УДК 378:004
DOI: 10.25686/978-5-8158-2474-4-2025-649-658

Метод быстрого распознавания неверного поворота фотографий детей для сайта «усыновите.рф»

Д. С. Потапов
Федеральный институт цифровой трансформации в сфере образования, Москва, Россия
Институт развития профессионального образования, Москва, Россия

Аннотация. Распространённой проблемой при загрузке пользовательских фотографий на сайт является их неправильная ориентация, чаще всего поворот на 90 градусов. Причиной обычно является отсутствие сенсора ориентации камеры или его неправильное срабатывание, игнорирование метаданных о повороте программным обеспечением или ошибочные действия пользователя. В работе представлен способ значительного ускорения проверки на поворот за счёт простейших преобразований изображения: обрезки и уменьшения изображения. Для решения проблемы ложного срабатывания на лежачих детях-инвалидах разработана процедура обучения свёрточной нейронной сети (СНС) для автоматического отсеивания изображений данного класса. Подход показал высокую точность решения поставленной задачи на реальных данных и был успешно интегрирован в программное обеспечение для обновления данных на сайте.

***Ключевые слова*:** распознавание неверной ориентации фотографий, компьютерное зрение, нейронные сети, распознавание изображений, дисбаланс классов, ROC-кривая

Fast method for wrong rotation recognition in orphan children photos for the website https://усыновите.рф

D. S. Potapov
Federal institute for digital transformation in education, Moscow, Russia
Institute for development of professional education, Moscow, Russia

Abstract. When a user uploads a photo to the web, a common problem is the wrong orientation of the file, usually a 90-degree rotation. This happens due to the absence of the automatic orientation sensor of the camera or its mistakes, due to ignoring of the orientation metadata by the processing software or because of the user faulty action. In this work we present a technique for a significant speed-up of the rotation check due to the use of simple image transformations: cropping and downsampling. We have developed a procedure for training of a convolutional neural network (CNN) for automatic elimination of false alarms coming from images of bedridden handicapped children. The proposed approach solves the formulated problem with high accuracy and has been successfully intergrated in the software for updating the website content.

***Keywords*:** recognition of wrongly oriented photos, computer vision, neural networks, image classification, imbalanced classes, ROC-curve

Введение

При поддержке веб-ресурсов с большим объёмом данных часто встаёт проблема автоматического нахождения ошибок в данных, поскольку ручная проверка невозможна. Проблема особенно актуальна в связи с тем, что даже 0,1 % ошибок уже могут быть замечены 50-500 пользователями в день. Хотя и часть проблем может быть решена на этапе ввода данных, это не всегда возможно, поскольку интерфейс ввода может быть несовершенен либо из-за того, что ошибки семантические и не могут быть легко обнаружены или требуют анализа всех данных в целом. Для сайта усыновите.рф данные собираются с органов опеки всех регионов России, частично проверяются региональными и федеральными операторами, а мы проводим дополнительные проверки на этапе загрузки на сайт. В качестве одной из проверок в данной работе предлагается распознавание неверно повёрнутых фотографий.

Задача определения ориентации фотографий давно известна в области компьютерного зрения [1-4]. Она подразумевает нахождение наиболее правильной ориентации из четырёх вариантов: 0°, +90°, -90°, 180°. Решение этой задачи предполагает использование признаков изображения, по которым можно определить корректную ориентацию фотографии. Если на изображении точно есть человек, то задачу можно решать, основываясь на обнаружении лица, поскольку в большинстве случаев лицо находится в естественной ориентации (лоб сверху, подбородок снизу).

В 2009 году был опубликован патент [5] на определение ориентации фотографии на основе детектора лиц того времени. Метод предполагает нахождение лиц на четырёх разных ориентациях фотографии и выбор ориентации с наибольшим количеством лиц или наибольшей уверенностью. В работе [4] сравнивается определение ориентации изображений лиц на основе детектора лиц и использование признаков из предобученной нейронной сети. Мы показываем, что такой метод работает медленнее, чем методы на основе детекторов лиц, поэтому мы применяем его только на этапе отсеивания ложных срабатываний.

Методы обнаружения лиц развивались благодаря их важным применениям. Во-первых, они используются в фотосъёмке для фокусировки на лицах и срабатывании затвора по улыбке. Во-вторых, большую популярность приобрела задача распознавания лиц. В связи с особенностями строения тела человека обычно лицо отклоняется не более чем на 45° в плоскости фотографии от вертикального положения. Традиционно детекторы обнаруживали лица, близкие к вертикальному положению в плоскости изображения, что навело нас на мысль, что эту особенность можно использовать для определения ориентации фотографии.

