Гибридная архитектура трансформера и свёрточной нейронной сети с многомасштабным механизмом деформируемого внимания в задаче семантической сегментации

Р.Р. Отырба [1], А.А. Сирота [1]
[1] Воронежский государственный университет, 394018, Воронеж, Россия, Университетская пл., д. 1

Аннотация

Предложена гибридная архитектура нейронной сети SegTwice для решения задачи семантической сегментации, которая сочетает в себе преимущества трансформеров и свёрточных нейронных сетей в рамках общей структуры кодер-декодер. Представлена оригинальная архитектура кодирующей сети TWICE-DA с иерархической структурой из четырех уровней. Вводятся и обосновываются новые архитектурные решения в блоках трансформера, имеющие отличие от известных аналогов. К ним относятся: модуль многомасштабного восприятия, модуль канального внимания, модуль деформируемого внимания и модуль свёрточной сети прямого распространения. Для задачи классификации изображений проведены эксперименты с целью оценки эффективности извлечения признаков TWICE-DA на разных по сложности наборах данных. Показано, что TWICE-DA демонстрирует высокое качество, превосходя большинство современных моделей по точности и вычислительной сложности. Осуществлена интеграция TWICE-DA в структуру сети семантической сегментации путём добавления легковесного MLP-декодера, что в итоге позволило реализовать заявленную архитектуру SegTwice. Эксперименты, проведённые на типовых аэрокосмических наборах данных LoveDA и Potsdam, показали, что предложенная сеть SegTwice демонстрирует конкурентоспособные показатели и не уступает в точности традиционным моделям и современным трансформерам, а в некоторых случаях превосходит их. Важно отметить, что SegTwice обучалась «с нуля», без предварительного обучения на больших наборах данных, что свидетельствует о её устойчивости к переобучению в условиях ограниченного объёма данных.

Ключевые слова: компьютерное зрение, семантическая сегментация, глубокие нейронные сети, свёрточные нейронные сети, трансформеры, механизм внимания.

Цитирование: **Отырба, Р.Р.** Гибридная архитектура трансформера и свёрточной нейронной сети с многомасштабным механизмом деформируемого внимания в задаче семантической сегментации / Р.Р. Отырба, А.А. Сирота // Компьютерная оптика. – 2026. – Т. 50, № 1. – 1686 – DOI: 10.18287/COJ1686.

Citation: Otyrba RR, Sirota AA. Hybrid architecture of transformer and convolutional neural network with a multi-scale deformable attention mechanism for semantic segmentation task. Computer Optics 2026; 50(1): 1686. DOI: 10.18287/COJ1686.

Введение

Семантическая сегментация (СС) – одна из наиболее сложных задач компьютерного зрения, которая заключается в разбиении изображения на сегменты с их последующей классификацией по категориям. Сегодня эта задача имеет широкий спектр приложений в различных областях, таких как робототехника, транспортные системы, медицинская диагностика, дистанционное зондирование Земли и т.д.

В настоящее время основные подходы к решению задачи СС базируются на использовании глубоких нейронных сетей, использующих общую архитектуру «кодер-декодер». В этой архитектуре кодирующая сеть извлекает признаки разной абстракции входного изображения, в то время как декодирующая сеть осуществляет постепенный процесс их декодирования и преобразует их в сегментированное изображение, в котором каждый пиксель имеет метку, указывающую на принадлежность к определенной категории.

Традиционно, начиная с Fully Convolutional Network (FCN) [1], задача СС решалась с помощью свёрточных нейронных сетей (СНС), которые способны эффективно извлекать локальные признаки изображения. Основной недостаток данного подхода заключался в неспособности сверточных сетей эффективно захватывать глобальную контекстную информацию, что часто является критически важным для понимания объектов, находящихся в различных частях сцены. Для преодоления этого ограничения предлагались различные техники увеличения рецептивного поля сети и обработки признаков в разных масштабах, как в моделях DeepLab [2 – 5] и FPN [6], разрабатывались специализированные многоветвевые архитектуры, как HRNet [7] и Res2Net [8], а также внедрялись первые механизмы внимания в сверточных слоях [9 – 11], которые повысили качество сегментации.

В свете прогресса в области обработки естественного языка и появления архитектуры трансформер [12] в последние годы наблюдается всплеск интереса к использованию другого подхода к решению задач компьютерного зрения. Он базируется на распространении предложенных в языковых трансформерах принципов механизма

самовнимания применительно к задачам анализа изображений. С момента появления первого Vision Transformer (ViT) [13] трансформеры зарекомендовали себя как мощный инструмент для решения широкого спектра задач компьютерного зрения, включая задачу СС.

Первым успешным примером применения ViT в области СС была архитектура SETR [14]. Однако, несмотря на достигнутые успехи, данный подход столкнулся с рядом ограничений, таких как высокая вычислительная сложность механизма самовнимания, недостаточная способность учитывать информацию в разных масштабах, сложность с обработкой локальных деталей и т.д. Для устранения этих недостатков активно предлагаются различные способы совершенствования ViT, включая иерархические архитектуры для обработки признаков на разных уровнях масштаба [15], а также введение новых локальных и глобальных «облегчённых» механизмов самовнимания [16 – 20]. Эти, а также ряд других приемов внесли значительный вклад в развитие моделей и методов компьютерного зрения, что позволило трансформерам занять превалирующие позиции по отношению к традиционным подходам в задачах классификации, детекции и семантической сегментации объектов.

В то же время, несмотря на значительные успехи, достигнутые в области семантической сегментации, кодирующие сети современных моделей трансформеров всё ещё имеют ряд ограничений.

1. Постоянное улучшение точности трансформеров достигается за счёт увеличения размеров моделей и роста их вычислительной сложности. Это затрудняет их использование в условиях ограниченных вычислительных возможностей.
2. Трансформеры демонстрируют снижение эффективности при работе с небольшими наборами данных. Это связано с тем, что они, как и было задумано, не обладают так называемой индуктивной предвзятостью (априорными предположениями о характере данных и их зависимостях). В отличие от СНС, которые используют локальные свёртки для обработки взаимосвязей между пикселями, трансформеры полагаются на механизм многоголового самовнимания (Multi–Head Self–Attention, MHSA), требующий значительного объёма данных для обучения. Всё это делает их склонными к переобучению в условиях малой обучающей выборки.
3. Трансформеры фокусируются на обработке признаков в пределах определённых масштабов на каждом уровне иерархии сети, но не учитывают возможную вариативность масштабов объектов в пределах одного слоя внимания. Это приводит к неспособности эффективно захватывать информативные признаки объектов различного размера. Это также снижает эффективность голов MHSA, поскольку они будут склонны выявлять схожие зависимости, что уменьшает разнообразие внимания и, в свою очередь, вызывает избыточность вычислений [21].
4. Трансформеры сталкиваются с высокими вычислительными затратами, вызванными избыточным вниманием, что снижает эффективность моделей и повышает риск переобучения. Попытки облегчить этот механизм приводят к ограничениям в его гибкости, что влечет пропуск важных ключей/значений.
5. Наконец, последние исследования показали, что свёрточные нейронные сети также могут достигать конкурентоспособных показателей эффективности в задачах обработки изображений и, в частности, в задаче СС при меньшей вычислительной сложности, о чём свидетельствуют результаты, представленные в таких работах, как ConvNeXt [22], MSCAN [23], RepLKNet [24] и SLaK [25].

Помимо этого, важным аспектом для достижения качественной СС является обеспечение эффективной передачи локальных признаков с кодирующей части сети в декодирующую. Несмотря на активные исследования в сфере кодирующих сетей, декодирующим сетям уделяется меньше внимания, что приводит к ряду ограничений в современных моделях.

Декодирующие сети в современных моделях часто основываются на свёрточных сетях. Однако они, как и сверточные кодировщики, имеют ограниченное рецептивное поле и не могут в полной мере учитывать глобальный контекст при декодировании извлечённых признаков, теряя таким образом важную информацию. Помимо этого, они проявляют недостаточную гибкость при работе с изображениями разных размеров.

В этом контексте предлагаются новые декодеры на основе трансформеров, которые, несмотря на свою эффективность, также сталкиваются с проблемами, характерными для трансформеров в кодирующих сетях. Параллельно, начиная с работы [16], активно исследуются декодеры, построенные исключительно на MLP-слоях. В отличие от предыдущих подходов, такие декодеры отличаются простотой и легковесностью, демонстрируя при этом высокую эффективность. Однако они выполняют агрегацию признаков только по каналам, не обеспечивая межпиксельного взаимодействия. Это означает, что их эффективность во многом зависит от наличия у кодирующей сети достаточно большого рецептивного поля.

