DOI 10.37882/2223-2966.2023.8-2.13

РАСПОЗНАВАНИЕ ФЕЙКОВОГО (ПОДДЕЛЬНОГО) ВИДЕОКОНТЕНТА, СИНТЕЗИРОВАННОГО С ПОМОЩЬЮ ТЕХНОЛОГИИ DEEPFAKE АЛГОРИТМА GENERATIVE ADVERSARIAL NETWORK (GAN)

Джуров Александр Андреевич
Аспирант,
Донской государственный технический университет
sashaz1696@yandex.ru
Черкесова Лариса Владимировна
Д.ф.–м.н., профессор,
Донской государственный технический университет
chia2002@inbox.ru
Ревякина Елена Александровна
К.т.н., доцент
Донской государственный технический университет
revyelena@yandex.ru

RECOGNITION OF FAKE (SPURIOUS) VIDEO CONTENT SYNTHESIZED USING DEEPFAKE TECHNOLOGY OF THE GENERATIVE ADVERSARIAL NETWORK (GAN) ALGORITHM

A. Dzhurov
L. Cherckesova
E. Revyakina

Summary. In the modern world, one of the main and actual problems is fake (false) content: news, videos, photos, etc. At the early stage of the development of DeepFake imitation technology, it was used mainly by amateur users to synthesize entertaining multimedia content by comparing people's facial expressions and phrases said, as a rule, by recognizable personalities, to create fake news that looks authentic. But the political situation has changed, and DeepFake technology has been used not only to compromise undesirable persons, but also for disinformation and political agitation, as an integral part of the information war. The purpose of study: software implementation of the video content recognition algorithm synthesized using the DeepFake technology of the Generative Adversarial Networks (GAN) algorithm with acceptable correctness and accuracy. The paper proposes the software implementation that analyzes video content and makes decision about its authenticity. The main architectures of the GAN algorithm are presented; the results and consequences of using DeepFake technology are considered. The analysis of the features of the Xception and ResNeXt models trained using neural networks is carried out. Methods: for the operation of the system, the appropriate neural networks were selected based on the results of their productivity. The software implementation uses ResNeXt and XceptionNet models, as well as a pre-trained human face recognition model BlazeFace, used for face recognition on extracted images. Results: the Deep_Fake_Recognizer–23 software tool has been created that recognizes fake video content synthesized using DeepFake technology using the GAN algorithm with acceptable correctness and accuracy.

Keywords: information war, false content, DeepFake, Generative Adversarial Networks (GAN) algorithm; neural networks, discriminator.

Аннотация. В современном мире одной из основных и актуальных проблем является фейковый (ложный) контент: новости, видео, фото и т.д. На раннем этапе развития технологии глубокого подражания (подделки) DeepFake она применялась в основном пользователями-любителями, для синтезирования развлекательного мультимедийного контента, путём сопоставления выражений лиц людей и фраз, сказанных, как правило, узнаваемыми личностями, для создания фейковых новостей, выглядящих как подлинные. Но политическая ситуация изменилась, и технология DeepFake стала использоваться не только для компрометации неугодных лиц, но и для дезинформации и политической агитации, как составная часть информационной войны. Цель исследования: программная реализация алгоритма распознавания видеоконтента, синтезированного с помощью технологии DeepFake алгоритма Generative Adversarial Networks (GAN), с приемлемой точностью. В работе предложена программная реализация, анализирующая видеоконтент и принимающая решение о его подлинности. Представлены основные архитектуры алгоритма GAN, рассмотрены результаты и последствия применения технологии DeepFake. Проведен анализ особенностей моделей Xception и ResNeXt, обученных с помощью нейронных сетей. Методы: для работы системы осуществлен выбор соответствующих нейронных сетей на основе результатов их производительности. В программной реализации использованы модели ResNeXt и XceptionNet и предварительно обученная модель распознавания человеческих лиц BlazeFace, применяемая для распознавания лиц на извлеченных изображениях. Результаты: создано программное средство Deep_Fake_Recognizer–23, распознающее фейковый видеоконтент, синтезированный по технологии DeepFake по алгоритму GAN с приемлемой точностью.

Ключевые слова: информационная война, ложный контент, DeepFake, алгоритм Generative Adversarial Networks (GAN); нейронные сети, дискриминатор.

Введение

В работе разрабатывается программная реализация, предназначенная для анализа видеоконтента Web-ресурсов и вывода решения о его поддельности или подлинности с приемлемой вероятностью, используя нейросетевые технологии искусственного интеллекта. Анализу подвергается использование интеллектуальной технологии Deepfake и алгоритма Generative Adversarial Networks (GAN) при создании фейкового видеоконтента с помощью алгоритмов его распознавания.

Технология Deepfake [1] представляет собой методику компьютерного синтеза изображения, основанную на искусственном интеллекте, которая используется для соединения и наложения существующих изображений и видео на исходные изображения или видеоролики. Искусственный интеллект использует синтез изображения человека, объединяя несколько кадров, на которых человек запечатлен с различных ракурсов и с различным выражением лица, после чего синтезирует из них видеоролик [2]. Технология Deepfake представляет собой данные, полученные с помощью синтеза, в содержании которых личность и его лицо из реального видеоряда может быть заменена на другую личность. Как правило, результаты имеют формат видеозаписи, аудио или фото. Технология DeepFake позволяет создавать и заменять различные элементы существующих видеороликов на другие элементы, с помощью искусственного интеллекта и обучения нейронных сетей. Однако всё чаще её используют в корыстных целях. Существуют факты злонамеренного использования технологии DeepFake [3], к которым относятся проблемы социальной стабильности и национальной безопасности, развития организованной преступности, рисков для репутации медийных лиц и обычных граждан, с последующим шантажом и вымогательством денег, и др. Многочисленные угрозы злоумышленников уже были неоднократно реализованы с помощью использования алгоритма Generative Adversarial Networks (GAN), и есть все основания полагать, что в ближайшее время количество поддельного видеоконтента будет только возрастать. Новая угроза использования дипфейков — это дезинформация в политике в рамках информационной войны. Дипфейки быстро создаются и легко распространяются в широкой аудитории [4].

Предлагаемая методология

Алгоритм GAN

В широко распространённом алгоритме Generative Adversarial Networksиспользуется нейросетевые технологии. В терминологии сферы искусственного интеллекта две применяемые искусственные нейронные сети (ИНС) носят название *синтезатора* (*генератора*) и *дискриминатора* (*детектора*) [5].

Генерирующий алгоритм, на вход которого поступают случайные данные, синтезирует уникальный контент. Другая ИНС, являющаяся дискриминатором, проверяет контент, чтобы убедиться, что он соответствует исходным данным. Такая конкуренция двух ИНС, по сути, и составляет основной принцип работы алгоритма GAN. Нейронная сеть — синтезатор выдаёт в качестве результата реалистичный видеоконтент, в том числе, с использованием медийных лиц. Рассматривая синтез изображений через призму математических вычислений, нейросети, синтезирующие картинки Web-контента(статические графические изображения) и видео (динамические изображения) не регистрируют различий, при том, что полученный видеоряд может использоваться для различных целей.