Задача обнаружения лиц приобрела популярность в связи с увеличением количества цифровых фотографий в начале 2000-х годов [6, 7]. В изначальной постановке подразумевалось обнаружение только околофронтальных лиц без значительного поворота в плоскости изображения. В широко известном методе Viola Jones [6] задача решается с использованием скользящего окна, быстро вычисляемых признаков Хаара, алгоритма машинного обучения AdaBoost и каскада бинарных классификаторов. Метод был реализован во многих библиотеках и устройствах из-за его высокой скорости работы (например, в OpenCV [8]).

Начиная с ключевой статьи, опубликованной в 2012 году [9], в которой была существенно повышена точность в задаче распознавания изображений, в мире сильно возросла популярность искусственных нейронных сетей. Постепенно они были адаптированы для решения задачи обнаружения лиц. Метод MTCNN [10] решает сразу несколько задач: обнаружение лица, предсказание прямоугольника лица и нахождение ключевых точек. За счёт дополнительных данных при обучении метод превосходит предыдущие алгоритмы по точности. Метод также отличается высокой скоростью, благодаря чему был реализован в нескольких библиотеках [11, 12]. В статье [13], вышедшей в 2020 году, описан подход RetinaFace для обнаружения лиц, в котором в дополнение к ключевым точкам, также моделируется 3D-форма лица. В работе [14] описан быстрый детектор лиц YuNet, который достигает наилучшего компромисса между точностью и скоростью. Метод включает в себя высокопроизводительную основу и упрощённый способ смешивания признаков с разных масштабов. Метод был интегрирован в библиотеку OpenCV [8].

Мы применили обнаружение лиц для распознавания неправильно повёрнутых фотографий на сайте усыновите.рф, и сразу выяснилась одна особенность данных: в банке данных детей-сирот присутствует значительное число фотографий лежачих детей-инвалидов, у которых голова ориентирована на кадре горизонтально, поэтому возникают ложные срабатывания. Мы решили их отсеивать с помощью методов распознавания изображений.

Задача распознавания изображений активно развивалась в последние 30-40 лет. В её традиционной формулировке необходимо классифицировать неизвестное модели изображение на несколько категорий (классов). Изначально рассматривались относительно простые задачи, например распознавание рукописных цифр от 0 до 9 [15]. Впоследствии задачи стали усложняться и размер изображений увеличивался. В работе [16] применяется метод скользящего окна для классификации подизображения на два класса (пешеход и фон) с использованием гистограмм ориентированных градиентов. В 2000-е годы стал популярным подход Bag of Visual Words [17], в котором локальные признаки изображения (например, SIFT [18]) агрегировались в дескриптор всего изображения без учёта положения признаков. В дальнейшем появились более качественные методы агрегации [19] и учёт приблизительного положения признаков на изображении [20, 21]. Сильный толчок к развитию методов распознавания изображений дало соревнование Pascal Visual Object Challenge [22], проходившее в 2008-2012 годах.

Основополагающая работа [9] практически совершила революцию в мире распознавания изображений. Этому способствовало появление больших коллекций изображений [23] и возможности значительного ускорения вычислений на графических картах (GPU). С этого момента искусственные нейронные сети начинают стремительно развиваться и с каждым годом появляются всё новые улучшения [24-27]. СНС для распознавания изображений в основном обучаются и тестируются на большой базе данных ImageNet [23], содержавшей изначально 1000 классов и 14 миллионов изображений и впоследствии расширенной до более 20 тысяч классов. Было показано, что, используя процедуру дообучения (fine-tuning), можно перенести визуальные знания о мире из больших наборов данных в СНС, обучающиеся на относительно небольших данных [28].

При обучении на небольших наборах изображений нейронные сети склонны к переобучению. Это частично компенсируется применением дообучения, но не полностью. Для решения проблемы переобучения предложено множество методов обогащения (аугментации) обучающей выборки. В их числе случайные вырезания подизображений, случайный поворот на небольшой угол, случайное отражение относительно вертикали, варьирование оттенка, добавление шума, размытие, сжатие Jpeg и прочее. В различных задачах также применяются Random Erasing [29] и Cutout [30] и более замысловатые методы MixUp [31] и CutMix [32].

В связи с тем что СНС требовательны к вычислительным ресурсам, **методы повышения быстродействия нейронных сетей** в последнее время бурно развиваются. В данной работе затрагивается повышение скорости объединённого метода на основе СНС только на этапе применения моделей обнаружения лиц и распознавания лежащих детей. В настоящее время используются каскады классификаторов [10], вычисления на GPU [9], квантизация [33] и дистилляция [34] нейронных сетей. Применение большинства из этих методов требует большего времени, дополнительных средств или выполнения некоторых условий, в то время как в данной работе предлагается простой подход на основе использования специфики тестовой выборки.

