

УДК 004.896

Классификация тонкостенных 3D-моделей с помощью нейросетей

Куприн Андрей Дмитриевич

kuprin.ad@bmstu.ru

МГТУ им. Н. Э. Баумана, Москва, Россия

Аннотация. Работа посвящена разработке метода автоматизированной классификации тонкостенных CAD-моделей на основе модифицированной архитектуры UV-net [1]. Исследование направлено на решение задачи идентификации геометрических объектов с малым отношением толщины стенок к общим габаритам, что позволяет переходить от объемных конечно-элементных сеток к двумерным аппроксимациям и существенно сокращает вычислительную сложность задач механики, аэродинамики и термоструктурного анализа. Предложенная архитектура сочетает сверточные (CNN) и графовые (GNN) нейросети для совместной обработки параметрических сеток граней и топологии модели в формате B-Rep. Ключевыми улучшениями исходной UV-net стали: интеграция механизма внимания на основе скалярного произведения, снижение размерности скрытых признаков, замена функций активации на более современные (Swish, Mish) [5], а также реализация системы автоматического дифференцирования на C++ без использования сторонних библиотек машинного обучения. Для обучения модели был разработан метод процедурной генерации искусственного набора данных, включающего три категории объёмов (малый, средний, большой) с общим количеством до 450 тыс. моделей. Экспериментально подтверждено повышение точности классификации до 94 % при сокращении времени обучения на 50 % и потребления памяти в 4 раза по сравнению с базовой реализацией на Python/PyTorch.

Ключевые слова: машинное обучение, свёрточные нейросети, графовые нейросети, классификация, тонкостенные модели, B-Rep, UV-net, автоматическое дифференцирование, генерация данных

Classification of thin-walled 3D models using neural networks

Kuprin Andrey Dmitrievich

kuprin.ad@bmstu.ru

BMSTU, Moscow, Russia

Abstract. This work presents a method for automated classification of thin-walled CAD models based on a modified UV-net architecture. The study addresses the challenge of identifying geometric objects with a low wall-thickness-to-overall-size ratio, enabling a transition from volumetric finite element meshes to 2D approximations, which significantly reduces computational complexity in mechanics, aerodynamics, and thermo-structural analysis. The proposed architecture combines convolutional (CNN) and graph neural networks (GNN) to jointly process parametric face meshes and model topology in B-Rep format. Key enhancements over the original UV-net include: integration of a scalar self-attention mechanism, reduction of hidden feature dimensionality, replacement of activation functions with modern alternatives (Swish, Mish), and implementation of an automatic differentiation system in C++ without third-party ML libraries. A procedural method for generating artificial datasets was developed, comprising three volume categories (small, medi-

um, large) with up to 450k models. Experiments demonstrated a classification accuracy of 94%, alongside a 50% reduction in training time and a 4-fold decrease in memory consumption compared to the baseline Python/PyTorch implementation.

Keywords: machine learning, convolutional neural networks, graph neural networks, classification, thin-walled models, B-Rep, UV-net, automatic differentiation, data generation

Введение. Автоматизация инженерных процессов в современных системах компьютерного проектирования (CAD) требует решения фундаментальной задачи классификации геометрических моделей, среди которых особое значение имеет идентификация тонкостенных объектов. Тонкостенные CAD-модели характеризуются соотношением толщины стенок к общим габаритам, что позволяет переходить от объемных конечно-элементных сеток к двумерным аппроксимациям и существенно сокращает вычислительную сложность. Однако автоматизация классификации таких моделей сопряжена с серьезными вызовами: геометрическое разнообразие деталей, вариативность топологических связей, семантическая неоднозначность граничных представлений (B-Rep) и отсутствие размеченных данных достаточного объема. Архитектура UV-net, объединяющая сверточные и графовые нейросети, является перспективным решением, но ее практическое применение ограничено проблемами масштабируемости, использованием устаревших функций активации и отсутствием механизмов фокусировки на информативных геометрических паттернах [1]. Целью данной работы является разработка модифицированной архитектуры UV-net, способной обеспечить высокую точность классификации при сокращении вычислительных затрат, с последующей интеграцией в промышленные CAD-системы и конечно-элементные решатели.

