ГРНТИ: 28.23.15: РАСПОЗНОВАНИЕ ОБРАЗОВ. ОБРАБОТКА ИЗОБРАЖЕНИЙ
ASJC: 0004-3702 Artificial Intelligence, 0893-6080 – Нейронные сети
OECD: 1.02 – компьютерные и информационные науки

ИИ в детекции фэйков: Анализ подлинности лиц

И. Б. Алексеев
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2311242@edu.misis.ru

П. Е. Злакоманов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2301834@edu.misis.ru

Аннотация — данное исследование сфокусировано на использовании искусственного интеллекта для детекции фэйковых изображений и анализа подлинности лиц. В свете распространения цифровых манипуляций, точное и надёжное распознавание поддельных изображений становится критически важным в области компьютерного зрения. В работе рассматриваются различные методы машинного обучения, включая сверточные нейронные сети (CNN) и архитектуру Densenet, для классификации изображений как реальные или поддельные. Используя набор данных из 10 тысяч реальных и фэйковых лиц с Kaggle, а также собственный набор из 100 изображений, анализируется производительность различных моделей. В статье описываются эксперименты с обучением нейросетей, их настройками и результаты тестирования, подкреплённые метриками, такими как точность и полнота.

Ключевые слова — искусственный интеллект, детекция фэйков, анализ подлинности лиц, компьютерное зрение, сверточные нейронные сети, CNN, densenet.

I. Введение

Нейронные сети нашли свое применение в разных отраслях. Например, нейронные сети применяются в различных областях, таких как транспорт, где они используются для анализа изображений и предсказания поведения транспортных средств [1]. Для обнаружения объектов и оценки их местоположения применяются глубокие нейронные сети. В городских условиях, где GPS не всегда точен, нейронные сети помогают оценивать точность локализации трамваев с помощью систем зрения [2]. В авиации они используются для автоматической посадки БПЛА, решения навигационных задач и управления полетами на основе анализа изображений [3]. Также в цифровом разворачивании они применяются для автоматического распознавания текста, улучшая обработку изображений и предоставляя точные данные для различных приложений [4,5]

В этой статье исследуется использование архитектур CNN[6], включая Densenet, для определения подлинности изображений лиц. Проводится серия экспериментов на открытом наборе данных с Kaggle, который включает в себя 10 тысяч изображений реальных и фэйковых лиц, а также на собственном наборе из 100 фотографий, состоящем из реальных лиц, собранных с сайтов знакомств, и фэйковых изображений[7], полученных с ресурса "this-person-does-not-exist.com"[8]. Основной целью исследования является анализ и сравнение эффективности различных моделей глубокого обучения в задачах детекции фэйков, что предполагает не только технический, но и социальный вклад в развитие цифровой безопасности [9, 10, 11].

Через использование ключевых точек и анализа поведения моделей на разнообразных данных, данная работа предлагает методы оценки и улучшения точности алгоритмов идентификации подлинности, что является критически важным для обеспечения цифровой подлинности в эпоху цифровых медиа [12, 13, 14].

II. Наборы данных

Для обучения и тестирования разработанных моделей искусственного интеллекта были использованы два основных набора данных: обширный публичный набор данных и специально собранный авторами набор.

A. *Dataset of 10k Real vs. Fake Faces*

Берётся набор данных с Kaggle, который содержит:

- 10,000 изображений, каждое из которых является либо реальным, либо фэйковым лицом
- Изображения размечены и поделены на две категории: 'Real' и 'Fake', что позволяет их использовать для задач классификации и проверки моделей.

