ПОДАВЛЕНИЕ ЛОЖНОПОЛОЖИТЕЛЬНЫХ ОБНАРУЖЕНИЙ ЛИЦ В ВИДЕОПОТОКАХ СИСТЕМ ВИДЕОНАБЛЮДЕНИЯ

А.Е. Сергеев[1], А.С. Конушин[1], В.С. Конушин[2]
[1]НИУ Высшая школа экономики, Москва, Россия,
[2]ООО «Технологии видеоанализа» Москва, Россия

Аннотация

Данная работа посвящена задаче фильтрации ложноположительных обнаружений лиц людей в видеопотоках систем видеонаблюдения. Предлагается два подхода для подавления ложноположительных обнаружений в фоновых участках кадра: первый заключается в адаптации детектора под наблюдаемый видеопоток, а второй представляет собой постобработку выхода детектора за счёт анализа частоты обнаружения похожих частей кадра. Мы используем в качестве базового каскадный детектор, но метод может быть применён к другим алгоритмам. Экспериментальное оценивание показывает, что предложенные методы улучшают и точность, и полноту, при этом время работы детектора сокращается на 47 %.

Ключевые слова: детекторы, распознавание образов, анализ изображений, алгоритмы машинного зрения.

Цитирование: **Сергеев, А.Е.** Подавление ложноположительных обнаружений лиц в видеопотоках систем видеонаблюдения / А.Е. Сергеев, А.С. Конушин, В.С. Конушин // Компьютерная оптика. – 2016. – Т. 40, № 6. – С. 958-967. – DOI: 10.18287/2412-6179-2016-40-6-958-967.

Введение

Алгоритм выделения лиц людей, или детектор лиц, является одним из важных элементов в разнообразных системах интеллектуальной видеообработки. Существующие решения этой задачи опираются на применение алгоритма выделения лиц в изображениях к отдельным кадрам видеоряда. Современные алгоритмы выделения лиц в изображениях обладают высокими характеристиками и широко используются в промышленных системах, например, для индексации изображений в поисковых системах и социальных сетях. Однако качество изображений лиц людей в видеопотоке с камер видеонаблюдения заметно ниже, чем в обычных изображениях с фотоаппаратов. Поэтому применение алгоритмов выделения лиц к видеоряду приводит либо к регулярным пропускам людей, либо к заметному числу ложноположительных срабатываний.

Для решения этой проблемы было предложено несколько подходов. Первый основан на сопоставлении или сопровождении лица человека между кадрами, что позволяет оценить правдоподобность пути перемещения объекта [1] и отфильтровать слишком короткие или физически невозможные траектории. Второй подход предлагает модифицировать алгоритмы выделения лиц специально для работы с видеорядом. Помимо признаков изображения, предлагается использовать информацию о движении, извлекаемую напрямую из видеоряда [2]. Самый популярный пример такого признака – оптический поток, описывающий движение точек между соседними кадрами [3]. Третий подход основан на использовании информации о геометрической конфигурации сцены. Если камера откалибрована и известно её положение относительно сцены, тогда можно наложить ограничения на допустимые размеры объектов и отфильтровать часть ложноположительных обнаружений [4]. В четвёртом подходе предлагается фильтровать обнаружения на основе дополнительной информации. Например, использовать алгоритм выделения движущихся объектов на основе вычитания фона. Такие алгоритмы строят маску переднего плана, определяя для каждого пикселя кадра, принадлежит он фону или движущемуся объекту. Для этого обычно анализируется и моделируется распределение значений цвета пикселей в предположении сохранения цвета фона и его преобладания в течение времени. Выделенные лица считаются ложноположительными срабатываниями, например, если доля пикселей переднего плана ниже заданного порога.

Все подходы, за исключением второго, применяются дополнительно к детектору лиц и не зависят от него. Поэтому их можно использовать совместно для любого существующего детектора лиц.

В данной работе предлагается два метода подавления ложноположительных срабатываний. Первый заключается в дообучении детектора на примерах, автоматически собираемых из видеоряда. Второй метод относится к четвёртому из рассмотренных подходов – фильтрации обнаружений за счёт дополнительной информации, а именно анализа частоты обнаружения похожих частей кадра.

1. Детектор лиц

Современные алгоритмы выделения объектов можно условно разделить на три основные группы: каскадные детекторы [5, 6, 7], детекторы на основе деформируемой модели частей объекта [8, 6, 9] и нейросетевые модели [10, 11].

Предлагаемые в данной работе методы могут применяться для любого современного детектора лиц. Для проведения экспериментальной оценки в качестве базового детектора был выбран каскадный детектор на основе интегральных признаков [7]. Использовалась реализация алгоритма Пиотра Доллара, т.к. она является свободно распространяемой и активно развивается [12 – 15]. Ниже базовый детектор будет описан более подробно.

Каскадные детекторы впервые вводятся в работе Виолы и Джонса [5]. Основными элементами алгоритма являются: метод скользящего окна для сведения за-

дачи выделения объектов к задаче бинарной классификации фрагмента изображения на объект и фон; каскад бинарных классификаторов для последовательной фильтрации гипотез; построение линейного классификатора в виде взвешенной линейной комбинации слабых признаков с помощью алгоритма бустинга [19]; новый вид признаков изображения, вычисляемых с помощью интегральных изображений в канале яркости. Работа Виола–Джонса стала одной из наиболее известных работ в области выделения объектов. Предложенный алгоритм впервые позволил решать задачу выделения лиц в видеопотоке на современных на тот момент персональных компьютерах с качеством, допускающим практическое применение.

Каскадный детектор на основе интегральных признаков [7] является развитием метода, предложенного Виолой и Джонсом. В исходном алгоритме в качестве признаков изображения использовались суммы значений яркости пикселов в заданной прямоугольной области. В используемом алгоритме признаки считаются не по 1 каналу яркости, а по 10 каналам. Три канала цвета в представлении CIE-LUV, один канал – норма градиента, и шесть каналов откликов детекторов краёв – фильтров Габора. Вместо каскада линейных классификаторов используется один линейный классификатор, но признаки в нём специально упорядочиваются, формируя «мягкий каскад» [20]. В отличие от исходного каскада завершение алгоритма может происходить после вычисления любого признака, а не по завершении одного из этапов, что снижает время работы в среднем. Также модифицируется алгоритм обучения, который включает несколько стадий поиска ложноположительных срабатываний, используемых как «сложные» отрицательные примеры (под сложностью примера здесь и дальше подразумевается то, что алгоритм допускает на нём ошибку). Суммарно всё это позволило существенно повысить характеристики алгоритма, пусть и за счёт повышения вычислительной сложности.

