

МРНТИ 50.05.13
УДК 004.852

<https://doi.org/10.51889/2021-4.1728-7901.11>

Н. Дүйсеғалиева^{1}, А. Адамова²*

¹*Казакский агротехнический университет им. С.Сейфуллина, г. Нур-Султан, Казакстан*

²*Astana IT University, г. Нур-Султан, Казакстан*

**e-mail: nasipzhan@mail.ru*

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АТТЕСТАЦИИ С ПОМОЩЬЮ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЯ

Аннотация

В нынешней пандемической ситуации Covid-19 все больше и больше уделяется внимание онлайн-удаленным доступам с использованием машинного обучения. Однако с проведением различных экзаменов, возникли проблемы. И если это будет нормой в дальнейшем, то необходимо какое-то решение. Разработка программного обеспечения для аттестации с помощью алгоритмов машинного обучения на Python, которая может контролировать обучающихся системой веб-камерой и микрофоном. Внедрение этого процесса в больших масштабах будет давать возможность выполнения объемной работы за короткий срок. В работе рассматриваются и анализируются функции, которые могут сделать систему контроля веб-различными технологиями и такие языки, как HTML3, CSS5, BOOTSTRAP5, Django, Python, PostgreSQL и необходимые библиотеки.

Ключевые слова: система онлайн-экзаменов, прокторинг, идентификация, обнаружение, машинное обучение, сверточные нейронные сети, аттестация, OpenCV. YOLO, DNN, DarkNet.

Аңдатпа

Н. Дүйсеғалиева¹, А. Адамова²

¹*С.Сейфуллин атындағы Қазақ агротехникалық университеті, Нұр-Сұлтан қ., Қазақстан*

²*Astana IT University, Нұр-Сұлтан қ., Қазақстан*

МАШИНАЛЫҚ ОҚЫТУ АЛГОРИТМДЕРІН ҚОЛДАНА ОТЫРЫП АТТЕСТАТТАУ БАҒДАРЛАМАСЫН ЖАСАҚТАУ

Қазіргі таңда Covid-19 пандемиясы жағдайында машиналық оқытуды қолдануға негізделген онлайн платформаларға ерекше назар аударылуда. Десек те, түрлі емтихандарды өткізу барысында бірқатар қиындықтар туындады. Егер бұл қалыпты жағдай болса, бұл қиындықтардың шешімі болу қажет. Білім алушыларды жүйелі веб-камера және микрофонмен бақылауға мүмкіндік беретін Python арқылы машиналық оқыту алгоритмінің көмегімен аттестацияға арналған бағдарламаны дайындау ұтымды шешім болмақ. Аталған үдерісті кең ауқымда енгізсек, бұл көлемді жұмысты қысқа мерзімде атқаруға жағдай жасайды. Мақалада веб-камера арқылы бақылау жүйесін жетілдіретін қызмет түрлері қарастырылып, талданады. Оған қоса, бағдарламаны дайындау барысында түрлі технологиялар мен HTML3, CSS5, BOOTSTRAP5, Django, Python, PostgreSQL, сонымен бірге қажетті кітапхана қоры қолданылады.

Түйін сөздер: онлайн емтихан жүйесі, прокторинг, сәйкестендіру, ашу, машиналық оқыту, конволюционды нейрондық желілер, аттестация, OpenCV. YOLO, DNN, DarkNet.

Abstract

PROCTORING SYSTEM USING MACHINE LEARNING ALGORITHMS TO DEVELOP ATTESTATION SOFTWARE

Duisegaliyeva N.¹, Adamova A.²

¹*S.Seifullin Kazakh AgroTechnical University, Nur-Sultan, Kazakhstan*

²*Astana IT University, Nur-Sulatan, Kazakhstan*

In this pandemic situation of Covid-19 pandemic, there is an increasing focus on online remote access using machine learning. However, there has been a great trouble in conducting examinations and if it is the way we are living is to be the new norm there needs to be some solution. Developing attestation software using machine-learning algorithms in Python that can monitor learners with a system webcam and microphone. Further implementing this process at a large scale will make it possible to complete large work in a short period. This paper focuses on the tests and analyzes the creation of functions that can make the existing webcam control system more advanced. In addition, in the development process, various technologies and languages are used such as HTML3, CSS5, BOOTSTRAP5, Django, Python, PostgreSQL and the necessary libraries.

