Инженерный вестник Дона, №12 (2023)
ivdon.ru/ru/magazine/archive/n12y2023/8949

Анализ нейронных сетей U-Net-Attention и SegGPT в задаче сегментации трещин на изображениях дорожного покрытия

М. Е. Яхимович[1], Х.А. Джабраилов[2], Р.А. Гематудинов[1], В.И. Марсов[1], Е.В. Марсова[1].

[1] *Московский автомобильно-дорожный государственный технический университет*
[2]*Московский Технический Университет Связи и Информатики*

Аннотация: В данной работе исследуются и сравниваются две нейронные сети - U-Net-Attention и SegGPT, использующие разные механизмы внимания, для поиска взаимосвязей между различными частями входных и выходных данных. Архитектура U-Net-Attention представляет собой нейросеть U-Net с дополнительным слоем внимания, данная нейросеть предназначена для сегментации изображений. Она имеет кодер и декодер, объединенные связями между слоями и связями, пропускающими скрытые слои, что позволяет передавать информацию о локальных свойствах карты признаков. Для улучшения качества сегментации в оригинальную архитектуру U-Net включен слой механизма внимания, который помогает усилить поиск интересующих нас признаков изображения. Модель SegGPT основана на архитектуре Visual Transformers и также использует механизм внимания. Обе модели обладают способностью фокусироваться на важных аспектах изображения и могут быть эффективными при решении различных задач. В работе производится сравнение их работы на примере сегментации трещин на изображениях дорожного полотна, для дальнейшей классификации состояния дорожного покрытия в целом. Таже произведен анализ и выводы о возможности использования архитектур Transformers для решения широкого спектра задач.

Ключевые слова: Машинное обучение, нейронные сети Transformer, U-Net-Attention, SegGPT, анализ состояния дорожного полотна, компьютерное зрение.

Введение

Сегодня все шире применяются нейронные сети для решения различных практических задач. Больших успехов достигли нейронные сети, которые используют архитектуру Transformers. На основе данной архитектуры были созданы так называемые большие языковые модели

(LLM), которые могут содержать десятки или сотни миллиардов параметров. Примером таких моделей могут выступать: BERT с 330 миллионами параметров, GPT-2 с 1,5 миллиарда параметров. Большие языковые модели (LLM) используют принципы и техники архитектуры Transformers для обработки различных типов данных. Суть данной архитектуры заключается в наличии механизма внимания, а также в том, что данная архитектура не требует наличия реккурентных или сверточных архитектур в своем составе. Механизм внимания — это ключевая концепция, позволяющая модели фокусироваться на определенных частях входных данных .

Одной из главных особенностей моделей, построенных на архитектуре Transformers, является их способность быть более эффективными и универсальными при решении сложных задач.

Особенность обучения и архитектуры нейросети Transformers позволяет понимать контекст и создавать связные предложения. Сегодня модели становятся все более многофункциональными, они успешно работают с различными типами данных, такими, как изображения и звуковые файлы. Кроме того, проводятся исследования, где предпринимаются попытки интегрировать данные, полученные в процессе обучения с подкреплением, в эти модели [1].

В данной работе будет произведено исследование двух нейронный сетей, которые используют разные механизмы внимания. Это позволит сравнить их работу, а также оценить возможности использования моделей Transformers, в качестве универсального средства решения различных задач.

Постановка задачи исследования.

Важно понимать различия в эффективности решения различных задач моделями, использующими архитектуру Transformers и узкоспециализированными нейросетями [2]. В статье решаются несколько задач:

1. Определение эффективности работы узкоспециализированных моделей по сравнению с универсальными моделями на основе архитектуры Transformers;
2. Определение возможностей нейросетей U-Net-Attention и SegGPT;
3. Описание преимуществ использования Transformers для создания агентов, то есть автономных систем, способных найти решения для конкретных отраслевых задач.

В данной работе сравниваются обученная модель SegGPT, с возможностью настройки на внешние признаки за счет подсказок, а также нейросеть U-Net-Attention, обученная на подготовленных изображениях трещин дорожного полотна [3].

В данной статье будет произведено исследование и сравнение следующих нейросетей:

1. Модель U-Net-Attention;
2. Модель SegGPT;

Для модели SegGPT, как было отмечено выше, используется предобученная нейросеть, которая общедоступна в хранилище моделей «Hugging Face». Основой для сравнения работы двух архитектур будут выступать дефекты в виде трещин, выявленные на фотографиях дорожного полотна, пример показан ниже на рисунке 1.

