

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
им. М. В. ЛОМОНОСОВА»
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ
КАФЕДРА МАТЕМАТИЧЕСКОЙ ТЕОРИИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(ДИПЛОМНАЯ РАБОТА)
специалиста

Исследование влияния инициализации слоев бэкбона в задаче обнаружения аномалий

Выполнил студент
611 группы
Щекалев Михаил Игоревич

подпись студента

Научный руководитель:
к.ф.-м.н., с.н.с.
Мазуренко Иван Леонидович

подпись научного руководителя

Москва
2022

Содержание

| | | |
|----------|--|-----------|
| 1 | Аннотация | 2 |
| 2 | Введение | 2 |
| 3 | Одноклассовая классификация | 3 |
| 3.1 | Классические одноклассовые ядерные методы | 3 |
| 3.2 | Глубокое описание данных опрными векторами | 5 |
| 3.3 | Трансферное обучение для одноклассовой классификации | 6 |
| 3.4 | Развитие глубокого подхода к одноклассовой классификации | 6 |
| 4 | Методы сравнения эмбеддингов | 7 |
| 4.1 | SPADE | 8 |
| 4.2 | PaDiM | 8 |
| 4.3 | PatchCore | 9 |
| 5 | Реконструкционные методы | 10 |
| 5.1 | CFLOW-AD | 10 |
| 6 | Влияние инициализации слоёв бэкбона | 10 |
| 6.1 | AlexNet | 11 |
| 6.2 | VGG | 11 |
| 6.3 | ResNet | 11 |
| 6.4 | SqueezeNet | 12 |
| 6.5 | DenseNet | 12 |
| 6.6 | Inception | 13 |
| 6.7 | ShuffleNet | 13 |
| 6.8 | MobileNet | 14 |
| 6.9 | ResNeXt | 14 |
| 6.10 | SE-сети | 14 |
| 6.11 | Wide ResNet | 14 |
| 6.12 | ResNeSt | 15 |
| 6.13 | IBN-Net | 15 |
| 6.14 | MnasNet | 15 |
| 6.15 | EfficientNet | 15 |
| 6.16 | RegNet | 16 |
| 6.17 | ConvNeXt | 17 |
| 7 | Заключение | 17 |
| | Список литературы | 18 |

1 Аннотация

В этой работе исследуется влияние логики инициализации и выбора архитектуры бэкбона для решения задачи обнаружения аномалий и подзадачи локализации аномалий. Показано, какие архитектуры и приёмы инициализации слоёв бэкбона показывают наилучшее качество, быстроедействие и баланс этих характеристик, полезный для промышленных приложений, для SOTA методов в данной области.

2 Введение

Человеческий глаз без проблем может увидеть нетипичный дефект для какого-то заранее определённого гомогенного множества объектов. В области машинного обучения эта задача называется задачей обнаружения аномалий и имеет широкое применение в промышленных приложениях, связанных, в частности, с работой с фото- и видеоматериалами. Однако, в реальной жизни аномалии, как правило, происходят крайне редко, что затрудняет постоянный мониторинг и снижает качество детекции человеком. Кроме того, в определённых приложениях, это может потребовать большое количество человеческих ресурсов, например, в контроле качества на производстве или при видеонаблюдении на территории. Компьютерная автоматизация обнаружения аномалий может обеспечить непрерывный и дешёвый контроль в промышленных и бизнес-процессах, при этом не требуя постоянного вмешательства человека.

В компьютерном зрении задача обнаружения аномалий сводится к присваиванию каждому обрабатываемому машиной изображению коэффициента аномальности. Задача локализации аномалий, как подзадача обнаружения, предполагает присваивание каждому пикселю или какому-то кусочку изображения коэффициентов аномальности, из которых потом выбираются самые проблемные. Это позволяет выделить построить карту аномальности изображения и выделить самые проблемные области.

За годы исследований подходов к решению задачи обнаружения аномалий выделилось множество принципиально разных методик. Для задач компьютерного зрения наибольший интерес представляют методы, так или иначе использующие свёрточные нейронные сети для извлечения пространственной информации из входных изображений. В этой работе основной упор делается на методы, которые с помощью свёрточных нейронных сетей преобразуют изображение или его кусочек в некое внутреннее представление числовой вектор, а затем считают по полученному вектору коэффициент аномальности, после чего принимается решение о признании изображения или кусочка аномальным или нормальным. Сети или подсети, преобразующие вход в вектор называются бэкбон-ом алгоритма (от англ. backbone - позвоночник).