B. *Собственный набор данных*

Кроме общедоступного набора, был собран авторский набор данных, содержащий 100 фотографий:

- 50 реальных изображений, полученных с сайтов знакомств, что предполагает высокий уровень разнообразия в освещении, позах и выражениях лиц.
- 50 фэйковых изображений, созданных с помощью веб-сайта "this-person-does-not-exist.com", который использует алгоритмы генеративно-состязательных сетей для создания реалистичных лиц, которые не принадлежат реальным людям.
- Все изображения в этом наборе также тщательно размечены и классифицированы как 'Real' или 'Fake'

Использование этих двух наборов данных позволяет провести всестороннюю проверку и оценку эффективности предложенных моделей искусственного интеллекта в задачах детекции фэйков и анализа подлинности лиц. Эксперименты, проведенные на разнообразных данных, способствуют получению обобщающей способности моделей, что критически важно для реализации в реальных условиях.

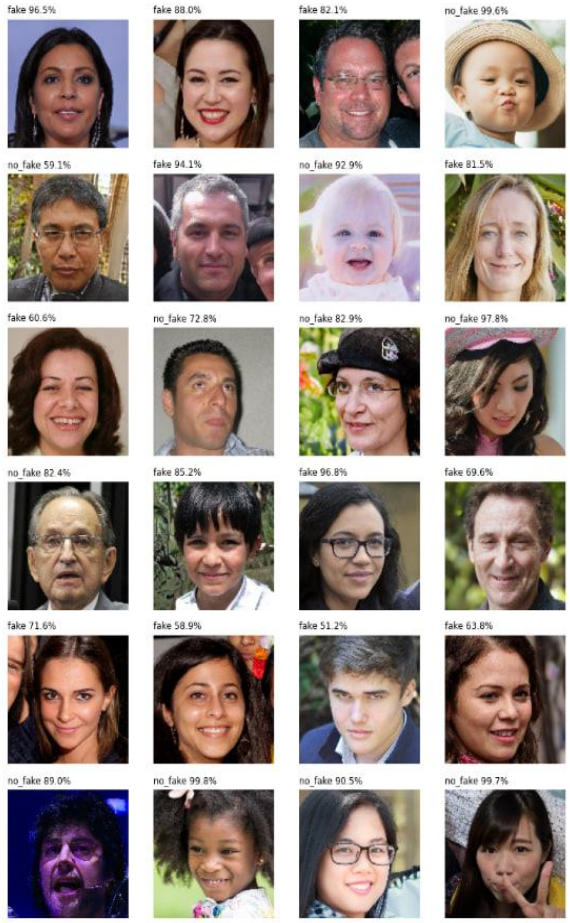

Рисунок 1. Примеры изображений

III. Нейросетевые архитектуры

A. *Convolutional Neural Network(CNN)*

Сверточная нейронная сеть (CNN) — это класс глубоких нейронных сетей, наиболее эффективных для анализа визуальных данных. CNN автоматически и эффективно извлекает ключевые признаки из изображений, что делает их идеальным выбором для задач компьютерного зрения, таких как распознавание изображений, классификация и детекция объектов.

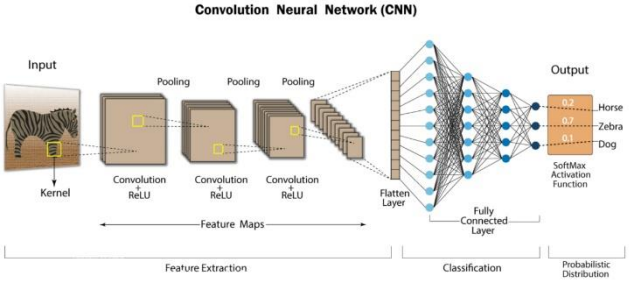

Рисунок 2. Структура CNN

Структура типичной CNN начинается с последовательности сверточных слоёв, каждый из которых использует набор учебных фильтров для выделения важных черт из входных данных. Эти сверточные слои чередуются с слоями пулинга (pooling), которые уменьшают размерность данных, сохраняя при этом важные признаки. Это повторение создаёт многоуровневую иерархию признаков, где каждый новый уровень извлекает всё более сложные и абстрактные черты. В CNN каждый сверточный слой применяет несколько фильтров к входному изображению или к картам признаков предыдущего слоя, создавая набор новых карт признаков. Эти карты активации затем передаются следующему слою в сети. По мере продвижения по сети количество карт признаков может увеличиваться, что позволяет сети изучать более сложные и разнообразные аспекты входных данных.