Чтобы по возможности устранить вышеперечисленные недостатки и улучшить качество сегментации, мы предлагаем новую гибридную модель SegTwice для СС, которая сочетает в себе достоинства трансформеров и свёрточных нейронных сетей и, на наш взгляд, способна успешно конкурировать с известными решениями в этой области.

Таким образом, целью настоящей работы является обоснование и исследование предлагаемой гибридной архитектуры глубокой нейронной сети для решения задач семантической сегментации изображений, а также ее сравнение с известными моделями в ходе компьютерных экспериментов.

1. Предлагаемая архитектура

Предлагаемая архитектура SegTwice представляет собой гибридное решение, объединяющее трансформеры и свёрточные нейронные сети в рамках общей структуры кодер-декодер, представленной на рис. 1. В основе этого подхода лежит стремление использовать сильные стороны обоих методов: свёрточные нейронные сети – для эффективного извлечения локальных признаков, а трансформеры – для захвата глобальных зависимостей. Введение свёрточных слоёв дополнительно усиливает индуктивную предвзятость сети, что способствует повышению её эффективности и снижению риска переобучения в условиях малой обучающей выборки.

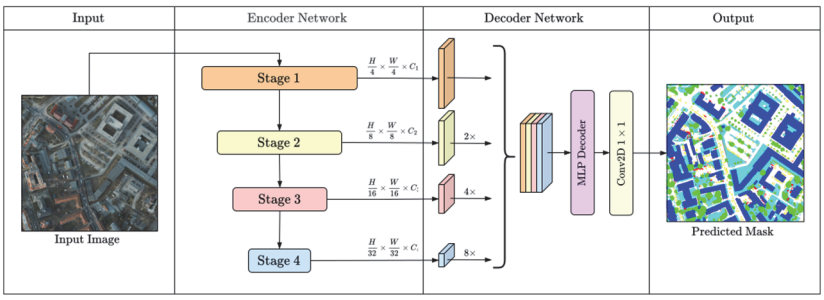

Рис. 1. Общий вид предлагаемой архитектуры SegTwice

1.1. Архитектура кодирующей сети

Прежде всего, в SegTwice предлагается реализовать новую кодирующую сеть TWICE-DA (от англ. Transformer With Integrated Multi-Scale Convolutional Extractor and Deformable Attention) с иерархической структурой из четырех уровней для извлечения и обработки признаков разной степени абстракции. Общий вид кодирующей сети представлен на рис. 2.

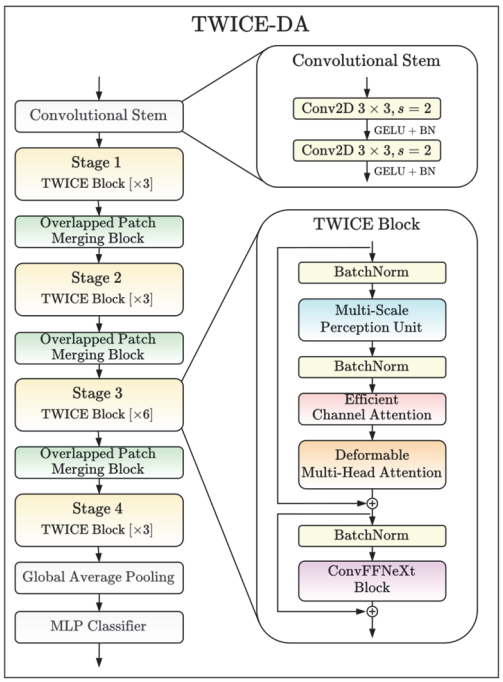

Рис. 2. Архитектура TWICE-DA

В архитектуре TWICE-DA вместо типичного извлечения патчей 7×7 или 16×16 (Patchify Stem) с использованием свёрточного слоя применяется последовательность из двух свёрточных слоёв 3×3 с шагом 2 (Convolutional Stem). Такой подход позволяет более плавно снижать размерность карт признаков, сохраняя больше важных деталей, что особенно существенно для задачи семантической сегментации.

Основной компонент архитектуры TWICE-DA – блок трансформера, который отличается несколькими ключевыми особенностями, которые реализованы в представленных далее и размещаемых последовательно внутренних модулях.

Модуль многомасштабного восприятия (Multi-Scale Perception Unit, MSPU) (рис. 3) является усовершенствованной альтернативой модуля локального восприятия (Local Perception Unit, LPU) [26], используемой в современных трансформерах. LPU обычно представляет собой разделяемую по глубине свёртку 3×3 (depthwise–свёртку) с

остаточной связью. Он размещается перед каждым механизмом самовнимания MHSA, чтобы усилить позиционную информацию о положении объектов на изображении, а также лучше учитывать мелкие детали и текстуры.

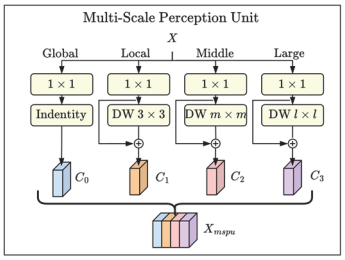

Рис. 3. Модуль многомасштабного восприятия

Предлагаемый модуль MSPU призван расширить возможности LPU, обеспечивая более гибкое и эффективное извлечение признаков при сравнительно небольшом увеличении числа параметров. Принципиальной особенностью MSPU является использование нескольких параллельных ветвей, каждая из которых отвечает за извлечение признаков на определённом масштабе.

На вход MSPU в 4 параллельные ветви поступают карты признаков $X \in \mathbb{R}^{C\times H\times W}$, где C – число каналов, а H и W– пространственные размеры карт. Первым этапом карты признаков проходят через линейные слои проекции. Эти слои осуществляют компрессию признаков по канальной размерности, уменьшая количество каналов в 4 раза:

$$\tilde{X}_i = \mathrm{Conv}^i_{1\times 1}(X), \quad i \in \{1,2,3,4\}, \tag{1}$$

где i – индекс ветви, а выходные карты $\tilde{X}_i \in \mathbb{R}^{C_r\times H\times W}$ имеют уменьшенную канальную размерность $C_r = C/4$. Это позволяет снизить объём вычислений для последующих ресурсоёмких операций, при этом сохраняя ключевые свойства признаков, а также обеспечить многогранность представлений признаков для учёта различных аспектов входных данных.

На следующем этапе каждая ветвь выполняет специализированную обработку входных карт признаков depth-wise-свёртками для извлечения информации на разных масштабах:

$$\mathrm{C}_i = \begin{cases} \tilde{X}_i, & i = 1 \\ \mathrm{DWConv}_{k_i}(\tilde{X}_i) + \tilde{X}_i, & i > 1 \end{cases}, \tag{2}$$

где k_i – размер ядра свёртки для ветви i, а остаточная связь (residual connection) $+\tilde{X}_i$ добавляет исходные признаки, предотвращая их потерю при фильтрации свёртками, а также улучшая сходимость модели.

Ветви MSPU выполняют следующие задачи:

1. *Ветвь Global* – передаёт результат линейной проекции без изменений, обеспечивая общее представление признаков на высоком уровне абстракции.
2. *Ветвь Local* – извлекает локальные признаки для анализа мелких деталей ядром 3×3.
3. *Ветвь Middle* – обрабатывает пространственные зависимости среднего уровня ядром среднего размера $m\times m$ (например, 7×7).
4. *Ветвь Large* – захватывает глобальные пространственные зависимости ядром большого размера $l\times l$ (например, 15×15).

Важно отметить, что размеры ядер для ветвей Middle и Large варьируются в зависимости от уровня иерархии сети (stage), чтобы адаптироваться к разной сложности пространственных зависимостей.

На последнем этапе результаты всех ветвей конкатенируются вдоль канальной оси, формируя многомасштабный набор признаков $X_{mspu} \in \mathbb{R}^{C\times H\times W}$:

$$X_{mspu} = \mathrm{Concat}(C_1, C_2, C_3, C_4). \tag{3}$$

Таким образом, модуль MSPU делает архитектуру более устойчивой к вариативности изображений, масштабу и положению объектов на сцене, а также способствует улучшению эффективности голов внимания MHSA, увеличивая разнообразие внимания.