(а) (б) (в)

Рисунок 1. Пример изображений из классов, которые нужно распознать:
а) правильная ориентация; б) неправильная ориентация; в) лежащий ребёнок. Designed by Freepik

Постановка задачи

Глобально рассматривается задача автоматического определения ориентации фотографии, содержащей человека. Мы делаем предположение, что на фотографии видно лицо, то есть человек повёрнут к камере на угол не более 90° относительно вертикали. Поскольку в наших интересах найти только фотографии с неправильной ориентацией, а исправлением ошибки уже занимается человек, для себя мы ставим задачу только распознавания фотографий с неправильной ориентацией, то есть повёрнутых на один из углов (-90°, +90° или 180°).

При решении задачи возникла необходимость повышения скорости работы метода распознавания неправильной ориентации. Изначально ставилась задача сократить время проверки базы до менее одной минуты без использования GPU, но в итоге нам удалось разработать ещё более быстрый метод.

В связи со спецификой данных ставится дополнительная задача – распознавание лежащих детей среди фотографий с горизонтальным лицом. Для этой задачи особых требований к скорости работы нет, поскольку количество ложных срабатываний очень мало по сравнению с общим объёмом изображений.

Описание методов

1. Метод распознавания неправильной ориентации фотографии на основе детектора лиц

Изначально алгоритм был следующим: применить детектор лиц; если на фотографии найдено лицо, то выдать результат «правильная ориентация»; иначе – «неправильная ориентация».

Первоначально на обработку 35’000 фотографий уходило 7 часов[1]. Впоследствии мы воспользовались библиотекой torch-MTCNN [11] и перенесли запуск кода на машину с графическим процессором, и время сократилось до 12 минут. При этом проведение остальных проверок при загрузке на сайт занимало менее полминуты. На компьютере администратора не было GPU, поэтому данная проверка не очень сочеталась с существующими процессами.

Для сокращения времени работы изначального алгоритма мы предположили, что можно обрезать края изображения, а также уменьшить размер изображения. В редких случаях, когда это приведёт к потере детекции, обнаружение лиц необходимо повторить на исходном изображении.

На большинстве фотографий лицо ребёнка находится близко к центру изображения, поэтому мы решили оценить область обрезки. Мы нашли лица на 1000 случайных изображениях из базы и сохранили координаты прямоугольников лиц в массивы left, top, right, bottom.

Прямоугольник задаётся координатами левого верхнего (left, top) и правого нижнего (right, bottom) углов с осью X, направленной вправо, и осью Y, направленной вниз. Далее мы вычислили нормализованные координаты прямоугольников:

$$left\_n[i] = left[i]/w[i],\ top\_n[i] = top[i]/h[i],\ right\_n[i] = right[i]/w[i],\ bottom\_n[i] = bottom[i]/h[i],$$

где w[i] и h[i] – ширина и высота изображения.

Когда на фотографии было найдено несколько лиц, выбирался наибольший по площади прямоугольник лица – обычно в базе присутствует только лицо одного ребёнка. Чтобы оценить область, в которой в большинстве случаев находится лицо, по массивам left_n и top_n мы посчитали 5-й квантиль, по массивам right_n и bottom_n – 95-й квантиль. Получились такие значения квантилей:

$$q\_left = 0.19,\ q\_top = 0.10,\ q\_right = 0.80,\ q\_bottom = 0.75.$$

Это означает, что более $0.95^4 \approx 81$ % лиц находится в области площадью $0.61 \cdot 0.65 \approx 0.40$ от исходного изображения.

Далее мы решили проверить, можно ли обнаруживать лица на уменьшенном изображении, чтобы сократить время работы детектора. Для каждого изображения из 1000 случайных мы рассмотрели масштабированные версии изображения с коэффициентом от 0.05 до 1.0 и шагом 0.025 и определили, на каком минимальном масштабе обнаруживается лицо (обрезка в данном случае не делалась). Далее мы взяли 97-й квантиль массива минимальных масштабов и получили масштаб 0.25, что примерно соответствует изображению размера 90x110. Площадь изображения при этом уменьшается в 16 раз, что существенно увеличивает скорость детекции лиц.

