество пикселей предсказанных как актер, но являющихся фоном; **FN** (false negative) - количество пикселей предсказанных как фон, но являющихся актером.


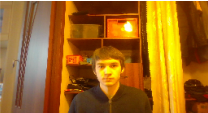




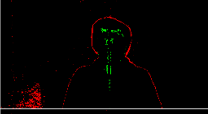
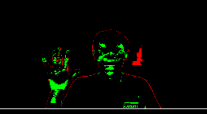
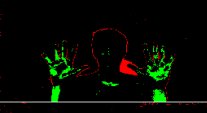

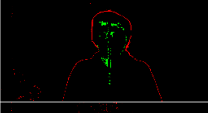
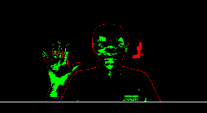
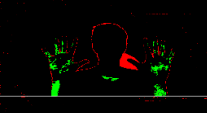
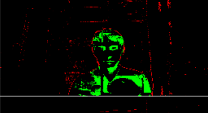

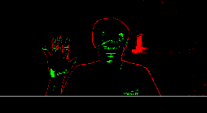
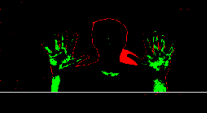
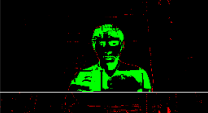
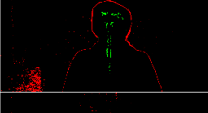
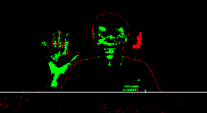
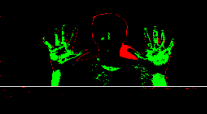
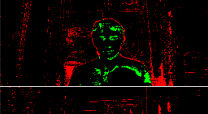
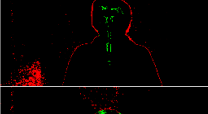
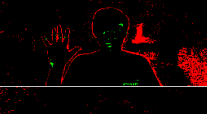
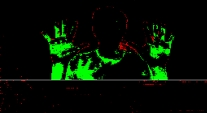
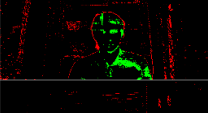
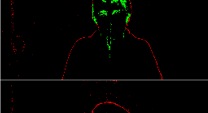

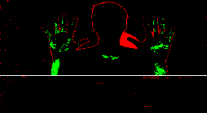
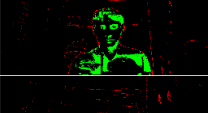
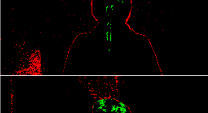

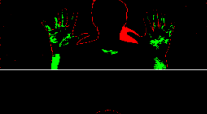

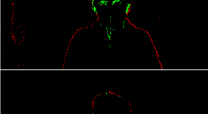
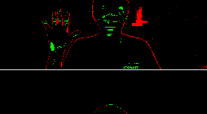

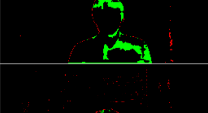
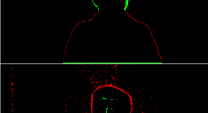
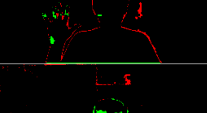
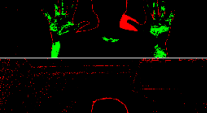
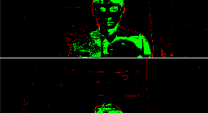
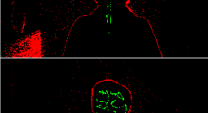

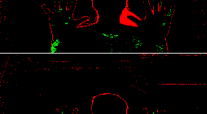
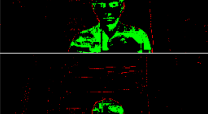
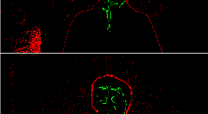
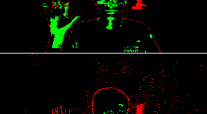

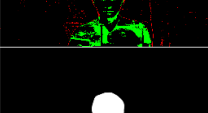

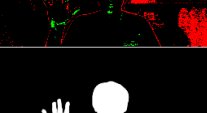




				1 - набор входных изображений
				2 - базовое вычитание
				3 - базовое вычитание
				4 - single gaussian
				5 - fuzzy GMM
				6 - eigenbackground subtraction
				7 - KNN
				8 - fuzzyAdaptiveSOM
				9 - MOG
				10 - PAWCS
				11 - ViBe
				12 - weighted median;
				13 - простое вычитание
				14 - образцовые маски

Рис. 1. Сравнение 12 различных видов алгоритмов.

Красным отмечены FP предсказания, зеленым - FN.

Алгоритм, представленный в строке 2, использует комбинацию метода сравнения текущего кадра с медианным фоном за N последних кадров с помощью оператора Лапласа и дифференциальной разницы по порогу [3]. Видно, что он порождает неприемлемые дефекты в сетях 1, 2 и 4, так как в значительной степени актер классифицируется как фон. Среднее значение F_1 метрики по всем изображениям составило 0.89.

Алгоритм, представленный в строке 3, использует метод подавления фона [4]. Видно, что он порождает неприемлемые дефекты в сетях 1, 2 и 4. Средний F_1 составил 0.89

Алгоритм, представленный в строке 4, моделирует каждый пиксель фона с помощью функции плотности вероятности, полученной с помощью набора обучающих кадров [5]. Видно, что он порождает неприемлемые дефекты в сетях 1 и 2. Средний F_1 составил 0.91.

Алгоритм, представленный в строке 5, основан на байесовском методе с использованием нечеткой модели (fuzzy model) и является усовершенствованием алгоритма Gaussian Mixture Model (GMM) для нестабильного фона [6]. Видно, что он порождает неприемлемые дефекты в сетях 1, 2 и 3. Средний F_1 составил 0.78.

Алгоритм, представленный в строке 6, основан на объединении нисходящей и восходящей информации в замкнутом контуре обратной связи, причем оба компонента используют статистический байесовский подход [7]. Видно, что он порождает неприемлемые дефекты в сетях 1 и 2. Средний F_1 составил 0.79.

Алгоритм, представленный в строке 7, основан на построении рекурсивных уравнений, которые используются для постоянного обновления параметров модели гауссовой смеси (MOG) и одновременного выбора подходящего количества компонентов для

каждого пикселя [8]. Видно, что он порождает неприемлемые дефекты в сетях 1 и 2. Средний F_1 составил 0.89.

Алгоритм, представленный в строке 8, базируется на самоорганизации через нейронные сети и нечеткой (fuzzy) модели для решения проблем принятия решений [9]. Видно, что он порождает неприемлемые дефекты в сети 2. Средний F_1 составил 0.90.

Алгоритм, представленный в строке 9, для выделения актер-фон использует адаптивный алгоритм с использованием плотности вероятности гауссовой смеси [10]. Видно, что он порождает неприемлемые дефекты в сетях 2 и 3. Средний F_1 составил 0.91.

Алгоритм, представленный в строке 10, использует непараметрический подход к моделированию фона на уровне пикселей. Важность каждой фоновой выборки оценивается на основе их повторяемости среди всех локальных наблюдений [11]. Видно, что он порождает неприемлемые дефекты в сетях 1 и 2. Средний F_1 составил 0.91.

Алгоритм, представленный в строке 11, хранит для каждого пикселя набор значений, взятых в прошлом в том же месте или по соседству. Затем он сравнивает этот набор с текущим значением пикселя, чтобы определить, принадлежит ли этот пиксель фону, и адаптирует модель, случайным образом выбирая значения для замены из фоновой модели. Когда обнаруживается, что пиксель является частью фона, его значение распространяется на фоновую модель соседнего пикселя [12]. Видно, что он порождает неприемлемые дефекты в сетях 2 и 4. Средний F_1 составил 0.85.

Алгоритм, представленный в строке 12, был реализован самостоятельно и основывается на сравнении средних значений и среднеквадратичных отклонений первоначального кадра фона, усредненного за N кадров, и текущего кадра, усредненного за M последних кадров [13]. Реализованный

алгоритм порождает неприемлемые для нашей задачи результаты в сетях 2 и 4. Кроме того, при быстром движении актера, за ним будет оставаться след. Средний F_1 составил 0.88.

Алгоритм, представленный в строке 13, реализует обычное вычитание предварительно сфотографированного фона. Данный алгоритм порождает неприемлемые дефекты во всех сетях. Средний F_1 составил 0.78.

Вывод. Все из рассматриваемых алгоритмов дают неприемлемые дефекты на хотя бы одном из рассматриваемых тестовых сетях. Были проведены попытки по улучшению рассматриваемых алгоритмов, путем их комбинации с другими методами, но значительного улучшения достигнуто не было.

Таким образом, использование метода вычитания фона для решения задачи по выделению актера на произвольном фоне не представляется возможным. Предположительно, посредственные результаты связаны с высоким уровнем шумов для бытовых видеокамер, а также невозможности их устранения в режиме реального времени. Принимая во внимание важность шумовой составляющей необходимо учитывать ее в дальнейшем, при построении синтетических тестовых данных.

1.2. Алгоритмы, основанные на использовании сверточной нейронной сети.

Еще одним возможным решением задачи о выделении человека на изображении без внешнего оборудования в режиме реального времени является вариант с использованием одноклассовой сегментационной сверточной нейронной сети. Ввиду актуальности данной проблемы в результате поиска было найдено множество сверточных нейронных сетей. Оставив из них только те сети, на вход которых поступает трехканальное изображение и на выходе получается одноканальная матрица вероятностей принадлежности пикселей к классу человека (маска актера) было проведен

визуальный анализ получаемых результатов. На этапе визуального анализа, ввиду неудовлетворительных результатов как по качеству, так и по скорости, были исключены из дальнейшего рассмотрения сети с архитектурами: DeepLabV3 [14] и FCN [15]. После чего для оставшихся нейронных сетей было проведено сравнение по качеству сегментации на тестовом наборе данных из 4-х изображений, описанном выше (рисунок 2).






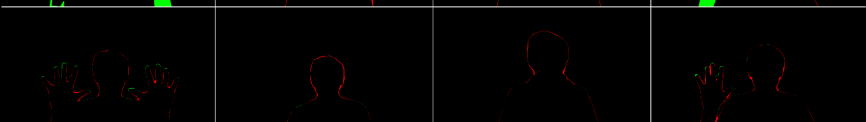

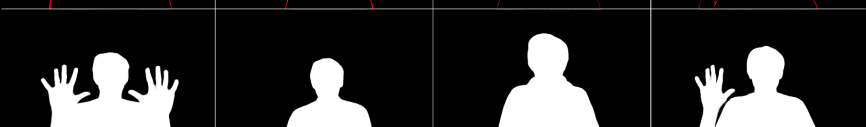
	1 - набор сравниваемых изображений		
	2 - GoogleMeet MediaPipe		
	3 - MODNet		
	4 - Nvidia Broadcast		
	5 - U2Net (U^2Net)		
	6 - UNet-ternaus		
	7 - RVM		
	8 - образцовые маски		

Рис. 2. Сравнение 6 алгоритмов использующих сверточную нейронную сеть.

Красным отмечены FP предсказания, зеленым - FN.

Алгоритм, представленный в строке 2, использует сверточную нейронную сеть с архитектурой MobileNetV3 с постобработкой получившейся маски с помощью билатерального фильтра [16]. Информации о данных на которых происходило обучение этой сети предоставлено не было. Данный алгоритм предназначен для работы на обычном процессоре. В результате средний F_1 по

всем четырем примерам составил 0.92. Тем не менее, ввиду специфики данного решения, было принято решение о его более детальном рассмотрении на последующих этапах.

Алгоритм, представленный в строке 3, предсказывает портретную сегментационную маску, граничные детали и итоговую альфа-маску через три взаимозависимые ветви нейронной сети [17]. Обучение этой сети происходило на наборе данных Adobe Matting Dataset и PPM-100. В результате средний F_1 по всем примерам составил 0.98. Стоит отметить, что среднее количество кадров в секунду составило 35 на видеокарте RTX 3060 Laptop.

Алгоритм, представленный в строке 4, предоставляет SDK без описания используемой архитектуры нейронной сети [18]. Кроме того, авторами не был описан набор данных на которых происходило ее обучение. В результате средний F_1 составил 0.99. Стоит отметить, что данный алгоритм позволяет достичь наиболее приемлемых результатов, обладает высоким быстродействием (способен обрабатывать 60 кадров в секунду), однако для его работы требуется видеокарта Nvidia серии 20 и выше.