2. Предложенные методы

В данном параграфе описываются два предложенных метода подавления ложноположительных срабатываний. Первый заключается в адаптации базового детектора лиц под обрабатываемый видеопоток за счёт дополнительного обучения. Второй метод позволяет фильтровать обнаружения детектора за счёт анализа частоты обнаружений похожих частей кадра.

Адаптация под обрабатываемый видеопоток

Основная идея первого метода заключается в автоматическом сборе ложноположительных срабатываний базового детектора лиц и использовании их для дополнительной настройки детектора.

Ранее было предложено несколько методов адаптации детектора под обрабатываемые данные. Например, в работе [16] обнаружение с низким уровнем уверенности детектора автоматически получает увеличение уверенности, если оно оказывается похоже на какое-нибудь предыдущее обнаружение с высокой уверенностью. В работе [17] рассматривается задача анализа дорожного движения, поэтому используется допущение о том, что участники движения придерживаются похожих и прогнозируемых траекторий. В рассматриваемой нами задаче выделения лиц людей подобные допущения некорректны.

Предлагаемый нами метод работает в два этапа. На первом этапе к видепотоку применяется базовый детектор. Некоторые ложноположительные срабатывания детектора идентифицируются, и из них составляется дополнительная отрицательная выборка, специфическая для обрабатываемого видеопотока. Дополнительная выборка добавляется к исходной обучающей выборке, и детектор переобучается. Пока идет процесс дообучения, видеопоток обрабатывается исходным детектором. По завершении дообучения для обработки применяется уже обновлённый, адаптированный к видеопотоку детектор.

Идентифицировать ложноположительные срабатывания мы предлагаем следующим образом. К видеопотоку применяется метод выделения движущихся объектов на основе вычитания фона. Анализируя маску переднего плана, можно выбрать кадры, на которых отсутствуют движущиеся объекты, а значит, все срабатывания детектора на этих кадрах ложноположительные. Такой подход позволяет определить только часть ложноположительных обнаружений. Но, как показывают проведённые эксперименты, этого достаточно для того, чтобы адаптированный, дообученный детектор превосходил базовый при работе с данным видеопотоком.

Все существующие методы вычитания фона дают ошибки в отдельных пикселях практически на каждом кадре видеопотока. Обычно такие ошибочные пиксели равномерно распределены по кадру. Поэтому использовать просто число точек переднего плана для оценки наличия движущихся объектов нельзя. Поэтому кадр разбивается на блоки, и число точек переднего плана подсчитывается для каждого блока.

Псевдокод алгоритма адаптации под видеопоток.

```
Вход:
    кадры I(n), где n - номер кадра,
    исходный_детектор,
    алгоритм_вычитания_фона.
Выход:
    адаптированный_детектор.

выбрано_кадров = 0;
номер_кадра = 1;
выбранные_кадры = пустое множество {};
пока выбрано_кадров < целевое_число_кадров:
   запустить алгоритм_вычитания_фона на кадре
   I(номер_кадра);
   разбить маску регулярной сеткой на блоки
   размерами блок_высота × блок_ширина;
   вычислить максимум M точек переднего плана по
   всем блокам;
   если M > максимум_точек_переднего_плана:
      добавить I(номер_кадра) в выбранные_кадры;
```

```
        увеличить выбрано_кадров на 1;
    увеличить номер_кадра на 1;
выборка_лиц = исходная_выборка_лиц;
выборка_фона = исходная_выборка_фона;
текущий_детектор = исходный_детектор;
повторить_итерацию = истина;
пока повторить_итерацию равно истина:
    запустить текущий_детектор на выбранные_кадры;
    если нет обнаруженных лиц:
        повторить_итерацию = ложь;
        прервать выполнение итерации;
    для искажение в набор_искажений:
        если выполнение искажение невозможно:
            перейти к следующей итерации;
        добавить в выборка_фона часть кадра,
        построенную по положению обнаружения
         и после выполнения преобразования искажение;
        обучить новый детектор на выборках;
        выборка_фона и выборка_лиц;
        записать новый детектор как текущий_детектор;
вернуть в качестве адаптированный_детектор текущий_детектор;
```

Использованные параметры:

целевое_число_кадров = 50; параметр задает число кадров, которые будут использованы как источник примеров фона в рамках процесса адаптации.

блок_ширина = блок_высота = 25;

максимум_точек_переднего_плана = 70 / 625;

параметры определяют чувствительность алгоритма выбора кадров, содержащих фон, для дальнейшей адаптации. Выбор размеров блока и максимального допустимого числа точек переднего плана производился на основе оценки размеров лица в видеозаписях, а также параметров работы алгоритма вычитания фона.

набор_искажений = всевозможные комбинации из независимых искажений, которые включают: сдвиги по одной из сторон в масштабе размера обнаружения – (– 0,5; 0; 0,5); изменения масштаба: 0,9 в степенях от 1 до 4; 1,0; 1,1 в степенях от 1 до 4. Сдвиги соответствуют перемещению центра ограничивающего прямоугольника, изменения масштаба сохраняют положение центра. Как показано во многих работах, использование подобных версий обучающих примеров наравне с исходными позволяет повысить характеристики детектора [6, 8].