В процессе создания фейкового видеоряда генерируется множество последовательных изображений, обусловленных необходимостью придавать естественность движениям людей, чтобы избежать резкого несогласованного движения частей тела от кадра к кадру. Такая плавность изображения на кадрах видеоряда достигается за счёт различных модификаций алгоритма GAN [6].

Чтобы придать естественность движениям объекта на видео и улучшить его трёхмерное изображение, в нейронную сеть нужно загрузить ряд фотографий объекта, сделанных с разных ракурсов. Если одинаково фотографировать людей, например, с бородой и без бороды, то получить точных результатов не удастся. Поэтому не стоит опасаться, что злоумышленники могут взять фотографии из социальных сетей и создавать дипфейки на основе подобных изображений [7].

Для того чтобы создать качественное искусственное изображение на основе фотографий, понадобится сделать ряд фотоснимков, выполненных с различных ракурсов, вручную создать трёхмерную модель, синтезировать множество отдельных изображений этой 3D–модели и загрузить их в нейросеть [8].

Алгоритм GAN используют две видеопоследовательности: первая — с изображением лица человека, которая используется для замены лица на второй видеопоследовательности, а вторая — исходная, в которой и выполняется замена лица. На качество результата влияет множество характеристик исходного файла и входных данных (например, разрешение и продолжительность видеофайла, выражение лица человека, используемое в синтезе, относительное сходство лиц, освещенность изображения в видеоклипе и т.д.). Процесс предполагает ряд этапов.

Этап 1 — происходит распознавание черт лица человека в видеокадрах, полученных из видеоролика. Для упрощения сложных вычислений, каждый кадр прове-

ряется, и некорректные и/или неудачные отбрасываются. К ним относятся образцы из множества, которые не содержат чётко определяемых человеческих черт. Возможны ситуации, когда лицо закрыто различными предметами, или смазано. На этой стадии можно улучшить качество конечного продукта.

Этап 2 — происходит процесс определения контуров человеческого лица в кадрах, полученных из второго видеоряда. Главным отличием является тот факт, что в данной ситуации, в каждом отдельном образце, необходимо извлечь все лица, находящиеся на картинке, даже если конкретное лицо будет нечётким или мутным, что обусловлено невозможностью провести замену лица человека без соблюдения правил, без которых добиться высокого качества результата не получится.

Этап 3 — обучение искусственной нейронной сети на наборах данных (датасетах), содержащих изображения на видеоряде. Для тренировки нейронной сети необходимо выбрать одну из возможных моделей обучения и подобрать её архитектуру. Обучение ИНС, в свою очередь, является базовой циклической процедурой, выполняемой относительно алгоритма GAN. От качества датасетов, используемых для обучения, зависит и качество работы нейронной сети.

Этап 4 — тренировка работы ИНС. Это наиболее время затратная и ресурсоемкая часть, по длительности она может занять как несколько суток, так и несколько недель. Чем больше продолжительность тренировок, тем лучшими окажутся результаты. На качество результатов, помимо длительности обучения и качества исходного материала, влияет и производительность оборудования.

По результатам обучения, выполненного на 4 этапе, проводится покадровое наложение сгенерированных лиц на изображения, полученные из исходного материала. Возможно использование нескольких режимов наложения.

Этап 5 — конечная стадия алгоритма GAN, заключающаяся в процессе наложения кадров на видеоряд получаемого видеоролика, с точно такой же частотой фреймов и звуковым сопровождением, которые были в исходном файле [9].

Каждая стадия работы алгоритма требует различных временных ресурсов как от человека, так и от компьютера. Время работы программного средства, покадрово извлекающего изображения из видео, может составить несколько минут, однако для проверки результатов человеку может потребоваться несколько часов.

Перечислим самые известные направления работы алгоритма GAN, которые активнее всего используются ИТ–сообществом: конвертация исходной картинки между состояниями (CycleGAN); создание изображения на основе текстового описания, напечатанного или даже написанного человеком от руки, а впоследствии распознанного интеллектуальным алгоритмом (процесс преобразования текста в изображение); создание изображения с очень высоким разрешением, что предполагает развитие классического алгоритма до идеала, и др.

Разрабатываемая интеллектуальная система состоит из двух нейронных сетей, генератора и дискриминатора (детектора), обучаемых по методу *backpropagation* (метод обратного распространения ошибки). Суть метода состоит в том, что распространение сигнала о неточности значений искомых входных и выходных точек (input-output) происходит в направлении, обратном прямому распространению сигнала, в стандарте метода прямого распространения ошибки [10].

На рис. 1 представлена схема работы алгоритма GAN.

Из множества случайных чисел (случайного шума из заранее выбранного распределения) генератор создает требуемую картинку, причём изображение должно быть максимально реалистичным. Синтез происходит на основе имеющегося набора данных. Далее данные передаются детектору, который представляет собой двоичный классификатор, который с наибольшей точностью определяет, является ли входная выборка реальной (в этом случае вывод скалярного значения равен 1) или ложной (вывод скалярного значения равен 0).

В детектор (дискриминатор) попадает образец, синтезированный генератором, и обрабатываемое изображение. В генератор поступает информация о причине, по которой дискриминатор определил текущую выборку как синтезированный контент.

Дискриминатор должен выполнять свою работу максимально качественно. Когда поддельный образец (созданный генератором) передается дискриминатору и производятся соответствующие вычисления, результаты всегда округляются не в пользу генератора. Однако он должен сгенерировать образцы таким образом, чтобы дискриминатор допустил ошибку, назвав его подлинным.

В конце каждой итерации детектор получает информацию от специального контролирующего блока о том, правильно ли он выполнил свою работу или нет. Такой блок называется блоком потерь или функцией Loss.

Важно отметить, что в генератор не поступают образцы из оригинального набора данных (датасета), он изменяет веса своих нейронов, ориентируясь и опираясь только на результаты, полученные от детектора.