Эффективный модуль канального внимания (Effective Channel Attention, ECA). MSPU создает представление данных с богатой разномасштабной информацией, однако важность каналов внутри этого представления может различаться. Мы предлагаем использовать легковесный модуль канального внимания ECA [27], чтобы динамически выделять наиболее информативные каналы, усиливая важные признаки и подавляя менее значимые (рис. 4). Достигается это путём формирования весов для каналов с использованием адаптивного усреднения для захвата глобальной пространственной информации и свёртки для межканального анализа.

Модуль сперва агрегирует пространственную информацию для каждого канала входных карт признаков $X_{mspu} \in \mathbb{R}^{C\times H\times W}$, вычисляя глобальное представление $\bar{z}$:

$$z_c = \frac{1}{H \cdot W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{mspu(c,i,j)}, \quad (4)$$

где $Z_{c,I,j}$ – значение признака на канале c в позиции (i, j), z_c – результат усреднения для канала c.

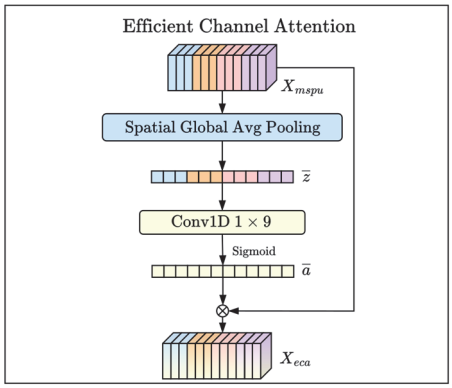

Рис. 4. Эффективный модуль канального внимания

Далее полученное глобальное представление $\bar{z}$ адаптивно взвешивается одномерной свёрткой 1×9, после чего нормализуется сигмоидальной функцией активации σ. В результате формируется вектор канального внимания $a \in \mathbb{R}^C$, выражаемый как:

$$\bar{a} = \sigma\big(\mathrm{Conv}_{1\times 9}(\bar{z})\big). \quad (5)$$

На последнем этапе вектор внимания взвешивает исходные карты признаков, что позволяет усилить важные каналы и подавить менее информативные:

$$X_{eca} = X_{mspu} \odot \bar{a}, \quad (6)$$

где $\odot$ обозначает поэлементное умножение.

В результате ECA позволяет сформировать более четко структурированную пространственно-канальную информацию, что может помочь в дальнейшем MHSA более точно сфокусироваться на важных областях изображения.

Модуль деформируемого внимания (Deformable Multi-Head Attention, DMHA), представленный на рис. 5, является усовершенствованной альтернативой стандартного модуля внимания MHSA. DMHA, вместо того чтобы учитывать все пиксели изображения при вычислении внимания, динамически фокусируется только на определённых релевантных областях изображения [28].

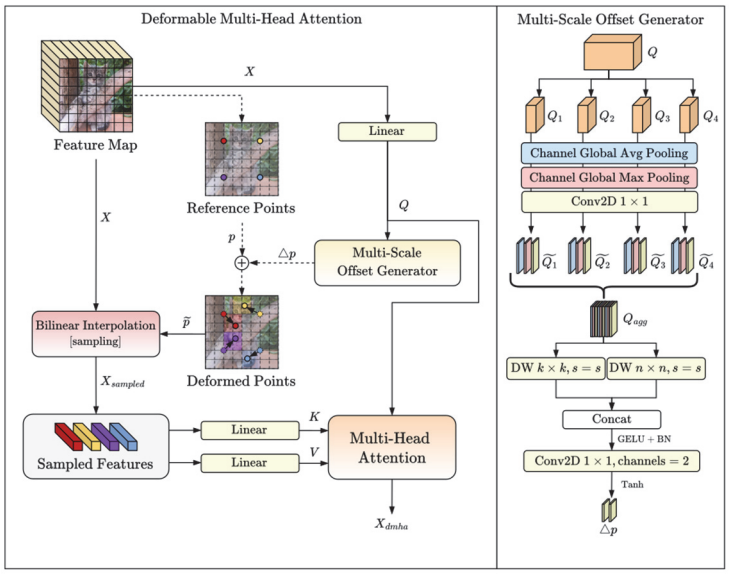

Рис. 5. Модуль деформируемого внимания

Реализация данного модуля имеет существенное значение для снижения вычислительных затрат при выполнении обработки информации в кодировщике TWICE-DA. На первом этапе DMHA формирует двумерную координатную сетку $p \in \mathbb{R}^{2\times H_G \times W_G}$, представляющую собой набор равномерно распределённых опорных точек (reference points) для входной карты признаков $X \in \mathbb{R}^{C\times H\times W}$. Каждая точка данной сетки определяется как:

$$p = \{(x, y) | x \in [0, W_G - 1], y \in [0, H_G - 1]\}, \quad (7)$$

где $H_G = H_G/r$ и $W_G = W_G/r$ – размеры сетки p, уменьшенной на коэффициент r. Это делается для сокращения вычислительных затрат и выделения только наиболее релевантных точек, которые будут задействованы в механизме внимания.

Для обеспечения стабильности и упрощения работы с координатами каждая точка полученной сетки масштабируется в диапазон [– 1, + 1] в соответствии с размерами H_G и W_G.

На втором этапе специальный генератор смещений θ_{offset}, построенный на основе последовательности свёрточных слоёв и нелинейных преобразований, использует так называемый входной запрос Q (*query*) для предсказания смещений $\Delta p \in \mathbb{R}^{2\times H_G \times W_G}$ точек сетки p, сдвигая их в более значимые области:

$$\begin{aligned} Q &= \text{Linear}(X), \\ \Delta p &= \theta_{offset}(Q), \\ \tilde{p} &= p + \Delta p. \end{aligned} \quad (8)$$

На третьем этапе формируются ключи K (*key*) и значения V (*value*) для механизма внимания (9). Этот процесс основан на выборке ключевых признаков из входной карты признаков X в соответствии с координатами, заданными сеткой опорных точек $\tilde{p}$. Для выборки используется билинейная интерполяция, которая обеспечивает плавную и непрерывную зависимость от координат. Такой подход делает процесс выборки дифференцируемым и позволяет эффективно обучать модель через обратное распространение ошибки.

Процесс выборки выражается как:

$$\begin{aligned} &\phi\big(z; (x, y)\big) = \sum_{(r_x, r_y)} g(x, r_x) g\big(y, r_y\big) z\big[r_y, r_x, :\big], \\ &K, V = \text{Linear}\big(\phi(X; \tilde{p})\big), \end{aligned} \quad (9)$$

где $\phi(\cdot,\cdot)$ – функция выборки признаков, использующая билинейную интерполяцию, $g(a, b) = max(0{,}1 - |a - b|)$ – билинейная весовая функция, которая определяет вклад каждой точки из окрестности (x, y) на сетке, $z\big[r_y, r_x, :\big]$ – значения карты признаков в точке $\big(r_x, r_y\big)$.

Для увеличения разнообразия K и V авторы данной работы предлагают делить каналы входного запроса Q на G групп, для каждой из которых генерируется свой набор G смещений. Такой подход схож с парадигмой многоголового внимания MHSA. Для эффективного использования вычислительных ресурсов веса генератора смещений являются общими для всех групп.

На последнем этапе происходит обычное вычисление многоголового внимания на основе полученных K и V, используя следующую формулу:

$$\widetilde{X^{(m)}} = \text{softmax}\left(\frac{Q^{(m)} K^{(m)T}}{\sqrt{d_k}}\right) V^{(m)}, \quad (10)$$

где m – индекс головы внимания, d_k – размерность, sotmax – функция софтмакс для нормализации внимания. Затем головы внимания конкатенируются и проходят через линейный слой, получая в итоге улучшенные представления исходных признаков:

$$X_{dmha} = \text{Linear}\left(\text{Concat}\big(\tilde{X}^{(0)}, \tilde{X}^{(1)}, \ldots, \tilde{X}^{(m)}\big)\right). \quad (11)$$

Мы заметили, что в оригинальном генераторе смещений есть недостаток: используется depthwise-свёртка с фиксированным размером ядра, которая обрабатывает входной запрос Q целиком. Это ограничивает способность эффективно захватывать информацию на разных уровнях детализации и увеличивает вычислительные затраты при работе с большими картами признаков. В ответ на эти ограничения мы предлагаем новый генератор смещений – Multi-Scale Offset Generator, который осуществляет многомасштабную обработку входного запроса Q для более качественного формирования смещений.

