УДК 004.056
doi: 10.26583/bit.2024.4.08

Александр С. Пикуль
Национальный исследовательский университет ИТМО,
Кронверкский пр–кт, 49, Санкт-Петербург, 197101, Россия
e-mail: aleksandrpikul134@gmail.com, https://orcid.org/0009-0000-2619-3822

АНСАМБЛЬ СОВРЕМЕННЫХ МОДЕЛЕЙ КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ ЗАДАЧИ ОБНАРУЖЕНИЯ ДИПФЕЙКОВ

Аннотация. Исследуется возможность применения современных архитектур компьютерного зрения для задачи обнаружения дипфейков. Рассматриваются следующие архитектуры: EfficientNet, Vision Transformer (ViT), VisionLSTM (ViL), VisionKAN и Mamba Vision. Новизна подхода заключается в применении и сравнении работы данных архитектур, а также в их объединении в парные ансамбли для повышения точности детекции дипфейков. В работе проведен эксперимент, основанный на применении нескольких архитектур для обработки изображений. Каждая архитектура применяется как отдельно, так и в составе ансамбля, состоящего из двух моделей. Набор данных для эксперимента был сформирован из фреймов видео с дипфейками. Полученные фреймы были подвержены различным аугментациям. Результаты экспериментов показали, что применение ансамблей современных архитектур повышает точность распознавания дипфейков. Ансамбль ViT и VisionLSTM показал F_1-меру 97,68%, что выше, чем применение данных архитектур по отдельности. Однако не все ансамбли привели к улучшению показателей. Например, комбинация Mamba Vision и VisionLSTM продемонстрировала снижение F_1-меры до 95,78% в сравнении с использованием Mamba Vision в одиночку. Результаты исследования полезны для специалистов, работающих в области компьютерного зрения, кибербезопасности и анализа мультимедийного контента. Применение предложенных архитектур и их ансамблей может быть эффективно в задачах по обнаружению дипфейков и других форм поддельного контента, что важно для защиты от информационных угроз.

Ключевые слова: дипфейк, компьютерное зрение, глубокие нейронные сети, ансамбли моделей машинного обучения, механизм внимания, рекуррентные нейронные сети, сверточные нейронные сети.

Для цитирования: ПИКУЛЬ, Александр С. АНСАМБЛЬ СОВРЕМЕННЫХ МОДЕЛЕЙ КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ ЗАДАЧИ ОБНАРУЖЕНИЯ ДИПФЕЙКОВ. Безопасность информационных технологий, [S.l.], т. 31, № 4, с. 116–127, 2024. ISSN 2074-7136. URL: https://bit.spels.ru/index.php/bit/article/view/1720. DOI: http://dx.doi.org/10.26583/bit.2024.4.08.

Aleksandr S. Pikul
ITMO University,
Kronverksky Ave., 49, St. Petersburg, 197101, Russia
e-mail: aleksandrpikul134@gmail.com, https://orcid.org/0009-0000-2619-3822

An ensemble of modern computer vision models for deepfake detection

Abstract. This article explores the potential use of modern computer vision architectures for the task of deepfake detection. The following architectures are considered: EfficientNet, Vision Transformer (ViT), VisionLSTM (ViL), Vision KAN, and Mamba Vision. The novelty of the approach lies in the application and comparison of these architectures, as well as their combination into paired ensembles to improve the accuracy of deepfake detection. The study conducted an experiment based on the application of multiple architectures for image processing. Each architecture was used both individually and as part of an ensemble consisting of two models. The dataset for the experiment was created from video frames containing deepfakes, and these frames were subjected to various augmentations. The experimental results

demonstrated that using ensembles of modern architectures improves the accuracy of deepfake recognition. The ensemble of ViT and VisionLSTM achieved an F_1-score of 97.68%, which is higher than the performance of these architectures when used individually. However, not all ensembles resulted in improved metrics. For example, the combination of Mamba Vision and VisionLSTM showed a decrease in F_1-score to 95.78% compared to using Mamba Vision alone. The research findings are valuable for professionals working in computer vision, cybersecurity, and multimedia content analysis. The proposed architectures and their ensembles can be effectively used in tasks related to deepfake detection and other forms of fake content, which is crucial for protection against information threats.
Keywords: deepfake, computer vision, deep neural networks, ensembles, attention mechanism, recurrent neural networks, convolutional neural networks.
For citation: PIKUL, Aleksandr S. An ensemble of modern computer vision models for deepfake detection. IT Security (Russia), [S.l.], v. 31, no. 4, p. 116–127, 2024. ISSN 2074-7136. URL: https://bit.spels.ru/index.php/bit/article/view/1720. DOI: http://dx.doi.org/10.26583/bit.2024.4.08.

Введение

Современные методы компьютерного зрения имеют огромный потенциал для решения различных задач. Одной из таких задач является обнаружение дипфейков. Дипфейк — это технология на основе глубоких нейронных сетей, которая используется для создания поддельного контента, представляющего угрозу личной безопасности.

