УДК 004.056+004.032.26+004.932

ПРИМЕНЕНИЕ ПИРАМИДАЛЬНОГО ВИЗУАЛЬНОГО ТРАНСФОРМЕРА ДЛЯ ОБНАРУЖЕНИЯ ПОДДЕЛЬНЫХ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

Зубков Павел Андреевич, Ильяшенко Илья Дмитриевич

Алтайский государственный университет, г. Барнаул
e-mail: pav.zubkoff@mail.ru, ilya-ilyash@yandex.ru

THE USE OF A PYRAMID VISION TRANSFORMER TO DETECT FAKE DIGITAL IMAGES

Zubkov Pavel A., Ilyashenko Ilya D.

Altai State University, Barnaul

Аннотация. На сегодняшний день большая часть изображений хранится и распространяется в цифровом виде. Простота использования и доступность программных инструментов и недорогого оборудования позволяет очень просто подделывать цифровые изображения, не оставляя практически никаких следов. Таким образом, в наше время мы не можем принимать подлинность и целостность цифровых изображений как должное. В данной работе предложено применение алгоритма глубокой нейронной сети, построенного на основе пирамидального визуального трансформера, для задачи обнаружения поддельных цифровых изображений. Было проведено обучение алгоритма на наборе данных с поддельными цифровыми изображениями. Произведены эксперименты, представлены результаты работы алгоритма. Проведена проверка работы алгоритма на изображениях с разными типами подделки. Выполнено сравнение результатов работы алгоритма с результатами других современных методов обнаружения подделок.

Ключевые слова: глубокие нейронные сети, механизм внимания, обнаружение поддельных цифровых изображений, пирамидальный визуальный трансформер, трансформер.

Abstract. To date, most of the images are stored and distributed digitally. The ease of use and availability of software tools and inexpensive equipment makes it very easy to fake digital images, leaving virtually no trace. Thus, nowadays we cannot take the authenticity and integrity of digital images for granted. In this paper, we propose the use of a deep neural network algorithm based on a Pyramid Vision Transformer for the task of detecting fake digital images. The algorithm was trained on a dataset with fake digital images. Experiments have been carried out, the results of the algorithm are presented. The algorithm was tested on images with different types of forgery. The results of the algorithm are compared with the results of other modern methods of detecting fakes.

Keywords: deep neural networks, attention mechanism, detection of fake digital images, Pyramid Vision Transformer, Transformer.

Для цитирования: Зубков П.А., Ильяшенко И.Д. Применение пирамидального визуального трансформера для обнаружения поддельных цифровых изображений // Проблемы правовой и технической защиты информации. 2023. №11. С. 21-28.

© Зубков П.А., Ильяшенко И.Д., 2023

For citation: Zubkov P.A., Ilyashenko I.D. The use of a Pyramid Vision Transformer to detect fake digital images // Legal and Technical Problems Information Protection. 2023. No. 11. P. 21-28.

Для решения задач компьютерного зрения, в том числе выявления поддельных цифровых изображений, часто используют картинки, представленные в виде трехмерного массива (высота, ширина, количество каналов), для которого применяются свертки. Однако, такой подход имеет свои недостатки:

– Не все пиксели одинаково полезны. Например, если поставлена задача сегментации, то объект на изображении важнее, чем фон.

– Свертки плохо работают с пикселями, которые удалены на большое расстояние друг от друга.

– Свертки мало эффективны в очень глубоких нейронных сетях.

Именно поэтому в рассматриваемом алгоритме, архитектура которого приведена на рисунке 1, для извлечения признаков из изображения используется алгоритм Pyramid Vision Transformer [1], процесс обработки которого начинается с размеров входного изображения, а затем оно подвергается обработке с использованием последовательности Transformer-блоков, уменьшающих масштаб изображения на каждой стадии.