Чтобы осуществить многомасштабную обработку, предлагается предварительно агрегировать входной запрос Q, чтобы снизить вычислительные затраты и позволить модели эффективнее фокусироваться на самых ключевых признаках. Для этого Q делится ровно на 4 части по канальной размерности $Q_i \in \mathbb{R}^{4\times\frac{C}{4}\times H\times W}$, каждая из которых индивидуально обрабатывается по каналам:

$$\tilde{Q}_i = \text{Concat}\big(f(Q_i)_{avg}(Q_i)_{1\times1}(Q_i)_{max}()\big), \quad (12)$$

где $\tilde{Q}_i \in \mathbb{R}^{3\times H\times W}$ – выходной тензор с тремя каналами, f_{max} – максимальный пулинг, f_{avg} – усредняющий пулинг, $f_{1\times1}$ – точечная (pointwise) свёртка. После этого все 4 обработанные части конкатенируются, образуя единый компактный тензор $Q_{agg} \in \mathbb{R}^{12\times H\times W}$:

$$Q_{agg} = \text{Concat}\left(\tilde{Q}_1, \tilde{Q}_2, \tilde{Q}_3, \tilde{Q}_4\right). \quad (13)$$

Результирующий Q_{agg} затем обрабатывается двумя параллельными depthwise-свёртками разного масштаба k_1 и k_2. Поскольку каждая опорная точка охватывает локальную область размером $r \times r$, генератор должен иметь рецептивное поле как минимум такого же размера, чтобы корректно сформировать смещения. Полученные результаты свёрток подвергаются конкатенации, нормализации, нелинейности, в результате чего формируется многомасштабный набор признаков $Z \in \mathbb{R}^{24 \times H_G \times W_G}$.

$$\begin{aligned} Z_1 &= \text{DWConv}_{h_1}\left(Q_{agg}, r\right), \\ Z_2 &= \text{DWConv}_{h_2}\left(Q_{agg}, r\right), \\ Z &= \text{GELU}\left(\text{LN}\left(\text{Concat}(Z_1, Z_2)\right)\right), \end{aligned} \quad (14)$$

где Z_1 и Z_2 – результаты depthwise-свёрток, h_1 и h_2 – размеры ядер свёрток, r – размер шага свёртки, LN – нормализация по слоям. Для h_1 мы предлагаем использовать ядро размером, близким к размеру области $r \times r$, а для h_2 – значительно больше. Такой подход позволяет эффективно сочетать локальную и глобальную информацию при формировании смещений. Важно отметить, что размеры ядер варьируются в зависимости от уровня иерархии сети.

Последним этапом точечной свёрткой осуществляется предсказания смещений $\Delta p \in \mathbb{R}^{2 \times H_G \times W_G}$:

$$\Delta p = \text{Tanh}\left(\text{Conv}_{1\times1}(Z)\right), \quad (15)$$

где Tanh – функция гиперболического тангенса для нормализации значений в пределах [– 1, + 1].

Таким образом, предложенный генератор обеспечивает более качественное формирование смещений за счет эффективной предварительной агрегации и многомасштабной обработки признаков.

Модуль свёрточной сети прямого распространения (Convolutional Feed-Forward Network, ConvFFNeXt) является завершающим компонентом в блоке TWICE. В трансформерах обработанная информация механизмом самовнимания обычно поступает в FFN блок, состоящий из двух полносвязных слоёв с нелинейной функцией активации. Задача FFN – обеспечение канального взаимодействия признаков, улучшая их информативность и выраженность, а также внесение большей нелинейности в модель.

Обычно трансформеры [16, 23, 26, 28] для обеспечения более качественного канального взаимодействия признаков и позиционного кодирования применяют технику «раздувания» (expansion) промежуточного представления в 4 раза, а также добавляют между полносвязными слоями в FFN блоке depthwise-свёртку 3×3, формируя таким образом инвертированный остаточный блок (Inverted Residual Block).

$$\begin{aligned} X &= \text{Linear}\left(\text{GELU}\left(\text{DWC}\left(\text{Linear}(X)\right)\right)\right), \\ \text{DWC}(X) &= \text{DWConv}_{3\times3}(X) + X. \end{aligned} \quad (16)$$

Однако такие преобразования существенно увеличивают вычислительную сложность модели. Мы предлагаем более облегчённый вариант FFN – модуль ConvFFNeXt, в котором depthwise-свёртка размещается перед слоями FFN, формируя более облегчённый блок ConvNeXt [22].

$$\begin{aligned} X_{fnn} &= \text{FFN}\left(\text{DWC}(X)\right) + X, \\ \text{DWC}(X) &= \text{BN}(\text{DWConv}_{3\times3}(X) + X), \\ FFN(X) &= \text{Conv}_{1\times1}\left(\text{GELU}\left(\text{Conv}_{1\times1}(X)\right)\right), \end{aligned} \quad (17)$$

где BN – слой батч–нормализации.

Коэффициент «раздувания» (expansion ratio, e.r.) уменьшен с 4 до 2, что позволяет существенно снизить вычислительную сложность модели при минимальной потере точности. Кроме того, в отличие от традиционного FFN-блока, мы применяем свёртки 1×1 для обеспечения более компактного и эффективного канального взаимодействия. Таким образом, ConvFFNet снижает количество параметров и ускоряет обучение, сохраняя при этом эффективность модели.

Батч-нормализация вместо нормализации по слоям. Одним из важнейших компонентов современных архитектур глубоких нейронных сетей является слой нормализации, который способствует более стабильному и быстрому обучению, обеспечивая центрирование и масштабирование активаций.

Следуя работе [29], вместо нормализации по слоям (LN), которую традиционно используют современные трансформеры для обработки последовательных данных, мы предлагаем использовать батч-нормализацию (BN), которая эффективно применяется в архитектурах компьютерного зрения, таких как CHC. В отличие от LN, которая нормализует активации на уровне каждого отдельного примера, BN оценивает статистику по всему батчу, что особенно эффективно при обработке изображений, где статистика по батчу обычно более стабильна. Всё это позволяет сети ускорить обработку данных, быстрее адаптироваться к изменениям в распределении данных, а также улучшить регуляризацию и сходимость. Использование BN в архитектуре TWICE-DA, как показали эксперименты, позволило ускорить работу модели на 15 %, а также значительно улучшить её сходимость и точность.

1.2. Архитектура декодирующей сети

В SegTwice мы реализуем легковесный декодер на основе MLP, который агрегирует признаки 4 уровней иерархии (stages), получаемые из TWICE-DA (рис. 1). Декодер содержит всего 0,5M параметров, что существенно снижает вычислительные затраты по сравнению с более сложными архитектурными решениями.

Сначала выходные карты признаков X_1, X_2, X_3, X_4 кодирующей сети TWICE-DA объединяются по канальной размерности и агрегируются. Для этого они предварительно приводятся к единой пространственной размерности X_1 $(\frac{H}{4} \times \frac{W}{4})$ с помощью билинейной интерполяции. Затем применяется MLP-слой в виде точечной свёртки для агрегации и сжатия признаков до 512 каналов:

$$X_i^{'} = \text{Upsample}\left(X_i, size = \left(\frac{H}{4} \times \frac{W}{4}\right)\right),$$
$$X_{agg} = \text{Conv}_{1\times1}\left(\text{Concat}(X_1, X_2^{'}, X_3^{'}, X_4^{'})\right), \quad (18)$$

где $i \in \{2,3,4\}$ – индекс уровня кодера, $X_i^{'}$ – увеличенные карты признаков уровня i, приведённые к единой размерности, X_{agg} – агрегированные признаки.

На последнем этапе признаки подаются в обычный свёрточный классификатор, который предсказывает сегментационную маску $M \in \mathbb{R}^{cls\times\frac{H}{4}\times\frac{W}{4}}$.

$$M = \text{Conv}_{1\times1}(X_{agg}). \quad (19)$$

Можно отметить, что реализованный декодер выполняет агрегацию признаков только по каналам и не обеспечивает межпиксельное взаимодействие. Однако использование столь простой архитектуры оправдано именно благодаря тому, что модель TWICE-DA обладает большим рецептивным полем и способна извлекать качественные разномасштабные признаки на каждом уровне иерархии. Эти признаки уже содержат важную информацию, которая позволяет декодеру эффективно работать без необходимости реализации сложных архитектурных решений. Такой подход снижает вычислительные затраты, сохраняя при этом высокую точность сегментации.