Глобально глубокие нейронные сети можно разделить на три основные группы: сверточные нейронные сети [1], рекуррентные нейронные сети [2], механизмы внимания [3]. В данной работе рассматривается применение современных методов компьютерного зрения с акцентом на их применимости к задаче обнаружения дипфейков. Основное внимание уделено архитектурам нейронных сетей: EfficientNet, Vision Transformer (ViT), VisionLSTM (ViL), VisionKAN, а также Mamba Vision. Данные подходы выбраны именно потому, что они сочетают в себе ключевые идеи сверток, рекуррентности и внимания. Каждая из данных архитектур представляет собой новаторский подход к анализу и обработке изображений, что делает их эффективными инструментами для задачи обнаружения дипфейков.

Помимо применения исходных архитектур, на их основе построены различные парные ансамбли, включающие в себя две ветви для обработки входного изображения. Предполагается, что такие ансамбли способны повысить точность детекции дипфейков, в сравнении с применением только одной архитектуры.

1. Архитектура EfficientNet

EfficientNet – архитектура свёрточных нейронных сетей, предложенная в 2019 г. в [4], показана на рис. 1.

Рис. 1. Архитектура EfficientNet

В основе EfficientNet лежит идея масштабирования Compound Scaling. При таком масштабировании увеличиваются сразу три параметра: глубина сети (количество слоёв), ширина (количество каналов), а также разрешение входного изображения.

EfficientNet состоит из MBConv блоков, которые впервые были представлены в модели MobileNetV2 [5]. На рис. 2 показана структура блока MBConv.

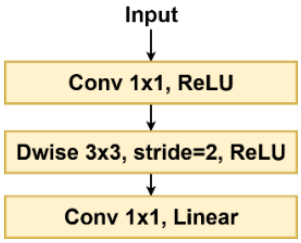

Рис. 2. Блок MBConv

2. Архитектура Vision Transformer

В 2021 г. в [6] описан визуальный трансформер (Vision Transformer, ViT). Используется для задач компьютерного зрения, при этом основные идеи построения архитектуры были заимствованы из классического трансформера, применяемого для работы с естественным языком.

Вначале изображение делится на патчи определенного размера, например, 16 на 16 пикселей, и каждый патч рассматривается как токен, как показано на рис. 3.

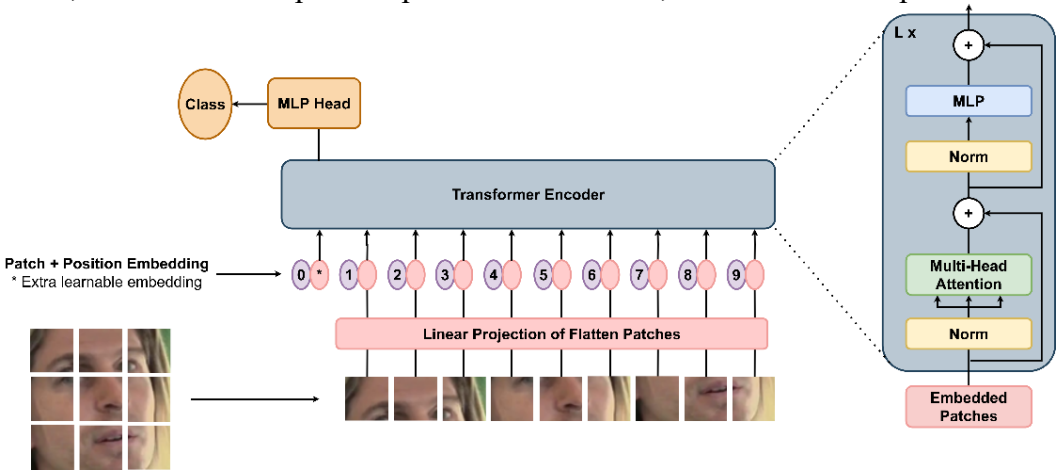

Рис. 3. Архитектура Vision Transformer (ViT)

Далее к каждому токену добавляются позиционные эмбеддинги. Это необходимо для сохранения пространственной информации о патчах изображения, таким образом, сохраняется положение каждого патча внутри исходного изображения.

После чего следует энкодер визуального трансформера, который состоит из следующих ключевых элементов:

1. Layer Normalization. Пусть a – вектор, к которому применяется нормализация. По вектору a вычисляется среднее арифметическое значение μ (среднее значение по всем элементам вектора a):

$$\mu = \frac{1}{N}\sum_{i}^{N} a_i, \tag{1}$$

где N – количество элементов в векторе. Вычисляется стандартное отклонение σ:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i}^{N}(\mu - a_i)^2}. \quad (2)$$

Слой нормализации Layer Normalization [7]:

$$LN(a) = \frac{g}{\sigma} \circ (a - \mu) + b, \quad (3)$$

где g, b – обучаемые вектора.