Keywords: online exam system, proctoring, identification, discovery, machine learning, convolutional neural networks, attestation, OpenCV, YOLO, DNN, DarkNet.

Введение

В современном мире с каждым днем растет интерес к технологиям для обучения и преподавания. Наряду с этим огромное влияние на жизнь всех людей оказала ситуация с пандемией Covid-19, в связи с чем, на сегодняшний день, все больше внимания уделяется онлайн-приложениям и программам удаленного доступа с использованием машинного обучения. Это приводит к увеличению спроса на онлайн-образование и дистанционное обучение. Расстояние или гибкость между обучающимися и преподавателями, а также между экзаменаторами и аттестуемыми преподавателями в среде онлайн-обучения могут, по сути, способствовать проблемам поддержания целостности онлайн-оценивания, эта проблема также была подчеркнута Холлистером и Беренсоном: «наиболее часто упоминаемая проблема онлайн-оценивания – это поддержание академической честности» [1]. Это вызвало как академические, так и неакадемические проблемы, такие как разработка и проведение онлайн-экзаменов, а также мониторинг поведения аттестуемых или обучающихся во время экзаменов [2]. Для проведения экзаменов в большинстве случаев организовывали одну ссылку на конференцию с помощью бесплатных приложений, таких как Zoom, Teams, Google Meet и т.д., используя которую обучающиеся могли войти в систему, в то время как преподаватели наблюдали за ними.

Онлайн-прокторинг – один из способов решения этой проблемы. Благодаря технологическим вспомогательным средствам, таким как блокировка компьютера или системы, мониторинг нажатия клавиш, возможность остановки или запуска теста и многие другие вспомогательные процессы контроля, которые теперь легко интегрированы в процесс мониторинга, онлайн-контроль теперь стал стать жизнеспособным решением. На сегодняшний день существует бесплатные и платные системы онлайн-прокторингов, которые находятся на очень ранней стадии, имеющие всевозможные функции контроля, но точность этих функций все еще находится под пристальным вниманием. Веб-камера, микрофон и совместное использование экрана – это три основных фактора, которые присущи проведению онлайн-экзаменов. Из них данные, собранные с веб-камеры, все еще не используются в полной мере, т.к. в современных системах онлайн-прокторинга реализован только процесс распознавания лиц.

В работе будут рассмотрены преимущества использования технологий алгоритмов машинного обучения для разработки программного обучения, для проведения аттестации преподавателей школ Республики Казахстан. Для этого разрабатывается более простая и надежная система, в которой будут использоваться Tensorflow 2.5 и библиотеки OpenCV, предназначенные для обучения и развертывания модели обнаружения лиц. Серверная часть будет полностью разработана с использованием фреймворка Django на Python3.

Основы проектирование системы

Благодаря технологиям удаленного контроля и наблюдения на основе искусственного интеллекта можно гарантировать, что учащиеся не будут вовлечены в мошенничество или недобросовестные методы во время экзамена. Вместе с тем прокторинг применяется не только в сфере образования, но и в других сферах, таких как: подбор персонала, онлайн-обучение, тестирование и аттестация персонала. В связи с этим в современном мире создания системы для прокторинга для аттестации является одной из широко исследуемых сфер. Прокторинг – система, позволяющая следить за тестированием или экзаменом в онлайн-режиме [3].

Планируется разработка веб-приложения, доступ к которому будет осуществляться легким и безопасным путем простого подключения к сети Интернет. Интерфейс для системы будет разработан с использованием языков программирования HTML3, CSS5 и некоторых компонентов начальной загрузки.