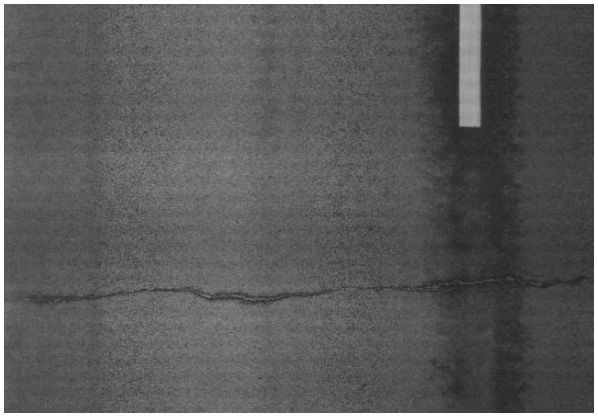

Рис. 1. – Пример дефекта в виде трещины.

Решение задачи.

Для решения поставленных задач необходимо сравнение моделей нейросетей U-Net-Attention и SegGPT с описанием механизмов внимания в них. В основе модели U-Net-Attention сверточный автоэнкодер с дополнительными связями между слоями, который используется для сегментации изображений, также для повышения качества сегментации добавлен дополнительный сверточный слой, реализующий механизм внимания[4]. Вторая модель представляет собой Transformers на основе модели VIT, данная модель именуется SegGPT [5, 3].

Причины, по которым были выбраны эти модели, следующие:

1.Обе модели обладают механизмом внимания, данный механизм позволяет нейросети сосредотачиваться на отдельных участках изображения, позволяя извлекать важную информацию и выделять в ней интересующие нас фрагменты.

2. Данные модели обладают достаточно большим входным окном, что позволяет захватывать больше признаков на изображении и находить заданные объекты.

Отличительная особенность модели U-Net заключается в захвате множества признаков высокого уровня на изображениях с одновременным сохранением локальной информации [6] .

Архитектура U-Net состоит из кодера и декодера, объединенных посредством связей между слоями. Другим важным элементом данной архитектуры является наличие связей, пропускающих скрытые слои между кодером и декодером, что позволяет передавать информацию о локальных свойствах карты признаков выходному слою [4, 7] .

Входное изображение проходит через кодировщик, который уменьшает его размер и извлекает заданные признаки. Затем декодер восстанавливает область сегментации, при этом сохраняя пространственную информацию.

Пропускающие соединения между слоями нейронной сети передают информацию о пространственных деталях изображения из кодировщика в декодер. Это позволяет сети объединять глобальную информацию и локальные детали.

Вышеуказанные особенности помогают повысить точность сегментации, особенно для маленьких объектов или тонких структур на изображении. Данная нейронная сеть имеет недостаток, связанный с потерей информации между слоями вследствие сжатия информации, для устранения этого недостатка добавляется слой внимания, который позволяет усиливать на выходе интересующие признаки.

Реализованный механизм внимания между кодером и декодером, позволяет нейросети произвести фокусировку на более сильных признаках изображения. Блок кода, отвечающего за внимание, показан ниже на рисунке 2.

```
def attention_gate(x, gate, n_intermediate_filters):

    x_shape=K.int_shape(x)
    Gate_shape=K.int_shape(gate)

    x_conv = Conv2D(n_intermediate_filters, (1,1), strides=(1,1), padding="same" )(x)
    gate_conv = Conv2D(n_intermediate_filters,  (1,1),  padding="same" )(gate)
    x_conv_shape=K.int_shape(x_conv)
    gate_conv_shape=K.int_shape(gate_conv)

    f = Activation("relu")(add([x_conv, gate_conv]))
    g = Conv2D(filters=1, kernel_size=1, strides=1, padding="same")(f)
    h = Activation("sigmoid")(g)
    h_shape=K.int_shape(h)

    upsample_psi=UpSampling2D(size=(x_shape[1]//h_shape[1],x_shape[2]//h_shape[2]))(h)

    return multiply([x, upsample_psi])
```

Рис. 2. – Пример кода врат внимания U-Net-Attention.

На вход нейросети U-Net-Attention подается изображение в формате двумерного массива. U-Net-Attention разработана для решения задач, таких, как сегментация объектов или областей на изображении.

На рисунке 3 ниже представлен результат сегментации и ручная разметка.

Рис. 3. – Автоматической сегментации (слева) и ручной разметки (справа).