2. Scaled Dot-Product Attention. Формально, механизм Scaled Dot-Product Attention (*SDPA*) описывается следующей формулой [7]:

$$SDPA(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (4)$$

где Q – матрица запросов (queries), K – матрица ключей (keys), V – матрица значений (values), K^T – транспонированная матрица ключей, d_k – размерность ключей.

Размерность ключей определяется следующей формулой:

$$d_k = d_v = \frac{d_{\text{model}}}{h}, \quad (5)$$

где d_{model} – это размерность векторного представления данных внутри модели, h – количество голов внимания.

3. Multi-Head Attention. Пусть $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, $W_i^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$, где W_i^Q – матрица параметров для запросов (queries), W_i^K – матрица параметров для ключей (keys), W_i^V – матрица параметров для значений (values), W_i^O – матрица параметров для объединенных выходов, i – номер головы внимания, h – количество голов внимания.

Для каждой головы i (где i=1,…,h) выполняются следующие шаги [8]:

$$Q_i = QW_i^Q, K_i = KW_i^K, V_i = VW_i^V. \quad (6)$$

Применение Scaled Dot-Product Attention:

$$O_i = SDPA(Q_i, K_i, V_i), \quad (7)$$

где O_i – это взвешенная комбинация значений V_i для каждой головы внимания, где веса определяются соответствующими запросами Q_i и ключами K_i через операцию Scaled Dot-Product Attention.

Все промежуточные выходы O_i конкатенируются с помощью Multi-Head Attention, чтобы сформировать объединенный вектор. Далее полученный вектор умножается на матрицу параметров W^O [8].

$$MHA(Q, K, V) = (O_1 \oplus O_2 \oplus \ldots \oplus O_h)\, W^O. \quad (8)$$

4. Feed-Forward layer. Формально полносвязный слой (Feed-Forward Layer) определяется следующей формулой:

$$FF(x) = W_2 ReLU(W_1 x + b_1) + b_2, \quad (9)$$

где x – входной вектор размерности d_{model}, $W_1 \in \mathbb{R}^{4d_{\text{model}} \times d_{\text{model}}}$ – первая матрица весов, $W_2 \in \mathbb{R}^{4d_{\text{model}} \times d_{\text{model}}}$ – вторая матрица весов, b_1 и b_2 – смещения (bias) для первого и второго слоев соответственно.

3. Архитектура VisionLSTM

В 2024 г. в [9] представлена архитектура VisionLSTM (ViL). На рис. 4 приведена архитектура предложенного подхода.

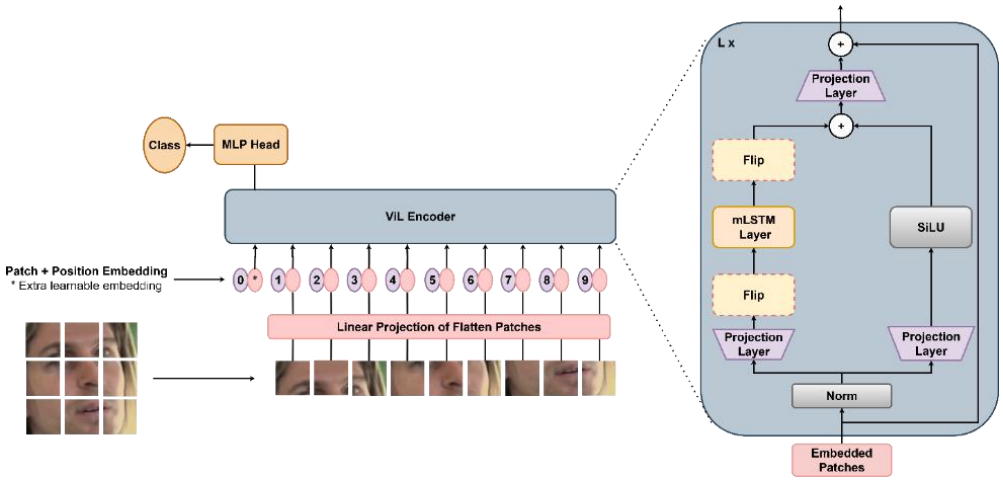

Рис. 4. Архитектура VisionLSTM

Исходное изображение разбивается на последовательность токенов. Стоит отметить, что данный шаг аналогичен тому, как разбиение входного изображения происходит в ViT.

Основу VisionLSTM составляют блоки xLSTM, а именно заимствованный из него блок mLSTM. В нем вместо скалярной ячейки памяти, которая применяется в LSTM, используется матрица, которая обновляется без предыдущих скрытых состояний сети. Для этого по аналогии с трансформерами введены запрос q_t, ключ k_t и значение v_t. Такое правило обновления называется правилом обновления ковариаций [10]:

$$\boldsymbol{C_t = f_t C_{t-1} + i_t v_t k_t^T}, \tag{10}$$

где f_t – forget gate, i_t – input gate; v_t и k_t – значения и ключи, аналогичные тем, что используются в трансформерах.