Процесс прокторинга на основе машинного обучения повторяется тысячи раз для разработки, обучения и уточнения каждого события, определенного в системе. Событие может быть единичным поведением или свидетельствовать о мошенничестве, краже контента или мошенничестве. Например, если кто-то смотрит влево за кадром, это можно рассматривать как одну точку данных, и эта конкретная часть видео сегментируется и помечается. Как только количество таких точек, данных с одинаковым поведением превышает установленный предел, инициируется непрерывное событие

построения, обучения и уточнения. Нижеперечисленные технологии искусственного интеллекта используются для удаленного контроля:

- ✓ Домашней страницы;
- ✓ Аутентификации учетной записи с проверкой изображения (Рисунок 1);
- ✓ Главной учетной записи пользователя и панели управления;
- ✓ Страницы пользовательских функций.

```
class RegisterView(TemplateView):
    template_name = "account/register.html"

    def dispatch(self, request, *args, **kwargs):
        if request.method == 'POST':
            username = request.POST.get('username')
            email = request.POST.get('email')
            password = request.POST.get('password')
            User.objects.create_user(username, email, password)
            return redirect("/")

        return render(request, self.template_name)
```

Рисунок 1. Обработка регистрации

База данных будет выполнена с использованием фреймворка Django и PostgreSQL. Подобная база данных сможет включать в себя три типа профилей пользователей: администратор, обучающийся, преподаватель. Отдельный доступ для входа и регистрации будет предоставлен для каждого пользователя. Также эта база данных будет иметь полную логику системы тестов, идентификацию пользователя и на основе этого будет функционировать доступ и работа конкретных функций пользователя. На данный момент в системе проктора реализованы следующие функции:

- ✓ Распознавание объектов;
- ✓ Распознавание лиц;
- ✓ Обнаружение движения глаз;
- ✓ Обнаружение рта.

Предлагаемая система включает в себя шесть основных компонентов, которые отображают ключевые аспекты оценки поведения обучающегося: проверка тестируемого, обнаружение звуков, активное окно и переключение обнаружения окон, оценка взгляда, идентификация телефона и человека. Комбинируя компоненты предлагаемой системы с многопоточными компонентами и применяя условия, можно классифицировать функции независимо от того, мошенничает тестируемый или нет в тот или иной момент во время экзамена. Чтобы оценить различные виды мошенничества при сдаче экзаменов, программа собирает мультимедийные данные, такие как аудио и видео файлы.

Для организации прокторинга и обеспечения эффективного способа организации аттестационных тестов в учреждениях можно использовать различные технологии машинного обучения.

Учёными из КСР Института инженерии и технологий были определены некоторые недостатки существующих онлайн-прокторингов, которые заключаются в том, что они позволяют преподавателю изучать движения обучающихся. Следует учитывать, что обучающиеся, в качестве подтверждения личности, могут показать перед камерой фотографии, не являющиеся их собственными, тем самым нарушая академическую честность. Размещение веб-камеры и расстояние до аттестуемого, а также качество передаваемого изображения, позволяет другому человеку сдавать экзамен вместо другого человека. Еще один недостаток таких систем в том, что они не идентифицируют фотографии или изображения, а просто распознают, что на представленных фотографиях изображен настоящий человек, что дает возможность мошенничать обучающимся. [4].

Проектирование подобных систем во многих случаях является затруднительным, поскольку изначально сложно учесть все детали, необходимые для разработки системы. Большинство существующих систем фокусируются на определении присутствия более одного человека в кадре, некоторые системы специализированы на выявлении фактов мошенничества, эти функции не очень точны и в некоторых случаях не позволяют выявить нарушения. Иногда, из-за неточности модели, система может работать некорректно и дать ложные результаты.

Поэтому более выгодным решением будет учет возникающих неопределенностей уже в процессе функционирования системы и последующая работа с ними. Основной целью является создание онлайн системы прокторинга с использованием машинного обучения на Python для мониторинга действий и выявления недостатков в существующей системе. С помощью этой системы появится возможность контролировать нескольких аттестуемых одновременно и более точно проводить идентификацию личности и вести подсчет количества человек, присутствующих в комнате помимо аттестуемого.

Алгоритмы машинного обучения могут анализировать ответы и выдавать в программе результаты проводимых работ, в данном случае аттестации. В области разработки программного обеспечения существует несколько подходов к программам, таких как усилий, безопасности, качества, ошибок, затрат и повторного использования.