Алгоритм, представленный в строке 5, использует сверточную нейронную сеть с новейшей архитектурой, которая представляет двухуровневую вложенную сеть типа UNet, направленную на сегментацию значимых объектов в кадре. Эта конструкция способна собирать больше контекстуальной информации из разных масштабов благодаря смешению рецептивных полей [19]. Существует несколько различных версий этой сети: обученной на наборе данных DUTS-TR либо Supervisely Person Dataset с разным количеством слоев в сети. Для тестирования использовалась версия с большим количеством слоев и обученная на наборе данных Supervisely. В

результате средний F_1 составил 0.88. Среднее количество кадров в секунду составило 30 на видеокарте RTX 3060 Laptop.

Алгоритм, представленный в строке 6, использует классическую архитектуру сверточной сети UNet с энкодером timm-efficientnet-b0 [23], обученную на наборе данных COCO, Pascal VOC, AI segment Human Matting и Mapillary Vistas Commercial [20]. В результате средний F_1 составил 0.99. Стоит отметить, что ввиду специфики обучения данной сети, возможно получение только бинарной результирующей маски. Количество кадров в секунду составило 30 на видеокарте RTX 3060 Laptop.

Алгоритм, представленный в строке 7, использует рекуррентную архитектуру нейронной сети для использования временной информации в видео с постобработкой билатеральным фильтром. В качестве основного и промежуточного блоков взята архитектура сетей DeepLabV3 и MobileNetV3 соответственно. Нейронная сеть обучалась на наборе данных VideoMatte240K, Distinctions-646 и Adobe Image Matting. Представленная архитектура сети способна выполнять задачи матирования и сегментации последовательно выдавая изображение сходное с результатами “хромакея” [21]. В результате средний F_1 составил 0.98. Количество кадров в секунду составило 60 на видеокарте RTX 3060 Laptop.

Также были визуально протестированы алгоритмы, используемые в видеоконференциях (Skype, Microsoft Teams, Zoom), работающие с использованием обычного процессора. Существенного преимущества выявлено не было, а учитывая их закрытость для коммерческого использования и сложности, возникшие при ручном тестировании, было принято решение не рассматривать их в дальнейшем.

Таким образом, при обзоре существующих решения были выделены шесть визуально приемлемых кандидатов, которые необходимо протестировать более детально.

2. Детальное тестирование выбранных алгоритмов

Для исследования особенностей каждого из алгоритмов выделения актера и определения наилучшего из них - необходимо провести сравнение, отличное от визуального. Сравнение при этом необходимо производить на наборе данных, приближенных к возникающим при проведении онлайн трансляций, с учетом особенности бытовых камер. Для этого необходимо изучить условия, возникающие при съемке в бытовых условиях на веб-камеру.

2.1 Изучение условий, возникающих при проведении онлайн-трансляций.

Изображения с бытовых камер, как правило, содержат большое количество шумов. Также бытовые камеры могут иметь неотключаемые настройки по автоматической регулировке параметров съемки. Используемая камера может немного изменять свою позицию в течении времени, освещение может изменяться во времени в результате локальных или глобальных возмущений. В кадр, как правило, попадает только верхняя часть актера, включая руки и пальцы. Фон, на котором происходит съемка актера, может быть произвольным, как со статическими, так и с динамическими объектами позади. Изучим более детально возникающие условия, описанные выше.

2.1.1 Изучение шумовой составляющей веб-камер.

Шум изображения - случайное изменение цветовой информации в изображении. Шум всегда присутствует в цифровых изображениях на этапах их получения, кодирования, передачи и обработки.

Основными этапами формирования шумов являются: тепловое непостоянство окружающей среды, электронные помехи, невозможность идеально усилить сигнал, ошибки при преобразовании фотонов в электроны,

ошибки при чтении. Ввиду отсутствия возможности определения природы шума, проведем исследование шумовой картины в целом, для различных камер с различным разрешением, для разного фона и разной освещенности.

2.1.1.1 Временное распределение шумов в зависимости от их пространственного расположения.

Исследуем временное распределение шумов в зависимости от их пространственного расположения. Для этого рассмотрим шумовое распределение для каждого пикселя между двумя последовательными кадрами на протяжении ста кадров на статичном фоне. Для этого построим статистику дельт по формуле 1 для $N \in [1, 100]$:

$$\Delta_{(x,y,N)} = C_{xy}(N) - C_{xy}(N - 1) \quad (2)$$

где $\Delta_{(x,y,N)}$ - разность цветовой компоненты C в пикселе с координатами (x,y) между N и $N-1$ кадрами.

При построении гистограммы для случайно выбранных пикселей (x,y) было замечено, что распределение представляет из себя гауссового. Таким образом, для визуализации полученной статистики проведем аппроксимацию гауссовым распределением на выборке $\Delta_{(x,y,N)}$, где $N \in [1, 100]$ для каждого пикселя (x,y) . Результат для некоторого пикселя, нормированный на единичную площадь изображен на рисунке 3.

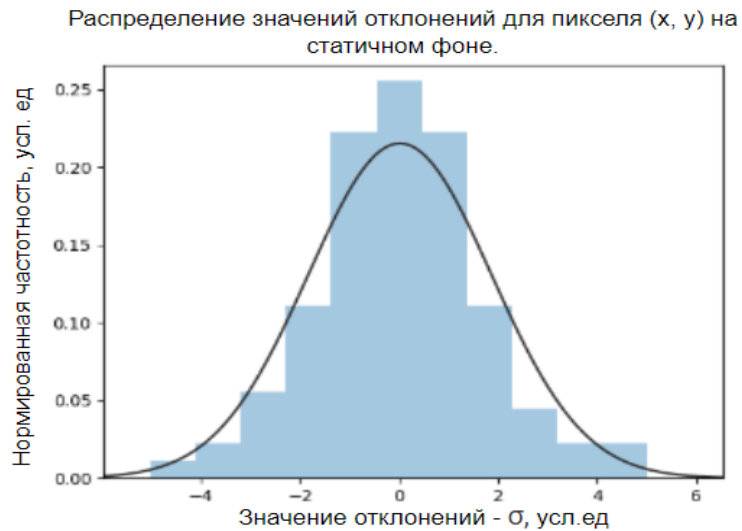


Рис. 3. Шумовое распределение для пикселя с координатами (x, y) за 100 кадров на статичном фоне для цветовой компоненты Y в цветовом пространстве YCrCb, полученное с помощью камеры модели Marshall CV344.

Для сравнения и анализа результатов было предложено использовать σ - среднеквадратичное отклонение, внутри которого находятся 68.2% отклонений от максимального значения. Проведем описанный выше эксперимент для различных камер с различным разрешением и изучим характерные особенности. Визуализируем σ для всех пикселей. Полученные результаты изображены на рисунке 4.

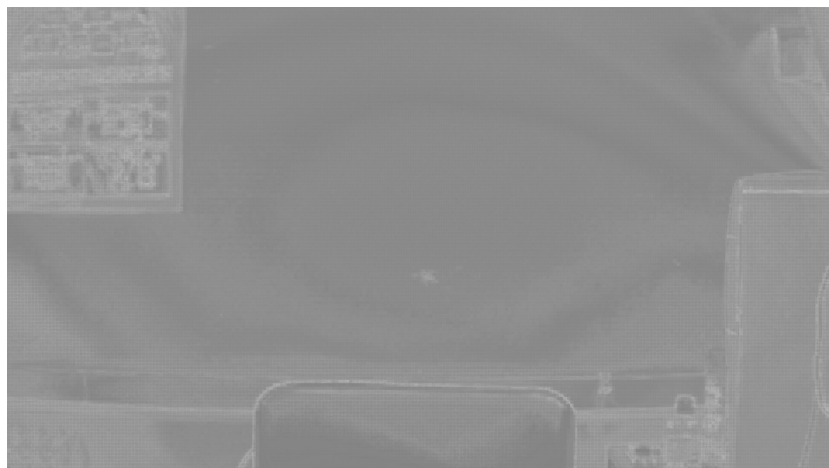


Рис. 4. Визуализация значений среднеквадратичного отклонения для серии из 100 снимков усиленного в 5 раз, снятых на камеру Marshall CV344, при

записи видео в формате MPEG для цветового пространства RGB и компоненты В с разрешением 2160x1440.

Из проведенного эксперимента установлено, что характерный размер пространственных шумов составляет от 2 до 3 пикселей и отчетливо проявляется в местах, где присутствует цветовой градиент - на границах. Кроме того, проанализировав гистограммы для всех цветовых компонент, было установлено, что цветовое пространство YCrCb имеет в среднем среднеквадратичное отклонение 2.5, тогда как пространство RGB около 4. Ввиду чего, в дальнейшем будем использовать цветовое пространство YCrCb.

Аналогичные исследования были проведены для камер: Marshall CV344, Marshall CV420-CS, Sony FDR-AX1 для различных фонов. И сделаны выводы о том, что в целом шумовая составляющая не зависит от их пространственного расположения, за исключением соседствующих областей с цветовым градиентом на границах. Это обусловлено смещением видеокамер на величины порядка 0-1.5 пикселя. Соответственно, возникает необходимость в эмуляции этого явления.

2.1.1.2 Пространственное распределение шумов.

Исследуем распределение шумов для двух последовательно снятых кадров на статичном фоне. Для этого построим гистограмму разностей для всех пикселей двух последовательно взятых статичных изображений и нормируем ее на единичную площадь. Полученный результат изображен на рисунке 5.

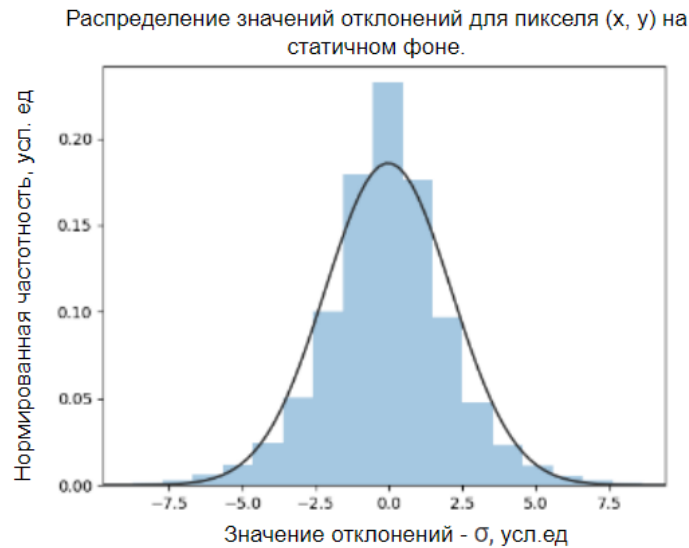


Рис. 5. Шумовое распределение для всех пикселей на статичном фоне для цветовой компоненты Y в цветовом пространстве YCrCb, полученное с помощью камеры модели Marshall CV344.

Из графика видно, что результаты по распределению и значениям совпадают с результатами, полученными при исследовании шумовой составляющей для конкретного пикселя. Исследуем максимально возможное отклонение пикселей друг от друга для двух последовательно взятых изображений на серии из 100 снимков. Результат изображен на рисунке 6.

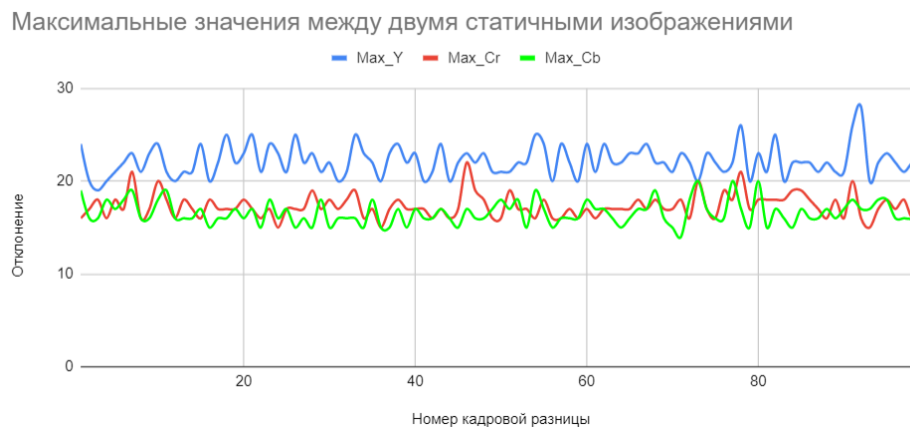


Рис. 6. Максимальные значения отклонений среди всех пикселей двух последовательных изображений при разрешении 2160x1440, снятые на камеру Marshall CV344 в цветовом пространстве YCrCb.

Из графика видно, что для некоторых пикселей разница значений может достигать 28, что является препятствием при использовании метода вычитания исходного фона.