В этой работе для мотивации основного исследования работы в секциях 3-5 обзревается математическая составляющая задачи обнаружения аномалий и подходов к её решению. В секции 6 исследуется зависимость результатов SOTA (state of the art - выс-

ший на данный момент уровень развития) методов обнаружения аномалий от выбора и инициализации бэкбона. В качестве бэкбона были попробованы реализации основных современных архитектуры, а так же реализации самых заметных и актуальных их вариаций.

3 Одноклассовая классификация

Задачу обнаружения аномалий так же можно интерпретировать как задачу бинарной классификации, где, условно, первый класс - нормальный, а второй - аномальный. Но, как уже было сказано выше, аномалии возникают редко, из-за чего для обучения классификатора в выборке может оказаться слишком мало аномальных примеров. Кроме того, аномалии могут каждый раз принимать всё новые и новые формы, с которыми классификатор не сможет справиться. Поэтому, как правило, при обучении моделей для решения задачи обнаружения аномалий используются только не аномальные изображения. Такой подход носит название одноклассовой классификации.

Если предположить, что все нормальные изображения лежат в каком-то подпространстве многомерного пространства признаков, то задача обнаружения аномалий сводится к поиску этого подпространства, либо к поиску подпространства, разделяющего нормальные и все остальные, то есть аномальные, изображения.

Такую задачу можно решать как классическими методами машинного обучения, так и методами глубокого обучения.

Пусть все изучаемые изображения представляют собой вектора в некотором N -мерном пространстве признаков, для простоты предположим, что это \mathbb{R}^N . Рассмотрим в признаковом пространстве множество нормальных изображений x_1, \dots, x_n и предположим, что существует подпространство $X \subset \mathbb{R}^N$, такое что $x_i \in X$, $i = 1, \dots, n$.

Тогда будем считать любое изображение \hat{x} , лежащее за пределами X , аномальным. В таких условиях идеальным решением задачи одноклассовой классификации будет классификатор $C : \mathbb{R}^N \rightarrow [0, 1]$, такой что:

$$\begin{aligned} \forall x \in X \quad C(x) &= 1, \\ \forall \hat{x} \in \mathbb{R}^N \setminus X \quad C(\hat{x}) &= 0 \end{aligned}$$

Иначе говоря, при обучении на конечной выборке нормальных изображений x_1, \dots, x_n в качестве решения мы получим $C(x) = P(x \in X)$.

Для решения задачи одноклассовой классификации могут быть использованы классические методы машинного обучения, в частности методы опорных векторов.

3.1 Классические одноклассовые ядерные методы

Для описания работы ядерных методов сначала введём следующие объекты. Рассмотрим положительно определённое ядро $k : X \times X \rightarrow [0, +\infty)$, гильбертовом про-

пространстве с воспроизводящим ядром \mathcal{V} ([1]) и отображение $\phi : X \rightarrow \mathcal{V}$, такие что $k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{V}}$.

Один из самых известных классических методов называется **одноклассовый метод опорных векторов (OC-SVM)** [2]. Этот способ ищет гиперплоскость w в гильбертовом пространстве с воспроизводящим ядром \mathcal{V} , которая лучше всего разделяет отображённую в \mathcal{V} обучающую выборку нормальных изображений x_1, \dots, x_n и начало координат.

Формально эту задачу можно записать как оптимизационную:

$$\begin{aligned} \frac{1}{\alpha n} \sum_{i=1}^n \xi_i - \text{dist}_{\mathcal{V}}(0, w) + \frac{1}{2} \|w\|_{\mathcal{V}}^2 &\rightarrow \min_{w, \xi}, \\ \langle w, \phi(x_i) \rangle_{\mathcal{V}} &\geq \text{dist}_{\mathcal{V}}(0, w) - \xi_i, \quad i = 1, \dots, n, \\ \xi_i &\geq 0, \quad i = 1, \dots, n \end{aligned}$$

Здесь ξ_1, \dots, ξ_n - набор штрафующих переменных, позволяющих сгладить границу, $\alpha \in (0, 1]$ - гиперпараметр, балансирующий жёсткость искомой границы, а последнее слагаемое - L^2 -регуляризация.