2. Сравнительные эксперименты

2.1. Детали реализации

На практике исследователи обычно представляют архитектуры в нескольких вариантах, различающихся по размеру, таких как Tiny (T), Small (S), Large (L) и другие. Однако ввиду ограничений вычислительных мощностей в данной работе мы предлагаем только одну небольшую версию – SegTwice–T.

Подробно значения гиперпараметров предлагаемой архитектуры представлены в таб. 1. Ниже приведены краткие описания каждого параметра:

– D_i: количество блоков на уровне Stage i;
– C_i: количество каналов в каждом блоке на уровне Stage i;
– R_i: коэффициент уменьшения сетки деформируемых точек на уровне Stage i;
– H_i: количество голов внимания модуля DMHA на уровне Stage i;
– G_i: количество групп деформируемых точек модуля DMHA на уровне Stage i;
– k_j: размеры ядер свёрточных слоёв модуля MSPU на уровне Stage i;
– h_j: размеры ядер свёрточных слоёв модуля Multi–Scale Offset Generator;
– E_i: коэффициент раздувания в модуле ConvFFNeXt на уровне Stage i.

Табл. 1. Детали реализации предлагаемой архитектуры

Этап	Размер выхода	Параметры	
Stage 1	$\frac{H}{4} \times \frac{W}{4}$	D = 3, C = 64, R = 8, H = 2,	G = 1, k = [3, 7, 21], h = [9, 15], E = 2.
Stage 2	$\frac{H}{8} \times \frac{W}{8}$	D = 3, C = 128, R = 4, H = 4,	G = 2, k = [3, 7, 15], h = [5, 11]. E = 2.
Stage 3	$\frac{H}{16} \times \frac{W}{16}$	D = 6, C = 256, R = 2, H = 8,	G = 4, k = [3, 7, 11], h = [3, 7], E = 2.
Stage 4	$\frac{H}{32} \times \frac{W}{32}$	D = 3, C = 512, R = 1, H = 16,	G = 8, k = [3, 5, 7], h = [3, 5], E = 2.

Стоит отметить, что SegTwice может легко масштабироваться путём увеличения количества блоков, каналов, голов внимания и других параметров, что позволит повысить её эффективность, но потребует существенно больше вычислительных ресурсов и памяти.

2.2. Классификация изображений

Прежде чем перейти к задаче сегментации, мы оптимизировали кодирующую сеть TWICE-DA в задаче классификации, чтобы лучше понять способности модели к извлечению и обработке признаков, а также оценить её конкурентоспособность в сравнении с современными архитектурами. Эксперименты проводились на наборах данных различной сложности, таких как CIFAR-100 [30] и Caltech-256 [31].

CIFAR-100 включает 50000 изображений размером 32×32, разделённых на 100 классов. Тестовый набор состоит из 10000 изображений. Исходный набор данных был разделён на две части: 45000 изображений для обучения и 5000 для валидации.

Caltech-256 содержит 30607 изображений, разделённых на 257 классов. Набор данных характеризуется высокой несбалансированностью классов, а изображения имеют различное разрешение и качество. Данные были разделён на три части: 24487 изображений для обучения и по 3060 для валидации и тестирования.

Для обучения модели был использован оптимизатор AdamW [32] с параметрами: скорость обучения (learning rate) – 0,0005, коэффициент регуляризации (weight decay) – 0,05, параметры β – (0,9, 0,98) и ε – 1e–9. Было применено расписание скорости обучения Cosine Annealing Warm Restarts [33], обновляющее скорость обучения каждые 5 эпох. В качестве функции ошибки использовалась стандартная кросс-энтропия со сглаживанием меток (label smoothing) на уровне 0,1. Из-за ограничений вычислительных ресурсов был использован размер батча 32. Для повышения эффективности обучения градиенты аккумулировались каждые 2 мини-пакета, что фактически позволило имитировать размер батча 64. Также применялся градиентный клиппинг (gradient clipping) с пороговым значением 5, чтобы избежать проблемы взрыва градиентов. Дополнительно применялись регуляризационные техники, включая Drop-Path [34], где вероятность увеличивалась линейно от 0 до 0,2 в зависимости от глубины, Dropout для ConvFFNeXt с вероятностью 0,2 и Attention Dropout с вероятностью 0,1.

Для аугментации данных использовались горизонтальное отражение и собственная модификация эффективного метода RandAugment [35], реализованная с помощью библиотеки Albumentations [36]. В ней предлагается разделить 20 различных трансформаций на несколько категорий: геометрические, освещение, дисторсия, цветовые, шум и резкость. Это необходимо для предотвращения случайного выбора трансформаций из одной категории, поскольку это может привести к потенциальному ухудшению качества данных. Кроме того, в соответствии с современными стандартами использовались методы регуляризации, такие как MixUp [37] и CutMix [38] с вероятностью 0,5. Все изображения были приведены к размеру 224×224×3 с использованием интерполяции Lanczos4.

Для оценки эффективности и конкурентоспособности TWICE-DA мы провели сравнительные эксперименты с рядом современных архитектур, включая три модели на основе сверточных нейронных сетей (EfficientNetV2-S [39], ConvNeXt-T [22], MSCAN-S [23]), три модели на основе трансформеров (MiT-B1 [16], Swin-T [17], Twins-SVT-S [18]), а также одну гибридную модель CvT-13 [19].

Исследования проводились с использованием фреймворков Pytorch, Pytorch Lighting и библиотеки timm [40]. Все модели обучались с нуля на протяжении 300 эпох с использованием GPU RTX 3060 Ti. Все эксперименты были проведены в рамках трёх независимых запусков, а полученные метрики были усреднены.

Результаты экспериментов представлены в табл. 2. В таблице для каждой модели указано количество параметров (в миллионах, M), вычислительная сложность (FLOPs), а также точность классификации на наборах данных CIFAR-100 и Caltech-256.

Табл. 2. Сравнительный анализ точности TWICE-DA

Модель	**Парам.**	**FLOPs**	**CIFAR**	**Caltech**
EfficientNetV2–S [39]	20,5M	2,8G	0,7849	0,7287
ConvNeXt–T [22]	28,0M	4,4G	0,7437	0,6444
MSCAN–S [23]	13,6M	2,6G	0,8100	0,7591
MiT–B1 [16]	13,3M	1,6G	0,7777	0,6349
Swin–T [17]	27,7M	4,3G	0,7621	0,6392
Twins–SVT–S [18]	23,7M	2,8G	0,7594	0,6313
CvT–13 [19]	19,7M	4,0G	0,7710	0,6643
TWICE-DA	13,1M	1,8G	0,8098	0,7441

Как показано в табл. 2, предлагаемая архитектура демонстрирует высокие и конкурентноспособные результаты как на наборе данных CIFAR-100, так и на более сложном наборе данных Caltech-256 TWICE-DA, обладая меньшим количеством обучаемых параметров (13,1M) и вычислительной сложностью (1,8G FLOPs), превосходит все сравниваемые модели, незначительно уступая только MSCAN-S (CIFAR-100: точность 0,8100 и Caltech-256: точность 0,7591). Стоит отметить, что модели на основе трансформеров, такие как MiT-B1, Swin-T, Twins-

SVT-S, показывают более низкие результаты на обоих наборах данных, что подтверждает сложность эффективного обучения трансформеров на относительно небольших и несбалансированных наборах данных.

Таким образом, результаты экспериментов свидетельствуют о том, что гибридная модель TWICE-DA обладает хорошей способностью к обобщению и извлечению признаков независимо от сложности набора данных, имея при этом небольшой размер и невысокую вычислительную сложность. Это делает её перспективной для применения на более сложных задачах, таких как семантическая сегментация.

2.3. Семантическая сегментация изображений

Тестирование предлагаемой архитектуры SegTwice для задачи семантической сегментации проводилось на популярных аэрокосмических наборах данных LoveDA [41] и Potsdam [42].

LoveDA содержит 5987 спутниковых снимков размером 1024×1024 из Китая, города Нанкин, Чанжоу и Ухань, охватывающих как городские, так и сельские районы. Набор данных содержит 7 классов объектов: задний фон, здания, дороги, водные объекты, бесплодные земли, лесные насаждения и сельскохозяйственные земли. Для обучения доступно 4191 изображение, оставшиеся изображения выделены для тестирования. Результаты тестирования были получены через специально выделенный сервер с соревнованием [43].