В результате обучения система учится оптимально реагировать на изменение внешних условий (в результате которых происходит увеличение/уменьшение времени отклика) и накапливает знания о том, как нужно изменить свое состояние, чтобы достичь максимального вознаграждения.



Рисунок 2. Системная архитектура

Все кадры, которые используются для аутентификации и мониторинга пользователя, берутся с веб-камеры, расположенной на сайте пользователя. Существует ряд функций и действий, которые находятся под контролем, причем каждая из этих функций и действий требует создания отдельных моделей машинного обучения. Так, лицо пользователя распознается и повторно извлекается из одного кадра выборки с заданной частотой. После прохождения первого сопоставления пользователь либо успешно входит в систему, либо не может войти в систему и поэтому вынужден выйти из нее. После прохождения фазы аутентификации следует итерация. Если процент несоответствия черт лица между новым сохраненным шаблоном (снимком, сделанным на этапе аутентификации содержащий информацию о лице и фоне пользователя), а выборка кадра в реальном времени превышает предварительно настроенное пороговое значение, выявляется подозрительное поведение и записывается видеоклип, который затем используется для дальнейшей проверки (Рисунок 2).

Методы исследования

Для получения входных данных с веб-камеры и преобразования полученных данных в несколько изображений используется библиотека алгоритмов компьютерного зрения OpenCV (Рисунок 3). После чего создаются журналы, описывающие различные действия человека, которые впоследствии можно будет использовать для определения случаев факта мошенничества.

Для достижения всех вышеупомянутых действий используются predetermined модули и пакеты Python, такие как tensorflow, opencv, dlib, keras, nltk, wget, sklearn, pyaudio, Speech_recognition, обученные модели и базовые модули.

Реализацию всех этих функций необходимо осуществить построив две модели машинного обучения. Первая модель – это реализация на основе классификатора Extra Trees Classifier для обнаружения фальсификации лиц.

```

class VideoCamera(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        (self.grabbed, self.frame) = self.video.read()
        threading.Thread(target=self.update, args=()).start()
    def __del__(self):
        self.video.release()
    def get_frame(self):
        image = self.frame
        _, jpeg = cv2.imencode('.jpg', image)
        return jpeg.tobytes()

    def update(self):
        while True:
            (self.grabbed, self.frame) = self.video.read()

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

```

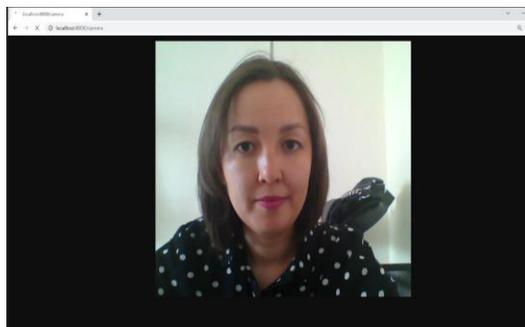


Рисунок 3. Модуль обработки данных с камеры OpenCV2

Следующим шагом является реализация модели на основе сверточных нейронных сетей для определения лицевых ориентиров. Модель обнаружения фальсификации лиц построена с использованием техники Extra Trees Classifier на основе набора данных от Kaggle, который состоит из реальных и поддельных изображений. Модель ориентира лица строится с использованием сверточных нейронных сетей, основанных на наборе данных от Kaggle, который состоит из изображений и точек ориентира лица. В этом случае используется предварительно обрабатываемый набор данных из Kaggle и устанавливаются переменные для обучения и тестирования. Затем разрабатываются сверточные нейронные сети [5].

Распознавание лиц

Для идентификации лиц на заданном изображении существуют несколько алгоритмов, которые можно использовать: Haar Cascade, распознавание лиц Eigen, распознавание лиц Fisher, Adaboost и DNN Open CV/модель Caffe.

Большинство систем распознавания лиц были созданы с использованием алгоритма LBP (Local Binary Pattern) и каскадного классификатора Хаара. Вместо того, чтобы генерировать некоторую графическую информацию из изображения, такую как LBP, каскадный классификатор Хаара смотрит непосредственно на связанные черты лица на основе прямоугольных блоков. Распознавания с использованием Haar Cascade методологии также делает все возможное, чтобы исключить ненужные затраты. Используемый датчик движения удерживает машину в режиме ожидания, что снижает общее энергопотребление [6].