Обучение U-Net-Attention производилось на изображениях трещин в различных объектах в асфальте, бетоне. Обучение осуществлялось на 3440 тыс. изображений из общего набора данных [rb.gy/gj9r8r] состоящего из 11.200 фотографий. Часть набора данных исключалась, как неподходящая для целей сегментации из-за низкого качества или слишком малых областей с дефектами. Для обучения использовались изображения из следующих наборов данных: CrackTree200 [8,9], Volker [10,8], Rissbilder [10,8], EugenMuller [11,8], SylvieChambon [clck.ru/37JN8S], CrackForest [12], CRACK500 [8,13,14], GAPS384 [8,15], DeepCrack [16]. Ниже в таблице 1 представлено процентное соотношение изображений из перечисленных выше наборов данных.

Таблица №1

Соотношение изображений для обучения из различных наборов данных.

Наименование набора данных	Количество изображений	Процент
CrackForest	118	3%

CRACK500	981	29%
GAPs384	152	4%
cracktree200	56	2%
DeepCrack	156	5%
Eugen_Muller	55	2%
Rissbilder	1146	33%
Sylvie_Chambon	56	2%
Volker	297	9%
Отрицательные примеры	423	12%
	3440	100%

Для повышения эффективности обучения, в функцию ошибки была добавлена функция потерь Дайса.

Формула 1 ниже показывает кросс-энтропию с коэффициент Дайса:

$$L_{bce}(y,\hat{y}) = -\frac{1}{n}\sum(y\log(\hat{y}) + (1-y)\log(1-\hat{y})) + (1 - \frac{2\sum_i^N y_i \hat{y}_i}{\sum_i^N y_i^2 + \sum_i^N \hat{y}_i^2})$$

Формула 1. – Функции ошибки U-Net-Attention

Второй моделью, с которой производится сравнение, является модель именуемая SegGPT, она построена на основе VIT(Visual Transformers) [3, 1].

Основные элементы архитектуры VIT:

1. Входное изображение разбивается на последовательность маленьких, фиксированных и неперекрывающихся фрагментов именуемых патчем. Каждый патч рассматривается, как отдельный вход нейронной сети. Пример изображен ниже на рисунке 4.

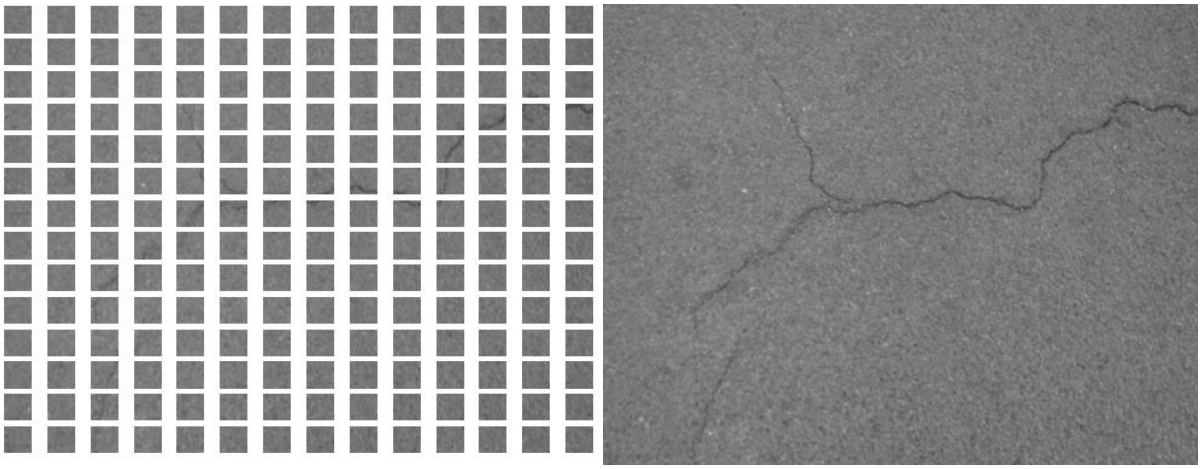

Рис. 4. – Пример входных данных для VIT и токенезации изображения.

3. Каждый патч изображения конвертируется в векторное представление.

4. К каждому вектору добавляется информация о позиции фрагмента на изображении. Это важно для сохранения пространственной информации и связи между блоками изображения.

3. Обработка векторов происходит в слое энкодера, который включает механизм внимания и полносвязные слои (соединения). Механизм внимания позволяет модели улавливать глобальные зависимости.