Вычисление гейтов [11]:

$$\boldsymbol{i_t = exp(w_i^T x_t + b_i)}, \tag{11}$$
$$\boldsymbol{f_t = exp(w_f^T x_t + b_f)}, \tag{12}$$
$$\boldsymbol{o_t = exp(W_o x_t + b_o)}, \tag{13}$$

где $\boldsymbol{o_t}$ – выходной гейт; $\boldsymbol{w_i}$, $\boldsymbol{w_f}$, $\boldsymbol{w_o}$ – матрицы весов для соответствующих гейтов; $\boldsymbol{x_t}$ – входной вектор на момент времени t; $\boldsymbol{b_i}$, $\boldsymbol{b_f}$, $\boldsymbol{b_o}$ – вектор смещений (bias) для каждого гейта.

Вычисление ключей, значений и запросов:

$$\boldsymbol{q_t = W_q x_t + b_q}, \tag{14}$$
$$\boldsymbol{k_t = \frac{1}{\sqrt{d}} W_k x_t + b_k}, \tag{15}$$

$$\boldsymbol{v_t = W_v x_t + b_v}, \tag{16}$$

где $\boldsymbol{W_q}$, $\boldsymbol{W_k}$, $\boldsymbol{W_v}$ – матрицы весов для запросов, ключей и значений; $\boldsymbol{b_q}$, $\boldsymbol{b_k}$, $\boldsymbol{b_v}$ – смещения для запросов, ключей и значений соответственно.

В конце VisionLSTM объединяются первый и последний токены, которые затем передаются на вход линейного классификационного слоя.

4. Архитектура VisionKAN

В 2024 г. в [12] была представлена архитектура нейронной сети под названием Kolmogorov-Arnold Networks (KAN), которая аналогична многослойному перцептрону (MLP). Классическая архитектура перцептрона основывается на теореме Цыбенко (Universal Approximation Theorem). Однако существует и другая теорема, связанная с аппроксимацией функций — теорема Колмогорова-Арнольда, на основе которой и была разработана KAN.

Теорема Колмогорова и Арнольда заключается в том, что аппроксимация непрерывной ограниченной функции от множества переменных сводится к нахождению полиномиального числа одномерных функций [13]:

$$f(x) = f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right). \tag{17}$$

Применяя данную теорему напрямую, оказывается, что KAN сложно масштабировать, то есть существует проблема увеличения сети с помощью дополнительных слоев, нейронов. В связи с данной проблемой была предложена обобщенная архитектура KAN, в которой на каждом слое построена матрица обучаемых параметров:

$$KAN(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \dots \circ \Phi_1 \circ \Phi_0)x. \tag{18}$$

На основе данной архитектуры была построена VisionKAN[1], которая предназначена для задач компьютерного зрения. Стоит отметить, что VisionKAN фактически полностью наследует классический ViT, только теперь в энкодере вместо MLP применяется KAN. На рис. 5 показана архитектура VisionKAN.

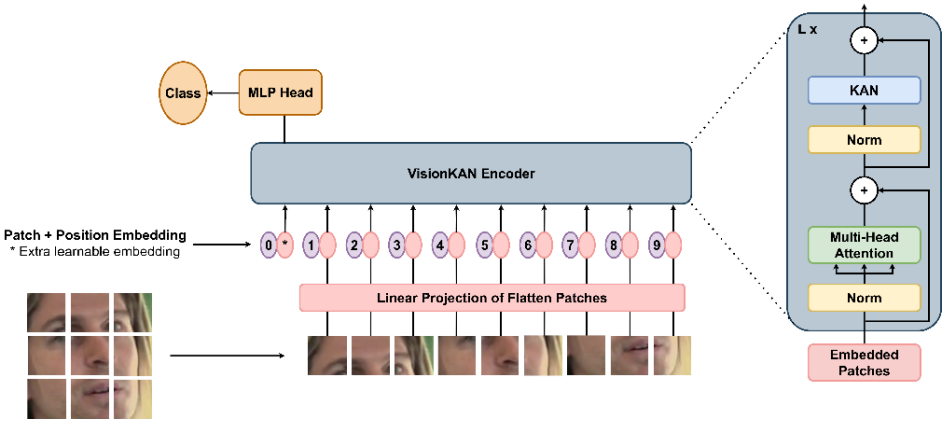

Рис. 5. Архитектура VisionKAN

[1]Vision-KAN. URL: https://github.com/chenziwenhaoshuai/Vision-KAN (дата обращения: 12.08.2024).

5. Архитектура Mamba Vision

В 2024 г. в [14] представлена Mamba Vision. По аналогии с трансформером и визуальным трансформером, основные архитектурные решения наследуются из классической Mamba [15], используемой для задач обработки естественного языка. На рис. 6 представлена архитектура Mamba Vision.