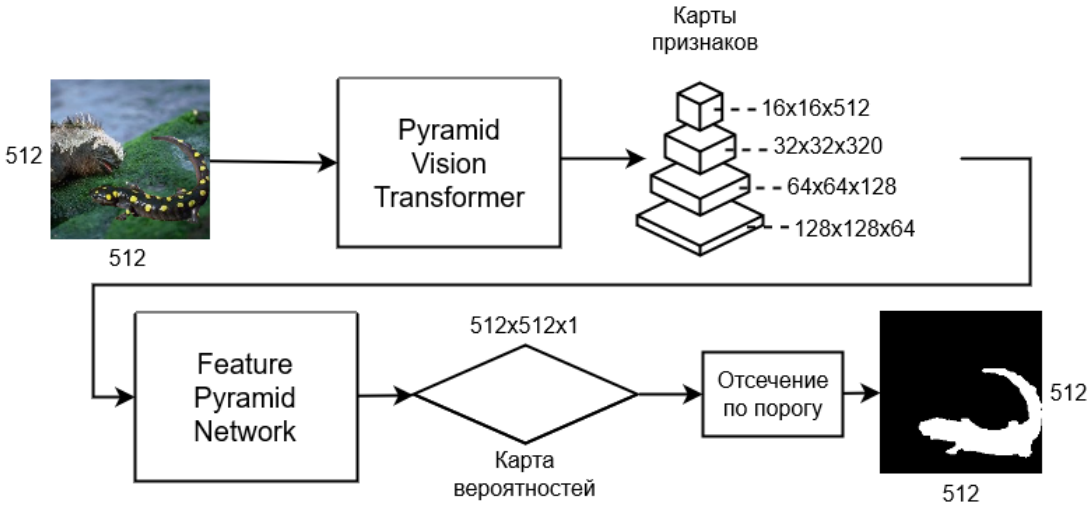

Рисунок 1. Архитектура алгоритма

На вход алгоритма, состоящего из энкодера и декодера, где энкодер – это сеть пирамидального визуального трансформера (Pyramid Vision Transformer, PVT) [1], а декодер – функциональная пирамидальная сеть (Feature Pyramid Network, FPN) [2], подается тензор размерностью [B, C, H, W], где B – batch size, C – количество каналов, H и W – высота и ширина изображения. В энкодере применяется слой внимания для вычисления весовых коэффициентов, которые отражают то, насколько важны различные пиксели и объекты на изображении для предсказания конкретного класса объекта, а также слой прямой передачи, используемый для объединения выходных данных из разных блоков на предыдущем уровне пирамиды в единый поток данных, который затем передается на следующий уровень. Таким образом происходит позиционное кодирование. Декодер, в свою очередь, при помощи сверток генерирует карты вероятностей отнесения каждого пикселя к классу подделки на каждом уровне рассмотрения,

после чего объединяет их в общую карту вероятностей.

На выходе алгоритма получается тензор [C, H, W], где C – количество классов. Полученный тензор состоит из вероятностей отнесения каждого пикселя к тому или иному классу. Так как класс один, то в тензоре вероятность только одного класса. Чтобы принять решение о принадлежности к классу, происходит отсечение вероятности по порогу 0,5. Если значение больше 0,5, то этот пиксель является измененным, если меньше – оригинальным.

В алгоритме Pyramid Vision Transformer [1], архитектура которого представлена на рисунке 2, была внедрена пирамидальная структура для анализа изображений на разных масштабах.