Проведя аналогичные исследования для той же камеры с разрешением 1920x1080 установлено, что шумовая составляющая уменьшилась, и максимальные значения для цветовой компоненты Y составили 24. Используя же линейную интерполяцию из разрешения 2160x1440 в 1920x1080 можно уменьшить максимальные значения на 25%. Результаты представлены на рисунке 7.

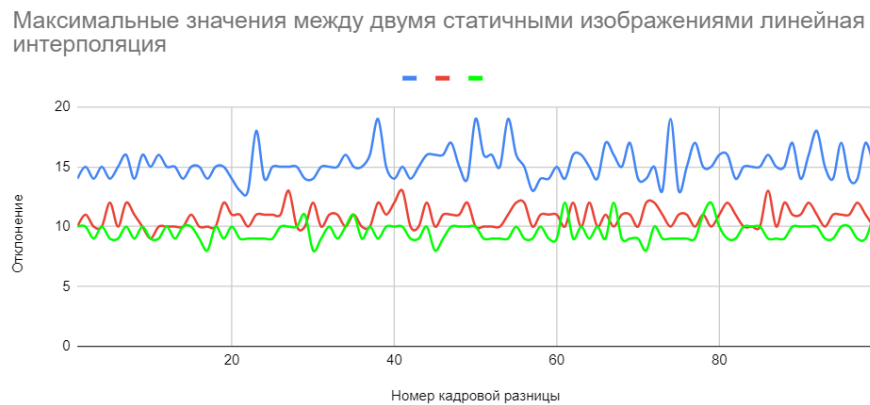


Рис. 7. Максимальные значения для серий изображений из ста снимков для разрешения интерполированного разрешения 1920x1080, снятые на камеру Marshall CV344 в формате YCrCb.

Проведем аналогичные измерения для камеры обычного ноутбука с разрешением 1920x1080 при хорошем освещении. Среднеквадратичное отклонение (σ) составило 5 единиц. Визуализация изображена на рисунке 8.

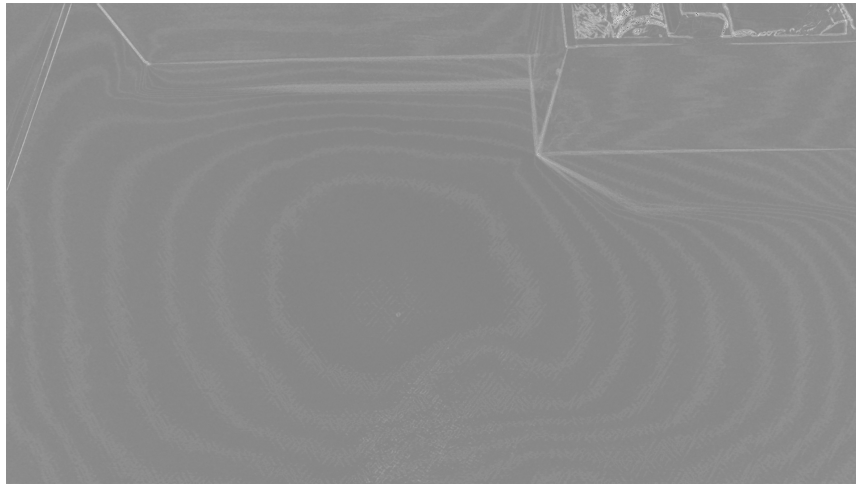


Рис. 8. Визуализация усиленных в 5 раз значений среднеквадратичного отклонения для серии из 100 снимков, снятых на камеру ноутбука GE76 в цветовом пространстве YCrCb с разрешением 1920x1080.

Рисунок 8 подтверждает замеченное в п. 2.1.1.1 предположение, о том, что шумы в основном содержатся на градиентных областях. В этих местах максимальные значения отклонений достигали до 60 единиц для цветовой компоненты Y. Результаты изображены на рисунке 9.

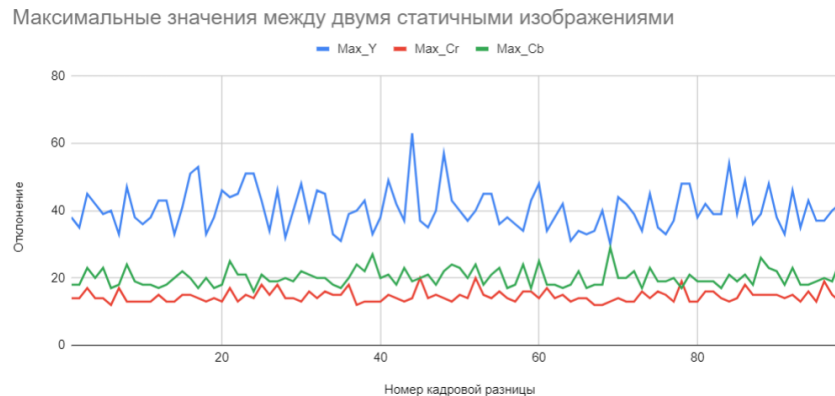


Рис. 9. Максимальные значения для серий изображений из ста снимков для камеры ноутбука GE76 при разрешении 1920x1080 в формате YCrCb.

Аналогичные исследования были проведены для камер: Marshall CV344, Marshall CV420-CS, Sony FDR-AX1 для различных фонов, характерных при проведении видеоконференций. При этом было установлено, что для камер типа Marshall CV344, Marshall CV420-CS, Sony FDR-AX1 по характеру

распределения шумов являются гауссовскими, тогда как для веб-камеры ноутбука GE76 шумы являются кластеризованными (рисунок 10).

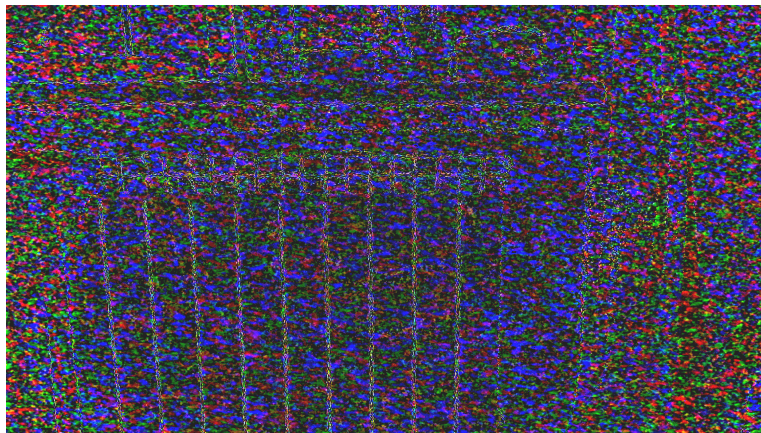


Рис. 10. Усиленные в 5 раз кластеризованные шумы для изображения полученного на камеру ноутбука GE76, как разность двух последовательных статичных кадров, с разрешением 1920x1080 в формате RGB.

Таким образом, при исследовании условий, возникающих при бытовой съемке, было установлено, что шумы характеризуются двумя типами: гауссовские и кластеризованные. Характерное среднеквадратичное отклонение находится в диапазоне [2-5] с максимальными значениями [25-50]. При этом отсутствует корреляция между пространственным расположением и уровнем шумов. Однако столь высокие максимальные значения шумов обусловлены смещением камер на величины порядка пикселя. Следовательно необходимо проводить эмуляцию смещения видеокамеры.

2.1.2 Изучение других особенностей веб-камер.

Не менее важный вклад при съемке вносят технологии автобаланса и автофокуса камер.

Автобаланс - это процесс автоматического изменения яркости всего изображения для нивелирования изменения яркости в результате внешних факторов. Как правило, внешним фактором является глобальное изменение

освещенности, например в результате включения света, либо захода актера в кадр. Зачастую автобаланс пытается сохранять постоянную яркость на последовательности изображений так, чтобы средняя яркость в кадре под номером t совпадала со средней яркостью в кадре под номером $t + 1$. Автобаланс можно эмулировать путем добавления этого значения к яркости $t + 1$ кадра.

Помимо глобального изменения освещенности существует локальное изменение освещенности. Локальное изменение освещенности существенным образом модифицирует изображение между t и $t + 1$ кадрами. Как правило, локальное изменение освещенности возникает в результате появления точечных источников освещения: таких как Солнце или настольная лампа и представляет серьезную проблему для методов дифференциального кеинга.

Автофокус - это адаптивная система, обеспечивающая автоматическую фокусировку объектива веб-камеры на один или несколько объектов съёмки. Как правило, полноценная технология автофокуса невозможна на обычных веб-камерах, однако в последних присутствует ее вариация в виде постоянного колебания около максимального значения резкости.

Таким образом, описанные выше условия съемки важны и их необходимо учитывать при разработке системы эмуляции бытовых условий.

2.1.3 Изучение характерных фоновых изображений.

При бытовых условиях видеосъемки возможны разнообразные варианты фонового изображения. Для создания набора данных наиболее близкого к условиям, возникающим при проведении интернет-трансляций, необходимо выделить характерные группы фоновых изображений.

Был проведен анализ 50 видео с онлайн-трансляций различных видеоблогеров YouTube и выявлены характерные особенности фонового изображения (рисунок 11).



Рис. 11. Примеры изображений взятых из видео для анализа фоновых условий различных видеоблогеров.

В результате проведенного анализа было установлено, что большинство людей использует статичный фон как с малым, так и большим количеством объектов позади. При этом у некоторых людей цветовая гамма актера совпадала с цветами фона, а у некоторых использовался монотонно зеленый фон. У части людей фон был динамический: на заднем фоне колыхалась штора, бегали домашние животные или ходили люди.

Таким образом, необходимо учитывать характерные фоновые условия при разработке системы по эмуляции бытовых условий.

2.2 Разработка системы тестирования алгоритмов выделения актера с эмуляцией бытовых условий.

Определившись с системой сравнения алгоритмов, необходимо собрать тестовые данные, на которых будет возможно проведение сравнения. Одним из требований является необходимость наличия надежной образцовой маски.

Существующие наборы тестовых данных не удовлетворяют этому критерию либо ввиду низкой точности образцовой маски (так как она была размечена человеком вручную), либо ввиду отсутствия в тестовых данных людей с пальцами рук, а также положениями, характерными при проведении онлайн трансляций. Таким образом, необходимо получить собственный набор тестовых данных, соблюдая наложенные критерии.

2.2.1 Получение образцовой маски актера.

Для получения тестовых данных с надежной образцовой маской при хорошем освещении снимался актер на зеленом фоне, в движении характерном при проведении видеоконференций. При этом, для разнообразия тестовых данных, актер снимался в различной одежде, один из примеров тестовых данных изображен на рисунке 12.

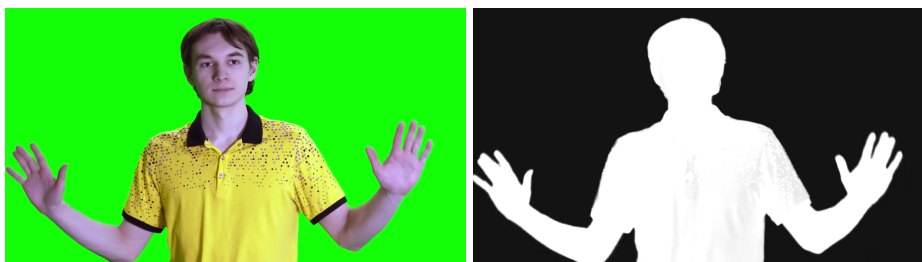


Рис. 12. Пример положения актера с полученной образцовой маской.

После этого производилось получение образцовой маски актера. Для этого с помощью алгоритма Ultimate производился “кеинг” актера по зеленому цвету (рисунок 12).

2.2.2 Получение набора тестовых данных.

Был произведен поиск фонов, с принадлежностью к выделенным в п. 2.1.3 классам. В результате найденные фоны были скомбинированы с изображением выделенного актера. Полученные изображения и соответствующие им образцовые маски изображены на рисунке 13.



Рис. 13. Актер, скомбинированный с различными фоновыми изображениями с соответствующими им образцовыми масками (слева направо, сверху вниз)

- 1, 2 - статичный фон с множеством объектов на заднем фоне;
- 3, 4 - статичный хорошо освещенный монотонный зеленый фон;
- 5, 6 - динамический фон с бегающими на заднем фоне собаками;
- 7, 8 - динамический фон с проходящими на заднем фоне сотрудниками;
- 9, 10 - статичный фон, схожий с цветовой гаммой актера;
- 11, 12 - динамический фон с колышущейся на заднем фоне занавеской.