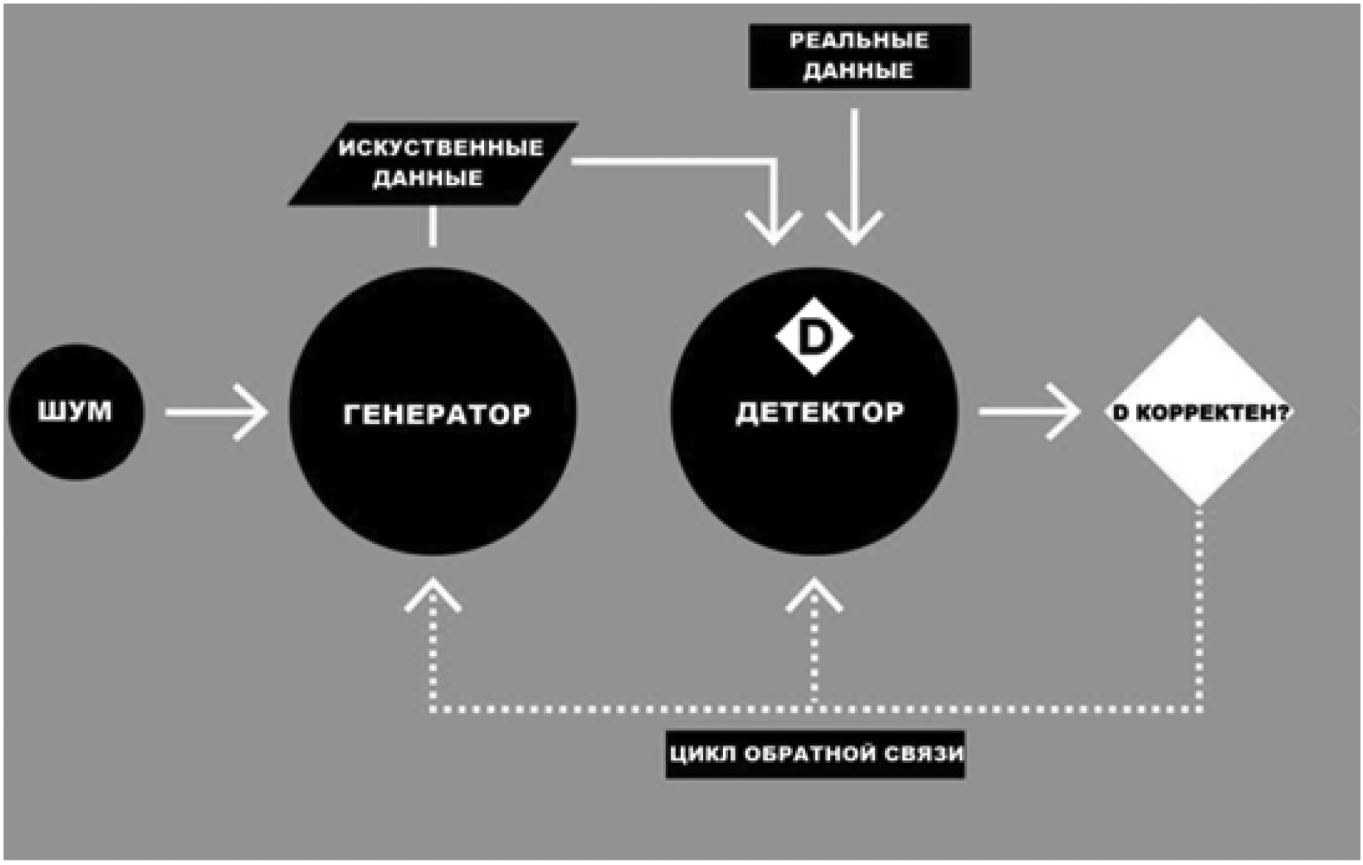

Рис. 1. Схема работы алгоритма Generative Adversarial Networks (GAN)

Рассмотрим выполнение обработки. Это формула оптимизации от минимального к максимальному, в которой генератор хочет минимизировать целевую функцию, в то время как дискриминатор хочет ее максимизировать:

$$V(D,G)=_{Ex\sim Pdata}(x)\times[\log D(x)]+_{Ez\sim PZ(Z)}\times[\log_{(1-)}D(G(Z))], \quad (1)$$

где $V(D,G)$ — целевая функция.

Различающей функцией является $D(G(Z))$, а функцией искусственной нейронной сети (ИНС) синтезатора является $G(Z)$. $PZ(Z)$— это вероятности, распределенные в скрытом пространстве (нормальное распределение), которое обычно является случайным (рандомным) гаусовским распределением. P_{data} — это распределение вероятностей обучающего набора данных.

На этапе выбора образца из P_{data}, детектор искусственной нейронной сети (ИНС — детектор) делает попытку его определить, как истинный образец. $G(Z)$ — это выборка, являющаяся результатом процесса синтеза, если $G(Z)$ указан в качестве входного параметра *input* для ИНС–детектора, который должен определить его подлинный или поддельный характер.

Дискриминатору необходимо уменьшить вероятность $D(G(Z))$ до нуля. Следовательно, он стремится максимизировать значение (1 — $D(G(Z))$, в то время как генератор хочет увеличить вероятность $D(G(Z))$ до 1, чтобы дискриминатор допустил ошибку при вызове сгенерированной выборки в качестве подлинной. Поэтому генератор стремится минимизировать значение данного выражения.

При такой постановке задачи градиент детектора будет принимать положительное значение, так как значение функции оптимизации будет возрастать с каждой итерацией, а градиент генератора станет антиградиентом, то есть его значение при каждом шаге будет уменьшаться.

В перечень известных архитектур входят: CycleGAN, StyleGAN, PixelRNN (авторегрессиваная модель), Text–2–Image (создание изображения на основе введенного текста), DiscoGAN (вариант архитектуры CycleGAN с незначительными изменениями, направленными на уменьшение потребления ресурсов).

Основные методы и средства разработки программного средства

При разработке программного средства использован язык программирования Python 3.11. В качестве среды разработки — IDE Microsoft Visual Studio Code. Применены следующие библиотеки языка Python:

— openCV — библиотека компьютерного зрения, которая предназначена для анализа, классификации и обработки изображений [11];
— NumPy — открытая бесплатная библиотека языка Python, предназначенная для работы с многомерными массивами [12];
— Pandas — высокоуровневая библиотека Python для анализа данных [13];
— PyTorch — современная библиотека глубокого обучения [14];
— Deepfakeutils — библиотека, включающая в себя модели обучения и инструменты, необходимые для создания и детектирования Deepfake–контента.

Программное средство должно принимать на вход тестовые данные, обрабатывать их, выводить итоги тестирования и генерировать файлы-результаты с вердиктом о синтезированности искомого видеоряда.

Используемые модели

При реализации программного обеспечения использованы предварительно обученные модели распознавания человеческих лиц на извлечённых изображениях. Такие модели (например, BlazeFace), применяющиеся в машинном обучении, были разработаны корпорацией Google с целью определения местонахождения основных точек человеческого лица относительно друг друга. Для разработки системы был осуществлён выбор искусственных нейронных сетей на основе характеристик их производительности и продуктивности, например, ИНС ResNeXt[15], и XceptionNet [16], применённые в рамках данного исследования.

Модели нейронных сетей обучались с использованием выбранного алгоритма ИНС на сервисе Google Cloud Platform, используемом разработчиками в сфере искусственного интеллекта и машинного обучения.

Каждая из моделей была протестирована на основе доступных тестовых видеоданных с целью проверки качества работы и поддержания точности. Как только окончательная модель проходит все проверки, она может быть запущена в производство, и, со временем, улучшает свои характеристики.