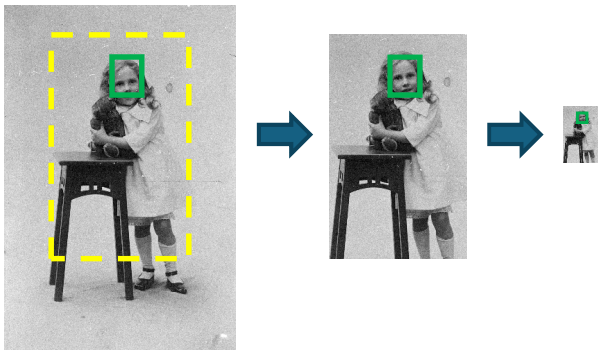

Рисунок 2. Иллюстрация уменьшения области применения детектора лиц за счёт уменьшения области поиска и уменьшения масштаба. Область обрезки такая большая, поскольку на фотографиях встречаются лица крупным планом. Площадь изображения сокращается в 40 раз. Photo by Museums Victoria on Unsplash

[1] Предполагаемая причина очень медленной работы – использование неэффективной реализации метода MTCNN.

Замеры показали, что 96.6 % лиц было найдено на уменьшенном изображении с небольшими временными затратами. Если же лицо не найдено, мы применяем детектор к оригинальному изображению.

Время работы метода составило 10 минут на 35’000 изображений без использования GPU. Замер показал, что по сравнению с изначальным алгоритмом программа ускорилась в 5 раз. Впоследствии процесс проверки данных был усовершенствован за счёт исключения повторной обработки одних и тех же фотографий. Дело в том, что фотографии детей обновляются в большинстве случаев не чаще одного раза в год, в среднем между двумя последовательными проверками обновляется порядка 300 фотографий. Достаточно проверять каждую фотографию только один раз, поэтому мы реализовали хранение md5-сумм проверенных фотографий. По сравнению с обнаружением лиц md5-сумма файла вычисляется практически мгновенно, примерно за 0.1 миллисекунду. Такой подход помог сократить время проверки в 100 раз.

Изначально мы предполагали, что на фотографиях с неправильной ориентацией лицо не будет найдено. В ходе исследований выяснилось, что детектор лиц при применении на фотографии с лицом, повёрнутым на 90°, в 22 % случаев выдаёт положительный результат обнаружения. Это связано с тем, что при обучении детектора использовались фотографии, снятые с произвольных ракурсов и, соответственно, в нём встречались лица, ориентированные в произвольном направлении. На рисунке 3 показана оценка вероятности обнаружения лица детектором torch-MTCNN [11] в зависимости от угла поворота.

Для уменьшения числа ошибок первого рода мы добавили дополнительную проверку на положение ключевых точек. Если угол между вектором правый-левый глаз ребёнка и осью X, направленной вправо, составляет более 45°, то выводится результат «неправильная ориентация». Даже с учётом такого улучшения ошибка первого рода составляла 11.8 % при замере на 2000 изображениях и поворотах ±90°. Дополнительный анализ показал, что на лицах повёрнутых на 90°, ключевые точки лица находятся неточно – часто с поворотом около 45°. Поэтому после первого нахождения ключевых точек мы выравниваем положение глаз с горизонталью с помощью поворота изображения и запускаем поиск ключевых точек повторно для уточнения их положения.

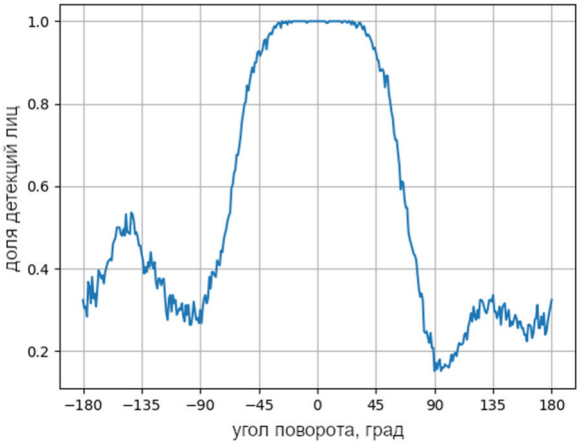

Рисунок 3. Статистика обнаружения лица в зависимости от угла ориентации лица. Направление горизонтальной оси соответствует повороту против часовой стрелки. Тест проведён на 250 фотографиях из базы детей с практически вертикальной ориентацией лица

По сравнению с методом [5] в изначальной версии метода требуется применение детектора лиц только в одной ориентации, что ускоряет обработку в 4 раза. Однако в некоторых библиотеках есть возможность отключения определения ключевых точек, что может компенсировать эти временные затраты. Далее мы приводим сравнение с методом [5], к которому также применим предложенный выше способ повышения скорости. Более точно в данном случае, если ни в одной ориентации не было найдено лицо, осуществляется поиск на изображении исходного размера.