В результате была получена серия из 30-ти изображений различного рода: 6 вариантов различных фонов с пятью вариантами различной одежды актера. Каждая серия представляла из себя 6 вариантов эмуляции бытовых условий: кластеризованные, либо гауссовские шумы с яркими, либо тусклыми локальными изменениями освещенности для них (подробнее об эмуляции бытовых условий будет рассказано в п. 2.2.3). При этом каждый вариант с разнообразной одеждой актера включал в себя последовательность из двух тысяч изображений в среднем. Стоит отметить, что в этой последовательности были использованы различные варианты поведения актера: наклоны в разные стороны, показы жестов, быстрое и медленное приветствие, корченье гримас, закрытие лица руками и другие.

2.2.3 Эмуляция бытовых условий.

В полученных выше скомбинированных изображениях отсутствуют условия, возникающие при проведении онлайн-трансляций, обнаруженных в пунктах 2.1.1 и 2.1.2 (шумы, автобаланс и автофокус). Таким образом, необходима эмуляция выявленных бытовых условий.

Для эмуляции использовались алгоритмические решения в виде добавления кластеризованных или гауссовских шумов. Для создания гауссовских шумов происходило сложение исходного изображения с массивом той же размерности, заполненного значениями из гауссовского распределения с $\sigma = 5$ и $\mu = 0$:

$$N(x; \mu, \sigma) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot e^{-\frac{(x-\mu)^2}{2 \cdot \sigma^2}} \quad (3)$$

Для создания кластеризованных шумов использовался собственно разработанный итеративный алгоритм. Изначально создавалась матрица, заполненная нулями, с разрешением, равным входному изображению, уменьшенному в 2^k раз (k выбиралось равное пяти). На каждой итерации происходило добавление к значению этого массива рандомизированных чисел из распределения $(-1; 1)$, после чего происходило масштабирование (увеличение) этого массива в 2 раза по каждой из размерностей. По истечении k итераций полученный массив добавлялся ко входному изображению.

Для эмуляции микросмещений видеокамеры использовался сдвиг входного изображения в случайную сторону на случайное дробное число пикселей из диапазона $[0; 1.5]$. Значения пикселей заполнялись билинейно интерполированными значениями пикселей исходного изображения.

Для эмуляции глобального изменения освещенности (автобаланса) входное изображение переводилось в формат YCrCb, затем к Y компоненте этого

изображения происходило добавление случайного значения из диапазона $[-15; 15]$.

Для эмуляции локального изменения освещенности использовался специальное двумерный массив, заполненный эмуляцией большого солнечного пятна. Из центра этого массива происходило его итеративное заполнение круговыми областями с некоторым константным значением с постоянно увеличивающимся радиусом круга. Максимальный радиус последнего круга выбирался так, чтобы значение в центре наложенных друг на друга кругов не превышало 60 и 120 яркостных дискрет для слабого и сильного освещения соответственно, а также так, чтобы диаметр последнего круга не превышал высоту изображения. В результате получается изображение, с яркостью, изменяющееся от центра к краю по закону $\frac{1}{r}$. После чего происходило сложение полученного массива с входным изображением на каждом кадре. При этом центр добавочного изображения на первом кадре помещался в левый нижний угол входного изображения, а затем двигался по закону $y = -x^2$, где $x = -\frac{width}{2} + (i * 10) \% width$, $width$ – ширина изображения, а i – номер кадра видеопоследовательности. В результате получалась эмуляция движения Солнца над горизонтом.

Для эмуляции автофокуса использовался фильтр Гаусса размерности 3×3 с малым значением σ .

В результате полученные серии изображений были готовы для проведения на них тестирования алгоритмов.

2.3. Тестирование алгоритмов.

2.3.1 Проведение сравнения алгоритмов.

2.3.1.1 Выбор метрик.

Для проведения тестирования алгоритмов была написана на языке Python функция, принимающая на вход образцовую маску и маску, полученную в результате работы рассматриваемого алгоритма. Выходом этой функции был массив, состоящий из значений четырех метрик.

В качестве стандартной метрики, выбранной из-за распространенности в научном сообществе (встречается практически во всех научных статьях), использовалась средняя абсолютная ошибка (MAE):

$$MAE(x, y) = 1 - \frac{\sum_{i=1}^n |x_i - y_i|}{n} \quad (4)$$

где x_i - значение i -ого пикселя образцовой маски в диапазоне $[0.0; 1.0]$, y_i - значение i -ого пикселя маски, полученной в результате работы исследуемого алгоритма, в диапазоне $[0; 255]$, n - количество пикселей в маске.

В качестве наглядно воспринимаемой метрики была использована небинарная версия меры Жаккара (IoU). Эту метрику можно описать как результат деления площади пересечения образцовой маски и результирующей маски на площадь их объединения. Небинарность связана с тем, что маски, получаемые в результате работы как алгоритма Ultimate, так и сравниваемых алгоритмов, были небинарные. В математическом виде небинарная версия метрики IoU записывается следующим образом:

$$IoU(x, y) = \frac{\sum_{i=1}^n x_i \cdot y_i}{\sum_{i=1}^n x_i + y_i - x_i \cdot y_i} \quad (5)$$

где x_i - значение i -ого пикселя образцовой маски в диапазоне $[0; 255]$, y_i - значение i -ого пикселя маски, полученной в результате работы алгоритма, в диапазоне $[0; 255]$, n - количество пикселей в маске.

2.3.1.2 Модификация метрик.

При проведении интернет-трансляций граничная область вокруг актера не обязана находиться с идеальной точностью. Ошибка в несколько пикселей не приводит к значительному визуальному ухудшению восприятия актера. В связи с этим используемые метрики для сравнения изображений были модифицированы путем исключения из рассмотрения граничной области вокруг актера. Производилось исключение области изображения, полученной в результате разности расширенной и суженной маски (на рисунке 14 исключаемая область отображается серым цветом). Получение суженной маски происходило с помощью трех итеративных морфологических преобразований по следующей формуле:

$$dst(x, y) = \min_{(x', y'): element(x', y') \neq 0} src(x + x', y + y') \quad (6)$$

Аналогично происходило получение расширенной маски, по формуле:

$$dst(x, y) = \max_{(x', y'): element(x', y') \neq 0} src(x + x', y + y') \quad (7)$$

где $dst(x, y)$ цвет пикселя с координатами (x, y) выходного изображения, $element(x', y')$ - цвет пикселя с координатами (x', y') структурного элемента морфологического преобразования (в нашем случае квадрат размером 3×3).

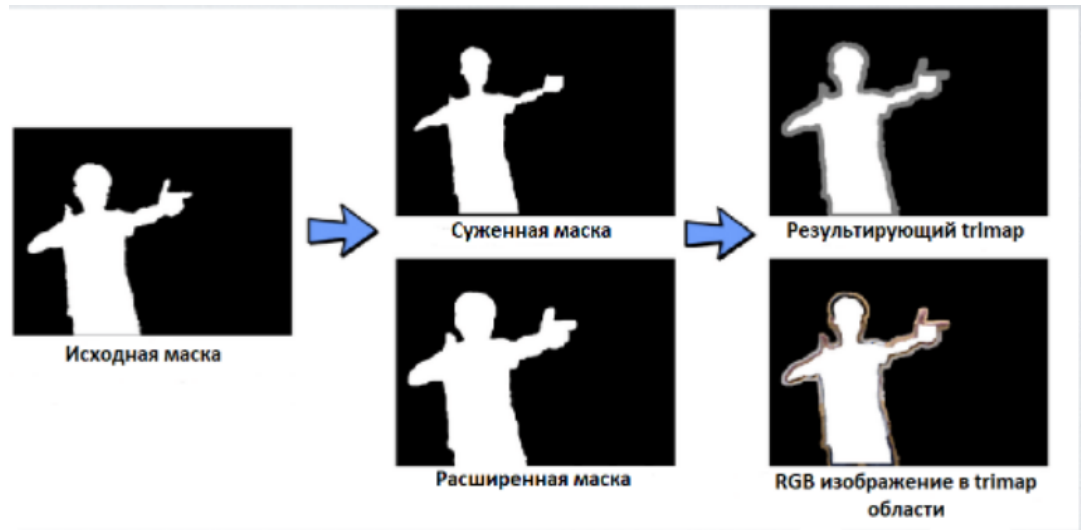
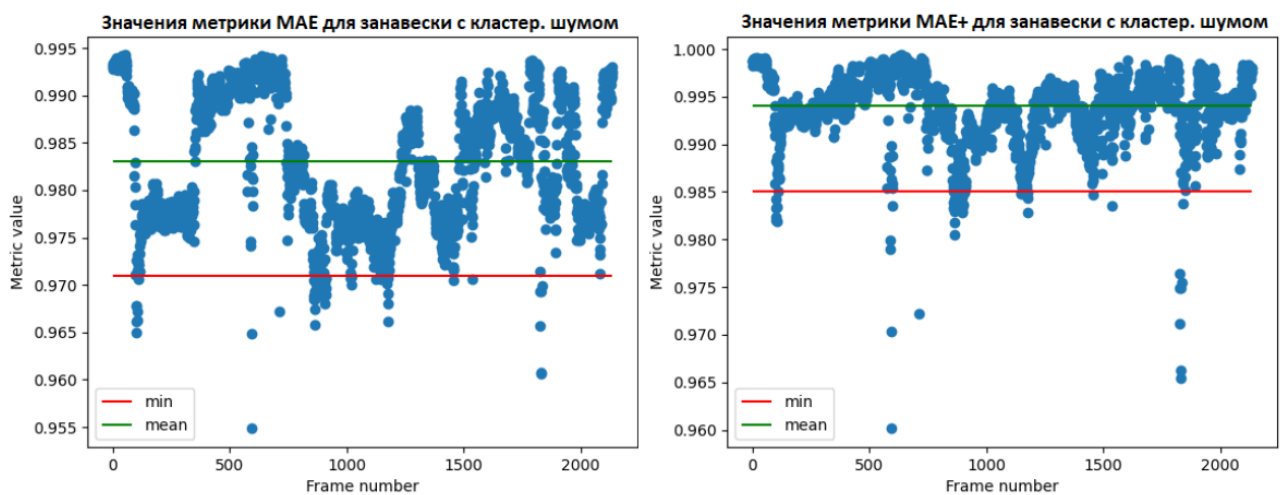


Рис. 14. Модификационная версия маски актера.

Такой подход применялся как к образцовой маске, так и к маске, полученной в результате работы алгоритма. После чего происходило сравнение этих масок по уже описанным метрикам, но без учета правильности предсказания внутри серой (исключаемой) области. В дальнейшем для обозначения модифицированных версий метрик MAE и IoU будет введено следующее обозначение: MAE+ и IoU+ соответственно.

2.3.1.3 Визуализация результатов.

Для наглядности полученных результатов проведем визуализацию полученных значений метрик. Для этого построим графики получаемых значений метрик в зависимости от номера серии изображений (рисунок 15).



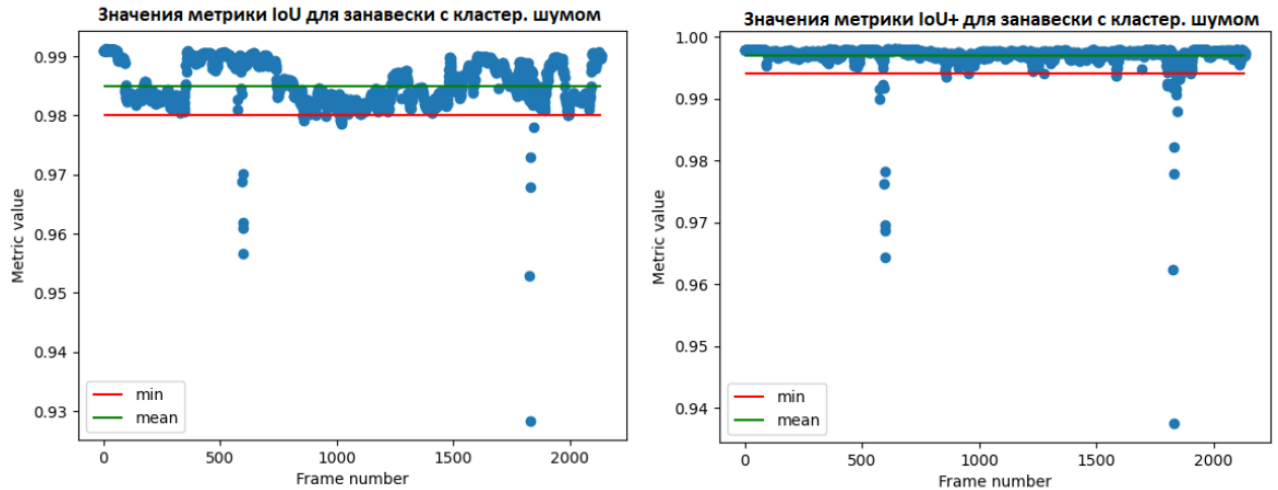


Рис. 15. Пример визуализации значения метрик MAE, MAE+, IoU, IoU+ для фонового изображения занавески с кластеризованными шумами для желтого варианта одежды актера для алгоритма Nvidia Broadcast.