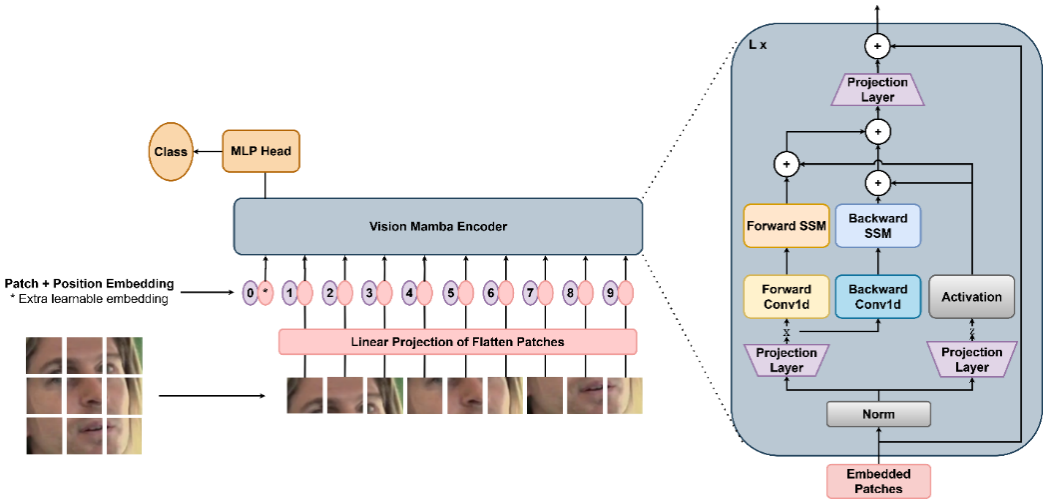

Рис. 6. Архитектура Mamba Vision

В основе Vim лежит модель пространства состояний (SSM), которая позволяет анализировать последовательности данных с помощью матриц состояний.

По аналогии с ViT входное изображение разбивается на патчи. Каждый патч проецируется в вектор, к которому добавляются позиционные эмбеддинги:

$$T_0 = \left[t_{cls};\ t_p^1 W;\ t_p^2 W;\ \ldots;\ t_p^j W\right] + E_{pos}, \tag{19}$$

где t_p^j – это j-й патч, W – обучаемая матрица, E_{pos} – позиционные эмбеддинги.

Формально принцип работы Mamba Vision можно описать следующим образом. На вход энкодера Vim подается последовательность токенов T_{l-1}:

$$T_l = Vim(T_{l-1}) + T_{l-1}. \tag{20}$$

Далее выходной токен T_L^O нормализуется:

$$f = Norm(T_L^O). \tag{21}$$

После чего следует многослойный перцептрон (MLP), который принимает на вход нормализованный вектор f:

$$\hat{p} = MLP(f). \tag{22}$$

В каждом блоке Vim выполняется двунаправленная обработка последовательности токенов. Сначала проводится нормализация токенов, далее – прямой проход через SSM, обратный проход через SSM, в конце результаты прямого и обратного проходов объединяются.

6. Ансамбль моделей

Ключевая идея исследования состоит в том, чтобы проверить, насколько точно современные методы компьютерного зрения способны выявлять дипфейки. Дополнительно, описанные методы можно комбинировать между собой с целью повышения точности распознавания дипфейков. Данный подход наследует концепцию ансамблей в машинном обучении, пример ансамбля показан на рис. 7.

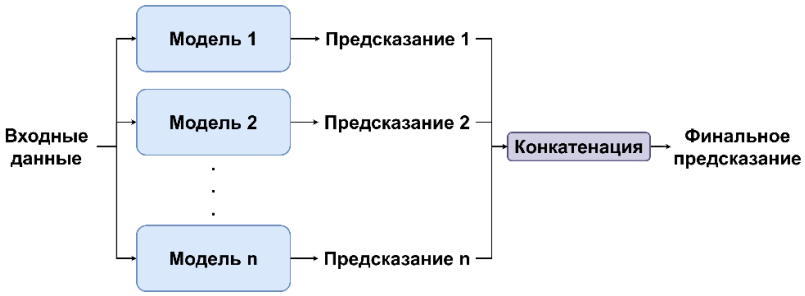

Рис. 7. Пример ансамбля в машинном обучении [16]

В рамках данной работы для построения ансамбля можно рассмотреть создание двух параллельных ветвей, как показано на рис. 8. Ансамбль состоит из двух моделей: «Модель 1» и «Модель 2». В качестве модели 2 используется VisionLSTM, так как она является единственной сетью LSTM, рассматриваемой в данной работе. В роли модели 1 будут выступать ViT, VisionKAN, Mamba Vision, EfficientNet. Каждая из моделей дает предсказание – вектор с размерностью 1 на 1. Полученные вектора объединяются в один вектор с размерностью 1 на 2. Результат конкатенации передается на «Финальную классификацию» – блок из двух классификационных слоев, на выходе которого извлекается вектор с размерностью 1 на 1. В конце, извлеченный вектор нормализуется с помощью сигмоиды. Таким образом, осуществляется «Финальное предсказание», в котором 0 означает, что на входе был подан оригинал, 1 – дипфейк.