Архитектура Inception, предложенная в 2015 году, не выбирает размер ядра, а использует одновременно несколько массивов, которые восстанавливаются в одно и то же время, и использует слияние для вывода каналов. Однако такая архитектура значительно увеличивает количество операций, которые нужно выполнить для вычисления в рамках отдельно взятого слоя.

Поэтому разработчики предложили хитрость: перед каждым свёрточным блоком рекомендуется делать свёртку с размером ядра 1x1, подающегося на вход свёрткам с большими размерами ядер [17]. Сверточный слой в прочих архитектурах обычно одновременно обрабатывает как пространственную информацию (корреляцию соседних точек внутри одного канала), так и данные между каналов, так как свёртка будет применена ко всем каналам сразу.

Архитектура Xception основана на теории, что обработка двух типов информации, непосредственно в последовательности, не приводит к снижению качества сети, и разлагает традиционную свертку на кросс–канальную (которая имеет дело только с межканальными корреляциями) и пространственную (которая имеет дело только с пространственными корреляциями внутри каждого канала). Получившаяся конструкция и составляет полный модуль Inception [18].

Архитектура Xception (рис. 2) основана на теории, что обработка двух типов информации непосредственно в последовательности не приводит к снижению качества сети, и разлагает традиционную свертку на кросс-канальную (которая имеет дело только с межканальными корреляциями) и пространственную (которая имеет дело только с пространственными корреляциями внутри каждого канала). Получившаяся на рисунке конструкция и составляет полный модуль Inception [19].

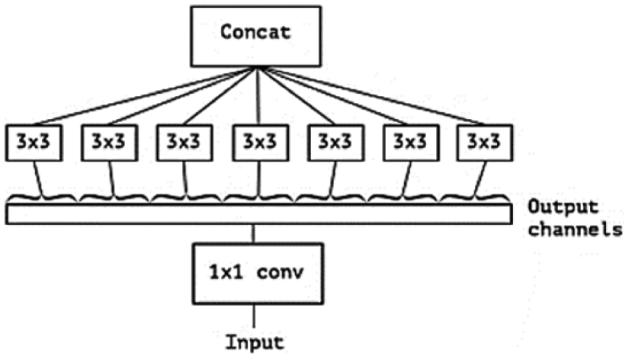

Рис. 2. Схема блока Xception

Вместо выполнения обычного алгоритма блока искусственной нейронной сети, последовательно выполняется два следующих шага [20]:

1. Нужно свернуть имеющийся тензор размером 1x1 подобно тому, как это выполнялось в блоке Inception, получив новый тензор. Это операция называется *pointwise convolution (поточечная свёртка)*.
2. Далее требуется свернуть каждый канал по отдельности, сверткой с параметрами 3 × 3 (в этом случае размерность не изменится, так как сворачиваются не все каналы вместе, как в обычном свёрточном слое). Это операция называется *depthwise spatial convolution (пространственная свертка по глубине)*.

Исходные данные, которые разделаются по глубине свёртки, выполняют первую пространственную свёртку относительно отдельно взятого канала, а затем выполняют свертку 1 × 1, тогда как модификация сначала выполняет свертку 1 × 1, а затем пространственную свёртку по отдельному каналу.

В исходном начальном этапе после первой выполненной операции характерна нелинейность. В исключении, в модели ResNeXt, в модифицированной разделяемой свёртке по глубине, промежуточной нелинейности нет. В ходе выполнения промежуточных тестов, сообществом разработчиков этой модели было доказано, что агрегированные преобразования превосходят стандартный модуль ResNet (сравнение суммы весов нейронных слоев) даже при условии ограничения сложности и размера модели, сохраняемых компьютером. Следует подчеркнуть, что хотя повысить точность нетрудно за счёт увеличения количества слоёв, методы повышения сложности на практике встречаются редко [20].

Модели нейронных сетей обучались с использованием выбранного алгоритма искусственной нейронной сети на сервисе Google Cloud Platform, широко используемом разработчиками в сфере ИИ и машинного обучения.

Сверточный слой в такой архитектуре обрабатывает внутриканальную и межканальную информацию последовательно, в рамках одного процесса. Это позволяет снижать нагрузку на ИНС, так как количество весов в рамках одного вычисления снижается. На рис. 3 изображена схема работы блока ResNeXt.

Метод основан на том, что мощность (размер серии преобразований) — это конкретное измеряемое значение (не являющееся константой), имеющее центральное значение наряду с измерениями ширины и глубины.

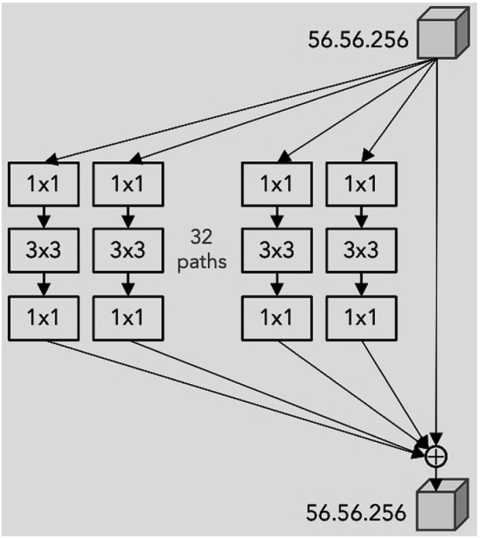

Рис. 3. Схема блока ResNeXt

Эксперименты показывают, что повышение производительности является более эффективным способом повышения точности, чем углубление или расширение изображения, особенно когда глубина и ширина начинают давать существующим моделям меньшие результаты при анализе функции потерь.

Простейшие нейроны в искусственной нейронной сети выполняют внутреннее произведение (взвешенную сумму), представляющее собой элементарное преобразование, выполняемое полносвязными и сверточными слоями [20].

Соответственно, чем больше вес внутри отдельно взятого слоя, тем больше вероятность, что характерное его признаки будут доминирующими при обучении и при дальнейшем распределении весов. Данная серия вычислений называется *wixi*. Эта операция отображена на рис. 4.

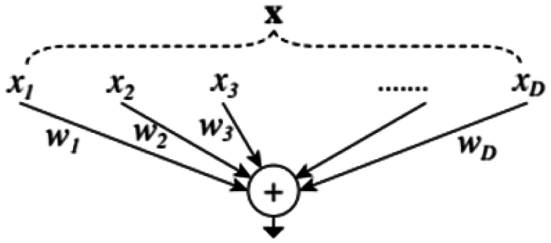

Рис. 4. Серия вычислений wixi

Задача определения ключевых точек имеет важное значение для распознавания и анализа человеческих лиц. Поэтому для решения этой проблемы были предложены различные методы. Классические методы, такие как *Random forest*, *Gradient boosting* и SVM, являются наиболее известными. Однако они сложны в реализации и не очень стабильны. В последние годы для решения задач компьютерного зрения были разработаны методы, связанные с высокоточными глубинными нейронными сетями. Решения на основе нейронных сетей быстро превосходят традиционные методы по точности, стабильности и общности модели в основных задачах компьютерного зрения.