Финальная версия алгоритма выглядит следующим образом:

1. Уменьшить изображение до площади 10’000 пикселей и обрезать поля относительного размера margin_left=0.19, margin_top=0.10, margin_right=0.20, margin_bottom=0.25;
2. Найти ориентацию с наибольшей уверенностью детектора YuNet [14] (метод [5]);
3. Если лица найдены и наилучшая ориентация 0°, то вернуть «правильная ориентация»;
4. Иначе применить метод [5] с детектором YuNet [14] к исходному изображению.

2. *Метод распознавания лежащих детей*

Как выяснилось на практике, алгоритм на основе обнаружения лиц на наших данных выдаёт много фотографий лежачих детей-инвалидов, у которых лицо ориентировано горизонтально. По согласованию с заказчиком было решено такие фотографии считать правильно ориентированными. Мы решили автоматически отсеивать случаи ложных срабатываний из числа потенциально неверно ориентированных фотографий.

Задача распознавания изображений в последние годы обычно решается с помощью нейронных сетей [9, 28]. В случае небольшего количества данных применяется дообучение [28] свёрточной нейронной сети (СНС), предобученной на базе ImageNet [23]. Дообучение предполагает инициализацию СНС с помощью весов, полученных в результате предобучения, и минимизацию функции потерь с более низким темпом обучения. Это позволяет сохранить некоторое количество априорных знаний о природе данных (изображений) и увеличивает точность распознавания. В отличие от [28] мы обновляем веса *всех* слоёв при обучении СНС, подобно тому, как это делают в методах распознавания кровного родства [35].

В качестве базовой модели (СНС) мы используем ResNet-50 [26] с весами IMAGENET1K_V2. Модель состоит из 50 слоёв. Слои объединяются в базовые блоки, состоящие из трёх слоёв: свёртка 1x1, свёртка 3x3 и свёртка 1x1. При приближении к последним слоям количество каналов свёртки увеличивается. Каждый базовый блок дублируется сквозной связью в обход этого блока.

Модель ResNet-50 предполагает входное изображение размером 224x224. Для приведения изображений к такому виду мы сначала масштабируем их к размеру 232x232. В качестве обогащения выборки используется отражение относительно вертикали с вероятностью 0.5 и вырезание случайного квадрата размером 224x224. Поскольку базовая модель решает задачу многоклассовой классификации, мы заменяем её последний полносвязный слой с 1000 выходами на полносвязный слой с одним выходом. В качестве функции потерь используются бинарные кросс-энтропийные потери (см. ниже), для её минимизации используется алгоритм Adam со стандартными параметрами в pytorch и темпом обучения 10^{-5}. Как это общепринято, мы перемешиваем обучающие примеры перед каждой эпохой.

Вручную разметили изображения на три класса: обычные (0), лежащие дети с вертикальным лицом (1) и лежащие дети с горизонтальным лицом (2). Был выбран протокол разметки, при котором дети, сидящие в коляске или на диване или с поднятой головой, не считались лежащими. Все обычные фотографии, которые имели неправильную ориентацию, были приведены к правильной ориентации. При обучении мы поворачиваем фотографии из классов 0 и 1 на 90 градусов влево или вправо, чтобы смоделировать ошибочную ориентацию. Фотографии из класса 2 оставляем как есть, так как их поворот на 180 градусов часто является неправдоподобным изображением.

Поскольку часть изображений из класса 1 также содержит детей-инвалидов, на этом этапе мы решили объединить классы 1 и 2, чтобы модель также отсеивала повёрнутые изображения из класса 1. Таких изображений примерно в 60 раз меньше, чем в классе 0, поэтому это не должно привести к большому количеству ошибок первого рода. Объединённый класс мы приняли за positive class.

В данной формулировке задача является сильно несбалансированной: в обучающей выборке приблизительно в 54 раза больше обычных фотографий, чем лежащих детей. При этом на этапе применения модели у нас в несколько раз больше лежащих детей, чем обычных. Поскольку допускается наличие небольшого количества ложных срабатываний, мы стремимся к точке на ROC-кривой, в которой ошибки первого и второго рода происходят одинаково часто: TPR ≈ TNR (true positive rate ≈ true negative rate), но мы немного упростили задачу и сразу выбрали целевое значение TNR = 0.97. Соответственно, при валидации модели мы используем значение сбалансированной точности BA = 0.5·(TPR+TNR) в точке TNR ≈ 0.97. Однако, при минимизации обычной кросс-

энтропийной функции потерь как функции от весов модели θ точность работы модели близка к 100 %, но модель всё время выдаёт класс 0, что приводит к низкому значению BA. Для корректировки дисбаланса в функцию потерь необходимо добавить вес положительного класса w, равный 54. Такая сбалансированная бинарная кросс-энтропийная функция потерь задаётся формулой

$$\text{BinaryCrossEntropyLoss}(x, y, \theta) = -(w \cdot y \cdot \log f(x, \theta) + (1 - y) \cdot \log(1 - f(x, \theta))) ,$$

где x – входное изображение; y – метка класса (0 или 1); $f(\cdot)$ – функция, которую задаёт нейронная сеть. На этапе применения модель выдаёт результат «обычная фотография» или «лежащий ребёнок» (нужно отсеять).