Детектор лица Виолы–Джонса основан на следующих идеях: интегральном представлении изображения, методе построения классификатора на основе алгоритма адаптивного бустинга (AdaBoost), и методе комбинирования классификаторов в каскадную структуру. Эти идеи позволяют построить детектор лица, способный работать в режиме реального времени. Метод главных компонент и вейвлет-преобразование используют для получения характеристик изображения. В задаче распознавания лиц они успешно применяются для сравнения компонент, характеризующих цветные изображения, с компонентами, описывающими неизвестные изображения. Этот детектор обладает крайне низкой вероятностью ложного обнаружения лица. Метод хорошо работает и обнаруживает черты лица даже при наблюдении объекта под небольшим углом, примерно до 30°. При угле наклона больше 30° вероятность обнаружения лица резко падает.

Метод Eigenfaces основан на методе главных компонент (Principal Component Analysis, PCA), который является неконтрольной статистической моделью, тем самым не совсем подходит для данной задачи. В свою очередь метод Fisherfaces дает определенную классовую линейную проекцию, что говорит о том, что он гораздо лучше подходит для возрастной, расовой и гендерной классификации [7]. Метод Fisherfaces уже реализован в OpenCV. OpenCV (Open Source Computer Vision Library) это библиотека алгоритмов компьютерного зрения, обработки изображений и

машинного обучения с открытым исходным кодом. Для того чтобы система классифицировала людей по возрастной, расовой и гендерной группам, необходимо обучить классификаторы модели распознавания Fisherfaces в OpenCV. Для обучения данных классификаторов необходим набор изображений разных людей. [8].

Возможности для распознавания лиц, не включающие модели обучения предоставляют глубинные учебные модули (DNN) в OpenCV, которые также поддерживают различные структуры глубинного обучения, такие как Tensorflow, Caffe, Darch и Darknet. В ходе исследования и анализа существующих модулей была выбрана модель обнаружения лиц, основанная на модуле DNN OpenCV, служащим для классификации изображений по обученной сети GoogLeNet из моделей Caffe. В этом модуле выделяется ряд преимуществ:

- Модуль DNN реализует только рассуждение, в то же время количество кода и компиляция, бегущей накладной головки гораздо меньше, чем в других учебных моделях.
- Модуль DNN обеспечивает ускорение встроенного и графического процессоров, не полагаясь на сторонние библиотеки. Если OpenCV используется в проекте, то с легкостью можно добавить возможности глубинного обучения в исходный элемент через модуль DNN.
- Модуль DNN поддерживает различные форматы сетевых моделей, тем самым пользователи могут напрямую использовать его без дополнительных сетевых моделей.
- Модуль DNN поддерживает несколько типов слоев сети, что в основном охватывает общие потребности в сети.

Многоразовый код для распознавания лиц

Модель Caffe основана на Single Shot-Multibox Detector (SSD) и использует архитектуру ResNet-10 в качестве основы. Определена функция поиска лиц по изображению, которая использует модель на основе DNN и возвращает выходные данные с координатами четырех углов лица. Входными данными является изображение в виде массива numpy и модели, а выходными данными является массив, состоящий из четырех координат лица. Во-первых, изображение, переданное в качестве входных данных, преобразуется в формат blob с помощью OpenCV. Затем с помощью модели определяется достоверность или значение вероятности наличия лица в кадре. Если это значение вероятности больше 0.5, то считается, что лицо обнаружено, тогда четыре координаты вокруг лица оцениваются и передаются в качестве выходных данных. Также определена функция для рисования прямоугольника, которая рисует прямоугольник на основе четырех координат, полученных в качестве выходных данных. Входными данными является изображение в виде массива numpy, а массив, содержащий четыре координаты изображения и прямоугольник, нарисованный над координатами, являются выходными данными. Прямоугольники рисуются по координатам с помощью функции прямоугольника OpenCV. Изображение и четыре координаты передаются в качестве входных данных в функцию прямоугольника.