В наших экспериментах мы использовали в качестве алгоритма вычитания фона алгоритм ViBe [18], так как он имеет высокие показатели скорости и качества работы. Для ускорения набора сложных негативных примеров можно использовать части кадра, определенные как фон, но подход с поиском полных фоновых кадров работает стабильнее, так как точность получения полностью фонового кадра выше (то есть собирать такие кадры дольше, нежели отдельные области, но алгоритм выбора совершает меньшее число ошибок). В наших экспериментах мы использовали наборы из 50 фоновых кадров на тестовую видеопоследовательность (включающую несколько тысяч кадров). Исходный обучающий набор включает 25000 положительных и 70000 отрицательных примеров. Процесс дообучения в среднем дает дополнительные 40000 примеров фона (учитывая вышеописанный процесс генерации дополнительных образцов) и завершается приблизительно через 10 итераций.

Идейно метод рассчитан на уменьшение числа ложноположительных обнаружений посредством «знакомства» детектора с такими примерами. Но большое число примеров из наблюдаемой сцены также дает положительный побочный эффект, так как в процессе обучения каскада выбираются правила, отбрасывающие большее число примеров из негативной выборки, детектор начинает быстрее исключать из рассмотрения фоновые примеры, характерные для обрабатываемого видеопотока. Таким образом, путь негативных примеров по каскаду сокращается и уменьшается время обработки. В результате алгоритм получает бонус в виде увеличения скорости работы.

Постобработка обнаружений

Второй метод является фильтрацией выхода алгоритма выделения лиц для уменьшения числа ложноположительных обнаружений на фоновых областях кадра – в основе лежит то, что подобные обнаружения показали себя нестабильными, то есть они обнаруживались раз в три или четыре последовательных кадра. Таким образом, выбор обнаружений для удаления (фильтрации) производится посредством подсчета числа кадров, где визуально похожая область в соответствующей части кадра не была выделена как лицо.

В дополнение к адаптации под наблюдаемую сцену мы предлагаем уменьшать число ложноположительных обнаружений путем их непосредственного поиска в выходе детектора. Метод основан на следующем наблюдении: некоторые ложноположительные обнаружения регулярно появляются и пропадают на определенных областях кадра. Причиной этого может быть зашумленность изображения или изменения освещения (например, тени от проходящих людей). Иногда эти ложные обнаружения могут быть идентифицированы с помощью алгоритмов вычитания фона, но часто и они подвергаются влиянию тех же проблем. Тем не менее, область ложных обнаружений визуально меняется слабо, то есть выглядит так же, как и на предыдущих кадрах, где этого ложного обнаружения в выходе алгоритма не наблюдалось. Таким образом, мы предлагаем отслеживать устойчивость обнаружений алгоритмом фиксированных областей кадра при условии их визуальной похожести, которая должна проверяться более устойчивым к изменениям освещения алгоритмом. Для реализации идеи предлагается ввести величину, выражающую для множества собранных с разных кадров визуально похожих областей, соответствующих одному положению (допускаются небольшие смещения), соотношение между общим числом выбранных кадров к числу тех кадров, где область была выбрана алгоритмом выделения как лицо. Таким образом, целевая частота выражает устойчивость обнаружения некоторой группы визуально похожих примеров, соответствую-

щих специфическому положению в сцене. Если частота в некоторый момент работы алгоритма становится ниже заданного порога (и, значит, выделение области как лица является неустойчивым), то такой пример добавляется в базу ложноположительных примеров. И тогда для каждого кадра выход алгоритма выделения лиц сначала сверяется с базой ложноположительных примеров, для того чтобы удалить такие примеры при условии визуальной похожести в заданном положении в кадре.

Визуальная похожесть должна проверяться алгоритмом, который устойчив к влиянию разнообразного шума, который может встретиться в потоках с камер видеонаблюдения. Простой алгоритм, предполагающий порог на величину суммарной абсолютной разницы между образцами, не удовлетворяет нашему требованию. Для решения этой проблемы мы использовали методы сопоставления границ: образцы сначала конвертируются в карты градиентов (используется величина градиента без учета направления), далее карты нормализуются путем деления на максимальное значение в каждой соответствующей карте. Величина похожести формируется как сумма абсолютных разностей в нормализованных картах. Этот результат также нормируется путем деления на число пикселей, чтобы убрать влияние размера сравниваемых областей.

Алгоритм предполагает использование некоторой базы обработанных примеров. Каждый пример в этой базе представлен следующим описанием:

- число обработанных кадров, которое задает время хранения примера в базе при условии его неактивности (то есть если в заданной области кадра нет визуальных совпадений с хранимым примером), далее будет называться время_жизни;
- число визуально похожих примеров в заданной области (число_совпадений);
- число визуально похожих примеров в заданной области, которые были выделены детектором как лица людей (число_обнаружений);
- сохраненная часть кадра и ограничивающий прямоугольник, соответствующий выделенному лицу на этой части кадра;
- флаг подавления (то есть индикатор того, что пример является ложноположительным и требуется его фильтрация).

Для примеров из базы проверяется визуальное сходство с текущим обрабатываемым кадром. Если для некоторого примера наблюдается визуальное сходство, то проверяется наличие обнаружения на обрабатываемом кадре, которое находится достаточно близко к сохраненному в базу. Ограничивающие прямоугольники обнаруженных лиц считаются близкими, если значение отношения их пересечения к объединению не меньше порога, равного 50 %. Это общепринятый критерий, который имеет отношение к мере Жаккара. Порог в 50 % начал активно использоваться в PASCAL VOC [21]. Соответствующие счетчики в базе обновляются в соответствии с результатами описанных операций (визуальная похожесть и совпадение обнаружений).

Время жизни примеров введено для того, чтобы удалять неактуальные образцы и уменьшить вычислительную нагрузку. Для того, чтобы реагировать на потенциальное ложное обнаружение без задержки на сбор статистики, предлагается использовать циклический буфер предыдущих кадров.