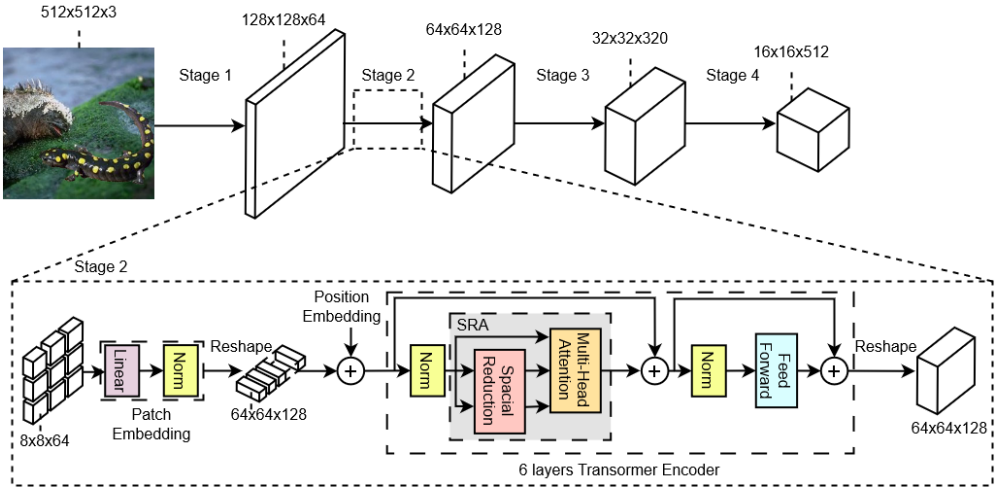

Рисунок 2. Архитектура Pyramid Vision Transformer [1]

Для эффективной обработки изображений была разработана новая схема агрегации, позволяющая объединять одинаковые объекты на разных масштабах. Данная модель использования информации о глобальном контексте изображения позволяет обнаруживать объекты на разных масштабах и повышать точность их обнаружения за счет механизма внимания.

Рассматриваемый алгоритм включает в себя четыре этапа, которые генерируют карты признаков на различных масштабах. Каждый этап имеет одинаковую архитектуру, которая включает в себя слой Patch Embedding и Li слоев Transformer Encoder (рисунок 2) [1].

На первом этапе изображение размером H×W×3 разбивается на $\frac{HW}{4^2}$ патча размером 4×4×3. Затем патчи проходят через линейную проекцию и преобразуются во встроенные патчи размером $\frac{HW}{4^2}\times C_1$. После этого патчи и Position Embedding, используемый для внесения информации о позиции каждого пикселя изображения в модель путем добавления вектора позиции к каждому пикселю, подаются на вход Transformer Encoder с количеством слоев – L_1. Выходные данные преобразуются в карту признаков F_1 размером $\frac{H}{4}\times\frac{W}{4}\times C_1$. Таким же образом получаются карты признаков F_2, F_3 и F_4, чьи шаги составляют 8, 16 и 32 пикселя по отношению к входному изображению, используя карту признаков предыдущего этапа в качестве входных данных.

Transformer Encoder на i этапе состоит из двух слоев: слоя внимания и слоя прямой передачи [3], как показано на рисунке 3.

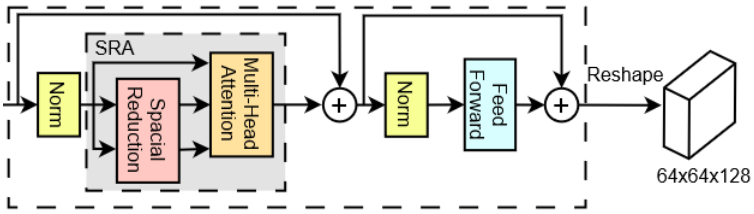

Рисунок 3. Transformer Encoder в PVT [1]

Слой внимания обеспечивает моделирование взаимоотношений и взаимодействий между различными частями изображения на разных масштабах рассмотрения. В PVT [1] слой внимания используется для вычисления весовых коэффициентов, которые отражают, то насколько важны различные пиксели и объекты на изображении для предсказания конкретного класса объекта. Слои внимания позволяют модели сосредоточиться на наиболее информативных признаках изображения.

Слой прямой передачи [3] помогает установить связь между различными уровнями пирамидальной структуры изображения. В частности, он используется для объединения выходных данных из разных блоков на предыдущем уровне пирамиды в единый поток данных, который затем передается на следующий уровень. Это позволяет выполнять масштабирование изображений и получать изображения с различными уровнями детализации.

Основная идея FPN [2], архитектура которого представлена на рисунке 4, заключается в том, чтобы использовать информацию со всех уровней пирамиды признаков, получаемой из PVT [1], для точной сегментации объектов на изображении. При этом основное внимание уделяется использованию контекстной информации больших объектов, которая может быть получена с помощью верхних слоев сети, и локальной информации мелких объектов с помощью нижних слоев сети.