Для дальнейшего сравнительного анализа алгоритмов было предложено использование двух характеристик - среднее значение (показано зеленой линией) и специальное “минимальное” значение (показано красной линией). Среднее значение бралось для оценки характерного качества работы алгоритмов. Минимальное значение необходимо для выявления худших случаев работы алгоритмов, поскольку именно худшие случаи наиболее сильно снижают оценку качества работы алгоритма. При визуальном анализе наихудших результатов были замечены характерные дефекты-выбросы, пример которых представлен на рисунке 16. Такие дефекты встречаются редко, но очень сильно ухудшают количественный результат. Для того, чтобы оценить часто встречающиеся наихудшие результаты было предложено отсечь два процента наихудших значений. Таким образом, минимальные значения каждого из анализируемых алгоритмов брались с двухпроцентным отсечением дефектов-выбросов.



Рис. 16. Пример дефекта-выброса с точностью 0.987 по метрике IoU+ для алгоритма UNet.

2.3.2 Анализ результатов.

В результате получилась сводная таблица результатов для всех рассматриваемых алгоритмов на предложенном наборе данных. Таблица содержит в себе средние и минимальные значения метрик для предложенных бытовых условий. Кроме того, таблица содержит производительности алгоритмов в виде возможности их работы на CPU и количества обрабатываемых кадров в секунду. Пример фрагмента получившейся таблицы изображен на рисунке 17.

Датасет	Произв-ть	GPU	FPS	Занавеска, кластер.				Занавеска, гаусс.				Занавеска, ярко, кластер.				Занавеска, ярко, гаусс.				Занавеска, тускло, кластер.				Занавеска, тускло, гаусс.			
				MAE	MAE+	IoU	IoU+	MAE	MAE+	IoU	IoU+	MAE	MAE+	IoU	IoU+	MAE	MAE+	IoU	IoU+	MAE	MAE+	IoU	IoU+	MAE	MAE+	IoU	IoU+
U2Net	+	30	Ср	0.743	0.748	0.971	0.983	0.73	0.735	0.94	0.964	0.735	0.741	0.968	0.98	0.725	0.73	0.965	0.978	0.742	0.747	0.972	0.984	0.728	0.733	0.968	0.98
			Мин	0.636	0.641	0.925	0.94	0.627	0.631	0.934	0.936	0.628	0.632	0.855	0.861	0.623	0.627	0.855	0.868	0.633	0.637	0.928	0.942	0.627	0.631	0.882	0.892
MediaPipe Selfie	-	60	Ср	0.943	0.953	0.959	0.968	0.942	0.952	0.954	0.963	0.94	0.951	0.954	0.963	0.94	0.95	0.951	0.96	0.943	0.953	0.958	0.967	0.941	0.951	0.954	0.963
			Мин	0.88	0.886	0.751	0.758	0.873	0.879	0.711	0.719	0.868	0.874	0.691	0.696	0.867	0.875	0.693	0.697	0.88	0.888	0.74	0.746	0.866	0.872	0.721	0.728
UNet-Teraus	+	30	Ср	0.989	0.998	0.991	0.998	0.989	0.998	0.991	0.998	0.989	0.999	0.992	0.998	0.988	0.998	0.991	0.998	0.989	0.999	0.992	0.998	0.988	0.998	0.991	0.998
			Мин	0.979	0.991	0.987	0.998	0.978	0.991	0.987	0.998	0.979	0.992	0.988	0.998	0.977	0.989	0.987	0.998	0.979	0.992	0.988	0.998	0.978	0.999	0.987	0.998
MODNet	+	35	Ср	0.71	0.715	0.981	0.984	0.717	0.722	0.98	0.993	0.703	0.708	0.981	0.994	0.711	0.716	0.98	0.993	0.709	0.714	0.981	0.994	0.716	0.721	0.98	0.993
			Мин	0.588	0.591	0.974	0.99	0.615	0.619	0.972	0.988	0.561	0.563	0.974	0.988	0.597	0.601	0.972	0.987	0.584	0.587	0.973	0.99	0.613	0.617	0.972	0.988
Nvidia	+	60	Ср	0.983	0.994	0.985	0.997	0.983	0.994	0.986	0.997	0.982	0.993	0.985	0.997	0.983	0.994	0.985	0.997	0.983	0.994	0.985	0.997	0.983	0.994	0.986	0.997
			Мин	0.971	0.985	0.98	0.994	0.971	0.986	0.981	0.995	0.969	0.983	0.979	0.992	0.97	0.984	0.98	0.992	0.97	0.984	0.98	0.993	0.971	0.985	0.98	0.994
RVM	+	60	Ср	0.975	0.987	0.981	0.995	0.976	0.987	0.982	0.995	0.974	0.986	0.981	0.994	0.975	0.987	0.981	0.994	0.975	0.987	0.981	0.994	0.975	0.987	0.982	0.995
			Мин	0.959	0.976	0.973	0.99	0.96	0.977	0.974	0.991	0.956	0.973	0.972	0.989	0.958	0.975	0.972	0.989	0.958	0.975	0.973	0.99	0.96	0.977	0.973	0.991
Датасет	Произв-ть			Собаки, кластер.				Собаки, гаусс.				Собаки, ярко, кластер.				Собаки, ярко, гаусс.				Собаки, тускло, кластер.				Собаки, тускло, гаусс.			
U2Net	+	30	Ср	0.767	0.773	0.973	0.984	0.75	0.756	0.972	0.983	0.766	0.771	0.974	0.985	0.745	0.75	0.971	0.983	0.765	0.771	0.973	0.985	0.749	0.754	0.971	0.983
			Мин	0.666	0.671	0.926	0.939	0.644	0.649	0.921	0.936	0.653	0.657	0.924	0.935	0.634	0.638	0.919	0.931	0.659	0.663	0.925	0.937	0.639	0.643	0.923	0.938
MediaPipe Selfie	-	60	Ср	0.941	0.952	0.963	0.974	0.944	0.954	0.962	0.973	0.944	0.954	0.964	0.975	0.946	0.956	0.964	0.975	0.942	0.952	0.964	0.975	0.944	0.954	0.963	0.974
			Мин	0.89	0.899	0.921	0.934	0.895	0.905	0.915	0.93	0.892	0.901	0.921	0.935	0.897	0.907	0.921	0.935	0.891	0.9	0.922	0.935	0.895	0.905	0.917	0.931
			Ср	0.987	0.997	0.991	0.998	0.986	0.994	0.981	0.994	0.987	0.997	0.981	0.994	0.988	0.995	0.991	0.998	0.987	0.997	0.991	0.998	0.986	0.995	0.991	0.998

Рис. 17. Сводная таблица результатов детального тестирования рассматриваемых алгоритмов на предложенном наборе данных.

Ввиду плохой наглядности полученных результатов, было принято решение по их визуализации в виде графиков. Были построены 30 графиков, набор из шести которых показывает качество работы конкретного алгоритма при различных фоновых и бытовых условиях на выбранном варианте одежды актера. Для построения конкретного графика использовались свечные графики, где “ценой” закрытия выступало среднее значение, полученное в результате работы алгоритма, а “ценой” открытия - специальное минимальное значение.

На рисунке 18 изображены результаты работы алгоритмов в фиксированном диапазоне [0.9; 1.0] по метрике IoU+ для первого варианта одежды актера. На рисунках 28, 29, 30, 31 приложения 1 изображены аналогичные результаты для других вариантов одежды актера.

Столбцы красного цвета показывают результаты работы алгоритма на фоне с колышущейся занавеской. Столбцы синего цвета - с бегающими на заднем фоне собаками. Для столбцов зеленого цвета тестирование происходило на зеленом фоне, а для желтых столбцов использовался фон офиса с людьми. Для столбцов розового цвета использовался желтый статичный фон похожий на цвет одежды актера. Столбцы серого цвета показывают результат работы алгоритма на реальном фоне, сфотографированном в кабинете автора.

При этом, каждая группа из шести разных фонов для некоторого алгоритма тестировалась при разных бытовых условиях, подписанных по оси X: кластеризованные либо гауссовские шумы с ярким либо тусклым локальным освещением.

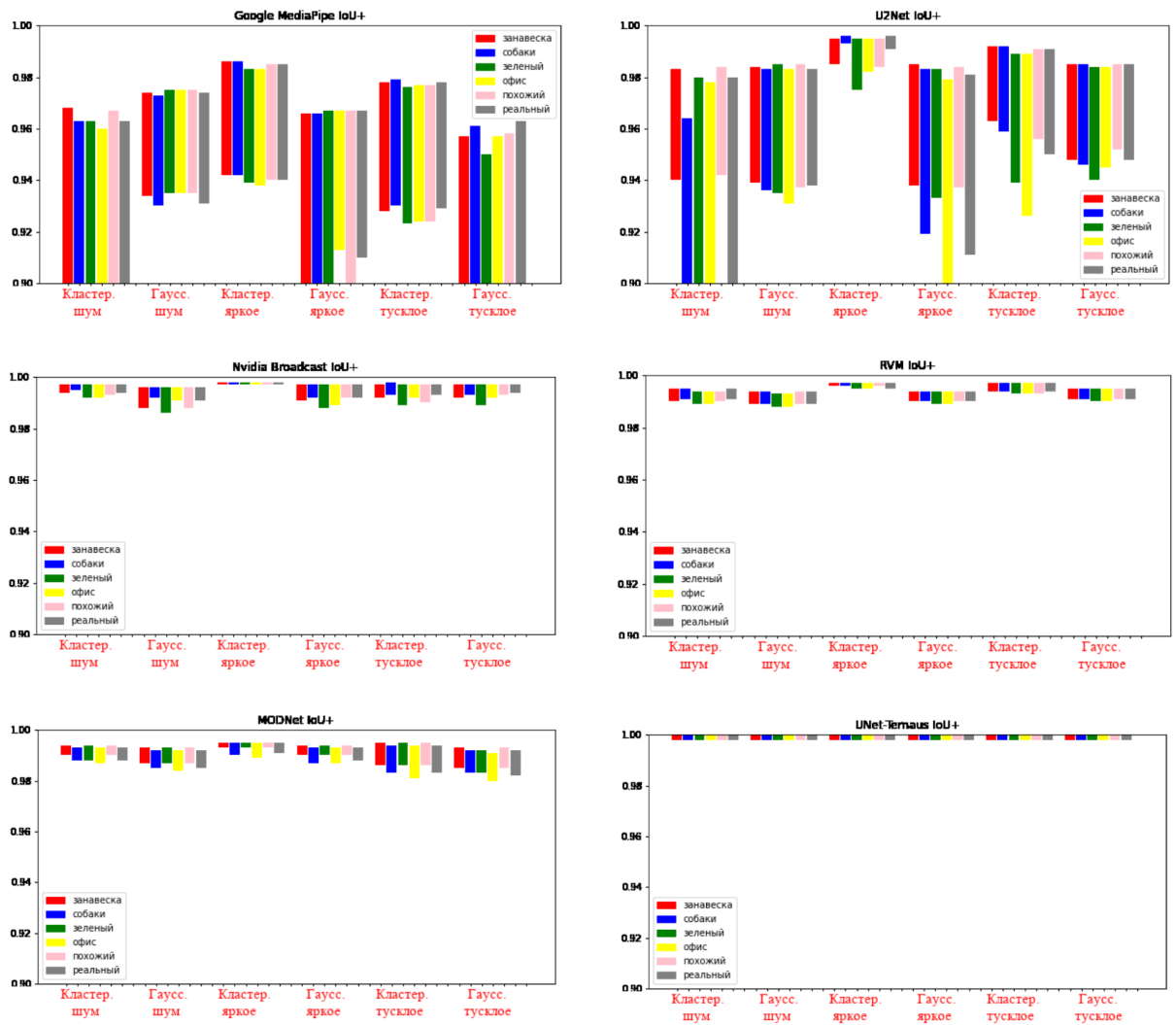


Рис. 18. Результат тестирования алгоритмов на предложенных тестовых данных для первого варианта одежды актера.

2.3.2.1 Исследование применимости алгоритмов.

В результате визуального анализа результатов можно определить наилучший по средним и минимальным значениям алгоритм - UNet. Тем не менее у каждого из рассматриваемых алгоритмов есть свои достоинства и недостатки. Следовательно необходимо сформулировать рекомендации к применению некоторого алгоритмов в зависимости от нужд пользователя.