Схема работы разработанного ПО

Алгоритм проверки видео на подделку состоит из обязательных шагов.

1. Загрузка предварительно обученных моделей ResNeXt и Xception, которая выполняется с помощью модуля *gdown*, представляющего собой одну из базовых библиотек по скачиванию файлов из сети Интернет.
2. Переход в модуль получения результатов предугадывания относительно модели ResNeXt. Выводом является величина отклонения в предугадывании.

3. Переход в модуль получения результатов предугадывания относительно модели Xception. Выводом является величина отклонения в предугадывании.
4. Вычисление среднего значения точности предугадывания, находится среднее арифметическое коэффициентов отклонений.
5. Если среднее значение точности предугадывания оказывается больше некоторого порогового значения (которое задается вручную), то выводом является заключение о том, что видео является поддельным — фейковым. Иначе — выводом является заключение о его подлинности.

Описанные шаги является действенным способом определения фейкового (поддельного или, наоборот, подлинного) видеоконтента, синтезированного с помощью технологии Deepfake алгоритма GAN, работающим с приемлемой точностью. Описанный алгоритм представлен в качестве блок-схемы на рис. 5.

В разработанном программном средстве также присутствует модуль получения результатов предугадывания отдельной модели.

Алгоритм его работы состоит из следующих шагов.

1. Извлечение видеокадров с лицами людей из искомого видеоряда.
2. Форматирование каждого кадра, проводящееся для конкретной модели.
3. Нормализация данных каждого видеокадра, и подача видеоряда на анализ.
4. Получение и анализ результатов обработки видеокадров.
5. Вывод результатов (в консоль или в качестве видеоряда).

Алгоритм представлен в качестве блок-схемы на рис. 6.

Вместе с этим, программная реализация включает в себя модуль анализа данных. Его работа состоит из нескольких этапов.

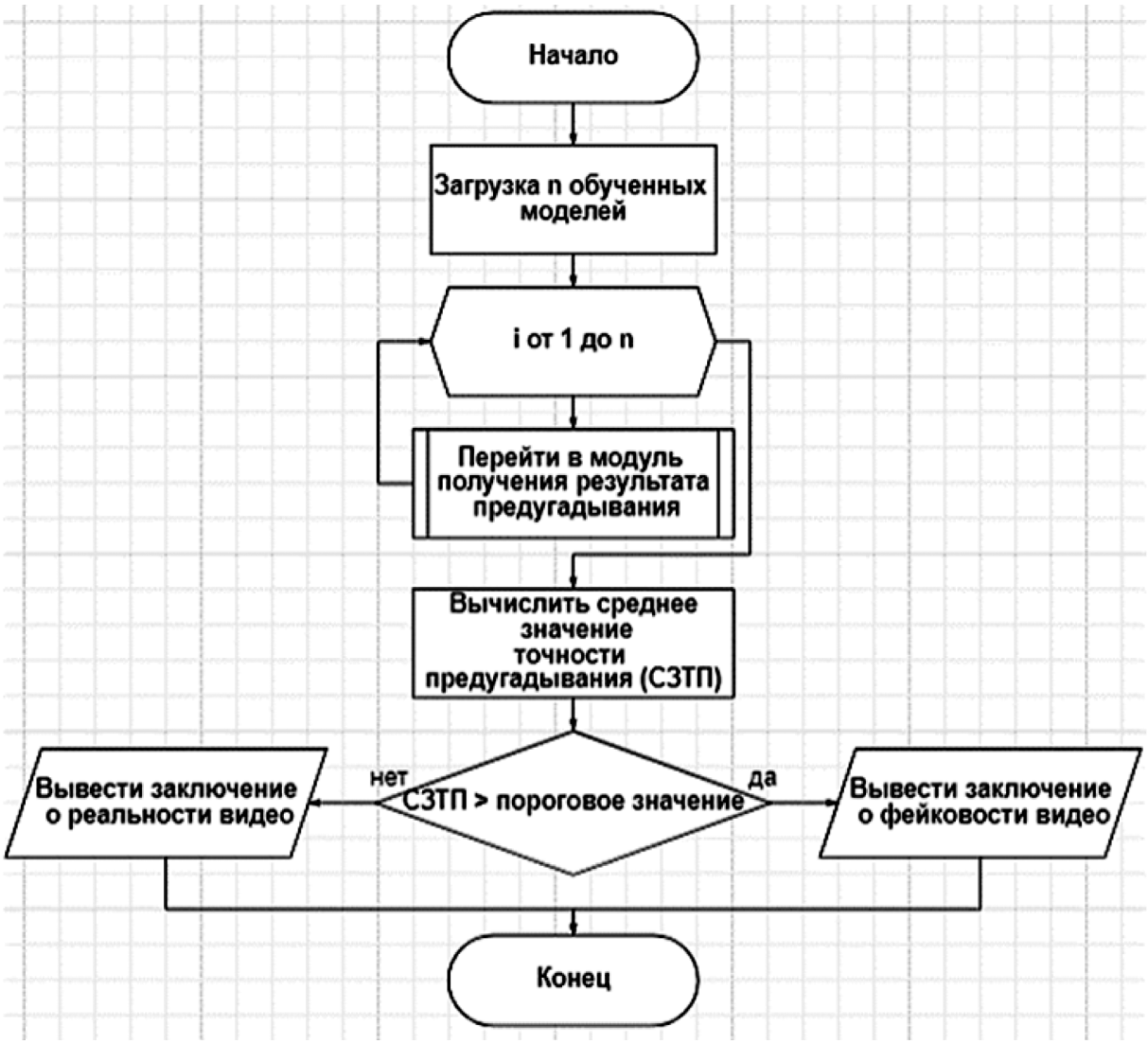

Рис. 5. Блок-схема работы основного алгоритма программы

1. Преобразование данных каждого видеокадра в тензор.
2. Если применяется модель искусственной нейронной сети Xception, то анализ происходит относительно этой модели, иначе — относительно модели ResNeXt.
3. Отключение градиентного спуска.
4. Выгрузка модели и запуск процесса предугадывания.
5. Нормализация данных с помощью сигмоиды.
6. Вывод среднего значения всех элементов массива.

Рис. 6. Блок-схема алгоритма получения результата предугадывания

Результаты исследований и их анализ

Функции разработанной программы

Программа обладает следующими функциональными возможностями.

1. Происходит подготовительная обработка образца — анализ и проверка видеоряда на предмет его поддельности (получение ответа на вопрос, является ли видеоконтент синтезированным с помощью технологии Deepfake алгоритма Generative Adversarial Network (GAN), или же он является подлинным.

Для этого на вход функции подается путь к видеоролику (в файловой системе). Образец проходит покадровую проверку на наличие лица человека в каждом отдельном фрейме. Если распознание прошло успешно, то данные добавляются в список. По желанию, можно оставить фиксированное количество семплов с наилучшим качеством среди представленных.