Псевдокод алгоритма выделения лиц с примененной постфильтрацией обнаружений.

```
Вход:
  кадры I(n), где n – номер кадра,
  детектор_лиц.
Выход:
  последовательность ограничивающих прямоугольников обнаружений для каждого кадра.

отслеживаемые_примеры = пустой массив записей;
буфер_истории = массив размера размер_истории;
вершина_истории = 1;
номер_кадра = 1;
выполнять до прерывания:
  обнаружения = результат работы детектор_лиц
  на текущем кадре I(номер_кадра);
  для каждого пример в отслеживаемые_примеры:
    если пример маркирован подавляемым:
      для каждого обнаружение в обнаружения:
        если обнаружение внутри пример:
          из сохраненной области кадра примера
          вырезать часть кадра X по границам
          обнаружения обнаружение;
          при условии визуально_похожи для
          обнаружение и часть кадра X:
            удалить обнаружение из множества
            обнаружения;
            сбросить пример. время_жизни в
            значение время_жизни_подавление;
    иначе (пример не подавляется):
      вырезать часть кадра в соответствии с
      положением пример;
      при условии визуально_похожи для пример
      и вырезанная часть кадра:
        увеличить пример.число_совпадений на 1;
        сбросить пример. время_жизни в значение
        время_жизни_совпадение;
        найти максимум отношения площадей
        пересечения к объединению среди пример
        и обнаружения;
        если найденный максимум больше чем
        порог_совпадения:
          увеличить пример. число_обнаружений
          на 1;
          пометить то обнаружение, где был
          достигнут максимум, как обработанное;
        вычислить численное отношение
        пример. число_обнаружений к
        пример. число_совпадений;
        если отношение меньше порог_частоты:
          пометить пример как подавляемый;
          сбросить пример. время_жизни в
```

```
                время_жизни_подавление;
        уменьшить пример.время_жизни на 1;
    для каждого обнаружение в обнаружения:
        если обнаружение помечено обработанным:
            перейти к следующей итерации;
        добавить запись новый_пример в массив
        отслеживаемые_примеры;
        сохранить текущий кадр в новый_пример;
        сохранить ограничивающий прямоугольник
        обнаружение в новый_пример;
        установить число_совпадений = 0;
        для всех непустых ячеек кадр в буфер_истории:
            вырезать часть кадра кадр, соответствующую
            прямоугольнику обнаружение;
            при условии визуальной_похожести для
            обнаружение и вырезанной части кадра:
                увеличить число_совпадений на 1;
        новый_пример.число_обнаружений = 1;
        записать число_совпадений + 1 в поле
        новый_пример.число_совпадений;
        вычислить 1 / (число_совпадений + 1);
        если отношение меньше порог_частоты:
            маркировать новый_пример подавляемым;
            сбросить новый_пример.время_жизни в
            время_жизни_подавление;
            удалить обнаружение из обнаружения;
        иначе:
            сбросить новый_пример.время_жизни в
            время_жизни_обнаружение;
    записать текущий кадр в буфер_истории по
    индексу вершина_истории;
    обновить вершина_истории значением
    1 + остаток от деления вершина_истории на
    размер_истории;
    для каждого примера пример в массиве
    отслеживаемые_примеры:
        если пример.время_жизни = 0:
            удалить пример из отслеживаемые_примеры;
    вернуть обнаружения как результат работы
    по выделению лиц на текущем кадре;
    увеличить номер_кадра на 1;
```

Использованные параметры:

размер_истории = 10;

Параметр влияет на размер окна поиска в предыдущих кадрах при появлении нового обнаружения. Смысл заключается в следующем: для каждого обнаружения мы проверяем, является ли область обнаружения визуально похожей на соответствующие области в циклическом буфере предыдущих кадров. Стоит обратить внимание на то, что далее мы считаем отношение числа визуальных совпадений к числу обнаружений, где последнее мы принимаем за 1 (текущий кадр). Это корректно только когда любое обнаружение в диапазоне размера циклического буфера сохранено в базе. Таким образом, должны быть выполнены следующие ограничения: время_жизни_совпадение больше размер_истории и время_жизни_обнаружение больше размер_истории. Помимо этого, следует согласовывать значение порог_частоты_обнаружений и значение размер_истории так, чтобы отношение 1 / (размер_истории + 1) было меньше, чем значение порог_частоты. В противном случае поиск визуальных соответствий в буфере никогда не приведет к подавлению обнаружения.

время_жизни_совпадение = 50;

время_жизни_подавление = 150;

Первый параметр задает число кадров, в течение которых пример будет удерживаться в базе примеров (при этом пример не ассоциирован с областью ложных обнаружений). После обработки данного числа кадров при условии того, что ни на одном из них не была зафиксирована визуальная похожесть с примеров в соответствующей области кадра, пример удаляется из базы. Этот параметр определяет время обработки каждого обнаружения – в рамках этого временного окна будет проверяться отношение между тем, сколько визуально похожих регионов было найдено, и тем, на какое число из этих областей среагировал детектор. Если есть серия визуально похожих регионов, но детектор нестабильно на неё реагирует (то есть редко), то регион сохраняется как область ложного обнаружения. Мы хотим подавлять ложные обнаружения как можно дольше, но для того, чтобы не хранить примеры, переставшие быть релевантными, подавляемые примеры удаляются из базы после числа кадров, равного второму параметру (и на этих кадрах нет визуальных соответствий данным примерам).

Таким образом, значение время_жизни_подавление больше значения время_жизни_совпадение.

порог_совпадения = 0,5;

Данный параметр задает порог на отношение между площадью пересечения ограничивающих прямоугольников и площадью их объединения. Это стандартный метод (и порог) сравнения ограничивающих прямоугольников. Ограничивающие прямоугольники требуется сравнивать, так как разметка положения лиц на кадрах и выход детектора лиц является наборами прямоугольников, ограничивающих лица.

порог_частоты = 0,2;

Данный параметр задает минимальное допустимое отношение между числом обнаружений и числом визуально похожих регионов кадра. Таким образом проверяется устойчивость выхода детектора на визуально похожих областях. Если детектор дает редкий неустойчивый выход, то дальнейшие обнаружения в соответствующем месте кадра при условии визуальной похожести будут подавлены. Параметр задает уровень устойчивости, предполагается значение меньше 0,5, так как мы хотим найти те примеры, на которые детектор реагирует меньше, чем в половине случаев.