- При использовании алгоритма UNet, ввиду специфики обучения, возможно получить только бинарную версию маски актера. Количество обрабатываемых кадров в секунду на видеокарте RTX 3060 Laptop с

входным и выходным разрешением 800x800 изображения не превышает тридцати. Кроме того, отсутствует готовое SDK решение для использования алгоритма. Тем не менее, для работы может использоваться карта любой серии и производителя, а получаемые результаты по качеству превосходят другие алгоритмы практически во всех протестированных ситуациях. Данное решение в большей степени подходит для разработчиков, пытающихся внедрить высококачественное выделение актера в существующие системы.

- Выделение актера с использованием алгоритма Nvidia Broadcast по качеству является одним из лучших. Кроме того, алгоритм имеет очень высокое быстродействие (не менее 60 кадров в секунду с разрешением 1920x1280). Также имеется готовое приложение для использования и возможность создания виртуальной камеры с изображением модифицированного фона. Тем не менее, для использования описанного решения необходима видеокарта производителя Nvidia серии 20+. Также минусом является то, что для разработчиков представлено только API на языке C. Описанное решение по выделению актера в большей степени подходит для пользователей, обладающих хорошим оборудованием, часто жестикулирующих и желающих получить отличное качество выделения как силуэта актера, так и его пальцев путем установки готового SDK.
- Алгоритм выделения актера GoogleMeet MediaPipe может использоваться на любом оборудовании с количеством кадров в секунду больше шестидесяти, с разрешением выходного изображения 256x256. Однако, данный алгоритм хорошо выделяет только поясничную часть актера, а руки и пальцы часто сегментируются неверно. Данный алгоритм используется в службе видеосвязи Google Meet, а также имеется API на языке Python, C, JavaScript и

следовательно, на основе этого решения можно с легкостью создать виртуальную камеру. Описанное решение подойдет для пользователей с разной квалификацией, не имеющих специализированного оборудования и во время проведения онлайн-трансляций редко жестикулирующих.

- Алгоритм выделения актера с использованием нейронной сети MODNet обеспечивает среднее качество выделения актера (по сравнению с другими сравниваемыми алгоритмами) с разрешением 512x512. Количество кадров, обрабатываемых в секунду на видеокарте 1080Ti, по словам авторов составило 67, в случае RTX 3060 Laptop - 35. Получаемая маска является матированной. Код с реализованной архитектурой на языке Python полностью доступен для использования. Таким образом, данное решение подойдет для продвинутых пользователей с хорошей видеокартой, желающих получить как хорошее качество выделения актера и рук, так и, ввиду матированной маски, детализированную маску волос.
- С помощью алгоритма Robust Video Matting (RVM) возможно получение хорошей маски выделенного актера с разрешением 1280x720 и количеством кадров в секунду около шестидесяти. Особенностью данного алгоритма является возможность обработки изображений разрешением 3840x2160. Полученная маска является матированной, что позволяет детализировать волосы. Несмотря на то, что отсутствует готовое приложение, создающее виртуальную камеру с результатом, существует реализованная версия описанной архитектуры на языке Python и C++, что делает возможным встраивание этого решения в существующие. Таким образом, представленное решение во всех аспектах превосходит схожее с ним - MODNet и подойдет как для продвинутых пользователей с хорошей видеокартой, желающих

получить хорошее качество выделения актера, рук и волос, так и для разработчиков, желающих внедрить это решение в свои сервисы интернет-трансляций, ввиду открытой лицензии.

- Решение по выделению актера с архитектурой U^2 Net позволяет обработать около тридцати масок актера с разрешением 320x320 в секунду на видеокарте RTX 3060 Laptop. Ввиду того, что архитектура сети предназначена для выделения наиболее значимых объектов в кадре, результаты выделения актера по минимальным значениям получились удовлетворительными только при игнорировании областей рук, так как по мнению авторов они не принадлежат актеру (ввиду специфики обучения). Таким образом, данный алгоритм не рекомендуется к использованию при текущих значениях обученных весов, Тем не менее, существует вероятность повторного анализа алгоритма при переобучении предложенной архитектуры сети на наборе данных с руками.

Таким образом, можно выделить 4 принципиально различных решения по выделению актера с использованием GPU, дающих неплохие результаты. На рисунке 19 показан результат итогового сравнения лучших алгоритмов на метрике IoU+ на фоновом изображении с бегущими животными.

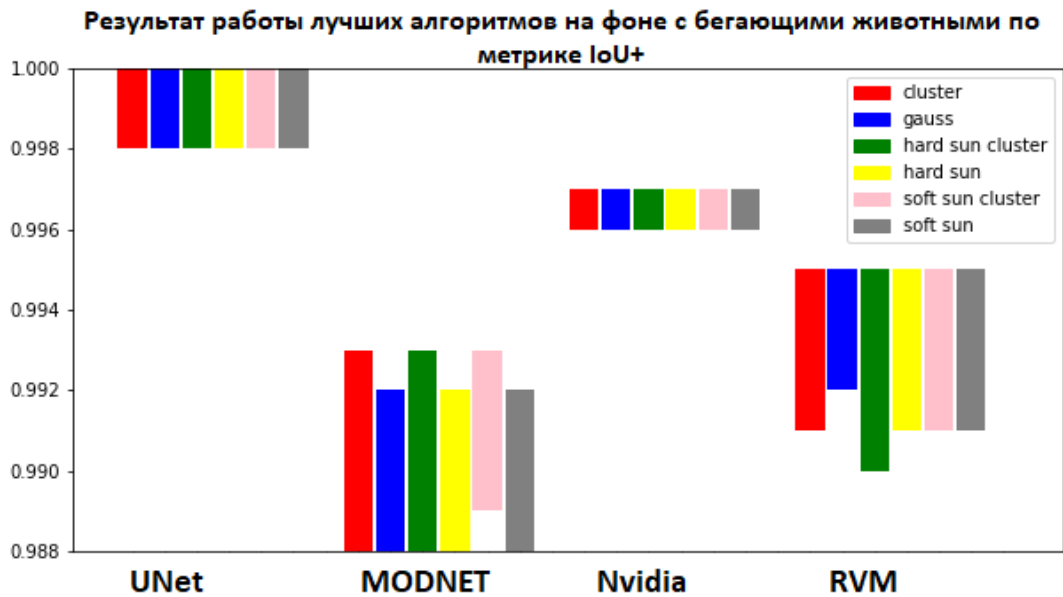


Рис. 19. Сравнение четырех лучших алгоритмов разделения актер-фон на наборе данных с бегающими животными для всевозможных бытовых условий.

Несмотря на отличные результаты, необходимо проанализировать двухпроцентные дефекты-выбросы, которые отбрасывались при подсчете минимальных значений метрик.

2.3.3 Анализ двухпроцентных дефектов-выбросов.

На первом этапе анализа двухпроцентных дефектов-выбросов происходила визуальная оценка изображений с характерными дефектами для выбранного алгоритма с определенным фоновым изображением при различных бытовых условиях. Выявленный анализ показал практическую независимость дефектов от бытовых условий и было принято решение анализировать дефектные изображения только для различных вариантах фонов и актера.

Из двухпроцентных дефектов-выбросов выбирались минимальные значения метрики IoU для каждого из алгоритмов при конкретном фоне, после чего полученные изображения визуально анализировались. В

результате проведенного анализа были выявлены характерные дефекты для каждого из алгоритмов, изображенные на рисунке 20.



Рис. 20. Характерные дефекты для четырех алгоритмов выделения актера для различной одежды актера на разных фоновых изображениях:

UNet, Nvidia Broadcast, MODNet, RVM слева направо.

При визуальном анализе было выявлено, что для алгоритма UNet характерными дефектами являются области около пальцев, а выбросами являются замкнутые области в результате объединения рук, как показано на рисунке 24 в первом столбце.

Для алгоритма Nvidia Broadcast выбросом является захват области фонового изображения, а к дефектам можно отнести особенность выделения области вокруг актера, в том числе и кистевой.

Алгоритм MODNet дает неприемлемые выбросы - такие как пропадание рук, кистей и пальцев. Иногда происходит захват значительной области фонового изображения.

Для алгоритма Robust Video Matting (RVM) характерны дефекты в виде пропадания кистевых областей рук, а в некоторых случаях и пальцев.

Таким образом, не смотря на то, что в среднем существующие на рынке решения позволяют достичь отличных результатов по выделению актера, существуют дефекты-выбросы, которые значительно портят визуальное восприятие реальности замены фона.

3. Разработка собственного алгоритма

Поскольку существующие на рынке решения не позволяли получить оптимальное качество было предпринято множество попыток по улучшению существующих решений с целью увеличения их качества.

3.1. Попытки улучшить существующие решения.

3.1.1 Улучшение классических алгоритмов.

3.1.1.1 Комбинирование дифференциального кеинга с оптическим потоком.

Изначально предпринимались попытки по улучшению алгоритма дифференциального кеинга. Это связано с тем, что как было отмечено выше, алгоритм дифференциального кеинга в теории должен работать хорошо, однако на практике, из-за шумов показывает неудовлетворительные результаты. Напротив, алгоритм распознавания движения (с помощью оптического потока) устойчив к шумам, но при исчезновении движения (остановке актёра) маска актёра пропадает. Таким образом, был предложен алгоритм заключающийся в комбинации этих методов. Основная идея заключается в том, что двигаться в кадре может только актер, поэтому любое движение автоматически означает, что это актер, а не фон.

На первом этапе для устранения шумов и гарантированного выделения фона пороговое значение дифференциального кеинга было максимизировано (до полного удаления фона). При этом на актёре неизбежно появлялись “дырки”, вызванные схожестью цветов фона и актёра (поскольку пороговое значение слишком большое). На втором этапе для устранения “дырок” использовалась информация о движении - все пиксели, в которых есть ненулевой вектор движения от предыдущего кадра к текущему, считаются актёром. При этом для исключения неверной классификации при остановке

актера для тех пикселей, где движения нет, использовалось значение маски из предыдущего кадра.

Предложенный алгоритм был реализован на языке C++ с использованием алгоритма Фарнбека [22] для обнаружения движения. Однако, на практике этот алгоритм оказался неприменим из-за возникновения эффекта оконтуривания вокруг актера (рисунок 21).

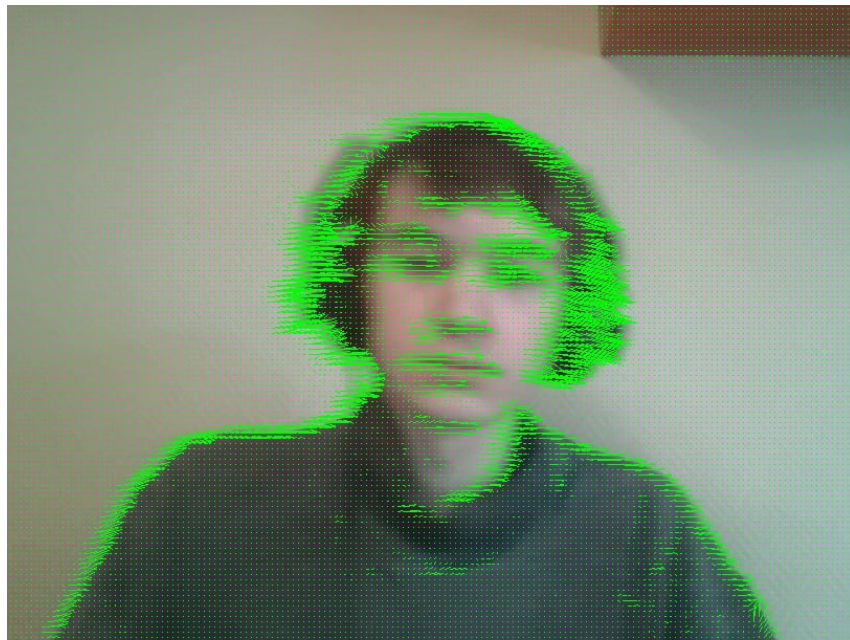


Рис. 21. Эффект оконтуривания вокруг актера, возникающий при использовании алгоритма анализа оптического потока Фарнбека.

Данный эффект связан с особенностью алгоритма Фарнбека - для определения движения используется пара соседних по времени кадров и в тех пикселях, где в момент времени t находился актер, а в момент времени $t+1$ фон возникает “движение”, хотя в этих пикселях актер уже отсутствует. Симметричная проблема возникает при использовании $t-1$ и t кадров.

Для решения этой проблемы было предложено брать 3 кадра: в моменты времени $t-1$, t , $t+1$; и отрисовывать кадр t , а маску движения отрисовывать лишь там, где имеется движение между $(t-1, t)$ и $(t+1, t)$ кадрами соответственно.