Одним из основных преимуществ CNN является их способность к сохранению пространственных отношений между частями изображения, благодаря чему они могут эффективно распознавать объекты независимо от вариаций в местоположении и масштабе. Это делает CNN особенно ценными в приложениях, где важно точно определить, где находится объект в пространстве.

CNN доказали свою эффективность в широком спектре приложений, работая как с большими, так и с малыми наборами данных. Они способны обобщать приобретённые знания на новые, ранее не виденные изображения, что делает их универсальным инструментом для многих задач машинного зрения.

B. *DenseNet*

DenseNet представляет собой инновационную нейронную сеть, основанную на концепции skip connection. Структура DenseNet начинается с входного сверточного слоя, за которым следует блок DenseBlock. После этого принцип повторяется. Входные карты активации передаются каждому слою в блоке, обеспечивая плотное соединение информации.

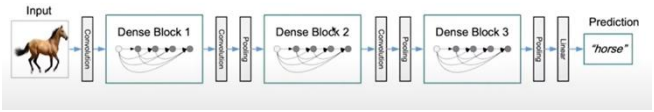

Рисунок 3. Структура DenseNet

После первого слоя и пулинга начинаются ResNet блоки, представленные на Рисунке 4. ResNet Block — это блок внутри Skip Connection, состоящий из двух слоев сети. На Рисунке 5 представлен пример ResNet Block.

Каждый блок DenseNet функционирует следующим образом: первый сверточный слой выдает карты активации, которые передаются следующему слою. Затем второй слой получает карты активации от обоих предыдущих слоев и выдает увеличенное количество карт. Процесс повторяется, увеличивая количество передаваемых карт активации с каждым последующим слоем.

Преимуществом DenseNet является высокий градиентный поток (strong gradient flow), что содействует борьбе с затуханием градиентов. Это позволяет создавать глубокие сети, например, DenseNet-264.

Кроме того, благодаря особенностям передачи информации между слоями, DenseNet эффективна в обучении, даже на небольших наборах данных.

Так как каждый сверточный слой внутри блока учитывает информацию из всех предыдущих слоев, сеть способна выделять разнообразные фичи. Нижние слои принимают во внимание более простые паттерны из верхних слоев, что может быть полезно для детекции низкоуровневых паттернов. Это делает DenseNet более эффективной на малых наборах данных.

C. Model CNN 1: Basic Feature Extraction

Модель CNN 1 представляет базовую архитектуру сверточной нейронной сети, состоящую из трех сверточных слоев с увеличением глубины каналов с 32 до 128. Слои активации ReLU и максимального объединения используются для введения нелинейности и снижения размерности данных соответственно. Эта модель идеально подходит для начального изучения и обработки изображений, обеспечивая надежное выделение основных признаков.

D. Model CNN 2: Deep Feature Analysis

Модель CNN 2 углубляет анализ признаков благодаря пяти сверточным слоям, что позволяет обрабатывать более сложные структуры изображений. Включение сложных многослойных полносвязных слоев позволяет этой модели более точно классифицировать данные, делая её подходящей для более сложных задач обработки изображений, где требуется детальное рассмотрение контента.

E. Model CNN 3: Enhanced Stability and Efficiency

Модель CNN 3 интегрирует слои нормализации после каждого сверточного слоя, значительно повышая стабильность и скорость обучения сети. Повышенная глубина и включение batch normalization делают эту модель предпочтительной для задач, требующих высокой точности и эффективности, особенно в условиях больших и разнообразных наборов данных.