Если конечное число кадров с лицами ненулевое, то запускается цикл, в котором каждый кадр форматируется. Изменяется его размер (изотопически), после чего стороны кадра уравниваются, он становится условно квадратным.

2. Вывод текстового заключения о поддельности или подлинности отдельно взятого видеоряда, с приемлемой точностью, с вероятностью, в процентах.

На основе анализа видеоряда делается текстовое заключение со сведениями, отражающими тот факт, является ли данный образец синтезированным с помощью технологии Deepfake алгоритма GAN, или нет. Вывод происходит в консоли.

3. Вывод текстового заключения о поддельности или подлинности любого произвольного количества видеороликов, находящихся в одной директории (папке или каталоге) файловой системы, с приемлемой точностью. Вывод текстового заключения относительно каждого видеоролика из предложенного списка.

4. Вывод видеозаключения о поддельности или подлинности отдельно взятого видеоряда с приемлемой точностью. Параметры нормализации являются рекомендуемыми разработчиками ResNeXt нейронной сети, для этого также создается отдельный объект класса нормализации. При анализе в качестве основного контейнера используется двумерный список, сгенерированный с помощью библиотеки NumPy. Инструмент по обработке образцов — сэмплов реализован в виде класса — наследника от основного класса.

Для выполнения основного функционала разработанного программного средства используются модули компьютерного зрения. Инициализируются объект–рекордер (для записи) и объект–плеер (для воспроизведения записи). Каждый объект покадрово проходит по образцу, создавая новый видеоролик, содержащий результат работы программного средства. По желанию, можно изменить его цветовую палитру, ускорить или замедлить темп его воспроизведения.

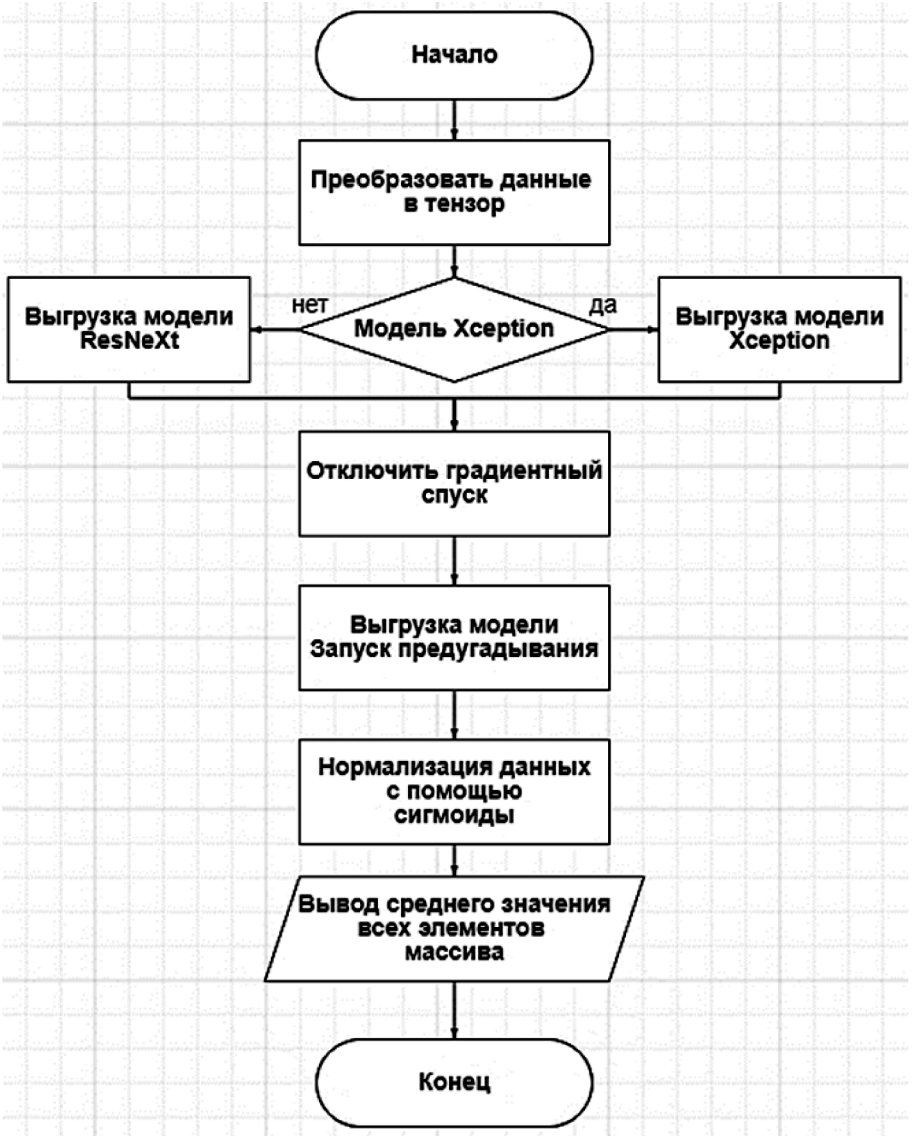

Рис. 7. Блок-схема алгоритма анализа данных

Принцип работы программного средства

Программа принимает на вход строку, являющуюся путём к видео в файловой системе, проверяет корректность введённых данных, а именно отсутствие повторяющихся функций, взаимоисключающих аргументов и неверных опций.

Далее, программа начинает свою работу на любом незагруженном логическом ядре, в противном случае будет выведено сообщение о возможном снижении производительности процесса тестирования. Далее проверяется возможность создания выходных файлов и чтение тестовых данных. Сбои во время этих действий маловероятны, однако может произойти ситуация, когда необходимых файлов нет. Программа не имеет графического интерфейса, поэтому запуск осуществляется через консоль. На рисунке 8 представлен пример запуска.

Если всё отработало успешно, то запускается тестирующая программа для всех тестовых данных, чтобы убедиться, что всё работает должным образом. Это делается только для начальных входных данных и только один раз.

Рассмотрим аргументы. Основными входными параметрами являются пути к видеороликам, которые могут быть представлены как строка, объект Path библиотеки Pathlib или объект Windows Path библиотеки OS.