Выявление наличия более чем одного человека и мобильного телефона

Основной задачей стоит определить является ли лицо перед камерой реальным человеком, для этого реальное зеркальное отражение тестируемого можем использовать как метод фальсификации лица. Существует несколько методов обнаружения фальсификации изображения (лица), среди них:

- ✓ DPM (*Deformable Part Model*) и R-CNN
- ✓ Faster R-CNN
- ✓ YOLO.

Модели деформируемых деталей (DPM) и сверхточные нейронные сети (R-CNN) – это два наиболее широко используемых инструмента для визуального распознавания объектов. DPM – это графическая модель (марковские случайные поля), а R-CNN – это нелинейные классификаторы «черного ящика». Объекты появляются на изображениях во всех масштабах. Стандартный способ справиться с этим фактом – запустить детектор в нескольких масштабах с использованием пирамиды изображений [9].

Предлагаемый метод FARCNN основан на структуре Faster R-CNN, служащий для защиты от фальсификации данных (подделки изображения лица) объединяет два этапа традиционного обнаружения подделки лиц (обнаружение и обрезка области лица, извлечение черт лица) в один этап. Исходные изображения без кадрирования загружаются в FARCNN, а ограничивающие рамки и оценки классификации выводятся напрямую, что удовлетворяет реалистичным сценариям

применения. В связи с тенденцией к машинному обучению R-CNN добились значительного улучшения характеристик обнаружения объектов, в отличие от традиционного обнаружения подделки лица, мы рассматриваем эту задачу как трехстороннюю классификацию для различия реального лица, поддельного лица и фона, которая сочетает в себе этап обнаружения лица и обнаружение фальсификации данных [10].

С помощью модели YOLOv3 на Tensorflow 2.0 можно определить количество человек в комнате, где тестируемый сдавал экзамен. Здесь предварительно обученная модель YOLOv3 может использоваться для классификации 80 объектов, она сверхбыстрая и почти такая же точная, как SSD, что позволяет подсчитать количество вещей в комнате [4].

Таблица 1. Характеристики методов распознавания фальсификации лиц

Методы	Преимущества	Недостатки	Время тестирования на изображение
DPM (Deformable Part Model) и R-CNN	<ul style="list-style-type: none"> ● обнаружение объекта 	<ul style="list-style-type: none"> ● больше ошибок локализации и фоновых ошибок ● энергозатратный 	50 секунд
Faster R-CNN	<ul style="list-style-type: none"> ● обнаружение малого или далекого объекта 	<ul style="list-style-type: none"> ● больше фоновых ошибок 	2 секунд
YOLO	<ul style="list-style-type: none"> ● скорость вывода ● обнаружение малого или далекого объекта ● нет перекрывающихся блоков 	<ul style="list-style-type: none"> ● больше ошибок локализации 	0,1 секунд

YOLOv3 использует несколько подфункций, которые определены как часть процесса создания DarkNet, и, наконец, возвращает модель KERAS в качестве выходных данных. Функция использует все ранее определенные функции и создает модель YOLOv3, после чего снова возвращает модель YOLOv3 в качестве выходных данных. Эта функция вызывается, и полученная модель сохраняется в переменной. Веб-камера запускается при помощи функции OpenCV VideoCapture. Каждый кадр веб-камеры снимается и обрабатывается, в следствии чего получается изображение (Рисунок 4). Размер полученного изображения изменяется до размера, поддерживаемого моделью YOLOv3. Это изображение обрабатывается на основе модели и ящиков объектов, оценка прогноза каждого класса и количество объектов каждого класса передаются в качестве входных данных. На основе полученных данных, если счетчик класса «человек» больше единицы, более одного человека печатается в качестве выходных данных (Рисунок 5), если счетчик класса «человек» равен нулю, ни один человек не печатается в качестве выходных данных, и если в классе обнаружен «мобильный телефон» (Рисунок 6), то обнаруженный телефон распечатывается в качестве выходных данных. Все эти выходные данные также сохраняются в файле журнала [5].