Распространённым методом регуляризации нейронных сетей является ранний останов, когда при обучении точность модели время от времени контролируется на валидационной выборке, и в качестве финальных выбираются веса с наилучшим качеством при валидации. Мы делаем контроль каждые 50 итераций при размере мини-батча в 32 изображения. Обучение длится не более 10 эпох, но мы останавливаем обучение, если нет улучшения качества на валидационной выборке в течение 20 раундов.

Результаты экспериментов

1. *Распознавание неправильной ориентации фотографии на основе детектора лиц*

Для сравнения методов определения ориентации фотографии на основе обнаружения лиц мы используем выборку из 2000 случайных фотографий детей[1]. Эти фотографии были проверены на правильность ориентации и отсутствие лежащих детей. 84 % фотографий имеют размер от 250x400 до 450x450, все фотографии ограничены размером 468x445. Далее были сделаны три копии данной выборки с поворотами на +90° (влево), на -90° (вправо) и на 180°. В таблице 1 показано сравнение точности и скорости различных методов. Сравнение проводилось на ноутбуке с процессором Intel Core i7-12700H с 14 физическими и 20 логическими ядрами с операционной системой Windows 11. Скорость работы замерялась на изображениях с поворотом 0°. В таблице 1 мы исследуем только внутренний параллелизм алгоритмов. Как показано в таблице 2, выполнение алгоритма в нескольких процессах (внешний параллелизм) даёт более существенный прирост скорости. Эксперимент проводился на 800 изображениях, приведены среднее время на обработку одного изображения и стандартное отклонение по трём запускам.

Таблица 1. **Сравнение скорости и точности различных методов (без использования GPU)**

Название метода	Сред. время обработки одного изображения, с			Точность срабатывания метода			
	1 ядро	2 ядра	4 ядра	0°	+90°	-90°	180°
torch-MTCNN	0.056	0.040	0.033	**99.90**	81.05	72.05	69.40
+ обрезка/уменьшение	**0.013**	**0.009**	**0.007**	**99.95**	79.10	70.20	65.95
torch-MTCNN + учёт положения точек	0.112	0.077	0.064	99.75	97.90	96.45	79.35
+ обрезка/уменьшение	0.070	0.049	0.041	99.75	97.70	96.25	78.40
RetinaFace + учёт положения точек	3.720	2.130	1.355	**99.90**	**99.95**	**100.00**	99.40
+ обрезка/уменьшение	2.120	1.220	0.775	**99.90**	**99.95**	**100.00**	99.40
Алгоритм [5] - torch-MTCNN x 4	0.208	0.146	0.121	99.60	**99.95**	99.75	**99.90**
+ обрезка/уменьшение	0.038	0.027	0.023	99.55	**99.90**	**99.90**	**99.90**
Алгоритм [5] - onnx-YuNet x 4	0.041	0.042	0.034	**99.95**	**100.00**	**100.00**	**99.95**
+ обрезка/уменьшение	0.016	0.017	0.015	**99.95**	**99.95**	**99.95**	**99.95**
Дообучение ResNet-50	0.101	0.053	0.035	**99.95**	**99.95**	**100.00**	**99.90**
Дообучение ResNet-34	0.076	0.041	0.027	**99.90**	**100.00**	**100.00**	**99.95**

[1] Среднее количество новых фотографий на одну проверку около 300, среди них в среднем около 0.03 % фотографий с неправильной ориентацией и 2 % фотографий лежащих детей. Фотографии с поворотом 180° достаточно редки.