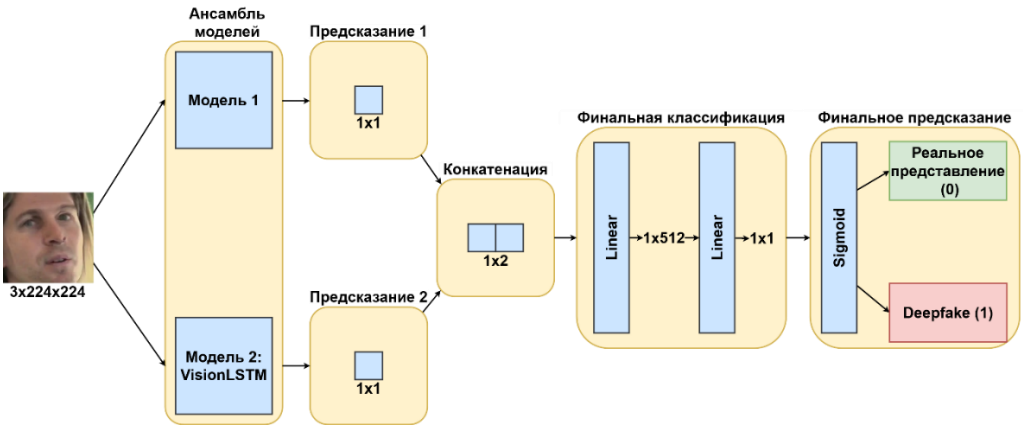

Рис. 8. Предложенная архитектура

Возможные варианты ансамблей для построенной архитектуры:

1. ViT + VisionLSTM.
2. VisionKAN + VisionLSTM.
3. Mamba Vision + VisionLSTM.
4. EfficientNet + VisionLSTM.

7. Описание набора данных

Для проведения экспериментов был выбран набор данных DFDC[2], содержащий в себе видео-образцы дипфейков. Из исходного набора данных был собран новый набор, содержащий в себе фреймы, извлеченные из исходных видео. К каждому фрейму были применены разного рода аугментации: размытие, шум, транспонирование, поворот, изменение размера. В результате аугментаций был получен набор данных, состоящий из 100 тыс. изображений дипфейков. Примеры образцов из набора данных показаны на рис. 9.

а) б) в) г)

Рис. 8. Образцы из собранного набора данных на основе DFDC. а) Исходное изображение, б) Изображение с небольшим поворотом, в) Поворот изображения на 90 градусов, г) Зашумленное изображение

В табл. 1 показано количество образцов по классам оригинал/дипфейк.

Таблица 1. Количество образцов в предложенном наборе данных

	Оригинал	Дипфейк
Обучение	47700	32300
Тестирование	5919	4027

В наборе данных наблюдается дисбаланс в сторону оригинальных изображений, поэтому в качестве метрики была выбрана F_1-мера:

$$F_1 = 2\frac{Recall \cdot Precision}{Recall + Precision} = \frac{TP}{TP + \frac{FP+FN}{2}}. \quad (23)$$

Положительный и отрицательный класс можно определить с помощью матрицы ошибок (рис. 9) следующим образом:

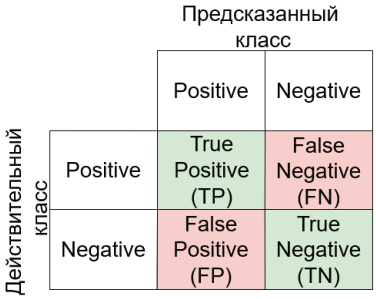

Рис. 9. Матрица ошибок

[2]The DeepFake Detection Challenge Dataset. URL: https://arxiv.org/abs/2006.07397 (дата обращения: 12.08.2024).

8. Тестирование

Тестирование проводилось в два этапа. Сначала все описанные архитектуры были обучены на предложенном датасете по отдельности. Далее тестировалась предложенная ансамблевая архитектура.

По результатам, представленным в табл. 2, 3 и 4, можно сделать вывод о том, что ансамбли позволяют повысить точность, в сравнении с архитектурами, применяемыми по отдельности. Такие результаты можно объяснить тем, что разные архитектуры позволяют учитывать разные признаки. На выходе же будет учитываться именно комбинация предсказаний разных моделей. Таким образом, повышается многообразие анализируемых признаков, что позволяет принимать более точные решения при классификации.