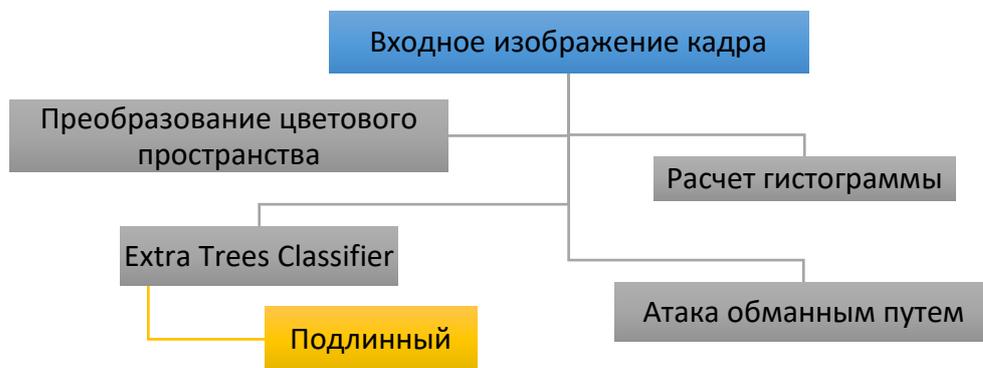


Рисунок 4. Блок-схема, описывающая различные этапы процесса обнаружения фальсификации лиц

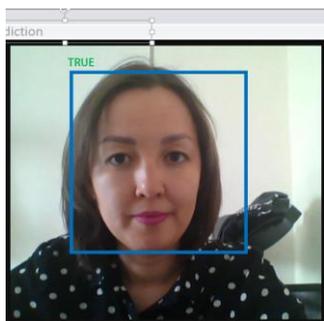


Рисунок 5. Выход для обнаружения подделки лица, когда фотография человека отображается перед веб-камерой

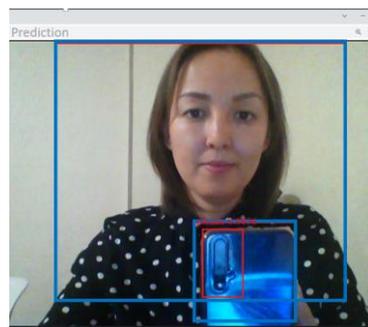


Рисунок 6. Выходные данные для определения присутствия более чем одного человека, когда перед веб-камерой находятся человек и мобильный телефон

Для начала необходимо импортировать файлы из python, которые содержат функции для обнаружения лиц. Затем определяется функция для расчета гистограммы изображения RGB. Изображение в виде массива numpy передается в качестве входных данных. Массив гистограмм создается при помощи функции calcHist в OpenCV. Массив гистограммы преобразуется в массив numpy и возвращается в качестве вывода. Модель обнаружения лиц хранится в переменной. Веб-камера запускается с помощью функции OpenCV VideoCapture, где каждый кадр веб-камеры снимается и обрабатывается, тем самым получается изображение. Преобразование цвета выполняется на полученном изображении с использованием функции OpenCV cvtColor. Гистограммы рассчитаны для преобразований двухцветного пространства. Теперь производятся дальнейшие расчеты и рассчитывается вероятность того, что изображение окажется ложным. Если значение вероятности больше или равно 0.7, то вывод отображается как поддельное изображение, иначе вывод отображается как реальное изображение. В случае, если отображаемый вывод является поддельным изображением, вывод сохраняется в файле журнала [5].

Анализируя аппаратное и программное обеспечения, которые необходимы для запуска системы прокторинга было решено создать веб-приложение, использующее протокол HTTPS для взаимодействия между клиентом и сервером. Технические требования для создания подобной системы перечислены в представленной ниже таблице 2.

Таблица 2. Технические требования

Процессор	4 ядра, тактовая частота 2.90 ГГц и выше
Платформа	32-х или 64-х разрядная
Оперативная память	10 ГБ и выше
Жесткий диск	80 ГБ свободного объема* и выше
Необходимые приложения	Firefox, Internet Explorer, Opera, Yandex, Google, Chrome, Safari
Среды разработки	Visual Studio code, PyCharm

Заключение

Данная работа дает представление о процессе проектирования и разработки системы онлайн-экзаменов с использованием веб-приложения и мониторинга пользователя на протяжении всего сеанса с использованием различных алгоритмов и библиотек машинного обучения. У данной формы контроля можно выявить как недостатки, так и преимущества. Однако, следует отметить два основных преимущества прокторинга, первое – объективная форма оценки знаний, второе – прокторинг является инструментом для внедрения принципов академической честности.