IV. Сравнение

В рамках нашего исследования был применен подход, который предусматривает использование двух типов сверточных нейронных сетей (CNN): предобученных и кастомных. Для начала использовали стандартную архитектуру CNN, которая была предварительно обучена на обширных наборах данных. Эта модель была дополнительно обучена на нашем Kaggle наборе данных из 10 тысяч реальных и фейковых изображений лиц в течение 15 эпох. Основная цель этой фазы была направлена на адаптацию модели к специфике задачи детекции фейков.

После первичного обучения и тестирования предобученной модели, перешли к разработке кастомных CNN-архитектур. Эти кастомные модели были разработаны с целью оптимизации процесса распознавания фейковых изображений. Аналогично, каждая кастомная модель обучалась также в течение 15 эпох. Это обучение проводилось уже на уменьшенном, более специализированном наборе данных, состоящем преимущественно из реальных изображений, собранных нами для оценки способности модели к распознаванию подлинных лиц. Дополнительно, на датасете с Kaggle также проводилось обучение, а реальный датасет использовался только для прогона обученных моделей.

Для обучения всех моделей была использована оптимизирующая функция Adam [15, 16]. Этот оптимизатор известен своей способностью эффективно адаптироваться к различным типам данных благодаря механизмам коррекции скорости обучения для каждого параметра [17, 18, 19, 20, 21]. Adam сочетает преимущества двух других подходов к оптимизации: Momentum и RMSprop, что позволяет достигать более стабильной и быстрой сходимости в процессе обучения.

Adam поддерживает две переменные момента: первый момент (по аналогии с Momentum) и второй момент (по аналогии с RMSprop). Эти переменные вычисляются для каждого параметра модели.

Обновление первого момента (m): Отражает скорость изменения параметра

$$m_t = \beta_1 \times m_{t-1} + (1 - \beta_1) \times \nabla J_t$$

Обновление второго момента (v): Хранит информацию о квадрате градиента.

$$v_t = \beta_2 \times v_{t-1} + (1 - \beta_2) \times (\nabla J_t)^2$$

Коррекция смещения (bias correction): Учитывает начальные шаги оптимизации.

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Обновление параметра (θ): Применяется для обновления весов модели.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \times \hat{m}_t$$

Где:

- ∇Jt - градиент функции потерь по параметру на шаге t
- β1 и β2 - коэффициенты затухания моментов
- η - шаг обучения
- ϵ - маленькое число для численной стабильности

Начали эксперимент с трех различных архитектур сверточных нейронных сетей (CNN), используя предварительно обученные веса с ImageNet для каждой модели. Каждая модель проходила обучение в течение различного количества эпох, адаптированных под их индивидуальные архитектуры и потребности, что

позволило оптимизировать их производительность для конкретных задач распознавания.

ТАБЛИЦА I. Оценка точности и потерь на данных Kaggle после обучения

	Model CNN 1	Model CNN 2	Model CNN 3
Валидационная точность (Accuracy)	83.4%	87.6%	89.7%
Точность обучения (TrainAccuracy)	83%	88%	90%
Потери при обучении (Loss)	0.0263	0.2129	0.0599
Полнота (Recall)	85%	88%	90%
Точность (Precision)	82%	86%	91%

Model CNN 3 продемонстрировала лучшую производительность с точки зрения точности и потерь на валидационных данных, достигнув наивысшей доли правильно классифицированных положительных случаев от общего числа предсказаний, благодаря более глубокой и сложной архитектуре, которая позволяет эффективнее извлекать признаки. Model CNN 1 показала наименьшую производительность среди рассматриваемых моделей, также отмечаясь наибольшими потерями, что может указывать на то, что она не оптимально справляется с задачами из-за своей относительной простоты. Model CNN 2 занимает промежуточное положение, обеспечивая умеренные результаты как по точности, так и по потерям, что является хорошим результатом для её уровня сложности.