Таблица 2. Общая таблица с результатами

Архитектура	F_1-мера, %
EfficientNet	97,41
Vision Transformer	94,28
VisionLSTM	96,69
VisionKAN	95,34
Mamba Vision	97,37
ViT + VisionLSTM	97,68
VisionKAN + VisionLSTM	97,05
Mamba Vision + VisionLSTM	95,78
EfficientNet + VisionLSTM	96,78

Таблица 3. Сравнение отдельно взятых архитектур ViT и VisionLSTM с их ансамблем

Архитектура	F_1-мера, %
Vision Transformer	94,28
VisionLSTM	96,69
ViT + VisionLSTM	**97,68**

Таблица 4. Сравнение отдельно взятых архитектур VisionKAN и VisionLSTM с их ансамблем

Архитектура	F_1-мера, %
VisionKAN	95,34
VisionLSTM	96,69
VisionKAN + VisionLSTM	**97,05**

Таблица 5. Сравнение отдельно взятых архитектур Mamba Vision и VisionLSTM с их ансамблем

Архитектура	F_1-мера, %
Mamba Vision	97,37
VisionLSTM	96,69
Mamba Vision + VisionLSTM	**95,78**

Таблица 6. Сравнение отдельно взятых архитектур EfficientNet и VisionLSTM с их ансамблем

Архитектура	F_1-мера, %
EfficientNet	97,41
VisionLSTM	96,69
EfficientNet + VisionLSTM	**96,78**

Тем не менее, не все ансамбли показали повышение точности – табл. 5 и 6. Дело в том, что Mamba Vision и EfficientNet имеют принципиально разные архитектуры, в отличие от ViT и VisionKAN. Ключевое отличие заключается в отсутствии механизма внимания как в Mamba Vision, так и в EfficientNet. Предположительно, именно из-за появления в двух

ветвях механизма внимания (ViT и VisionKAN) повышается и точность распознавания дипфейков.

Стоит отметить, что после объединения моделей процесс обучения значительно замедляется, так как появляются две ветви – становится больше обучаемых параметров. Для облегчения архитектуры можно применить методы сжатия нейронных сетей, например, прунинг или квантование. Такие методы позволяют сократить количество обучаемых параметров без потери точности классификации.

Заключение

Каждая из описанных архитектур обладает уникальными особенностями, позволяющими эффективно анализировать изображения. Рассмотренные подходы сочетают в себе свертки, рекуррентные сети и механизмы внимания. Наилучший результат по F_1-мере показала сеть EfficientNet (97,41%). Близкие результаты показала сеть Mamba Vision (97,37%). Стоит отметить, что в экспериментах применялась облегченная версия Mamba Vision, в то время как EfficientNet – самая мощная в семействе EfficientNet.

В ходе экспериментов показана эффективность применения ансамблей современных архитектур компьютерного зрения с точки зрения точности. Например, классический ViT работает с $F_1 = 94{,}28\%$, ансамбль ViT + VisionLSTM работает с $F_1 = 97{,}68\%$. Однако, несмотря на значительные улучшения в точности, комбинированные архитектуры требуют больше вычислительных ресурсов и времени на обучение.

Таким образом, применение современных методов компьютерного зрения для обнаружения дипфейков является перспективным направлением, способным повысить точность обнаружения достоверности визуального контента. Будущие исследования необходимо направить на оптимизацию архитектуры.

СПИСОК ЛИТЕРАТУРЫ:

1. LeCun Y., Bengio Y. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks. 1998, p. 255–258. URL: https://www.semanticscholar.org/paper/Convolutional-networks-for-images%2C-speech%2C-and-time-LeCun-Bengio/563e821bb5ea825efb56b77484f5287f08cf3753 (дата обращения: 23.08.2024).
2. Hochreiter S., Schmidhuber J. Long Short-Term Memory, in Neural Computation, v. 9, no. 8, p. 1735–1780, 15 Nov. 1997. DOI: 10.1162/neco.1997.9.8.1735.
3. Luong T., Pham H., Manning C. Effective Approaches to Attention-based Neural Machine Translation. In Proceedings of EMNLP. 2015, p. 1412–1421. DOI: https://doi.org/10.48550/arXiv.1508.04025.
4. Mingxing T., Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. The 36th International Conference on Machine Learning. 2019, p. 6105–6114. URL: https://arxiv.org/pdf/1905.11946 (дата обращения: 23.08.2024).
5. Sandler M., Howard A., Zhu M., Zhmoginov A., Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA. 2018, p. 4510–4520. DOI: 10.1109/CVPR.2018.00474.
6. Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly S., Uszkoreit J., & Houlsby N. ArXiv, 2020. URL: https://arxiv.org/abs/2010.11929 (дата обращения: 23.08.2024).
7. Ba J., Kiros J.R., & Hinton G.E. Layer Normalization. ArXiv, 2016. URL: https://arxiv.org/abs/1607.06450 (дата обращения: 23.08.2024).
8. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez Aidan N., Kaiser L., Polosukhin I. Attention is All You Need. ArXiv, 2017. URL: https://arxiv.org/abs/1706.03762 (дата обращения: 23.08.2024).
9. Alkin B., Beck M., Pöppel K., Hochreiter S., Brandstetter J. Vision-LSTM: xLSTM as Generic Vision Backbone. ArXiv, 2024. URL: https://arxiv.org/html/2406.04303v2 (дата обращения: 23.08.2024).
10. Dayan P., Willshaw D.J. Optimising synaptic learning rules in linear associative memories. Biological Cybernetics. 1991, p. 253–265. DOI: https://doi.org/10.1007/BF00206223.
11. Gers F.A., Schmidhuber J., Cummins F. Learning to forget: Continual prediction with LSTM. Neural Compututation. Neural Computation. 2000, p. 2451–2471. DOI: 10.1162/089976600300015015.