Это частично помогло устранить проблему оконтуривания, однако вектора сдвигов, ввиду взятия трех кадров, а также шумовой составляющей - определялись неточно. В результате возникают дефекты вдоль контура актера, что делает разработанный алгоритм неприемлемым. Ввиду отсутствия идей и результатов, дальнейшие попытки по комбинированию предложенных методов не предпринимались.

3.1.2 Улучшение алгоритмов, основанных на методах нейронной сети.

3.1.2.1 Дообучение одних сетей на наборе данных других.

На основе информации о характерных видах дефектов, полученной из п.2.3.3, предпринимались попытки улучшения алгоритмов, работающих с использованием нейронных сетей.

Изначально возникло предположение, что дообучение одной нейронной сети на обучающих данных другой приведет к исключению дефектов-выбросов ввиду большего разнообразия актеров. Для дообучения были выбраны нейронные сети с открытым исходным кодом: UNet, RVM, MODNet, U^2 Net. После чего, проанализировав наборы данных, на которых происходило обучение выбранных сетей, было замечено, что в обучающих данных для сетей UNet и U^2 Net практически отсутствовали люди с руками и пальцами. При обучении остальных сетей использовались всевозможные варианты актеров. Таким образом, среди кандидатов на дообучение (UNet и U^2 Net) была выбрана сеть с архитектурой UNet. Этот выбор связан с высоким качеством выделения актера, выявленном при проведенном тестировании. Дообучение проводилось на наборе данных сети RVM. Набор данных, на которых происходило обучение сети RVM, состоит из изображений актеров на черном фоне с соответствующими им маскам, в разрешении SD либо HD. Суммарный объем пар изображение+маска составил 6 и 60 гигабайт

соответственно. При обучении сети RVM происходило замена черного фона другим (из пятисот возможных).

Дообучение нейронной сети UNet с кодером `timm-efficientnet-b0` [23] происходило на разрешении HD (1920x1080), масштабированным в 800x800. Продолжительность обучения составила 240 часов на видеокарте Quadro 5000. Дообучение происходило для всех весов сети с оптимизатором AdamP со скоростью обучения $lr = 0.0001$. Размер “батча” составил 8 для тренировки и валидации. В качестве метрики использовалась IoU, а в качестве функции потерь выступала комбинация в виде Jaccard Loss [24] и Binary Focal Loss [25] с коэффициентами 0.1 и 0.9 соответственно. По прошествии 30 эпох валидационная метрика IoU вышла на постоянное значение, и спустя 50 эпох было принято решение по прекращению дообучения сети.

После дообучения методами визуальной оценки на изображениях, для которых случались дефекты-выбросы для исходной сети, происходило сравнение результатов с дообученной сетью. Несмотря на то, что замкнутые области теперь сегментируются верно, существенного улучшения выявлено не было. По-прежнему оставались дефекты на пальцах, а также появились дефекты по контуру актера, которых раньше не было. Пример полученного изображения изображен на рисунке 22.



Рис. 22. Результирующее изображение после дообучение сети UNet на обучающих данных RVM.

3.2. Разработанный оптимизированный алгоритм.

3.2.1 Описание разработанного алгоритма.

В процессе анализа дообученных алгоритмов и в процессе выполнения дообучения появилась идея дообучать алгоритмы на конкретном фоне, поскольку дообучение сети на относительно небольшом наборе изображений занимает не очень много времени. Идея алгоритма заключается в дообучении нейронной сети на серии изображений конкретного актера на конкретном фоне.

Для этого на первом этапе происходит подготовка обучающей выборки актера. Для получения этой выборки происходит видеосъемка конкретного актера в движении, характерном при проведении видеоконференций, в различной одежде на зеленом фоне (рисунок 23). Этот процесс аналогичен описанному в пункте 2.2.3. В результате, по снятому видео методами «хромакея» получается образцовая маска.



Рис. 23. Первый этап. Пример съемки актера на зеленом фоне.

На втором этапе, перед началом трансляции вещающая камера снимает фон без актера. После чего происходит замена зеленого фона сфотографированным для всех изображений, полученных на первом этапе, и затем, аналогично пункту 2.2.2 эмуляция бытовых условий. Затем по серии

полученных комбинированных изображений выполняется дообучение нейронной сети.

Руководствуясь результатами, полученными при сравнении алгоритмов, в качестве дообучаемой сети использовалась сеть с архитектурой UNet. Время, необходимое для дообучения этой сети варьируется в зависимости от мощности видеокарты, но в среднем для достижения стабильно отличных результатов достаточно дообучения, продолжительностью не более 15 минут на видеокарте уровня RTX 3060 Laptop. В зависимости от желаемого качества продолжительность обучения может быть уменьшена или увеличена.

3.2.2 Тестирование разработанного алгоритма.

Описанный выше алгоритм был реализован на языке Python и протестирован на наборе данных, полученных в главе 2. Дообучение происходило на конкретно выбранном фоне для трех вариантов одежды актера, на видеокарте RTX 3060 Laptop в течение 15 минут. Тестирование производилось на четвертом варианте одежды для каждого из фонов.

На рисунке 24 изображены результаты сравнения разработанного алгоритма (VanNet) по сравнению с четырьмя, выявленными раньше (лучшими решениями по метрике IoU на фоновом изображении с занавеской). Предпочтение метрики IoU вместо IoU+ связано с неотличимостью результатов алгоритмов UNet и разработанного алгоритма (VanNet). В результате можно видеть характерное преимущество разработанного алгоритма по средним и минимальным значениям. Аналогичные результаты были получены на других вариантах фонового изображения.

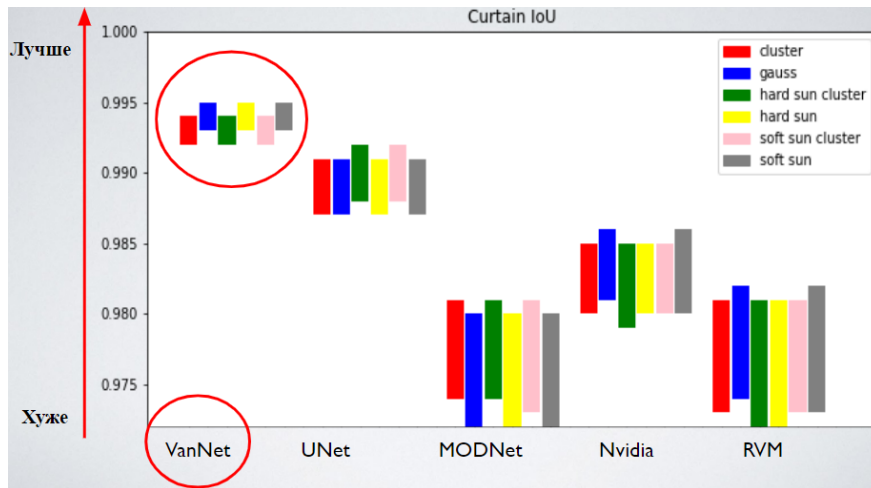


Рис. 24. Сравнение разработанного алгоритма (VanNet) с существующими лучшими решениями по метрике IoU на фоновом изображении занавески.

На рисунке 25 представлены графики ошибок работы дообученного алгоритма по метрикам IoU и IoU+ для одного из вариантов фонов и актера.

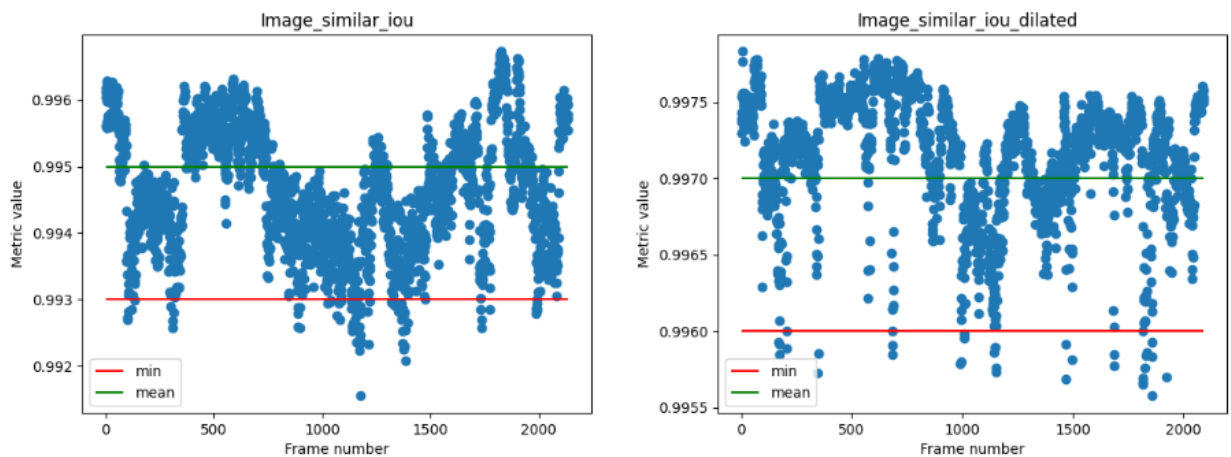


Рис. 25. Пример визуализации значения метрик IoU и IoU+ для фонового изображения кабинета автора.

Из рисунка видно, что на графике нет характерных дефектов-выбросов. В результате проведенного анализа было обнаружено, что в модифицированной (дообученной) версии алгоритма отсутствуют существенные выбросы - абсолютное минимальное значение метрик не опускается ниже 99%, в то время как раньше оно падало до 97%.

На рисунке 26 представлен кадр с минимальным значением метрики IoU.

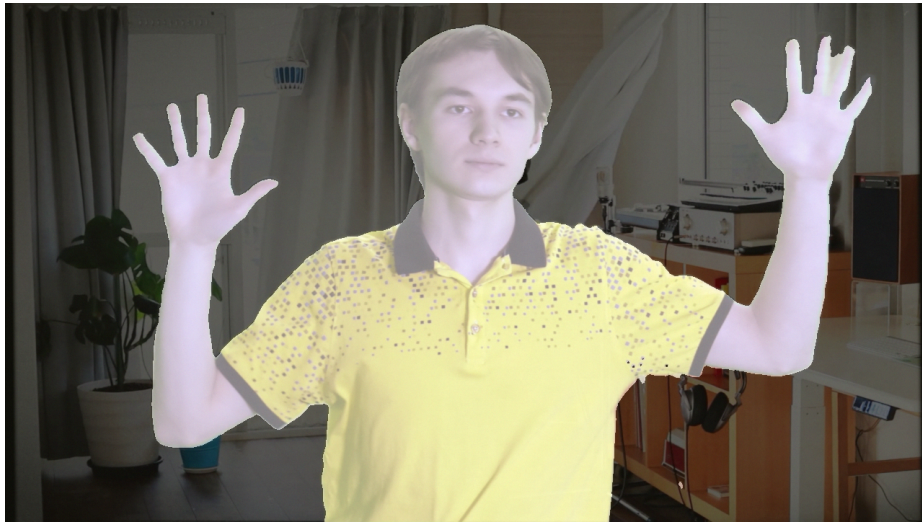


Рис. 26. Характерные дефекты для разработанного алгоритма VanNet.

Характерными дефектами являются пальцы рук, но они не столь заметны, как раньше (на рисунке “слиплись” средний и безымянный пальцы), и использование постпроцессинга в виде билатерального фильтра позволяет достичь качественно новых результатов.

Для проведения тестирования в реальных условиях был сфотографирован фон без актера и в течении 15 минут производилось дообучение нейронной сети на полученных скомбинированных изображениях. Затем была произведена видеосъемка актера в одежде, отличной от одежды, в которой актер снимался для образцовых роликов, в движении, за рабочим местом, и это видео было обработано дообученным алгоритмом. Результирующее видео с выделенным актером изображено на рисунке 27 (характерный кадр).



Рис. 27. Пример кадра из видео с результатом работы разработанного алгоритма VanNet в реальных условиях.

Визуально сравнив результаты, полученные с помощью разработанного алгоритма VanNet, с результатами, полученными с использованием алгоритма UNet, можно отметить существенное отличие как в области пальцев, так и в целом. При анализе двух видеопоследовательностей экспертной группой, состоящей из 5 человек, во всех случаях предпочтение было отдано разработанному алгоритму. Таким образом, при тестировании в реальных условиях была получена высокая точность разделения актер/фон, несмотря на то, что использование предлагаемого алгоритма требует дополнительного времени для подготовки к конкретным условиям съемки.