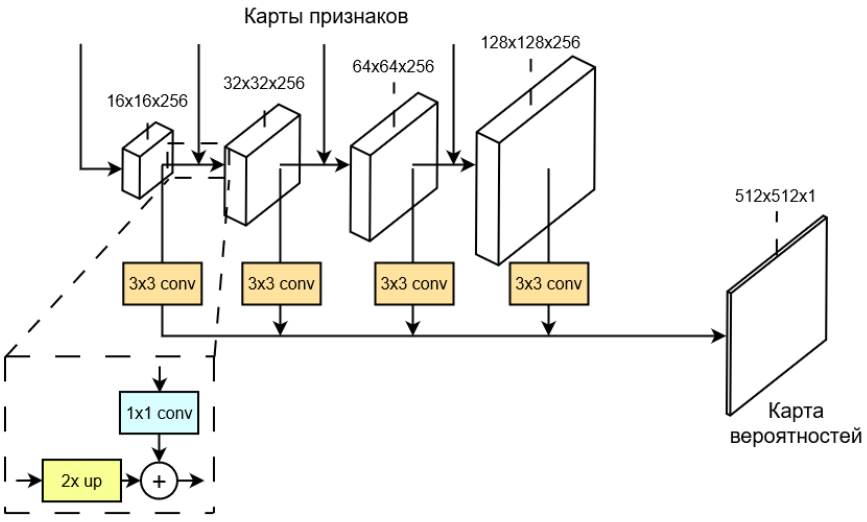

Рисунок 4. Архитектура декодера FPN [2]

В рассматриваемом алгоритме обнаружения поддельных цифровых изображений сеть FPN используется как декодер и, соответственно, применяется путь сверху вниз и соединения, где соединения соответствуют картам признаков, полученным в результате работы алгоритма PVT.

Нисходящий путь улучшает разрешение признаков за счет увеличения

пространственного размера, но при этом сохраняется точность семантики благодаря использованию карт признаков более высоких уровней. Далее эти признаки расширяются благодаря соединениям с картами признаков из PVT. Горизонтальные соединения объединяют карты признаков одинакового пространственного размера из PVT и FPN. Благодаря этим связям признаки с высоких уровней используются для идентификации больших областей изображения, а признаки с низких уровней позволяют уточнять границы и детали изображения.

Используя полученные признаки декодер при помощи сверток генерирует карты вероятностей отнесения каждого пикселя к классу подделки на каждом уровне рассмотрения, после чего объединяет их в общую карту вероятностей. Затем происходит отсечение по порогу 0,5. Если вероятность больше 0,5, то пиксель является поддельным.

В качестве базы данных для проведения экспериментов была выбрана CASIA ITDE V2 [4], которая содержит 12323 изображения и делится на два набора: подлинные и поддельные. Подлинный набор содержит 7200 изображений, а набор с подделками – 5123. Набор с поддельными изображениями был разделен на три выборки: обучающую (4084 изображения), проверочную (511 изображений) и тестовую (511 изображений).

Размеры изображений в CASIA ITDE V2 различны: они варьируются от 320×240 пикселей до 800×600 пикселей. В базе данных помимо изображений формата JPEG присутствуют BMP и TIFF. Кроме того, изображения JPEG рассматриваются с различными коэффициентами сжатия.

Поддельные изображения созданы с помощью операций Copy-Move и Image Splicing в программе Adobe Photoshop. После операций подделки используется постобработка изображений, чтобы скрыть явные следы фальсификации. Пример поддельного цифрового изображения, полученного при помощи использования метода Copy-Move, и бинарной маски измененной области представлены на рисунке 5, где слева находится подделка, а справа – бинарная маска измененной области.

Рисунок 5. Пример из базы данных CASIA ITDE V2 [4]

Перед началом обучения была проведена корректировка масок, так как в исходном наборе они были трехканальными со значениями яркости [0, 255], а для обучения требовались одноканальные маски со значениями [0, 1].