Псевдокод алгоритма проверки условия визуальной похожести двух изображений.

```
Вход:
    два изображения А и Б одинаковых размеров.
Выход:
    истина при условии похожести, иначе ложь.
```

при необходимости перевести А и Б в цветовое пространство градаций серого;
вычистить карты величин градиентов Г(А) и Г(Б);
независимо нормализовать карты градиентов путем деления на максимальный элемент, если он отличен от нуля, получаем НГ(А) и НГ(Б);
вычислить карту абсолютной разницы элементов АР = |НГ(А) – НГ(Б)|;
вычислить сумму С всех элементов в АР;
нормализировать сумму путём деления на число точек в изображениях, получаем НС;
вернуть истинность НС < порог ПНС как результат;

Использованные параметры:
ПНС = 0,065;
Параметр задает чувствительность алгоритма проверки визуальной похожести. Чем больше значение, тем более значительные различия допускаются в карте градиентов. Порог выбирался на основе примеров кадров с камер.

3. Экспериментальные результаты

Для экспериментального оценивания требуется размеченная база примеров, включающая в себя видеопоследовательности с камер видеонаблюдения. К сожалению, примеры, которые мы смогли обнаружить, имели слишком низкое разрешение (потому что не предназначались для задачи обнаружения лиц) или неподходящие углы съемки (например, PETS и Caviar: http://www.cvg.reading.ac.uk/PETS2009/a.html и http://groups.inf.ed.ac.uk/vision/CAVIAR/). Нам не удалось найти в общем доступе подходящей для тестирования базы примеров, поэтому мы собрали и разметили собственную базу. База состоит из нескольких видеозаписей суммарной длиной около 8000 кадров.

При разметке базы мы использовали правила, предложенные в работе [6]:

- каждое лицо на каждом кадре должно быть размечено;
- каждое из размеченных лиц, которое перекрыто или находится слишком далеко (использовался минимальный допустимый размер в 32 × 32 пикселя), а также те лица, которые являются изображениями на плакатах, футболках и так далее, должны быть помечены как игнорируемые.

Второе правило зачастую игнорируется при составлении обучающих и тестовых коллекций для детекторов объектов. И, как показано в работе [6], это приводит как к понижению характеристик детекторов, так и к некорректному сравнению друг с другом характеристик работы разных алгоритмов.

Для оценки детекторов строятся кривые зависимости точности от полноты. Точность (precision) является характеристикой, описывающей, насколько часто алгоритм дает ложные обнаружения (наилучшее значение равно 1 в случае отсутствия ложных обнаружений). Формально точность является отношением числа корректных обнаружений к общему числу обнаружений (корректные и ложные обнаружения). Полнота (recall) является характеристикой, описывающей, насколько много размеченных объектов сумел найти алгоритм (наилучшее значение равно 1 в случае обнаружения всех объектов в базе). Полнота задается отношением числа корректных обнаружений к числу всех размеченных объектов в базе. В качестве интегральной характеристики качества используется площадь под кривой зависимости точности от полноты (AUC). Кривые строятся за счёт варьирования порога на уверенность обнаружения, выдаваемую детектором для каждого обнаружения.

Результаты экспериментальной оценки базового детектора представлены в таблице ниже и на рис. 1. Для сравнения алгоритм обнаружения лиц из открытой библиотеки OpenCV, реализующий детектор Виолы и Джонса, с параметрами по умолчанию дает 14 % точности при 34 % полноты.

Табл. Результаты экспериментального оценивания

Метод	**Площадь под кривой, AUC**
Базовый каскадный детектор	0,497
+ вычитание фона	0,506
+ постобработка	0,513
Дообученный каскадный детектор	0,616
+ постобработка	0,622

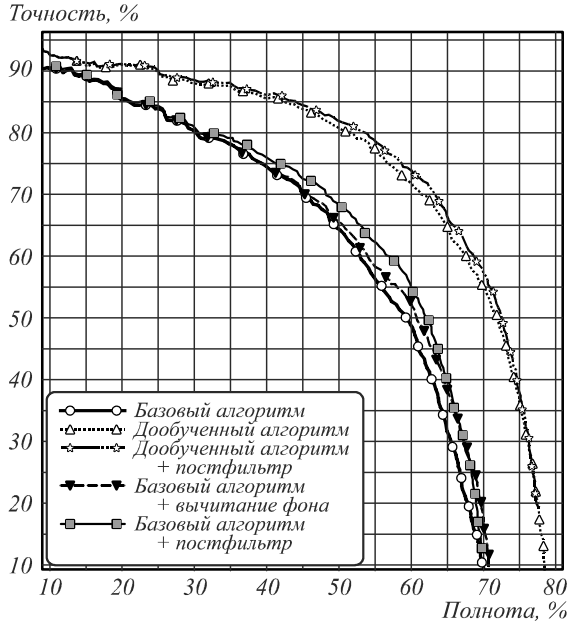

Рис. 1. Графики качества работы алгоритмов

В дополнение к описанным методам в таблице присутствует алгоритм, основанный на вычитании фона (то есть удалении тех обнаружений, на которых алгоритм вычитания фона нашел слишком много фоновых пикселей; выбран порог, дающий лучший результат). Это показывает, что более аккуратное использование алгоритма вычитания фона (не напрямую на выходе алгоритма, а для сбора расширенной негативной выборки для последующей адаптации детектора) дает значительное улучшение качества. При

этом стоит отметить, что адаптация под сцену дала увеличение характеристики полноты, что фактически невозможно для алгоритмов, которые основаны на постобработке результатов работы исходного детектора (в таком случае можно только удалить ложные обнаружения, но не добавить новые). Как уже было упомянуто ранее, адаптированный алгоритм также улучшил показатели скорости работы: исходный алгоритм обрабатывал один кадр размером 1280×800 пикселей за 0,135 секунды в однопоточном приложении (Intel Core i7 950 на частоте 3,07 ГГц), адаптированный улучшил показатель до 0,071 (– 47 %).

Применение дополнительного обучения может привести к нежелательной ситуации, где получаемый алгоритм действительно работает на новых данных лучше, но теряет в общей способности правильно классифицировать лица. Для того, чтобы экспериментально убедиться в том, что предложенный метод не имеет проблем с обобщающей способностью, мы провели сравнение на независимой выборке изображений лиц, собранной из публичных источников в сети Интернет (8000 изображений, одно лицо на изображение). Экспериментальная проверка на собранной базе показала, что обобщающая способность детектора, адаптированного под видеопоток, не имела значительных негативных эффектов. Графики представлены на рис. 2. Значения площади под кривой соответствуют 0,867 для исходного и 0,865 для дообученного алгоритма.