12. Liu Z, Wang Y, Vaidya S, Ruehle F, Halverson J, Soljačić M, Hou TY, Tegmark M. Kan: Kolmogorov-arnold networks. ArXiv, 2024. URL: https://arxiv.org/abs/2404.19756 (дата обращения: 23.08.2024).
13. Kolmogorov A.K. On the Representation of Continuous Functions of Several Variables by Superposition of Continuous Functions of One Variable and Addition. Doklady Akademii Nauk SSSR. 1957, p. 369–373.
14. Zhu L., Liao B., Zhang Q., Wang X., Liu W., Wang, X. Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model. ArXiv. 2024. URL: https://arxiv.org/abs/2401.09417 (дата обращения: 23.08.2024).
15. Gu A., Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. ArXiv, 2023. URL: https://arxiv.org/abs/2312.00752 (дата обращения: 23.08.2024).
16. Huyen C. Designing Machine Learning Systems: An Iterative Process for Production-ready Applications. O'Reilly Media, Incorporated. 2022. – 367 p. ISBN-13: 1098107969, 9781098107963.

REFERENCES:

[1] LeCun Y., Bengio Y. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks. 1998, p. 255–258. URL: https://www.semanticscholar.org/paper/Convolutional-networks-for-images%2C-speech%2C-and-time-LeCun-Bengio/563e821bb5ea825efb56b77484f5287f08cf3753 (accessed: 23.08.2024).
[2] Hochreiter S., Schmidhuber J. Long Short-Term Memory, in Neural Computation, v. 9, no. 8, p. 1735–1780, 15 Nov. 1997. DOI: 10.1162/neco.1997.9.8.1735.
[3] Luong T., Pham H., Manning C. Effective Approaches to Attention-based Neural Machine Translation. In Proceedings of EMNLP. 2015, p. 1412–1421. DOI: https://doi.org/10.48550/arXiv.1508.04025.
[4] Mingxing T., Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. The 36th International Conference on Machine Learning. 2019, p. 6105–6114. URL: https://arxiv.org/pdf/1905.11946 (accessed: 23.08.2024).
[5] Sandler M., Howard A., Zhu M., Zhmoginov A., Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA. 2018, p. 4510–4520. DOI: 10.1109/CVPR.2018.00474.
[6] Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly S., Uszkoreit J., & Houlsby N. ArXiv, 2020. URL: https://arxiv.org/abs/2010.11929 (accessed: 23.08.2024).
[7] Ba J., Kiros J.R., & Hinton G.E. Layer Normalization. ArXiv, 2016. URL: https://arxiv.org/abs/1607.06450 (accessed: 23.08.2024).
[8] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez Aidan N., Kaiser L., Polosukhin I. Attention is All You Need. ArXiv, 2017. URL: https://arxiv.org/abs/1706.03762 (accessed: 23.08.2024).
[9] Alkin B., Beck M., Pöppel K., Hochreiter S., Brandstetter J. Vision-LSTM: xLSTM as Generic Vision Backbone. ArXiv, 2024. URL: https://arxiv.org/html/2406.04303v2 (accessed: 23.08.2024).
[10] Dayan P., Willshaw D.J. Optimising synaptic learning rules in linear associative memories. Biological Cybernetics. 1991, p. 253–265. DOI: https://doi.org/10.1007/BF00206223.
[11] Gers F.A., Schmidhuber J., Cummins F. Learning to forget: Continual prediction with LSTM. Neural Compututation. Neural Computation. 2000, p. 2451–2471. DOI: 10.1162/089976600300015015.
[12] Liu Z, Wang Y, Vaidya S, Ruehle F, Halverson J, Soljačić M, Hou TY, Tegmark M. Kan: Kolmogorov-arnold networks. ArXiv, 2024. URL: https://arxiv.org/abs/2404.19756 (accessed: 23.08.2024).
[13] Kolmogorov A.K. On the Representation of Continuous Functions of Several Variables by Superposition of Continuous Functions of One Variable and Addition. Doklady Akademii Nauk SSSR. 1957, p. 369–373.
[14] Zhu L., Liao B., Zhang Q., Wang X., Liu W., Wang, X. Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model. ArXiv. 2024. URL: https://arxiv.org/abs/2401.09417 (accessed: 23.08.2024).
[15] Gu A., Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. ArXiv, 2023. URL: https://arxiv.org/abs/2312.00752 (accessed: 23.08.2024).
[16] Huyen C. Designing Machine Learning Systems: An Iterative Process for Production-ready Applications. O'Reilly Media, Incorporated. 2022. – 367 p. ISBN-13: 1098107969, 9781098107963.

Поступила в редакцию – 23 августа 2024 г. Окончательный вариант – 30 октября 2024 г.
Received – August 23, 2024. The final version – October 30, 2024.