Таблица 2. **Сравнение скорости алгоритмов при запуске в параллельных процессах**

Метод	Среднее время обработки одного изображения, с				
	1 процесс	2 процесса	4 процесса	8 процессов	Опер. пам. на 1 процесс, Мб
[5] - torch-MTCNN + обр./уменьш.	0.047 ± 0.007	0.023 ± 0.003	0.014 ± 0.001	0.015 ± 0.001	312
[5] - onnx-YuNet + обр./уменьш.	0.019 ± 0.004	0.009 ± 0.000	0.011 ± 0.000	0.005 ± 0.001	38

Поскольку точность для лидирующих алгоритмов оказалась очень близка к 100 %, мы дополнительно протестировали лучшие алгоритмы на 30253 изображениях с правильной ориентацией. Результаты приведены в таблице 3. Было замечено, что при применении обрезки и уменьшения, метод onnx-YuNet начинает выдавать больше ложных срабатываний на угле 0°. Поэтому мы применили следующее улучшение: если для уменьшенного изображения ориентация определена как неправильная, то проводится дополнительное тестирование на исходном изображении. Это сокращает количество ложных срабатываний на изображениях с правильной ориентацией, но практически не увеличивает время работы, поскольку на большинстве изображений в базе лицо ориентировано правильно. Более точно в тесте с 8 процессами алгоритм обрабатывает одно изображение в среднем за 5 мс.

Таблица 3. **Измерение точности работы методов на большой выборке**

Название метода	Точность срабатывания метода			
	0°	+90°	-90°	180°
Алгоритм [5] - torch-MTCNN x 4	99.63	99.93	99.82	99.88
+ обрезка/уменьшение	99.69	99.90	99.88	99.79
Алгоритм [5] - onnx-YuNet x 4	**99.88**	**99.98**	**99.98**	**99.94**
+ обрезка/уменьшение	99.74	**99.96**	**99.96**	99.84
+ повторная проверка при угле ≠ 0°	**99.89**	**99.96**	**99.96**	99.84

Таблица 4. **Качество методов распознавания лежащих детей**

Метод	Валидационная выборка			Тестовая выборка		
	BA, %	TNR, %	TPR, %	BA, %	TNR, %	TPR, %
ResNet50, дообучение, weight=54	97.40	97.01	97.80	96.57	97.00	96.14
+ оба поворота при обучении	**97.95**	97.01	98.90	**96.64**	97.14	96.14
+ случайный поворот на ± 10°	**97.95**	97.01	98.90			

2. Распознавание лежащих детей

Мы взяли набор фотографий детей из 31000 изображений (из открытого источника) и разбили его на обучающую, валидационную и тестовую выборку в отношении 10:3:10.

В таблице 4 показаны результаты экспериментов. Мы проводили тестирование на тестовой выборке только в случае, если сбалансированная точность (BA) превышает качество, достигнутое в предыдущих экспериментах. Было замечено, что использование обоих поворотов в обучающей выборке немного улучшает точность. Более точно мы дублируем обучающую выборку и каждый второй раз выбираем поворот на 90°, противоположный предыдущему (в случае классов 0 и 1).

Обсуждение результатов

1. *Распознавание неправильной ориентации фотографии на основе детектора лиц*

Эксперименты показали, что хотя первая версия алгоритма достаточно точно срабатывает на вертикальных лицах, количество ошибок на фотографиях с неправильной ориентацией довольно большое – 20-30 %. Значительное улучшение точности было достигнуто за счёт применения метода [5]. Для повышения скорости алгоритма был использован детектор YuNet [14]. Для сокращения числа ложных срабатываний предложено проводить дополнительную проверку на изображении исходного размера в случае, если распознана неправильная ориентация.

Несмотря на то что метод RetinaFace [13] показал достаточно высокую точность, а также автоматический внутренний параллелизм, он работает примерно в 70 раз медленнее, и его применение без GPU не подходит под требования задачи.

Алгоритмы распознавания неверной ориентации на основе ResNet также показывают высокую точность. Однако, согласно замерам, на 4 ядрах скорость алгоритма на основе YuNet в 2.5 раза превышает скорость алгоритма на основе ResNet, даже при использовании легковесной сети ResNet-34. Поэтому в задачах, где допускается некоторое количество ложных срабатываний, разметка данных затруднительна и есть требования к скорости работы, метод на основе YuNet является предпочтительным. Дальнейшее улучшение точности возможно в случае применения нейронных сетей для отсеивания ложных срабатываний.

2. Распознавание лежачих детей-инвалидов

Метод показал способность отсеивать 96.14 % лежащих детей при отсеивании неверно ориентированной фотографии в 2.86 % случаев. Такая точность достаточна в нашем сценарии использования.

Поскольку мы распознавали всех лежащих детей, а не только детей-инвалидов, как подлежащих отсеиванию, это влияет на количество пропусков, но таких изображений в базе немного. Замер качества работы объединённого метода на независимой выборке из 31000 изображений показал ошибку первого рода 10 % и ошибку второго рода 0,05 %.