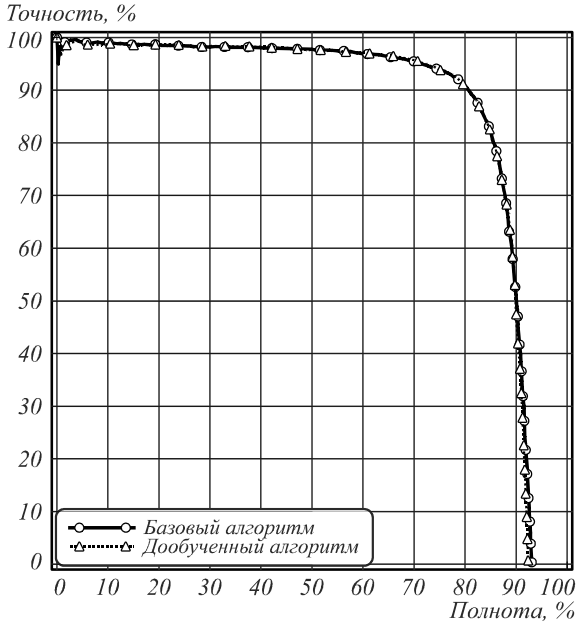

Рис. 2. Результаты на независимой выборке

Как можно заметить из представленных графиков, базовые показатели качества работы алгоритмов различаются на разных тестовых наборах. Это обусловлено тем, что базы имеют различающуюся сложность с точки зрения выделения на них лиц. Кроме того, базе с кадрами с камер видеонаблюдения результаты соответствуют хуже. Это ожидаемый результат, так как изображения с видеокамеры имеют худшее качество, если сравнивать их с фотографиями. Обучение базового алгоритма на примерах из видеозаписей предположительно должно дать более качественный детектор, но, как упоминалось, есть большие трудности с получением большого количества разнообразных данных соответствующего происхождения (обучающая выборка значительно больше валидационной). Минимальная разница между качеством работы дообученного и базового алгоритма на независимой выборке также является ожидаемой. Предлагается метод адаптации детектора под новые данные с целью уменьшения числа ложноположительных обнаружений. Алгоритм улучшает качество работы на данных, на которых производится адаптация, что и является основной задачей. Проблемой может быть то, что улучшение качества на новых данных может стоить потери обобщающей способности на новых данных. Как показывает эксперимент, этой проблемы не возникло.

Помимо этого, мы решили проверить качество работы адаптированного детектора в ситуации, когда адаптация производилась давно, но сцена изображается та же (без учета незначительных изменений угла направления камеры). Для этого мы дообучили детектор на видеопоследовательности, отснятой в течение солнечного дня, но тестирование производили на видеопотоке, соответствующем вечерней записи, где значительно изменилось освещение. Результаты проверки показывали, что алгоритм сохранил как улучшенные показатели качества работы (менее выраженные), так и повышенную производительность (рис. 3).

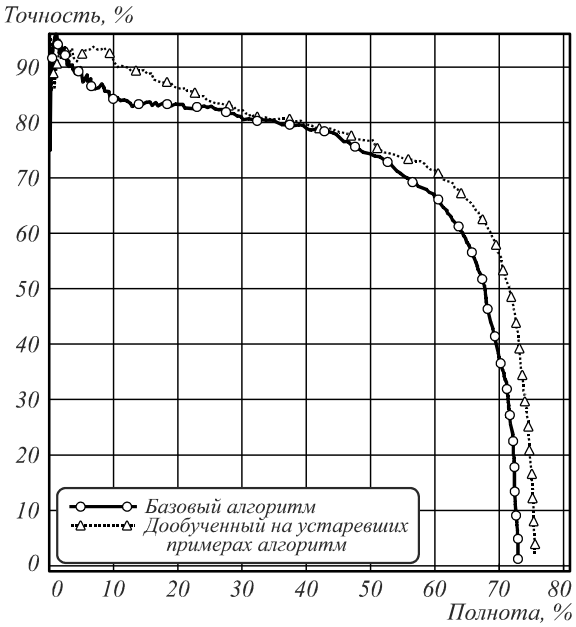

Рис. 3. Результаты адаптации на устаревших примерах

Заключение

В данной работе предлагаются два метода для повышения качества работы базового детектора лиц в видеопотоке. Первый метод заключается в автоматическом сборе части ложноположительных срабатываний базового детектора и его последующей адаптации к видеопотоку с использованием собранных при-

меров. Данный метод применим ко всем алгоритмам, которые могут быть переобучены в процессе использования за допустимое время. Второй метод является постобработкой выхода детектора, позволяющей отфильтровать ложноположительные обнаружения на фоновых областях кадра. Фильтрация опирается на анализ частоты выделения детектором визуально похожих примеров из кадров видеопотока. Метод применим ко всем алгоритмам, которые имеют ложные обнаружения на фоне, т.е. на данный момент ко всем существующим детекторам лиц.

Рис. 4. Пример разницы работы алгоритмов: сверху – базовый алгоритм, снизу – дообученный с постобработкой; слева сверху квадратов указана уверенность детектора

Экспериментальная оценка предложенных методов проведена с использованием каскадного детектора на интегральных признаках [7]. Оценка показала, что предложенные методы не только увеличивают показатели точности и полноты выделения лиц, но и уменьшают время работы на кадр видеопотока.

Благодарности

Работа выполнена при поддержке гранта РФФИ №15-31-20596.