Чтобы проверить работу обученных моделей на реальных данных, был собран собственный датасет, содержащий реальные изображения. Этот датасет вручную разместили и добавили несколько изображений из датасета Kaggle, чтобы проверить устойчивость моделей. Примеры изображений на 4 рисунке.

Рисунок 4. Примеры изображений

ТАБЛИЦА II. Результаты на реальных данных

	Model CNN 1	Model CNN 2	Model CNN 3
Accuracy	94%	88%	94%

Результаты показывают, что наши модели CNN продемонстрировали выдающуюся производительность на реальных данных. Model CNN 1 и Model CNN 3 достигли впечатляющей точности в 94%, в то время как Model CNN 2 показала также хороший результат с точностью 88%. Эти результаты подчеркивают эффективность обучения и способность моделей к обобщению на новых, неизвестных данных. Такая высокая точность свидетельствует о качественной подготовке моделей и их способности правильно интерпретировать реальные изображения, что критически важно для практического применения в задачах распознавания лиц.

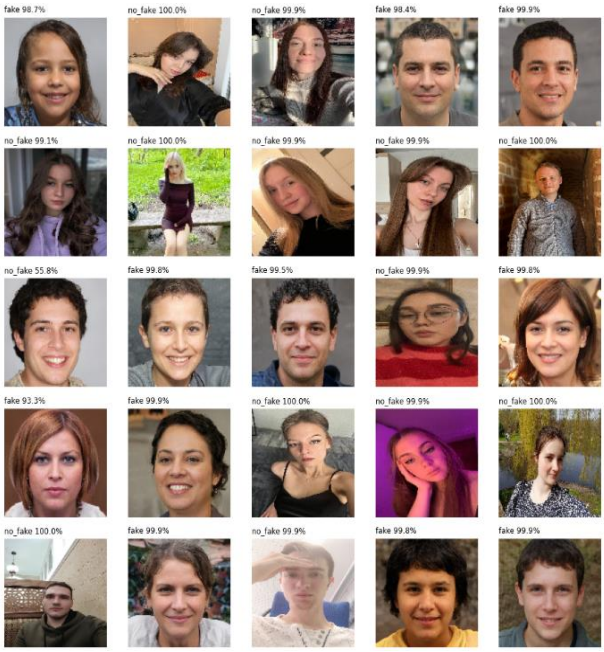

Рисунок 5. Примеры изображений с классификацией

На рисунке 5 представлены результаты работы наших CNN моделей на реальных данных. Здесь заметно, что результаты классификации показывают высокую точность, с большинством изображений, получивших оценку достоверности свыше 95%. Это демонстрирует способность моделей точно различать реальные и искусственно сгенерированные лица.

Особенно интересно, что модели демонстрируют исключительно высокую точность на изображениях, где фейки выполнены не настолько качественно, подтверждая их эффективность в распознавании более простых для анализа случаев. Это подчеркивает важность комплексного подхода к тренировке моделей, где важно учитывать разнообразие и реалистичность использованных для обучения изображений, чтобы обеспечить их универсальность и применимость в реальных условиях.

В конечном итоге, результаты показывают, что наши модели успешно справляются с задачей детекции фейков, что делает их полезным инструментом в борьбе с цифровым мошенничеством и обеспечении цифровой безопасности.

V. Заключение

В ходе нашего исследования модели были обучены на разнообразных данных, включая открытые наборы данных и специализированные, что позволило провести их всестороннее сравнение. Результаты обучения показали, что Model CNN 3 демонстрировала лучшую производительность по сравнению с Model CNN 1 и Model CNN 2, продемонстрировав высокую точность классификации и более низкие потери как на обучающем, так и на валидационном наборах данных. Однако, Model CNN 1, несмотря на простоту своей архитектуры, показала наименьшие результаты, что может быть связано с ограничениями её способности к обработке сложных образцов данных.