Материалы и методы. В основе метода лежит модифицированная архитектура UV-net, обрабатывающая CAD-модели в формате B-Rep через два параллельных потока: сверточные сети для анализа геометрии граней и ребер, и графовые сети для анализа топологии. Геометрические признаки извлекаются из параметрических UV-сеток граней (размерностью $16 \times 16 \times 7$) и U-сеток ребер (16×6) с использованием каскадов сверточных слоев (Conv2D и Conv1D), пакетной нормализации и функций активации ReLU.

Топологические признаки обрабатываются через механизм передачи сообщений ([3]) в графе смежности:

$$h_v^{(k)} = \phi^{(k)} \left((1 + \varepsilon^{(k)}) h_v^{(k-1)} + \sum_{u \in N(v)} f_{\theta}(h_{uv}^{(k-1)}) \odot h_u^{(k-1)} \right),$$

где $h_v^{(k)}$ — признак грани v на шаге k ; ϕ — перцептрон; f — отображение из пространства ребер в пространство граней.

Для агрегации признаков внедрен механизм скалярного внимания [2]:

$$\text{Attention}(Q, K, V) = \text{SoftMax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V.$$

Для обучения модели был разработан метод генерации искусственного набора данных с параметризованными объектами (цилиндры, конусы, трубы,

крылья, шестерни) с использованием библиотеки Gmsh и OpenCASCADE. Данные предобрабатывались с построением графов смежности и UV-сеток, а загрузка ускорялась через OpenMP. Реализация на C++ с использованием ArrayFire обеспечила аппаратное ускорение и снижение потребления памяти. Обучение проводилось методом стохастического градиентного спуска с динамическим изменением скорости обучения:

$$\lambda_i = 0,005 + \lambda \cdot \left(1 - \frac{i}{E}\right),$$

где E — номер эпохи.