Отделение данных от начала координат в гильбертовом пространстве с воспроизводящим ядром равнозначно нахождению полупространства, где лежит большинство нормальных данных. Таким образом, данные, лежащие за пределами этого полупространства, то есть, $\langle w, \phi(\hat{x}) \rangle_{\mathcal{V}} < \text{dist}_{\mathcal{V}}(0, w)$, являются аномалиями. Величина $s(x) = \langle w, \phi(x) \rangle_{\mathcal{V}}$ объявляется коэффициентом аномальности, изображение объявляется аномальным, если $s(x) > \tau$, где $\tau > 0$ - некий пороговый гипер-параметр.

Метод **SVDD** [3] похож на OC-SVM, только здесь для отделения нормальных данных вместо гиперплоскости ищется наименьшая гиперсфера в \mathcal{V} с центром $C \in \mathcal{V}$ и радиусом $r > 0$, такая что она содержит образы обучающей выборки нормальных изображений x_1, \dots, x_n . Оптимизационная задача для SVDD формулируется следующим образом:

$$\begin{aligned} \frac{1}{\alpha n} \sum_{i=1}^n \xi_i + r^2 &\rightarrow \min_{r, C, \xi}, \\ \|\phi(x_i) - C\|_{\mathcal{V}}^2 &\leq r^2 + \xi_i, \quad i = 1, \dots, n, \\ \xi_i &\geq 0, \quad i = 1, \dots, n \end{aligned}$$

Аналогично OC-SVM, здесь ξ_1, \dots, ξ_n - набор штрафующих переменных, $\alpha \in (0, 1]$ - гиперпараметр, контролирующий баланс между объёмом гиперсферы и штрафующими ξ_i .

Входные точки, выпадающие из сферы, то есть, $\|\phi(\hat{x}) - C\|_{\mathcal{V}}^2 > r^2$, являются аномалиями. Величина $s(x) = \|\phi(x) - C\|_{\mathcal{V}}^2$ объявляется коэффициентом аномальности, изображение объявляется аномальным, если $s(x) > \tau$, где $\tau > 0$ - некий пороговый гипер-параметр.

Обе поставленные в OC-SVM и SVDD оптимизационные задачи могут быть решены различными методами квадратичного программирования, такими как SMO ([4]).

Однако, множество проблем может возникать при использовании этих методов. Помимо того, что для работы этих методов требуется проводить преобразования признаков ([5]), вычислительная сложность ядерных методов растёт квадратично [6] при росте размера обучающей выборки. Кроме того, с ростом размерности входных данных, экспоненциально увеличивается вычислительная сложность. Для промышленного использования таких методов требуется применять методы снижения размерности, но тогда есть риск сильно потерять в качестве и, кроме того, дополнительные ресурсы уйдут на вычисление новых представлений сниженной размерности.

3.2 Глубокое описание данных опорными векторами

Метод глубокого описания данных опорными векторами (D-SVDD), предложенный в [7] схож с выше описанными классическими ядерными методами, однако, использует нейронные сети для отображения обучающей выборки в минимальную гиперсферу в гильбертовом пространстве с воспроизводящим ядром.

Определим, дополнительно, нейронную сеть с $L \in \mathbb{N}$ скрытых слоёв как параметрическое отображение $F_\theta : X \rightarrow \mathcal{V}$, где θ_j - веса j -го слоя, $j = 1, \dots, L$. $\theta = (\theta_1, \dots, \theta_L)$ - матрица весов нейронной сети.