Литература

1. **Verma, RC.** Face detection and tracking in a video by propagating detection probabilities / R.C. Verma, C. Schmid, K. Mikolaqczyk // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2003. – Vol. 25(10). – P. 1215-1228. – DOI: 10.1109/TPAMI.2003.1233896.
2. **Park, D.** Exploring weak stabilization for motion feature extraction / D. Park, C.L. Zitnick, D. Ramanan, P. Dollár // CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 2013. – 2013. – P. 2882-2889. – DOI: 10.1109/CVPR.2013.371.
3. **Walk, S.** New features and insights for pedestrian detection / S. Walk, N. Majer, K. Schindler, B. Schiele // IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010), June 13-18, 2010, San Francisco, California, USA. – 2010. – P. 1030-1037. – DOI: 10.1109/CVPR.2010.5540102.
4. **Kolarow, A.** APFel: The intelligent video analysis and surveillance system for assisting human operators / A. Kolarow, K. Schenk, M. Eisenbach, M. Dose, M. Brauckmann, K. Debes, H.-M. Gross // 2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). – 2013. – P. 195-201. – DOI: 10.1109/AVSS.2013.6636639.
5. **Viola, P.** Robust real-time face detection / P. Viola, M.J. Jones // International Journal of Computer Vision. – 2004. – Vol. 57(2). – P. 137-154. – DOI: 10.1023/B:VISI.0000013087.49260.fb.
6. **Mathias, M.** Face detection without bells and whistles / M. Mathias, R. Benenson, M. Pedersoli, L. Van Gool // 13th European Conference on Computer Vision (ECCV 2014), Zürich, Switzerland, September 6-12, 2014. – 2014. – P. 720-735. – DOI: 10.1007/978-3-319-10593-2_47.
7. **Dollár, P.** Integral channel features / P. Dollár, Z. Tu, P. Perona, S. Belongie // Proceedings of the British Machine Vision Conference. – 2009. – P. 91.1-91.11. – DOI: 10.5244/C.23.91.
8. **Felzenswalb, P.** A discriminatively trained, multiscale, deformable part model / P. Felzenswalb, D. McAllester, D. Ramanan // IEEE Conference on Computer Vision and Pattern Recognition, June 24-26, 2008 (CVPR 2008). – 2008. – P. 1-8. – DOI: 10.1109/CVPR.2008.4587597.
9. **Zhu, X.** Face detection, pose estimation and landmark localization in the wild / X. Zhu, D. Ramanan // IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 2012 (CVPR 2012). – 2012. – P. 2879-2886. – DOI: 10.1109/CVPR.2012.6248014.
10. **Szegedy, C.** Deep neural networks for object detection / C. Szegedy, A. Toshev, D. Erhan // Advances in Neural Information Processing Systems. – 2013. – P. 2553-2561.
11. **Girshick, R.** Rich feature hierarchies for accurate object detection and semantic segmentation / R. Girshick, J. Donahue, T. Darrell, J. Malik // Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. – 2014. – P. 580-587. – DOI: 10.1109/CVPR.2014.81.
12. **Appel, R.** Quickly boosting decision trees – Pruning underachieving features early / R. Appel, T. Fuchs, P. Dollár, P. Perona // Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. – 2013. – Vol. 28. – P. 594-602.
13. **Dollár, P.** Fast feature pyramids for object detection / P. Dollár, R. Appel, S. Belongie, P. Perona // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2014. – Vol. 36(8). – P. 1532-1545. – DOI: 10.1109/TPAMI.2014.2300479.
14. **Dollár, P.** The fastest pedestrian detector in the west / P. Dollár, S. Belongie, P. Perona // Proceedings of the British Machine Vision Conference. – 2010. – P. 68.1-68.11. – DOI: 10.5244/C.24.68.
15. **Dollár, P.** Crosstalk cascades for frame-rate pedestrian detection / P. Dollár, R. Appel, W. Kienzle // Proceedings of the 12th European Conference on Computer Vision. – 2012. – Part II. – P. 645-659. – DOI: 10.1007/978-3-642-33709-3_46.

16. **Jain, V.** Online domain adaptation of a pre-trained cascade of classifiers / V. Jain, E. Miller // Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. – 2011. – P. 577-584. – DOI: 10.1109/CVPR.2011.5995317.
17. **Wang, M.** Automatic adaptation of a generic pedestrian detector to a specific traffic scene / M. Wang, X. Wang // Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. – 2011. – P. 3401-3408. – DOI: 10.1109/CVPR.2011.5995698.
18. **Barnich, O.** ViBe: A universal background subtraction algorithm for video sequences / O. Barnich, M. Van Droogenbroeck // IEEE Transactions on Image Processing. – 2011. – Vol. 20(6). – P. 1709-1724. – DOI: 10.1109/TIP.2010.2101613.
19. **Freund, Y.** A decision-theoretic generalization of on-line learning and an application to boosting / Y. Freund, R.E. Schapire // Journal of Computer and System Sciences. – 1997. – Vol. 55(1). – P. 119-139. – DOI: 10.1006/jcss.1997.1504.
20. **Bourdev, L.** Robust object detection via soft cascade / L. Bourdev, J. Brandt // Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. – 2005. – Vol. 2. – P. 236-243. – DOI: 10.1109/CVPR.2005.310.
21. **Everingham, M.** The Pascal visual object clasees (VOC) challenge / M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman // International Journal of Computer Vision. – 2010. – Vol. 88(2). – P. 303-338. – DOI: 10.1007/s11263-009-0275-4.

Сведения об авторах

Сергеев Александр Евгеньевич, 1992 года рождения, в 2015 году окончил МГУ имени М.В. Ломоносова. Аспирант НИУ ВШЭ. Научные интересы: компьютерное зрение, выделение объектов на изображениях и в видео. E-mail: *aesergeev@hse.ru* .

Конушин Вадим Сергеевич, 1985 года рождения, в 2007 году окончил МГУ имени М.В. Ломоносова. Работает в ООО «Технологии видеоанализа». E-mail: *vadim@tevian.ru* .

Конушин Антон Сергеевич, 1980 года рождения, в 2002 году окончил МГУ имени М.В. Ломоносова. В 2005 году защитил кандидатскую диссертацию в ИПМ имени М.В. Келдыша РАН. Доцент НИУ ВШЭ и МГУ имени М.В. Ломоносова. Научные интересы: компьютерное зрение, машинное обучение. E-mail: *akonushin@hse.ru* .

Поступила в редакцию 16 января 2016 г. Окончательный вариант – 10 октября 2016 г.