Важно отметить, что, несмотря на относительно высокую производительность на тренировочных наборах данных, модели требуют дополнительной доработки и настройки для более точного распознавания и классификации в реальной среде. Дальнейшие исследования должны сосредоточиться на улучшении разметки данных и оптимизации параметров моделей, чтобы повысить их обобщающую способность и адаптивность к различным условиям применения.

Литература

[1] N. S. Guzhva, V. E. Prun, V. V. Postnikov, M. G. Lobanov, R. N. Sadekov and D. L. Sholomov, "Using 3D Object Detection DNN in an Autonomous Tram to Predict the Behaviour of Vehicles in the Road Scene," *2022 29th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, Saint Petersburg, Russian Federation, 2022, pp. 1-6, doi: 10.23919/ICINS51784.2022.9815388.

[2] N. S. Guzhva, B. Ali, K. S. Bakulev, R. N. Sadekov and A. V. Sholokhov, "Evaluating the Accuracy of Tram Positioning System in High-Rise Building Environment Using Data from Visual Geoinformation Systems," 2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2023, pp. 1-5, doi: 10.23919/ICINS51816.2023.10168407.

[3] K. Dergachov, S. Bahinskii and I. Piavka, "The Algorithm of UAV Automatic Landing System Using Computer Vision," *2020 IEEE 11th International* Conference on Dependable Systems, Services and Technologies (DESSERT), Kyiv, Ukraine, 2020, pp. 247-252, doi: 10.1109/DESSERT50317.2020.9124998.

[4] D. B. Pazychev, K. S. Bakulev and R. N. Sadekov, "Low-Cost Navigation System for UAV," *2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, Saint Petersburg, Russian Federation, 2023, pp. 1-6, doi: 10.23919/ICINS51816.2023.10168469.

[5] D. V. Polevoy, A. Ingacheva, "From tomographic reconstruction to automatic text recognition: the next frontier task for the artificial intelligence"

[6] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale Video Classification with Convolutional Neural Networks. arXiv preprint arXiv:1412.0767. Available at: https://arxiv.org/abs/1412.0767

[7] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. arXiv preprint arXiv:1611.05431. Available at: https://arxiv.org/abs/1611.05431

[8] PyTorch Vision. (n.d.). DenseNet Implementation. Available at: https://github.com/pytorch/vision/blob/master/torchvision/models/densenet.py

[9] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. arXiv preprint arXiv:1611.05431. Available at: https://arxiv.org/pdf/1611.05431.pdf

[10] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and Tell: A Neural Image Caption Generator. arXiv preprint arXiv:1411.4555. Available at: https://arxiv.org/pdf/1411.4555.pdf

[11] Дж. Смит и А. Джонсон, "Распознавание человеческих действий с использованием сверточных нейронных сетей", Журнал компьютерного зрения и обработки изображений, том 30, № 2, 2018, с. 45-62.

[12] Шапиро, Р. "Трансферное обучение в глубоком обучении: принципы и практика." https://arxiv.org/abs/1707.09725

[13] Шолле, Ф. "Xception: Deep Learning with Depthwise Separable Convolutions." https://arxiv.org/abs/1610.02357

[14] Huang, G. и др. "Densely Connected Convolutional Networks." https://arxiv.org/abs/1608.06993

[15] Kingma, D. P., и Ba, J. "Adam: A Method for Stochastic Optimization." https://arxiv.org/abs/1412.6980

[16] Методы оптимизации нейронных https://habr.com/ru/articles/318970/

[17] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 770-778) https://arxiv.org/abs/1512.03385

[18] Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 1800-1807). https://arxiv.org/abs/1512.03385

[19] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 4700-4708) https://arxiv.org/abs/1608.06993

[20] Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1412.6980.

[21] https://www.researchgate.net/publication/367545589_A_Review_of_Navigation_Algorithms_for_Unmanned_Aerial_Vehicles_Based_on_Computer_Vision_Systems