В продолжении данного исследования нам удалось сбалансировать ошибку первого и второго родов за счёт прямого предсказания правильности ориентации. Была получена ошибка первого рода 0 % и ошибка второго рода 0,05 %.

Выводы

В данной работе представлены два метода. Первый метод быстро распознаёт неверно повёрнутые изображения с помощью детектора лиц, затрачивая на это 5 мс на одно изображение без использования GPU. Второй метод отсеивает ложные срабатывания первого метода, так что итоговые ошибки первого и второго родов составляют 10 % и 0.05 % соответственно.

Список литературы

1. Vailaya, A., Zhang, H., Yang, C., et al. Automatic image orientation detection / A. Vailaya, H. Zhang, C. Yang, et al. *IEEE Transactions on Image Processing*, 2002, 11(7), 746-755.

2. Joshi, U., & Guerzhoy, M. Automatic photo orientation detection with convolutional neural networks. *14th IEEE Conference on Computer and Robot Vision (CRV)*, 2017, pp. 103-108.

3. Tolstaya, E. Content-based image orientation recognition. *GraphiCon 2007*, pp. 158-161.

4. Shima, Y., Nakashima, Y., & Yasuda, M. Detecting orientation of in-plain rotated face images based on category classification by deep learning. *TENCON 2017 - 2017 IEEE Region 10 Conference*, pp. 127-132.

5. US Patent US7565030B2

6. Rowley, H. A., Baluja, S., & Kanade, T. Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 1998, 20(1), 23-38.

7. Viola, P., & Jones, M. J. Robust real-time face detection. *Int. journal of computer vision*, 2004, 57(2), 137-154.

8. Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

9. Krizhevsky, A., Sutskever, I., & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012, 25.

10. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal processing letters*, 2016, 23(10), 1499-1503.

11. Tim Esler. Facenet-pytorch library. https://github.com/timesler/facenet-pytorch

12. Iván de Paz Centeno. ipazc/mtcnn: v1.0.0. *Zenodo*, 2024, doi: 10.5281/zenodo.13901378

13. Deng, J., Guo, J., Ververas, E., Kotsia, I., & Zafeiriou, S. Retinaface: Single-shot multi-level face localisation in the wild. *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2020, pp. 5203-5212.

14. Wu, W., Peng, H., & Yu, S. Yunet: A tiny millisecond-level face detector. *Machine Intelligence Research*, 2023, 20(5), 656-665.

15. LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 1989, 2.

16. Dalal, N., & Triggs, B. Histograms of oriented gradients for human detection. *2005 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886-893.

17. Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. Visual categorization with bags of keypoints. *Workshop on statistical learning in computer vision, ECCV 2004*, vol. 1, no. 1-22, pp. 1-2.

18. Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004, 60(2), 91-110.

19. Perronnin, F., & Dance, C. Fisher kernels on visual vocabularies for image categorization. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8.

20. Lazebnik, S., Schmid, C., & Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *2006 IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 2169-2178.

21. Krapac, J., Verbeek, J., & Jurie, F. Modeling spatial layout with fisher vectors for image categorization. *2011 IEEE International Conference on Computer Vision*, pp. 1487-1494.

22. Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. The Pascal Visual Object Classes challenge: A retrospective. *International Journal of Computer Vision*, 2015, 111(1), 98-136.

23. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248-255.

24. Simonyan, K., & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *2014 arXiv preprint*. *arXiv:1409.1556.*

25. Ioffe, S., & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *2015 International Conference on Machine Learning*, pp. 448-456.

26. He, K., Zhang, X., Ren, S., & Sun, J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778.

27. Deng, J., Guo, J., Xue, N., & Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. *2019 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4690-4699.

28. Oquab, M., Bottou, L., Laptev, I., & Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1717-1724.

29. Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. Random Erasing Data Augmentation. *2020 AAAI Conference on Artificial Intelligence*, Vol. 34, No. 07, pp. 13001-13008.

30. DeVries, T., & Taylor, G. W. Improved regularization of convolutional neural networks with cutout. *2017 arXiv preprint. arXiv:1708.04552*.

31. Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. mixup: Beyond empirical risk minimization. *2017 arXiv preprint*. *arXiv:1710.09412*.

32. Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. *2019 IEEE International Conference on Computer Vision*, pp. 6023-6032.

33. Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., & Keutzer, K. A survey of quantization methods for efficient neural network inference. In: *Low-power computer vision*, 2022, pp. 291-326.

34. Hinton, G., Vinyals, O., & Dean, J. Distilling the knowledge in a neural network. 2015, *arXiv:1503.02531*.

35. Shadrikov, A. Achieving better kinship recognition through better baseline. *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pp. 872-876.