Для работы программы необходимо ввести требуемый режим (его номер). Первый режим представляет собой вывод заключения о поддельности видео в виде текстовой строки; второй режим выводит строку с заключением о поддельности (фейковости) произвольного количества образцов, при работе третьего режима пользователь получает видеозаключение с вердиктом программы.

```
PS F:\Programm> & C:/Users/roone/AppData/Local/Programs/Python/Python39/python.exe f:/Programm/detector.py
----------------------------------------------------------------------------------------------------
Choose what you want:
1 - Get text result about one video
2 - Get text result about few video
3 - Get video result about one video
Mode: 1
----------------------------------------------------------------------------------------------------
Welcome to "Deeptector"! Start working!
----------------------------------------------------------------------------------------------------
Choose video to analyze:
        1 - examples\abbalbisk.mp4
        2 - examples\capitain_lore.mp4
        3 - examples\donald_thrump.mp4
        4 - examples\krylova.mp4
        5 - examples\me_at_zoo.mp4
        6 - examples\morgan_freeman.mp4
        7 - examples\queen_elizabeth.mp4
        8 - examples\robert_downey.mp4
        9 - examples\vladimir_putin.mp4
        10 - examples\vladimir_zelenskiy.mp4
Enter the number: 1
----------------------------------------------------------------------------------------------------
Load Xception pre-trained model...
        Done!
----------------------------------------------------------------------------------------------------
Load ResNext pre-trained model...
        Done!
----------------------------------------------------------------------------------------------------
Face samples:
        64
----------------------------------------------------------------------------------------------------
Model prediction:
        0.11014726758003235
----------------------------------------------------------------------------------------------------
Face samples:
        64
----------------------------------------------------------------------------------------------------
Model prediction:
        0.07376221567392349
----------------------------------------------------------------------------------------------------
Comparing prediction and threshold:
        0.08350665807823768 < 0.3
----------------------------------------------------------------------------------------------------
examples\abbalbisk.mp4 - REAL!
----------------------------------------------------------------------------------------------------
Good Bye!
```

Рис. 8. Работа программы в первом режиме

На рисунке 9 представлен пример работы второго режима программы.

Видеоролики, которые были проанализированы детектором, являются или поддельными (дипфейками) или подлинными оригинальными видео различного уровня качества, как самого видео, так и степени точности подделывания.

Из результатов запуска видеороликов можно сделать вывод, что детектор ошибся в одном случае из трёх (рисунок 10). В среднем, проверка одного видеоролика занимает от 7 до 30 секунд, в зависимости от размера и длительности видеоконтента. Вся тестовая сессия длится, в среднем, 90 секунд.

Третий режим работает аналогично первому. Основное отличие заключается в том, что в конце выводится

```
Mode: 2
--------------------------------------------------------------------------------------------------------------------------------
Welcome to "Deeptector"! Start working!
--------------------------------------------------------------------------------------------------------------------------------
Choose videos you want to analyze:
        1 - examples\abbalbisk.mp4
        2 - examples\capitain_lore.mp4
        3 - examples\donald_thrump.mp4
        4 - examples\krylova.mp4
        5 - examples\me_at_zoo.mp4
        6 - examples\morgan_freeman.mp4
        7 - examples\queen_elizabeth.mp4
        8 - examples\robert_downey.mp4
        9 - examples\vladimir_putin.mp4
        10 - examples\vladimir_zelenskiy.mp4
Choose number (press q to exit): 1
Choose number (press q to exit): 3
Choose number (press q to exit): 7
Choose number (press q to exit): q
--------------------------------------------------------------------------------------------------------------------------------
Face samples:
        64
--------------------------------------------------------------------------------------------------------------------------------
Model prediction:
        0.3829379081726074
--------------------------------------------------------------------------------------------------------------------------------
Face samples:
        64
--------------------------------------------------------------------------------------------------------------------------------
Model prediction:
        0.4184581935405731
--------------------------------------------------------------------------------------------------------------------------------
Face samples:
        64
--------------------------------------------------------------------------------------------------------------------------------
Model prediction:
        0.7207198143005371
--------------------------------------------------------------------------------------------------------------------------------
[0.3829379081726074, 0.4184581935405731, 0.7207198143005371]
--------------------------------------------------------------------------------------------------------------------------------
Comparing prediction (ResNeXt) and threshold:
        0.3829379081726074 > 0.3
--------------------------------------------------------------------------------------------------------------------------------
Comparing prediction (ResNeXt) and threshold:
        0.4184581935405731 > 0.3
--------------------------------------------------------------------------------------------------------------------------------
Comparing prediction (ResNeXt) and threshold:
        0.7207198143005371 > 0.3
--------------------------------------------------------------------------------------------------------------------------------
abbalbisk.mp4 - FAKE!
donald_thrump.mp4 - FAKE!
queen_elizabeth.mp4 - FAKE!
--------------------------------------------------------------------------------------------------------------------------------
Good Bye!
```

Рис. 9. Работа программы во втором режиме

результирующее видео. В данном режиме вводиться дополнительный аргумент, который представляет собой имя файла–результата. Пример работы программы приведен на рисунке 10.

Хранение значений кадров с лицами и соответствующих им имен файлов происходит в оперативной памяти. Проведено тестирование программного средства для различных файлов, включая обработку исключительных ситуаций.

Таким образом, реализовано программное средство для проверки видео на предмет его синтезированности с помощью технологии Deepfake алгоритмаGAN (генеративно-состязательных сетей: Generative Adversarial Network).

Заключение

Технологии глубокого подражания быстро развиваются. Точность получаемых данных постоянно растёт.