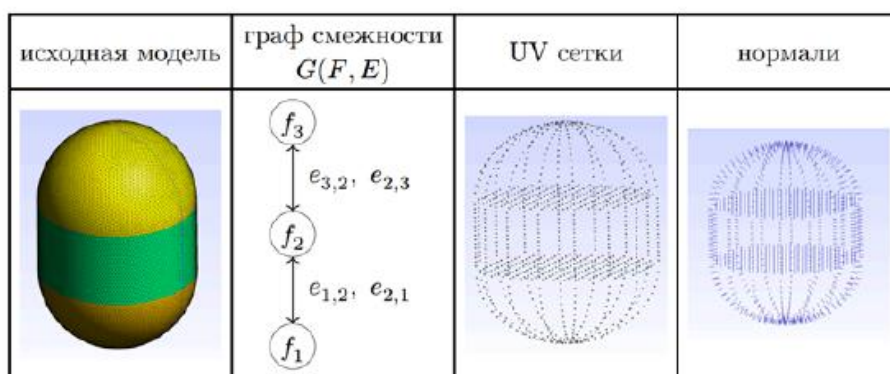


Рис. 1. Пример графа смежности и UV-сеток для модели бака

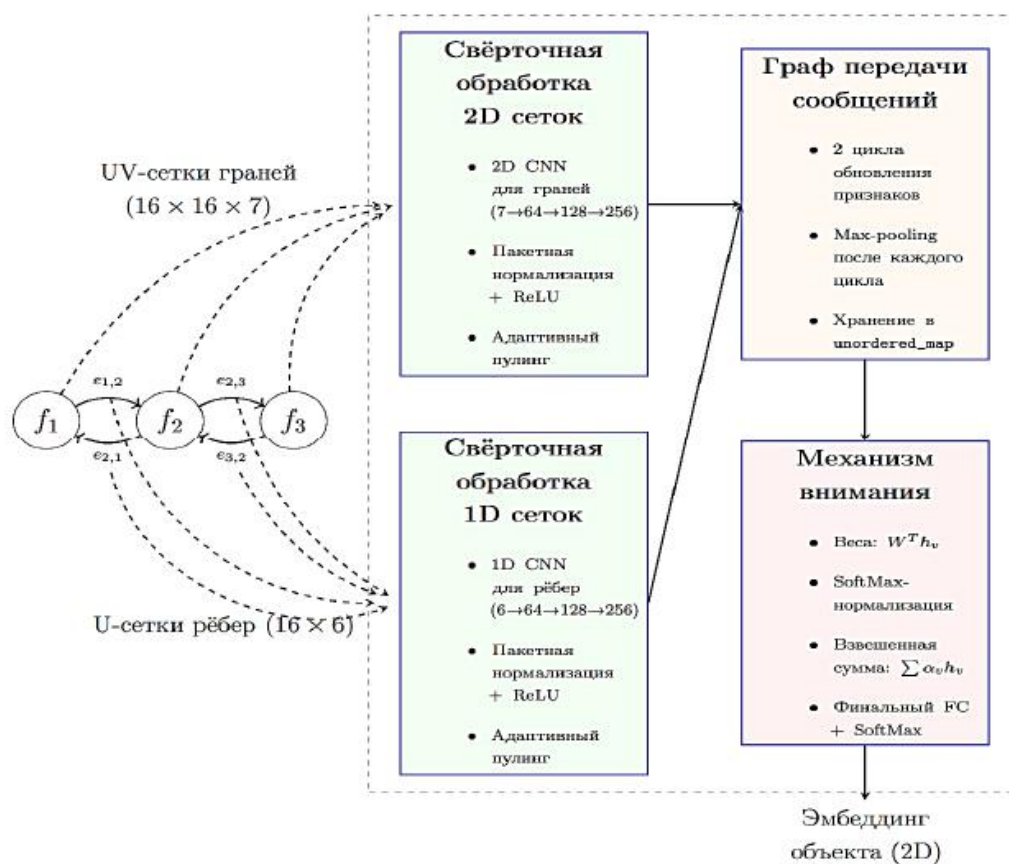


Рис. 2. Архитектура модифицированной UV-net

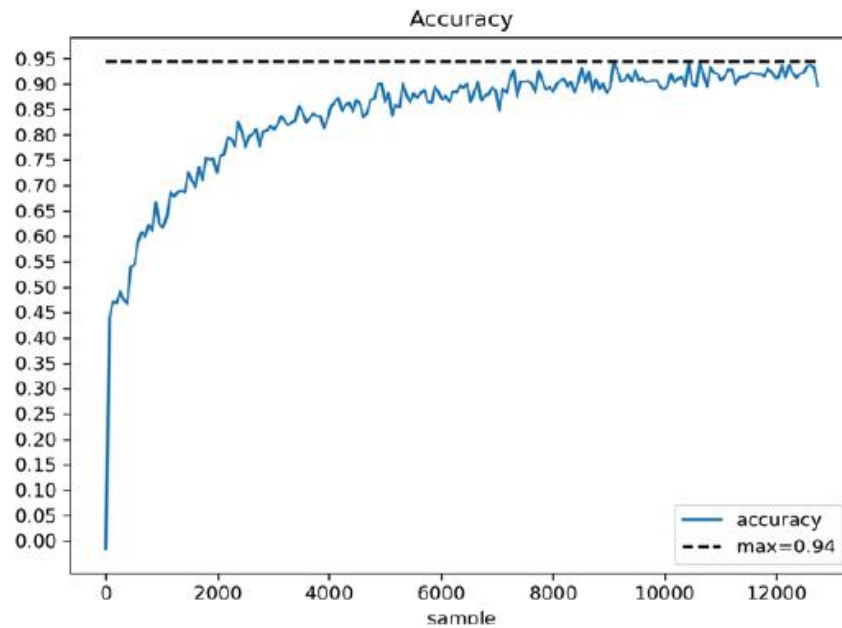


Рис. 3. Графики точности в процессе обучения

Результаты. Эксперименты проводились на трех наборах данных: малом (9k моделей), среднем (90k) и большом (450k). Модель, обученная на среднем наборе, достигла точности 99,8 % на тестовых данных аналогичного распределения и 94 % на разнородных данных, что на 6 % выше базовой реализации. На малом наборе наблюдалось переобучение (точность 96 % на своих данных, но падение до 10 % на больших данных). Реализация на C++ сократила время обучения в 2 раза и потребление памяти в 4 раза по сравнению с Python/PyTorch. Анализ матрицы ошибок показал асимметрию: точность для класса тонкостенных моделей составила 93 %, для толстостенных — 82 %, что связано с дисбалансом данных. Графики обучения демонстрируют устойчивую сходимость и рост метрик F1 и precision до 0,85 и 0,71 соответственно.