5. На последнем слое энкодера добавляется классификационный блок, который использует суммированные представления патчей для классификации или предсказаний.

Особенности обучения модели SegGPT заключаются в следующем:

1. При обучении происходит случайное перекрашивание пикселей, что заставляет модель реагировать на пространственные задачи отношения сегментов. Происходит фокусировка на нужных участках изображения, которые соответствует искомому запросу.

2. Объединение изображения контекста и входного изображения. Использование изображения контекста помогает найти нужные участки на нём. Изображение контекста и результата, пример показан ниже на рисунках 5 и 6.

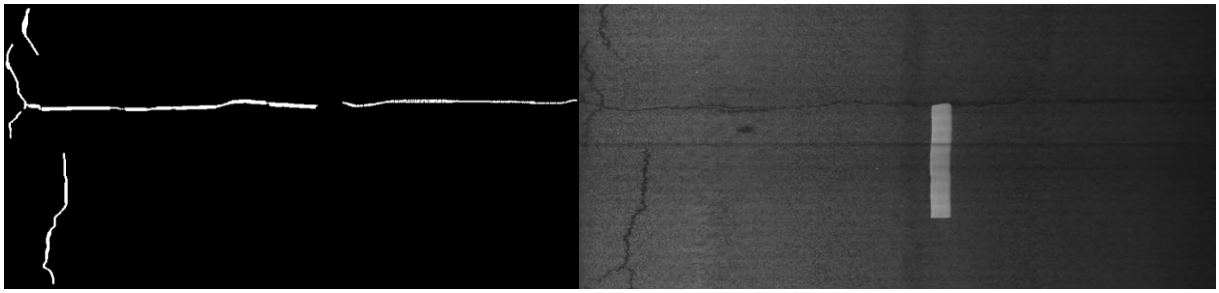

Рис. 5. – Изображение контекста и изображение-оригинал.

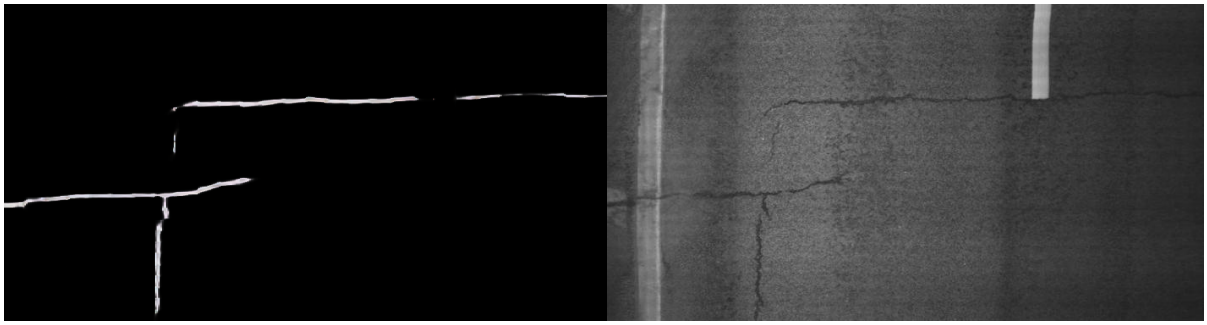

Рис. 6. – Результат сегментации и изображение-оригинал.

Результат

Для оценки качества сегментации моделей необходимо использовать различные формулы расчета точности сегментации, которые позволяют измерить и сравнить результаты сегментации. Для этой цели мы рассмотрели изображение, где приведены результаты сегментации нейросетью и ручной сегментации, как показано на рисунках выше.

В процессе сравнения эффективности использовались шесть формул расчета точности сегментации, которые дают различные результаты при оценке точности сегментации. Ниже приведены формулы этих метрик. Они включают следующие показатели: истинно положительные (TP), ложно положительные (FP), истинно отрицательные (TN) и ложно отрицательные (FN) предсказания [17][8].

Для сравнения использовались пять метрик, которые дали различные результаты. Вот три из них: индекс Жаккара, коэффициент Дайса и

пиксельная точность [17]. Ниже представлены данные формулы 2,3,4. Надо отметить, что у метрики пиксельная точность есть недостатки. Это связано с тем, что области изображений могут содержать очень маленькие сегментированные области, что может приводить к высоким процентам точности.

$$Jaccard\ Index\ (IoU) = \frac{TP}{TP + FP + FN}$$

Формула 2 – индекс Жаккара.

$$Dice = \frac{2TP}{2TP + FP + FN}$$

Формула 3 – коэффициент Дайса.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

Формула 4 – пиксельная точность.