Potsdam содержит 24 изображения размером 6000×6000 из города Потсдам, Германия. Данные представлены в двух форматах: ортофотоизображения (TOP) в формате RGB и цифровая модель поверхности (DSM) с каналом ближнего инфракрасного диапазона (NIR). Для экспериментов используются только TOP-изображения, а DSM исключается. Набор данных содержит 6 классов объектов: здания, низкая растительность, деревья, автомобили, непроницаемая поверхность (асфальт, бетон, дороги) и захламлённая территория (мусор, строительные материалы, смешанные объекты). Для экспериментов используется тип с границами. Данные разделены на 24 изображения для обучения и 14 изображений для тестирования (изображения 2_13, 2_14, 3_13, 3_14, 4_13, 4_14, 4_15, 5_13, 5_14, 5_15, 6_13, 6_14, 6_15 и 7_13). Для валидации были использованы 3 снимка (изображения 4_10, 5_11, 7_11). Все снимки были предварительно разбиты на патчи размером 1024×1024 с 50 % перекрытием.

Из-за ограничений вычислительных ресурсов обучение проводилось с нуля, то есть без использования предварительно обученной кодирующей сети, поэтому настройки и параметры остались практически идентичными, за исключением следующих. В качестве функции ошибки использовалась комбинация функции Дайса и фокальной ошибки со сглаживанием меток (label smoothing) на уровне 0,1. Размер батча равен 6, дополнительно использовался режим аккумулирования градиентов каждые 2 мини-пакета. Для объективного оценивания качества сегментации нами не применялись сложные стратегии аугментации, только простейшие, такие как: случайная вырезка размером 512×512, отражение по горизонтали и вертикали, повороты на 90 градусов. В ходе экспериментов SegTwice обучалась в течение 300 эпох. Эксперименты были проведены в рамках трёх независимых запусков, а полученные метрики были усреднены.

В табл. 3 представлены результаты экспериментов на наборе данных LoveDA по метрике степень пересечения изображений IoU для каждого класса, а также усреднённое значение mIoU. Цифры классов LoveDA имеют следующую расшифровку: 1 – задний фон, 2 – здания, 3 – дороги, 4 – водные объекты, 5 – бесплодные земли, 6 – лесные насаждения и 7 – сельскохозяйственные земли. На рис. 7 представлена визуализация предсказаний SegTwice на тестовых примерах набора данных LoveDA.

В табл. 4 представлены результаты экспериментов на наборе данных Potsdam. Метрика F1-Score традиционно используется для оценки результатов на этом наборе данных, и её значения приведены для каждого из классов. Также приводятся усреднённые значения mF1 и mIoU по классам. Цифры классов Potsdam имеют следующую расшифровку: 1 – непроницаемая поверхность, 2 – здания, 3 – низкая растительность, 4 – деревья, 5 – автомобили и 6 – захламлённая территория. На рис. 8 представлена визуализация предсказаний SegTwice на тестовых примерах Potsdam.

Исходя из результатов, можно сделать вывод, что предлагаемая архитектура SegTwice демонстрирует высокие и конкурентоспособные показатели точности на обоих вышеупомянутых наборах данных.

На наборе данных LoveDA модель достигла 51,27 по метрике mIoU, что превосходит такие традиционные модели на основе свёрточных нейронных сетей, как DeepLabV3+ (47,62) и HRNet (49,79), а также такие современные архитектуры на основе трансформеров, как SegFormer (49,14) и UpperNet (Swin–T) (50,00). SegTwice лишь немного уступает таким моделям, как AerialFormer–T (52,00), UpperNet (ViT–L12×4) (52,38) и MTP (54,17), которые обладают на порядок большим количеством параметров.

На наборе данных Potsdam модель достигла 74,0 по метрике mIoU, что превосходит такие традиционные модели на основе свёрточных нейронных сетей, как DeepLabV3+ (66,8), DANet (65,3) и SCAttNet V2 (68,3). Хотя SegTwice немного уступает SegFormer, по многим метрикам она остаётся достаточно близкой к показателям этой модели. В целом, некоторые уступки по метрикам объясняются высокой сложностью данных, на которых модель достаточно трудно обучать с нуля. В наборе данных присутствует также сложный класс «захламлённая территория», для которого даже предобученные модели демонстрируют низкое качество сегментации. Многие авторы предпочитают исключать этот класс для улучшения результатов.

Важно подчеркнуть, что представленные результаты были получены без предварительного обучения TWICE-DA на ImageNet, а также с использованием простейшего MLP-декодера. Однако несмотря на это модель успешно

справляется с улавливанием как глобального, так и локального контекста, эффективно сегментируя как крупные объекты, так и мелкие детали. Мы ожидаем, что предварительное обучение и добавление более сложных архитектурных решений для декодера (например, другие стратегии взаимодействия и агрегации признаков, механизмы внимания и так далее) повысят качество сегментации.

Табл. 3. Сравнительный анализ точности SegTwice и других современных моделей на наборе данных LoveDA

Модель	**Кодер**	**Параметры**	**Классы (IoU)**							**mIoU**
			1	**2**	**3**	**4**	**5**	**6**	**7**	
DeepLabV3+ [5]	ResNet50	39,6M	43,0	50,9	52,0	74,4	10,4	44,2	58,5	47,62
HRNet [7]	HRNet-W48	75,9M	44,6	55,3	57,4	74,0	11,1	45,3	60,9	49,79
SegFormer [16]	MiT-B1	13,7M	42,2	56,4	50,7	78,5	17,2	45,2	53,8	49,14
UperNet [44]	Swin-T	60,0M	43,3	54,3	54,3	78,7	14,9	45,3	59,6	50,00
AerialFormer-T [45]	Swin-T	42,7M	45,2	57,8	56,5	79,6	19,2	46,1	59,5	52,00
UperNet [44]	ViT-L12×4	80,6M	46,2	60,6	57,3	76,9	16,1	47,5	62,2	52,38
MTP [46]	InternImage-XL	335,0M	46,8	62,6	59,0	82,3	17,5	47,6	63,4	54,17
SegTwice	**TWICE-DA**	**13.5M**	**43,3**	**56,2**	**56,8**	**79,2**	**15,2**	**44,4**	**63,6**	**51,27**

Табл. 4. Сравнительный анализ точности SegTwice и других современных моделей на наборе данных Potsdam

Модель	**Кодер**	**Параметры**	**Классы (F1–Score)**						**mF1**	**mIoU**
			1	**2**	**3**	**4**	**5**	**6**		
DeepLabV3+ [5]	ResNet50	39,6M	89,3	92,8	83,3	78,4	88,2	31,6	77,3	66,8
DANet [10]	ResNet18	12,6M	88,5	92,7	78,8	85,7	73,7	43,2	77,1	65,3
SCAttNet V2 [47]	ResNet50	26,6M	90,0	94,0	84,1	79,8	89,1	33,6	78,4	68,3
SegFormer [16]	MiT-B1	13,7M	92,9	96,4	86,9	88,1	95,2	58,9	86,4	78,0
UperNet [44]	Swin-T	60,0M	93,5	97,0	87,4	88,6	96,1	56,9	86,6	78,5
AerialFormer-T [45]	Swin-T	42,7M	93,5	96,9	87,2	89,0	95,9	62,5	87,5	79,5
SegTwice	**TWICE-DA**	**13,5M**	**91,1**	**95,6**	**85,1**	**86,0**	**91,1**	**52,2**	**83,5**	**74,0**

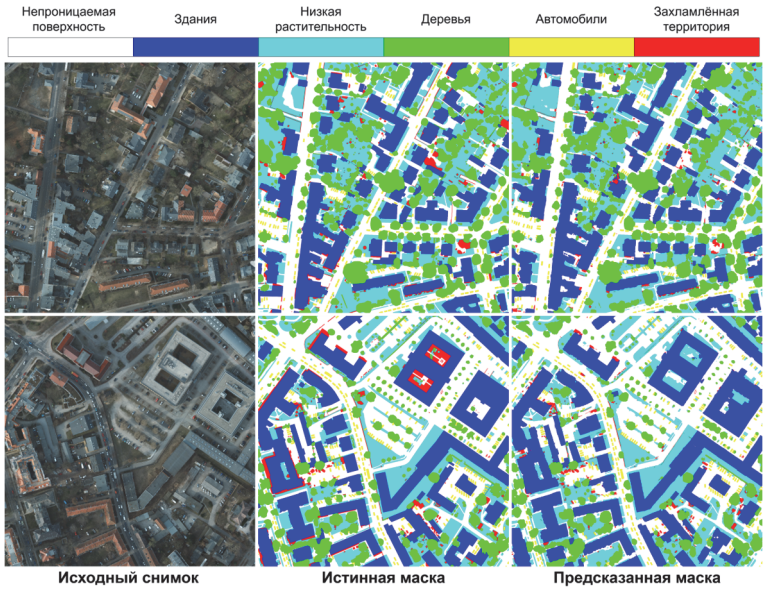

Рис. 7. Визуализация результатов предсказаний SegTwice на тестовых примерах набора данных LoveDA

Заключение

В данной работе была предложена новая гибридная архитектура SegTwice для задачи семантической сегментации, которая сочетает в себе преимущества трансформеров и свёрточных нейронных сетей.