На вход алгоритм принимает изображения размером 512×512 пикселей. Поскольку изображения в исследуемом наборе имеют разные размеры, то их необходимо привести к одному размеру. Отсюда если размер изображения меньше, то оно увеличивается путем добавления нулей по краям случайным образом, чтобы соответствовать размеру 512×512. Если же изображение больше, чем 512×512, то из него вырезается случайная область нужного размера.

В качестве метрик для оценки работы алгоритма были выбраны: Accuracy,

Precision, Recall, F1-score и Intersection over Union (IoU) [5, 6, 7].

Алгоритм написан на языке программирования Python с использованием библиотек PyTorch и MMSegmentation [8]. Метод оптимизации – AdamW, где learning_rate = 6×10-5, weight_decay = 0,01, остальные гипермараметры взяты по умолчанию.

Обучение модели проходило в 320 тысяч итераций на 4084 изображениях с различными видами фальсификаций и на соответствующих им масках. Кроме того, ко всем изображениям добавлялось 4 вида аугментаций: случайное изменение яркости, отражение по горизонтали или вертикали, повороты на малые углы, изменение масштаба. Все эти аугментации способствовали увеличению набора данных в обучающей выборке.

На рисунке 6 показаны графики метрик Accuracy и IoU при обучении алгоритма, которые вышли на плато.

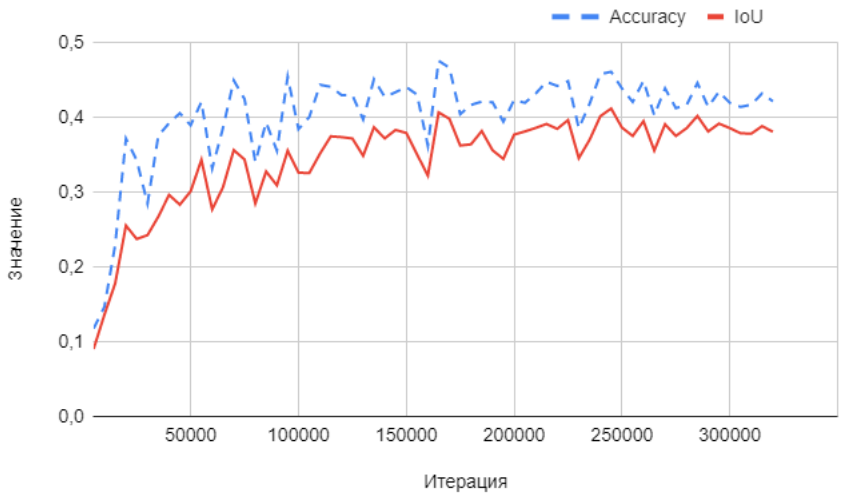

Рисунок 6. Графики метрик при обучении

На рисунке 7 представлен результат работы модели, где слева – это поддельное изображение, в центре – бинарная маска поддельной области, а справа – предсказанная моделью бинарная маска измененного фрагмента. Белые пиксели – это поддельные пиксели, черные – оригинальные.

Таблица 1 отражает результаты метрик после проведения эксперимента.

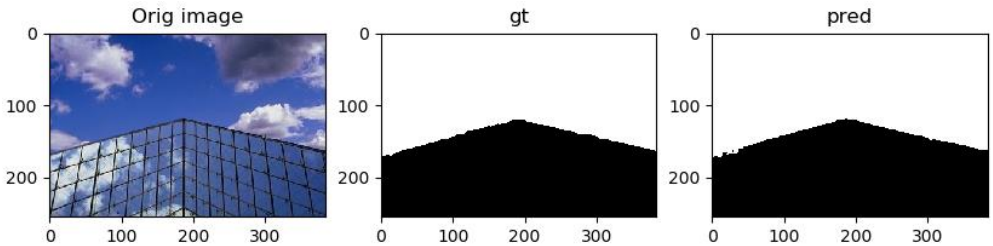

Рисунок 7. Результат работы модели

Таблица 1. Результаты метрик после проведения тестирования

Accuracy, %	F_1-score, %	Precision, %	Recall, %	IoU, %
44,09	53,11	66,77	44,09	36,16

Алгоритм, обученный с использованием аугментаций, был проверен на разных типах поддельных цифровых изображений. В таблице 2 представлены

результаты метрик для сравнения обнаружения подделок типа Copy-Move и Image Splicing реализованным алгоритмом.