Как было отмечено в работе, использование технологий алгоритмов машинного обучения для разработки программного обучения для проведения аттестации преподавателей школ Республики Казахстан является преимуществом при разработке надежной системы прокторинга. Реализация предлагаемого веб-приложения осуществляется легким и безопасным путем простого подключения к

сети Интернет. Также следует отметить, что интерфейс для системы разработан с использованием языков программирования HTML3, CSS5, что облегчает обучение работы с ним.

Предлагаемая система включает в себя шесть основных компонентов, которые отображают ключевые аспекты оценки поведения, обучающегося: проверка тестируемого, распознавание звуков, активное окно и переключение обнаружения окон, оценка взгляда, идентификация телефона и человека. Было разработано несколько функций для наблюдения за веб-камерой, такие как модель YOLOv3 была построена с использованием весов YOLOv3, и эта модель использовалась вместе с OpenCV для определения количества людей в прямой трансляции веб-камеры, а также для обнаружения наличия мобильного телефона. Можно создавать модели с более новыми или более продвинутыми конфигурациями нейронной сети. Обладая лучшими аппаратными возможностями, можно создавать более точные модели, обучая их большему количеству эпох. В комбинации компоненты этой системы взаимодействуют с многопоточными компонентами, при этом применяются условия возможности классифицирования функций независимо от того, мошенничает ли тестируемый в тот или иной момент проведения экзамена. Для этого программа собирает мультимедийные данные: аудио и видео файлы.

В дальнейшем планируется исследование и разработка моделей для распознавания звука, положения глаз и рта, при этом не исключается возможность повторного использования уже разработанных исходных алгоритмов, методологии машинного обучения. Также есть возможности для улучшения частоты кадров прямой трансляции веб-камеры, получаемой с помощью OpenCV.

References:

- 1 Cramp, J., Medlin, J. F., Lake, P., & Sharp, C. (2019). *Lessons learned from implementing remotely invigilated online exams*. *Journal of University Teaching & Learning Practice*, 16(1), 10. Retrieved from <https://ro.uow.edu.au/jutlp/vol16/iss1/10/>
- 2 CAUDIT/ACODE (2020). *CAUDIT/ACODE Forum on e-exams*. Retrieved April 14, 2020, from <https://www.acode.edu.au/>
- 3 Brown, V. (2018). *Evaluating technology to prevent academic integrity violations in online environments*. *Online Journal of Distance Learning Administration*, 21(1). Retrieved from <https://www.westga.edu/~distance/ojdla/spring211/brown211.html>
- 4 DR.B.Kalaavathi1, Sangeetha M, Chowmiya S B, Vaishnavi V, Pooja Shree K (April, 2021) *Automating Online Proctoring System/ Director and Professor, KSR Institute for Engineering and Technology, Tiruchengode.*
- 5 S. Harish1, D. Rajalakshmi, T. Ramesh, S. Ganesh Ram, M. Dharmendra (2021), *New Features for Webcam Proctoring Using Python and Opencv*, ISSN: 2237-0722
- 6 Anirban Chakraborty, Shilpa Sharma (2021), *Real Time Face Detection And Recognition System Using Haar Cascade Classifier And Neural Networks*
- 7 http://docs.opencv.org/trunk/modules/contrib/doc/facerec/tutorial/facerec_gender_classification.html
- 8 Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, Chi Zhang (Aug 2021), *DeFRCN: Decoupled Faster R-CNN for Few-Shot Object Detection*
- 9 Haonan Chen , Yaowu Chen, Xiang Tian And Rongxin Jiang (December 9, 2019), *A Cascade Face Spoofing Detector Based on Face Anti-Spoofing R-CNN and Improved Retinex LBP/ Digital Object Identifier 10.1109/ACCESS.2019.2955383*
- 10 Ross Girshick, Forrest Iandola, Trevor Darrell, Jitendra Malik, Microsoft Research (2015), *UC Berkeley Deformable Part Models are Convolutional Neural Networks.*