В рамках общей модели предложена оригинальная архитектура кодирующей сети TWICE-DA. Эксперименты на задаче классификации изображений для оценки эффективности извлечения признаков предлагаемой кодирующей сети на разных по сложности наборах данных показали, что TWICE-DA демонстрирует высокое качество, превосходя большинство современных моделей по точности и вычислительной сложности. Эти результаты подтверждают её способность извлекать высококачественные признаки и успешно обобщаться даже на относительно небольших и несбалансированных данных.

Интеграция TWICE-DA для задачи семантической сегментации с добавлением легковесного MLP-декодера позволила реализовать гибридную архитектуру SegTwice. Предложенная архитектура, несмотря на меньший размер модели и отсутствие предварительного обучения, не уступает в точности традиционным моделям и современным трансформерам, а в некоторых случаях превосходит их. Это было подтверждено результатами экспериментов, проведенных на наборах данных LoveDA и Potsdam. Эксперименты показали, что SegTwice, успешно

сегментируя как крупные, так и мелкие объекты, демонстрирует конкурентоспособные показатели на обоих наборах данных.

Исходя из полученных результатов, в ходе дальнейших исследований целесообразно рассмотреть вопросы оптимизации кодирующей сети для реализации разных вариантов моделей (Tiny (T), Small (S), Large (L)) с использованием в том числе предварительного обучения на крупных наборах данных например, ImageNet-1K. Важным направлением, по нашему мнению, станет разработка более сложных архитектурных решений для декодирующей сети.

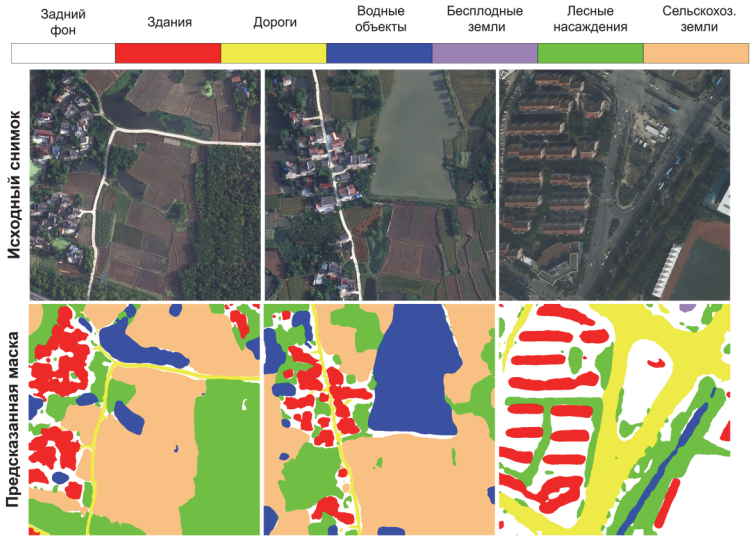

Рис. 8. Визуализация результатов предсказаний SegTwice на тестовых примерах набора данных Potsdam

References

[1] Long J, Shelhamer E, Darrell T. Fully Convolutional Networks for Semantic Segmentation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015: 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

[2] Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL, Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. arXiv Preprint. 2016. Source: <https://arxiv.org/pdf/1412.7062>.

[3] Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence 2018; 40(4): 834–848. DOI: 10.1109/TPAMI.2017.2699184.

[4] Chen LC, Papandreou G, Schroff F, Adam H. Rethinking Atrous Convolution for Semantic Image Segmentation. arXiv Preprint. 2017. Source: <https://arxiv.org/pdf/1706.05587>. DOI: 10.48550/arXiv.1706.05587.

[5] Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder–Decoder with Atrous Separable Convolution for Semantic Image Segmentation. arXiv Preprint. 2018. Source: <https://arxiv.org/pdf/1802.02611>. DOI: 10.48550/arXiv.1802.02611.

[6] Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature Pyramid Networks for Object Detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017: 936–944. DOI: 10.1109/CVPR.2017.106.

[7] Wang J, Sun K, Cheng T, Jiang B, Deng C, Zhao Y, Liu D, Mu Y, Tan M, Wang X, Liu W, Xiao B. Deep High–Resolution Representation Learning for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 2021; 43(10): 3349–3364. DOI: 10.1109/TPAMI.2020.2983686.

[8] Gao SH, Cheng MM, Zhao K, et al. Res2Net: A New Multi–Scale Backbone Architecture. IEEE Transactions on Pattern Analysis and Machine Intelligence 2021; 43(2): 652–662. DOI:10.1109/TPAMI.2019.2938758.

[9] Oktay O, Schlemper J, Folgoc LL, et al. Attention U–Net: Learning Where to Look for the Pancreas. arXiv Preprint. 2018. Source: <https://arxiv.org/pdf/1804.03999>. DOI: 10.48550/arXiv.1804.03999.

[10] Fu J, Liu J, Tian H, et al. Dual Attention Network for Scene Segmentation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019: 3141–3149. DOI: 10.1109/ CVPR.2019.00326.

[11] Huang Z, Wang X, Huang L, et al. CCNet: Criss–Cross Attention for Semantic Segmentation. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) 2019: 603–612. DOI: 10.1109/ICCV.2019.00069.

[12] Vaswani A, Shazeer NM, Parmar N, et al. Attention is all you need. arXiv Preprint. 2017. Source: <https://arxiv.org/pdf/1706.03762>. DOI: 10.48550/abs/1706.03762.

[13] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv Preprint. 2020. Source: <https://arxiv.org/pdf/2010.11929>. DOI: 10.48550/arXiv.2010.11929

[14] Zheng S, Lu J, Zhao H, et al. Rethinking Semantic Segmentation from a Sequence–to–Sequence Perspective with Transformers. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2021: 6877–6886. DOI: 10.1109/CVPR46437.2021.00681.

[15] Wang W, Xie E, Li X, et al. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) 2021: 548–558. DOI: 10.1109/ICCV48922.2021.00061.

[16] Xie E, Wang W, Yu Z, A, et al. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. arXiv Preprint. 2021. Source: <https://arxiv.org/pdf/2105.15203>.