Заключение

В ходе выполнения дипломной работы были получены следующие результаты:

- Произведен обзор распространенных методов и алгоритмов выделения актера. Изучены существующие алгоритмы на основе алгоритмов вычитания фона и с использованием нейронных сетей.
- Исследованы особенности съемки актера при проведении интернет-трансляций и видеоконференций: произведена оценка качества изображения, уровня шумов, стабильности фона и освещения. Создан набор тестовых данных с эмуляцией бытовых условий.
- Разработана система тестирования и протестирован 21 алгоритм выделения актера. Сравнение проводилось в два этапа - на первом этапе было отсеяно 15 алгоритмов из-за неприемлемо высоких ошибок. На втором этапе, на 24-х сериях изображений были протестированы шесть алгоритмов, основанных на нейронных сетях.
- Проведено исследование и выбор оптимальных метрик для сравнения алгоритмов. Разработана модифицированная версия этих метрик - без учета граничной области вокруг актера. Определены преимущества и недостатки сравниваемых алгоритмов. Сформированы рекомендации по их использованию.
- Безуспешно предприняты попытки модификации алгоритмов дифференциального кеинга для работы в условиях бытовых съемок. Предприняты попытки по улучшению существующих алгоритмов с использованием технологии нейронной сети.
- Разработан оптимизированный алгоритм, существенно улучшивший качество разделения актер-фон в сложных случаях. В результате качество выделения актер фон возросло на 0.5% в среднем случае и на

4% в худших случаях (по разработанной метрике IoU Dilated). Визуальная оценка показала преимущества разработанного алгоритма в реальных условиях съемок.

- Работа представлена на международной студенческой конференции, где получила диплом III степени.
- Определены дальнейшие планы развития работы, а именно оптимизация производительности и встраивание разработанного алгоритма в виртуальную студию All'Mix.

Список используемой литературы

1. Efficient Graph-Based Image Segmentation –
<https://people.cs.uchicago.edu/~pff/papers/seg-ijcv.pdf>
2. Normalized Cuts and Image Segmentation –
<https://people.eecs.berkeley.edu/~malik/papers/SM-ncut.pdf>
3. Segmentation and tracking of piglets in images –
<https://link.springer.com/content/pdf/10.1007/BF01215814.pdf>
4. Reliable Background Suppression for Complex Scenes –
<https://aimagelab.ing.unimore.it/imagelab/pubblicazioni/vssn61c-prati.pdf>
5. Review and Evaluation of Commonly-Implemented Background Subtraction Algorithms – <https://ieeexplore.ieee.org/document/4760998>
6. Multimedia and Signal Processing –
<https://link.springer.com/book/10.1007/978-3-642-35286-7>
7. A Bayesian computer vision system for modeling human interactions –
<https://ieeexplore.ieee.org/document/868684>
8. Efficient adaptive density estimation per image pixel for the task of background subtraction –
<https://sciencedirect.com/science/article/pii/S0167865505003521>
9. A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection –
https://researchgate.net/publication/220372111_A_fuzzy_spatial_coherence-based_approach_to_backgroundforeground_separation_for_moving_object_detection

10. Improved Adaptive Gaussian Mixture Model for Background Subtraction –
<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.4658&rep=rep1&type=pdf>
11. Improved background subtraction based on word consensus models –
<https://ieeexplore.ieee.org/document/8266565/>
12. ViBe: A Universal Background Subtraction Algorithm for Video Sequences –
https://www.researchgate.net/publication/224206851_ViBe_A_Universal_Background_Subtraction_Algorithm_for_Video_Sequences
13. Robust background subtraction in HSV color space –
https://www.researchgate.net/publication/228957857_Robust_background_subtraction_in_HSV_color_space
14. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets –
<https://arxiv.org/pdf/1606.00915.pdf>
15. Fully Convolutional Networks for Semantic Segmentation –
<https://arxiv.org/pdf/1411.4038.pdf>
16. Background Features in Google Meet –
<https://ai.googleblog.com/2020/10/background-features-in-google-meet.html>
17. MODNet: Real-Time Trimap-Free Portrait Matting via Objective Decomposition – <https://arxiv.org/pdf/2011.11961.pdf>
18. Nvidia Broadcast Video Effects SDK –

<https://developer.nvidia.com/maxine-getting-started>

19. U^2 -Net: Going Deeper with Nested U-Structure for Salient Object Detection – <https://arxiv.org/pdf/2005.09007.pdf>
20. Binary segmentation of people with UNet – https://github.com/ternaus/people_segmentation
21. Robust High-Resolution Video Matting with Temporal Guidance – <https://arxiv.org/abs/2108.11515>
22. Two-Frame Motion Estimation Based on Polynomial Expansion – <https://www.diva-portal.org/smash/get/diva2:273847/FULLTEXT01.pdf>
23. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks – <https://arxiv.org/pdf/1905.11946.pdf>
24. On Power Jaccard Losses for Semantic Segmentation – <https://www.scitepress.org/Papers/2021/103040/103040.pdf>
25. Focal Loss for Dense Object Detection – <https://arxiv.org/abs/1708.02002>

Приложение 1.

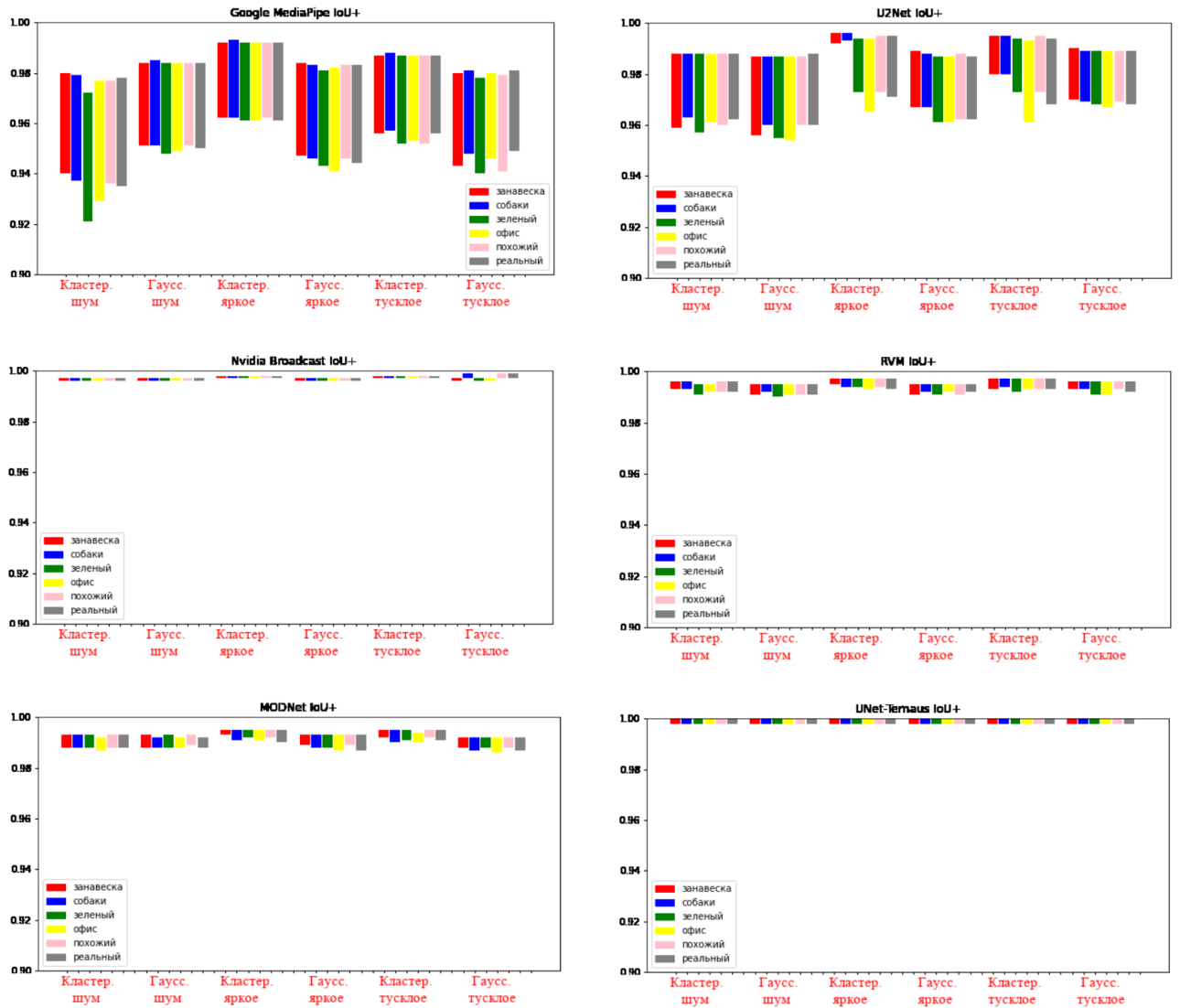


Рис. 28. Результат тестирования алгоритмов на предложенных тестовых данных для второго варианта одежды актера.



Рис. 29. Результат тестирования алгоритмов на предложенных тестовых данных для третьего варианта одежды актера.



Рис. 30. Результат тестирования алгоритмов на предложенных тестовых данных для четвертого варианта одежды актера.

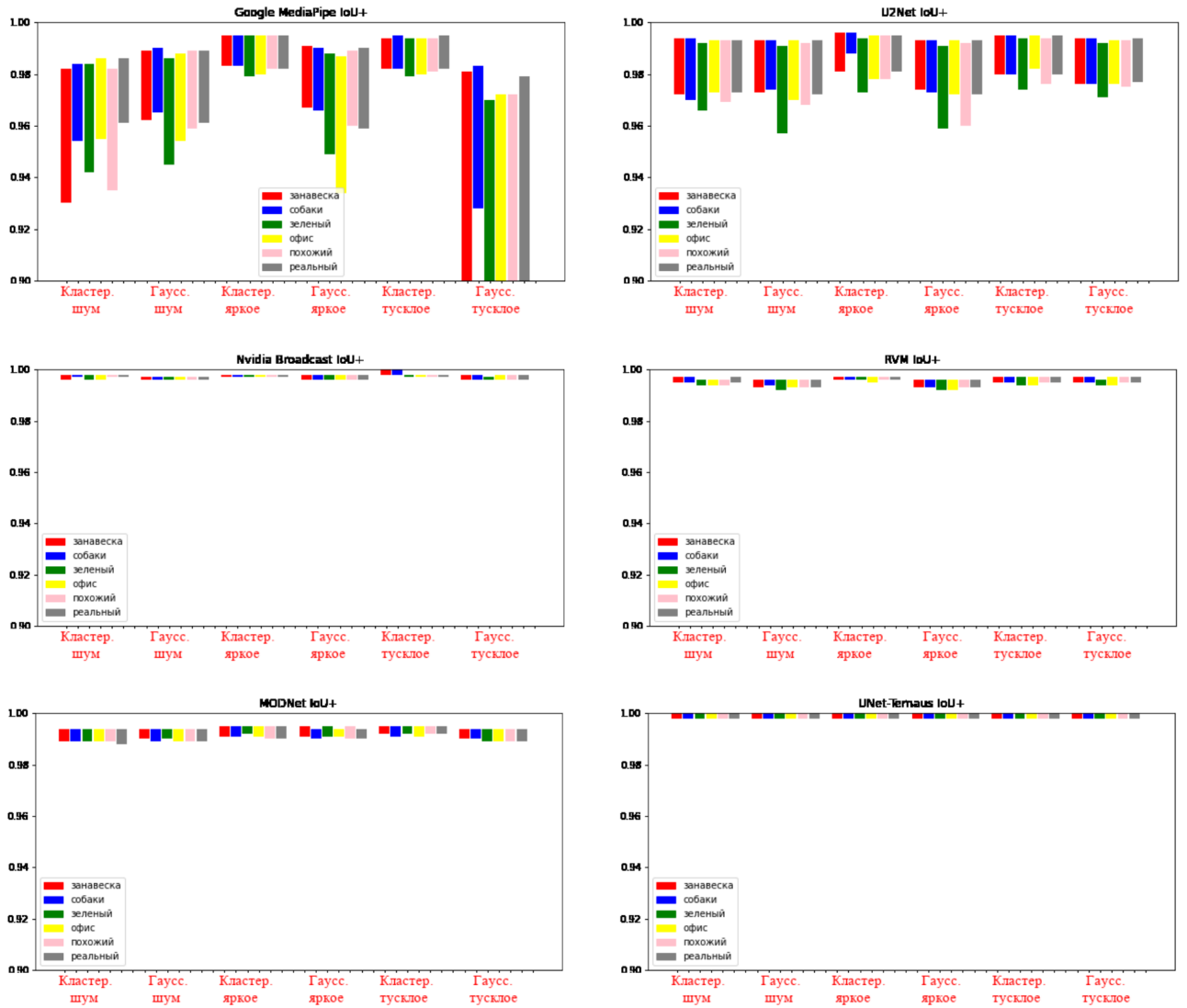


Рис. 31. Результат тестирования алгоритмов на предложенных тестовых данных для пятого варианта одежды актера.