Обсуждение полученных результатов. Полученные результаты демонстрируют не только количественное улучшение метрик, но и раскрывают важные аспекты практической применимости архитектуры в реальных инженерных задачах. Наблюдаемая асимметрия в точности классификации между тонкостенными и толстостенными объектами (93 % против 82 %) указывает на фундаментальную особенность самих данных — геометрические признаки толстостенных моделей распознаются хуже [4], что требует дополнительных исследований в области выделения инвариантных признаков для данного класса объектов. Успешное применение механизма внимания подтвердило его способность выделять критические топологические связи. Стоит отметить, что реализация на C++ не только улучшила производительность, но и выявила новые возможности для оптимизации, например использование разреженных представлений для графов смежности позволило бы работать с особо крупными моделями без потери информации, что является перспективным направлением для будущих работ.

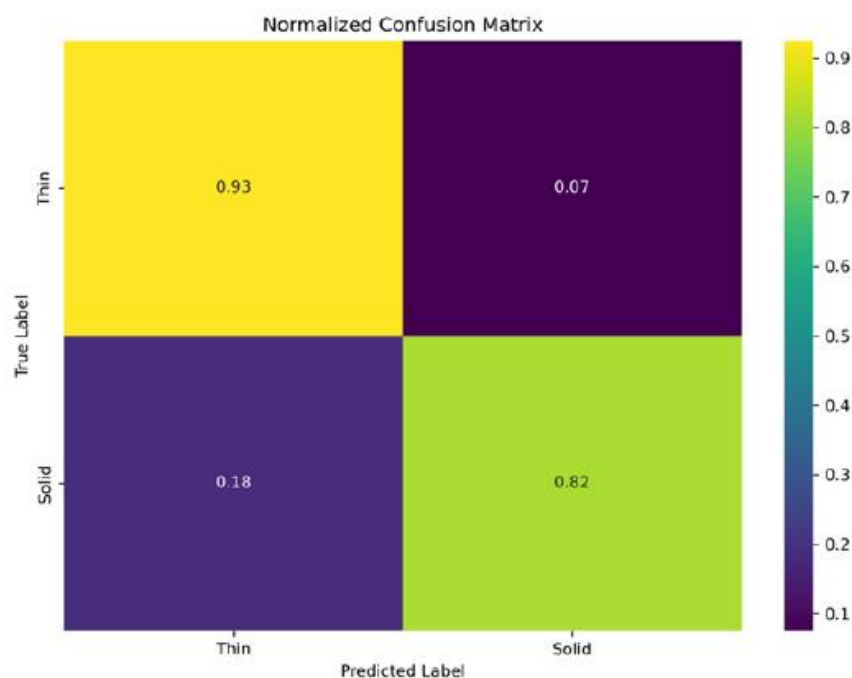


Рис. 4. Нормализованная матрица ошибок

Заключение. Разработанная модифицированная архитектура UV-net показала высокую эффективность в задаче классификации тонкостенных CAD-моделей, превзойдя исходную версию по точности, скорости и потреблению ресурсов. Ключевыми факторами успеха стали: интеграция механизма внимания, оптимизация размерности признаков, использование современных функций активации и низкоуровневая реализация на C++. Перспективы работы включают расширение онтологии классов (например, добавление класса «частично тонкостенные модели»), улучшение балансировки данных, интеграцию семантических признаков материалов и внедрение методов обработки реальных промышленных моделей. Предложенный подход открывает возможности для автоматизации выбора генераторов сеток в CAD-системах и конечно-элементных решателях.

СПИСОК ИСТОЧНИКОВ

- [1] Jayaraman P.K. UV-Net: Learning from Boundary Representations. *Arxiv:2006.102112021*. <https://doi.org/10.48550/arXiv.2006.10211>
- [2] Гудфеллоу Я., Бенджио И., Курвилль А. *Глубокое обучение*. Москва, ДМК, 2018, 652 с.
- [3] Sarker I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, 2021, vol. 2, art. no. 420. <https://doi.org/10.1007/s42979-021-00815-1>
- [4] Рашка С., Мирджалили В. *Python и машинное обучение*. Санкт-Петербург, ООО «Диалектика», 2020, 848 с.
- [5] Rasamoelina A.D., Adjailia F., Sincak P. A review of activation function for artificial neural network. *IEEE 18th world symposium on applied machine intelligence and informatics (SAMI)*, IEEE, 2020, pp. 281–286.