[17] Liu Z, Lin Y, Cao Y, et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) 2021: 9992–10002. DOI: 10.1109/ICCV48922.2021.00986.
[18] Chu X, Tian Z, Wang Y, et al. Twins: Revisiting Spatial Attention Design in Vision Transformers. Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2021) 2021; 34: 9355–9366.
[19] Wu H, Xiao B, Codella N, et al. CvT: Introducing Convolutions to Vision Transformers. arXiv Preprint. 2021. Source <https://arxiv.org/pdf/2103.15808>. DOI: 10.48550/arXiv.2103.15808.
[20] Dong X, Bao J, Chen D, et al., CSWin Transformer: A General Vision Transformer Backbone with Cross–Shaped Windows. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022: 12114–12124. DOI: 10.1109/CVPR52688.2022.01181.
[21] Liu X, Peng H, Zheng N, et al. EfficientViT: Memory Efficient Vision Transformer with Cascaded Group Attention. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2023: 14420–14430. DOI: 10.1109/CVPR52729.2023.01386.
[22] Liu Z, Mao H, Wu CY, et al. A ConvNet for the 2020s. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022: 11966–11976. DOI: 10.1109/CVPR52688.2022.01167.
[23] Guo M, Lu C, Hou Q, et al. SegNeXt: Rethinking Convolutional Attention Design for Semantic Segmentation. arXiv Preprint. 2022. Source: <https://arxiv.org/pdf/2209.08575>. DOI: DOI:10.48550/arXiv.2209.08575.
[24] Ding X, Zhang X, Han J, Ding G. Scaling Up Your Kernels to 31×31: Revisiting Large Kernel Design in CNNs. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022: 11953–11965. DOI: 10.1109/CVPR52688.2022.01166.
[25] Liu S, Chen T, Chen X, et al. More ConvNets in the 2020s: Scaling up Kernels Beyond 51x51 using Sparsity. arXiv Preprint. 2022. Source: <https://arxiv.org/pdf/2207.03620>. DOI: 10.48550/arXiv.2207.03620.
[26] Guo J, Han K, Wu H, et al. CMT: Convolutional Neural Networks Meet Vision Transformers. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022: 12165–12175, DOI: 10.1109/CVPR52688.2022.01186.
[27] Wang Q, Wu B, Zhu P, et al. ECA–Net: Efficient Channel Attention for Deep Convolutional Neural Networks. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2020: 11531–11539, DOI: 10.1109/CVPR42600.2020.01155.
[28] Xia Z, Pan X, Song S, Li LE, Huang G. DAT++: Spatially Dynamic Vision Transformer with Deformable Attention. arXiv Preprint. 2023. Source: <https://arxiv.org/pdf/2309.01430>. DOI: 10.48550/arXiv.2309.01430.
[29] Yao Z, Cao Y, Lin Y, Liu Z, Zhang Z, Hu H. Leveraging Batch Normalization for Vision Transformers. 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW) 2021: 413–422. DOI: 10.1109/ICCVW54120.2021.00050.
[30] Krizhevsky A, Hinton G. Learning Multiple Layers of Features from Tiny Images. Computer Science Department University of Toronto Tech. Rep. 2009; 1(4): 7.
[31] Griffin G, Holub A, Perona P. Caltech–256 Object Category Dataset. Technical Report 7694, California Institute of Technology, 2007.
[32] Loshchilov I, Hutter F. Decoupled Weight Decay Regularization. arXiv Preprint. 2017. Source: <https://arxiv.org/pdf/1711.05101>.
[33] Loshchilov I, Hutter F. SGDR: Stochastic gradient descent with warm restarts. arXiv Preprint. 2016. Source: <https://arxiv.org/pdf/1608.03983>. DOI: 10.48550/arXiv.1608.03983.
[34] Larsson G, Maire M, Shakhnarovich G. FractalNet: Ultra–Deep Neural Networks without Residuals. arXiv Preprint. 2017. Source: <https://arxiv.org/pdf/1605.07648v4>. DOI: 10.48550/arXiv.1605.07648.
[35] Cubuk ED, Zoph B, Shlens J, Le QV. Randaugment: Practical automated data augmentation with a reduced search space. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) 2020: 3008–3017. DOI: 10.1109/CVPRW50498.2020.00359.
[36] Buslaev A., Iglovikov VI, Khvedchenya E. Albumentations: Fast and Flexible Image Augmentations. Information. 2020; 11(2): 125. DOI: 10.3390/info11020125.
[37] Zhang H, Cisse M, Dauphin YN, Lopez–Paz D. mixup: Beyond Empirical Risk Minimization. arXiv Preprint. 2017. Source: <https://arxiv.org/pdf/1710.09412>. DOI: 10.48550/arXiv.1710.09412.
[38] Yun S, Han D, Chun S, et al. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) 2019: 6022–6031. DOI: 10.1109/ICCV.2019.00612.
[39] Tan M, Le QV. EfficientNetV2: Smaller Models and Faster Training. arXiv Preprint. 2021. Source: <https://arxiv.org/pdf/2104.00298>.
[40] Wightman R. PyTorch Image Models. GitHub repository. 2019. Source: <https://github.com/rwightman/pytorch–image–models>. DOI: 10.5281/zenodo.4414861.
[41] Wang JA. Remote Sensing Land–Cover Dataset for Domain Adaptation Semantic Segmentation. arXiv Preprint. 2021. Source: <https://arxiv.org/pdf/2110.08733>.
[42] 2D Semantic Labeling Contest – Potsdam. Source: <https://www.isprs.org/education/benchmarks/UrbanSemLab/2d–sem–label–potsdam.aspx>.
[43] LoveDA Semantic Segmentation. Source: <https://codalab.lisn.upsaclay.fr/competitions/421#learn_the_details–overview>.
[44] Xiao T, Liu Y, Zhou B, Jiang Y, Sun J. Unified Perceptual Parsing for Scene Understanding. arXiv Preprint. 2018. <https://arxiv.org/pdf/1807.10221>. DOI: 10.48550/arXiv.1807.10221.
[45] Yamazaki K, Hanyu T, Tran M, et al. AerialFormer: Multi–resolution Transformer for Aerial Image Segmentation. arXiv Preprint. 2023. Source: <https://arxiv.org/pdf/2306.06842>. DOI: 10.48550/arXiv.2306.06842.
[46] Wang D, Zhang J, Xu M, et al. MTP: Advancing Remote Sensing Foundation Model via Multitask Pretraining. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 2024; 17: 11632–11654. DOI: 10.1109/JSTARS.2024.3408154.
[47] Li H, Qiu K, Chen L, et al. SCAttNet: Semantic Segmentation Network With Spatial and Channel Attention Mechanism for High–Resolution Remote Sensing Images. IEEE Geoscience and Remote Sensing Letters 2021; 18(5): 905–909. DOI: 10.1109/LGRS.2020.2988294.

Сведения об авторах

Отырба Ростислав Русланович – аспирант кафедры технологий обработки и защиты информации факультета компьютерных наук Воронежского государственного университета.
E-mail: *otyrba@cs.vsu.ru* ORCID iD: https://orcid.org/0000–0002–0412–2465

Сирота Александр Анатольевич – д-р техн. наук, проф., заведующий кафедрой технологий обработки и защиты информации факультета компьютерных наук Воронежского государственного университета.
E-mail: *sir@cs.vsu.ru* ORCID iD: https://orcid.org/0000–0002–5785–8513

ГРНТИ: 28.23.15; 28.23.37
Поступила в редакцию 18 февраля 2025 г. Окончательный вариант – 05 апреля 2025 г.

Hybrid architecture of transformer and convolutional neural network with a multi–scale deformable attention mechanism for semantic segmentation task

R.R. Otyrba [1], A.A. Sirota [1]
[1] Voronezh State University, 394018, Russia, Voronezh, Universitetskaya Square 1

Abstract

A hybrid neural network architecture, SegTwice, is proposed for the semantic segmentation task. It combines the strengths of transformers and convolutional neural networks within a unified encoder–decoder framework. The original architecture of the encoding network, TWICE-DA, is presented, featuring a hierarchical structure with four levels. New architectural solutions are introduced and justified within the transformer blocks, which differ from known analogs: a multi–scale perception unit, a channel attention module, a deformable attention module, and a convolutional feedforward network module. Experiments on image classification tasks were conducted to assess the feature extraction effectiveness of TWICE-DA on datasets of varying complexity. It is shown that TWICE-DA demonstrates high quality, outperforming most modern models in terms of accuracy and computational complexity. The integration of TWICE-DA into the semantic segmentation network structure is achieved by adding a lightweight MLP decoder, ultimately realizing the SegTwice architecture. Experiments conducted on standard aerospace datasets, LoveDA and Potsdam, revealed that the proposed SegTwice network demonstrates competitive performance, matching traditional models and modern transformers in accuracy, and in some cases, outperforming them. Notably, SegTwice was trained "from scratch" without pre–training on large datasets, highlighting its resilience to overfitting in scenarios with limited data.

Keywords: computer vision, semantic segmentation, deep neural networks, convolutional neural networks, transformers, attention mechanism.

Citation: Otyrba RR, Sirota AA. Hybrid architecture of transformer and convolutional neural network with a multi–scale deformable attention mechanism for semantic segmentation task. Computer Optics 2026; 50(1): 1686. DOI: 10.18287/COJ1686.

Author's information

Rostislav Ruslanovich Otyrba – PhD student, Department of Information Security and Processing Technologies, Faculty of Computer Sciences, Voronezh State University.
E-mail: *otyrba@cs.vsu.ru* ORCID iD: https://orcid.org/0000–0002–0412–2465

Alexander Anatolyevich Sirota – DSc in Technical Sciences, Head of the Department of Information Security and Processing Technologies, Faculty of Computer Sciences, Voronezh State University.
E-mail: *sir@cs.vsu.ru* ORCID iD: https://orcid.org/0000–0002–5785–8513

Received February 18, 2025. The final version – April 05, 2025.