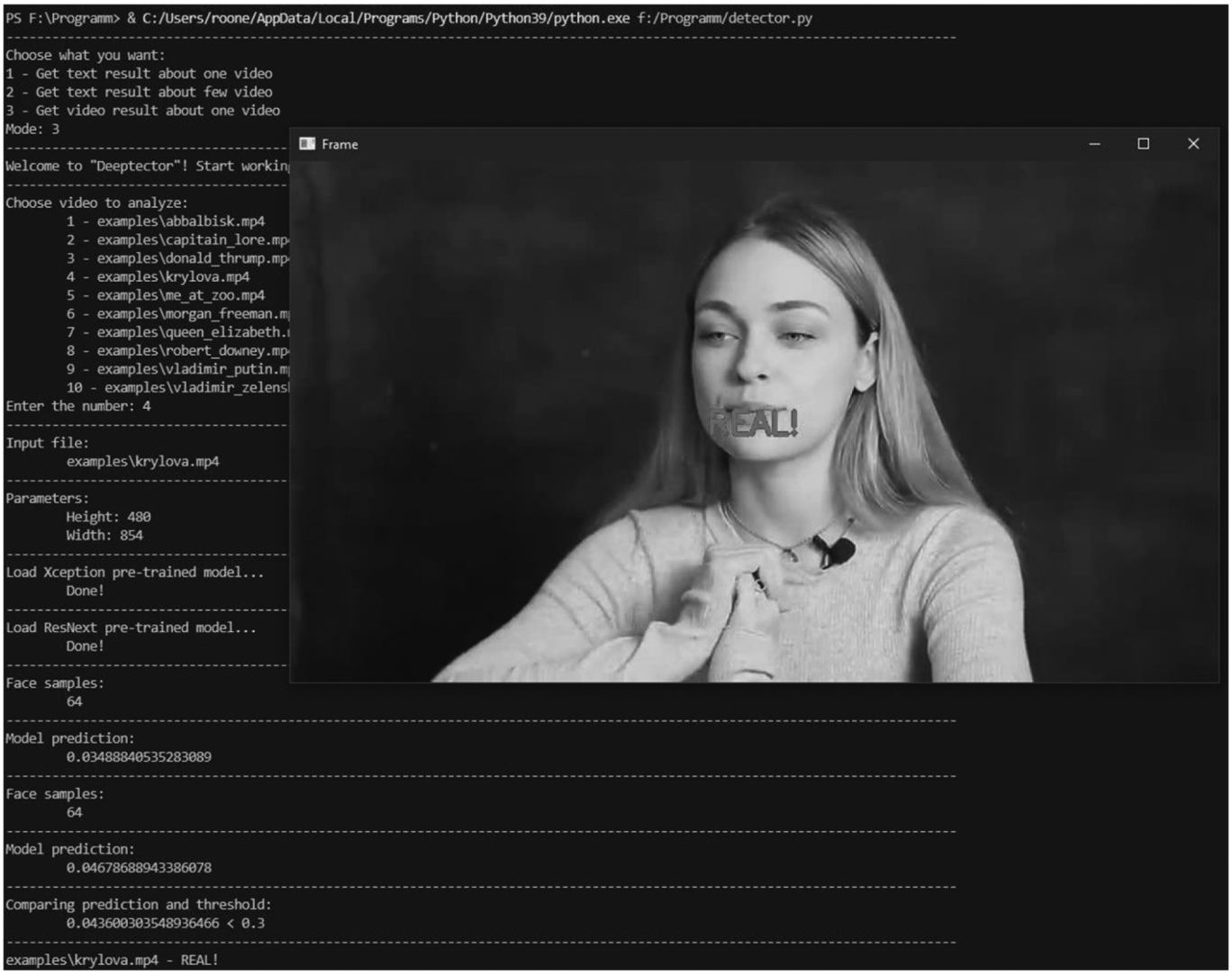

Рис. 10. Демонстрация работы программы в третьем режиме

Алгоритмы обнаружения мошенничества совершенствуются. Предполагается, что в будущем многие российские ИТ— компании будут предлагать услуги по борьбе с данной угрозой [20].

В настоящее время двумя факторами, всё ещё препятствующими широкому распространению дипфейков, являются сравнительно низкий уровень совершенства алгоритмов и высокая стоимость конечного продукта. Однако с ростом индустрии развлечений эти два показателя будут возрастать, по мере выхода Deepfake на массовый рынок. Угрозы будут только возрастать [21].

Авторы внесли свой скромный вклад в создание импортозамещающего отечественного программного продукта, направленного на борьбу с угрозой, которую представляет собой распространение ложного фейкового видеоконтента.

Результатом проделанной работы является оценка (процент вероятности) подлинности или поддельности видеоряда, проходящего проверку, т.е. — даётся ответ на вопрос, является ли видеоконтент синтезированным с помощью технологии DeepFake алгоритма GAN, или настоящей видеосъёмкой, в подлинности которой сомневаться не приходится.

Достоинством работы является её высокая актуальность, злободневность и практическая направленность на распознавание заведомо ложного поддельного видеоконтента, направленного на обман мировой общественности.

Полученные результаты можно использовать как в средствах массовой информации, на телевидении, в Интернете, — везде, где возникают сомнения в правдивости и подлинности содержания видеоконтента, предлагаемого массовому зрителю и/или пользователю, так и в частном порядке, для борьбы со злоумышленниками — мошенниками, пытающимися, для достижения своих неблаговидных целей, обмануть доверие простых людей.

ЛИТЕРАТУРА

1. Барабанщиков, В.А. Deepfake в исследованиях восприятия лица. Москва: ИНГН, 2018. 176 с.
2. Баранова, Е.К. Информационная безопасность и защита информации с нуля до полного понимания. Москва: Риор, 2018. 400 с.
3. Lyu S. Deepfake Detection: Current Challenges and Next Steps // IEEE International Conference on Multimedia & Expo Workshops (ICMEW). 2020. P. 1–6. DOI: 10.1109/ICMEW46912.2020.9105991.
4. Xinyi Z., Reza Z. A Survey of Fake News: Fundamental Theories, Detection Methods and Opportunities // ACM Computing Surveys. 2020. Vol. 53, Iss. 5. P. 1–40. DOI: 10.1145/3395046.
5. Dash A., Ye J., Wang G. A review of Generative Adversarial Networks (GANs) and its applications in a wide variety of disciplines // arXiv preprint.2021.
6. Ярочкин В.И. Информационная безопасность. Москва: Академический проект, 2018. 544 с.
7. Крон Д. Глубокое обучение в картинках. Визуальный гид по ИИ. СПб.: Питер, 2016. 416 с.
8. Стюарт Р. Искусственный интеллект. Современный подход к решению актуальной проблемы. Москва: МГИУ, 2017. 272 с.
9. Курцвейл Р. Как создать разум: секрет человеческого мышления раскрыт. СПб.: BHV, 2019. 368 с.
10. Songyuan L., Fan M., Chen R. Overview of generative adversarial networks. J Phys Conf Ser. 2021.
11. Howse J., Minichino J. Learning OpenCV 4 Computer Vision with Python 3: Get to grips with tools, techniques, and algorithms for computer vision and machine learning. 3rdEditio, 2020. P. 372.
12. Johansson R. Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib. 2019. DOI: 10.1007/978-1-4842-4246-9.
13. Лемешевский С.В. Введение в библиотеку Pandas. Институт математики НАН Беларуси. 2020.
14. Макмахан Б., Рао Д. Знакомство с PyTorch. Глубокое обучение при обработке естественного языка. 2020. ISBN:978-5-4461-1241-8.
15. Zhou T., Zhao Y., Wu J. ResNeXt and Res2Net Structures for Speaker Verification. Microsoft Corp., USA. 2020. https://doi.org/10.48550/arXiv.2007.02480
16. Jain A.K. Artificial Neural Networks: a Tutorial // Computer. 1996. Vol. 29. № 3. P. 31–44. DOI: 10.1109/2.485891.
17. Do N.Q. Phishing webpage classification via deep learning-based algorithms: an empirical study// Applied Sciences. 2021. Vol. 11. № 19.32 p. DOI: 10.3390/app11199210.
18. Letou K. Host-based Intrusion Detection and Prevention System (HIDPS) // International Journal of Computer Applications. 2013. Vol. 69. № 26. P. 28–33. DOI: 10.5120/12136-8419.
19. Mitchell T.M. Machine learning // New York: McGraw-hill. 1997. Vol. 1. № 9. 414 p. ISBN 0071154671, 9780071154673.
20. Модификация классического квантового протокола bb84, повышающая его характеристики Ляшенко К.А., Поркшеян В.М., Черкесова Л.В., Ревякина Е.А., Енгибарян И.А., Бурякова О.С., Решетникова О.А. Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2023. № 2. С. 100–115.
21. Goodfellow I. Deep learning // MIT press. 2016. 800 p. ISBN 9780262337373, 0262337371.