Из результатов следует, что алгоритм лучше распознает подделки, созданные при помощи метода Image Splicing. Это происходит из-за того, что склеенные изображения имеют между собой большой ряд отличительных признаков нежели подделки типа Copy-Move, на которых скопированные части имеют схожие характеристики, что делает их трудно обнаружимыми.

В таблице 3 приведено сравнение реализованного алгоритма с другими современными методами на примере базы данных CASIA ITDE V2 [4] с применением метрики F1-score.

Таблица 2. Сравнение метрик для разных типов подделок

Тип подделки	Accuracy, %	F1-score, %	Precision, %	Recall, %	IoU, %
Copy-Move	13,42	17,84	26,61	13,42	9,80
Image Splicing	67,91	77,24	89,55	67,91	62,93

Таблица 3. Сравнение с современными методами по метрике F1-score

Модель	F_1-score, %
ELA [9]	21,40
NOI1 [10]	26,30
MFCN [11]	54,10
RGB-N [12]	40,80
Реализованный алгоритм	53,11

Из проведенного сравнения результатов работы алгоритмов следует, что реализованный алгоритм обнаружения поддельных цифровых изображений демонстрирует один из лучших показателей со значением метрики F1-score = 53,11%.

Предложенный в работе алгоритм может быть использован криминалистами для обнаружения поддельных цифровых изображений при расследовании каких-либо инцидентов. Кроме того, его могут использовать рядовые пользователи в ситуациях, когда они сомневаются в достоверности цифровых изображений.

Библиографический список

1. Wang W., Xie E. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions // arXiv.org: сайт. URL: https://arxiv.org/abs/2102.12122 (дата обращения: 23.11.2023).

2. Lin T., Dollar P. Feature Pyramid Networks for Object Detection // arXiv.org: сайт. URL: https://arxiv.org/abs/1612.03144 (дата обращения: 23.11.2023).

3. Vaswani A., Shazeer N. Attention Is All You Need // Advances in neural information processing systems, 2017. – pp. 5998–6008.

4. Dong J., Wang W. CASIA Image Tampering Detection Evaluation Database // IEEE China Summit and Int. Conf. on Signal and Inf. Proc., 2013. – pp. 422–426.

5. Arias S., Duran J. Measuring Performance Metrics of Machine Learning Algorithms for Detecting and Classifying Transposable Elements // Processes, 2020. – pp. 1–19.

6. Goutte C., Gaussier E. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation // Lecture Notes in Computer Science, 2005. – pp. 345–359.

7. Rezatofighi H., Tsoi N. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression // arXiv.org: сайт. URL: https://arxiv.org/abs/1902.09630 (дата обращения: 23.11.2023).

8. OpenMMLab Semantic Segmentation Toolbox and Benchmark // arXiv.org: сайт. URL: https://github.com/open-mmlab/mmsegmentation (дата обращения: 23.11.2023).

9. Krawetz N. A Picture's Worth... Digital Image Analysis and Forensics Version 2 // Black Hat Briefings, 2007. – pp. 1–31.

10. Mahdian B., Saic S. Using noise inconsistencies for blind image forensics // Image and Vision Computing, 2009.

11. Salloum R., Ren Y. Image Splicing Localization Using a Multi-Task Fully Convolutional Network (MFCN) // arXiv.org: сайт. URL: https://arxiv.org/abs/1709.02016 (дата обращения: 23.11.2023).

12. Zhou P., Han X. Learning Rich Features for Image Manipulation Detection // arXiv.org: сайт. URL: https://arxiv.org/abs/1805.04953 (дата обращения: 23.11.2023).