Ниже показаны формулы 5, 6 для метрики точности и отзывчивости. Данные методики используют, когда данные несбалансированны [8].

$$Precision = \frac{TP}{TP + FP}$$

Формула 5 – точность.

$$Recall = \frac{TP}{TP + FN}$$

Формула 6 – отзывчивость.

Эти метрики позволяют измерить разные аспекты качества сегментации и помогают в оценке результатов. Важно анализировать результаты каждой метрики, чтобы получить полное представление о качестве сегментации нашей модели. Ниже показана таблица 2 с результатами

Таблица №2

Результаты сравнения работы нейросетей.

#метрики	U-Net-Attention	VIT(SegGPT)
Precision score	0,594	0,569
Recall score	0,6674	0,569
accuracy	0,9901	0,9914
Dice coef	0,588	0,537
iou	0,426	0,376

Как видно из таблицы 2, наиболее высокий результат получила нейросеть U-Net-Attention, так как она была обучена на изображениях дефектов дорожного полотна.

Выводы

Сравнение работы двух нейросетей с разными механизмами внимания показало, что модель, обученная для узкоспециализированных задач, является более эффективной. U-Net-Attention превзошел SegGPT по результатам сравнения с использованием различных метрик. Но полученный результат зависит от конкретной задачи и не является критичным. SegGPT, в свою очередь, обладает большей универсальностью в решении различных задач в области компьютерного зрения, что делает его перспективным для исследований в этой области.

Также стоит отметить, что проделанная работа с использованием механизма подсказок позволила в данной модели решить задачу автоматизации обработки изображений асфальтобетонного покрытия,

полученных автомобильно-дорожным сканером для оценки состояния дорожного полотна.

Литература(References)

1. Olson, David L.; and Delen, Dursun (2008); Advanced Data Mining Techniques. Springer, 1st edition (February 1, 2008), page 138, ISBN 3-540-76916-1

2. Dosovitskiy A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929. 2020., arXiv.URL: arxiv.org/abs/2010.11929

3. Reed S. et al. A generalist agent. arXiv preprint arXiv: 2205.06175. 2022, arXiv.URL: arxiv.org/abs/2205.06175

4. Oktay O. et al. Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999. 2018., arXiv.URL: arxiv.org/abs/1804.03999

5. Xinlong Wang and Xiaosong Zhang and Yue Cao and Wen Wang and Chunhua Shen and Tiejun Huang SegGPT: Segmenting Everything In Context 2023,arXiv.URL: arxiv.org/abs/2304.03284

6. Ronneberger O., Fischer P., Brox T. U-net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer International Publishing, 2015. pp. 234-241.

7. Y.-J. Cha, W. Choi and O. Büyüköztürk, Computer-Aided Civil and Infrastructure Engineering, vol. 32(5), pp. 361-378, May 2017.

8. Kulkarni S. et al. CrackSeg9k: a collection and benchmark for crack segmentation datasets and frameworks. European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022. pp. 179-195.

9. Zou Q. et al. Pattern Recognition Letters. 2012. Т. 33. №. 3. pp. 227-238.

10. Pak M., Kim S. Crack detection using fully convolutional network in wall-climbing robot. Advances in Computer Science and Ubiquitous Computing: CSA-CUTE 2019. Springer Singapore, 2021. pp. 267-272.

11. Ham S. et al. Journal of Korean Tunnelling and Underground Space Association. 2021. T. 23. №. 6. pp. 549-558.

12. Shi Y. et al. IEEE Transactions on Intelligent Transportation Systems. 2016. T. 17. №. 12. pp. 3434-3445.

13. Yang F. et al. IEEE Transactions on Intelligent Transportation Systems. 2019. T. 21. №. 4. pp. 1525-1535.

14. Zhang L. et al. 2016 IEEE international conference on image processing (ICIP). IEEE, 2016. pp. 3708-3712.

15. Eisenbach M. et al. How to get pavement distress detection ready for deep learning? A systematic approach. 2017 international joint conference on neural networks (IJCNN). IEEE, 2017. pp. 2039-2047.

16. Liu Y. et al. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. Neurocomputing. 2019. T. 338. pp. 139-153.

17. Vaswani A. et al. Advances in neural information processing systems. 2017. T.30.URL:proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Дата поступления: 19.11.2023

Дата публикации: 11.01.2024