REDUCING BACKGROUND FALSE POSITIVES FOR FACE DETECTION IN SURVEILLANCE FEEDS

A.E. Sergeev[1], A.S. Konushin[1], V.S. Konushin[2]
[1]National Research University Higher School of Economics, Moscow, Russia,
[2]Video Analysis Technologies LLC, Moscow, Russia

Abstract

This paper addresses a problem of false positive detection filtering in surveillance video streams. We propose two methods. The first one is based on automatic hard negative mining from a video stream, which is then used for fine-tuning of the baseline detector. The second one is the detector output filtering by analyzing the frequency of detection of visually similar samples. We demonstrate the proposed methods on cascade-based detectors, but they can be applied to any detector that can be trained in a reasonable amount of time. Experimental results show that the proposed methods improve both the precision and recall rate, as well as reducing the computational time by 47%.

Keywords: detectors, pattern recognition, image analysis, machine vision algorithms.

Citation: Sergeev AE, Konushin AS, Konushin VS. Reducing background false positives for face detection in surveillance feeds. Computer Optics 2016; 40(6): 958-967. DOI: 10.18287/2412-6179-2016-40-6-946-958-967.

Acknowledgements: The work was partially funded by RFBR, grant No. 15-31-20596.

References

[1] Verma RC. Schmid C, Mikolaqczyk K. Face detection and tracking in a video by propagating detection probabilities. IEEE Transactions on Pattern Analysis and Machine Intelligence 2003; 25(10): 1215-1228. DOI: 10.1109/TPAMI.2003.1233896.
[2] Park D, Zitnick CL, Ramanan D, Dollár P. Exploring weak stabilization for motion feature extraction. CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 2013: 2882-2889. DOI: 10.1109/CVPR.2013.371.
[3] Walk S, Majer N, Schindler K, Schiele B. New features and insights for pedestrian detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010), June 13-18, 2010, San Francisco, California, USA 2010: 1030-1037. DOI: 10.1109/CVPR.2010.5540102.
[4] Kolarow A, Schenk K, Eisenbach M, Dose M, Brauckmann M, Debes K, Gross H-M. APFel: The intelligent video analysis and surveillance system for assisting human operators. 2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) 2013: 195-201. DOI: 10.1109/AVSS.2013.6636639.

[5] Viola P, Jones MJ. Robust real-time face detection. International Journal of Computer Vision 2004; 57(2): 137-154. DOI: 10.1023/B:VISI.0000013087.49260.fb.
[6] Mathias M, Benenson R, Pedersoli M, Van Gool L. Face detection without bells and whistles. 13th European Conference on Computer Vision (ECCV 2014), Zürich, Switzerland, September 6-12, 2014: 720-735. DOI: 10.1007/978-3-319-10593-2_47.
[7] Dollár P, Tu Z, Perona P, Belongie S. Integral channel features. BMVC 2009: 91.1-91.11. DOI: 10.5244/C.23.91.
[8] Felzenswalb P, McAllester D, Ramanan D. A discriminatively trained, multiscale, deformable part model. CVPR 2008: 1-8. DOI: 10.1109/CVPR.2008.4587597.
[9] Zhu X, Ramanan D. Face detection, pose estimation and landmark localization in the wild. CVPR 2012: 2879-2886. DOI: 10.1109/CVPR.2012.6248014.
[10] Szegedy C, Toshev A, Erhan D. Deep neural networks for object detection. Advances in Neural Information Processing Systems 2013: 2553-2561.
[11] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR 2014: 580-587. DOI: 10.1109/CVPR.2014.81.
[12] Appel R, Fuchs T, Dollár P, Perona P. Quickly boosting decision trees-pruning underachieving features early. ICML 2013; 28: 594-602.
[13] Dollár P, Appel R, Belongie S, Perona P. Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 2014; 36(8): 1532-1545. DOI: 10.1109/TPAMI.2014.2300479.
[14] Dollár P, Belongie S, Perona P. The fastest pedestrian detector in the west. BMVC 2010: 68.1-68.11. DOI: 10.5244/C.24.68.
[15] Dollár P, Appel R, Kienzle W. Crosstalk cascades for frame-rate pedestrian detection. ECCV'12 2012; II: 645-659. DOI: 10.1007/978-3-642-33709-3_46.
[16] Jain V, Miller E. Online domain adaptation of a pre-trained cascade of classifiers. CVPR 2011: 577-584. DOI: 10.1109/CVPR.2011.5995317.
[17] Wang M, Wang X. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. CVPR 2011: 3401-3408. DOI: 10.1109/CVPR.2011.5995698.
[18] Barnich O, Van Droogenbroeck M. ViBe: A universal background subtraction algorithm for video sequences. IEEE Transactions on Image Processing 2011; 20(6): 1709-1724. DOI: 10.1109/TIP.2010.2101613.
[19] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 1997; 55(1): 119-139. DOI: 10.1006/jcss.1997.1504.
[20] Bourdev L, Brandt J. Robust object detection via soft cascade. CVPR 2005; 2: 236-243. DOI: 10.1109/CVPR.2005.310.
[21] Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A. The Pascal visual object clasees (VOC) challenge. International Journal of Computer Vision 2010; 88(2): 303-338. DOI: 10.1007/s11263-009-0275-4.

Authors' information

Alexander Evgenievich Sergeev (b. 1992), graduated from Lomonosov Moscow State University in 2015. Currently graduate student at NRU Higher School of Economics. Research interests are computer vision and object. E-mail: *aesergeev@hse.ru* .

Vadim Sergeyevich Konushin (b. 1985) graduated from Lomonosov Moscow State University in 2007 and currently work at «Video Analysis Technologies» LLC. E-mail: *vadim@tevian.ru* .

Anton Sergeyevich Konushin (b. 1980) graduated from Lomonosov Moscow State University in 2002. In 2005 successfully defended his PhD thesis in M.V. Keldysh Institute for Applied Mathematics RAS. He is currently associate professor at NRU HSE and Lomonosov Moscow State University. Research interests are computer vision and machine learning. E-mail: *akonushin@hse.ru* .

Received January 16, 2016. The final version – October 10, 2016.