Метод D-SVDD обучает нейронную сеть F_θ , подбирая параметры θ и, одновременно с этим, минимизирует объём гиперсферы, ограничивающей выборку нормальных данных, в пространстве \mathcal{V} . В виде оптимизационной задачи это можно записать как:

$$\frac{1}{n} \sum_{i=1}^n \|F_\theta(x_i) - C\|^2 + \frac{1}{\lambda} \sum_{j=1}^L \|\theta_j\|^2 \rightarrow \min_{\theta}$$

Первое слагаемое минимизирует среднее квадратичное отклонение отображённых нейронной сетью нормальных точек от центра C искомой гиперсферы. Второе слагаемое - L^2 -регуляризация с параметром $\lambda > 0$. Входные точки, выпадающие из сферы, то есть, $s(x) = \|\phi(\hat{x}) - C\|_{\mathcal{V}}^2 > r^2$, где $r > 0$ - радиус гиперсферы, являются аномалиями. Величину $s(x) \in [0, +\infty]$ можно считать коэффициентом аномальности для данного метода.

Будем называть функцию, упомянутую в задаче выше,

$$L_c = \|F_\theta(x) - C\|^2$$

центрирующей функцией потерь.

Эта оптимизационная задача может решаться стохастическим градиентным спуском и его вариациями, такими как Adam [8]. При использовании стохастического градиентного спуска вычислительная сложность растёт линейно при росте объёма обучающей выборки, что решает проблему с классическими методами, упомянутую выше.

Одна из главных проблем D-SVDD состоит в том, что оптимальное решение для указанной функции потерь может результировать в коллапс, при котором все входные

данные отображаются в одну и ту же точку, то есть $\forall x F_\theta(x) = a$, из-за чего отделимость нормальных и аномальных данных будет невозможна.

3.3 Трансферное обучение для одноклассовой классификации

Альтернативой D-SVDD стало использование предобученных представлений. Было показано ([9]), что представления, обученные на датасете ImageNet, дают хорошее качество и на других обучающих датасетах, домены которых слабо связаны с доменами ImageNet. Эта методика обобщается на любые крупные датасеты и называется трансферное обучение. Мы будем называть предобученные нейронные сети, позволяющие получить готовые внутренние представления, бэкбоном.

В [10], помимо тестирования различных бэкбонов для трансферного обучения под решение задачи одноклассовой классификации, был предложен **метод совместной оптимизации (JO)**. Во избежание получения тривиального решения $\forall x \in \mathbb{R}^N F_\theta(x) = a$ в отсутствие штрафов за ошибку было предложено обучать сеть F_θ с центрирующей функцией потерь совместно с бэкбон-классификатором, предобученным на ImageNet, с кросс-энтропией $CE(p, y) = -y \log(p) - (1 - y) \log(1 - p)$ в качестве функции потерь. Если соединить обе функции в одно целое, то получится следующая оптимизационная задача:

$$\sum_{i=1}^n CE(SMax(WF_\theta(x_i)), y_i) + \alpha \sum_{i=1}^n \|F_\theta(x_i) - c\| \rightarrow \min$$

где $SMax(x) = \frac{e^x}{\sum_{i=1}^n e^{x_i}}$ - софтмакс, W - финальный линейный слой

Y?

Стоит отметить, что JO требует большой объём обучающей выборки, а так же совместное обучение может снизить качество конкретно в задаче обнаружения аномалий, что несомненно является большим минусом в текущем контексте.

3.4 Развитие глубокого подхода к одноклассовой классификации

Для борьбы с **катастрофическим коллапсом** в [11] были предложены методы **PANDA** - pretrained Anomaly Detection Adaptation, представляющие собой 3 различных методики:

PANDA-Early - введение верхнего ограничения на число эпох в процессе обучения, например 15 на датасете CIFAR10. Экспериментально было показано, что уменьшение числа эпох при увеличении объёма датасета работает для большинства изученных датасетов.

PANDA-SES - каждые k эпох обученные модели сохраняются. Затем для каждой из сохранённых моделей $F_{\theta_t, t}$ и каждого из входных изображений \hat{x} считается величина $s(\hat{x}) = \|F_{\theta_t, t}(\hat{x}) - C\|^2$, а затем нормируется на среднее значение этой величины для всех входных изображений. Для конкретного изображения максимум полученных

величин, взятый по всем сохранённым моделям, объявляется коэффициентом аномальности. Таким образом, грубо достигается наилучшее разделение между нормальными и аномальными данными.

PANDA-EWC использует методику упругого закрепления весов (EWC, [12]), то есть, для всех весов бэкбона считается матрица Фишера, в которой диагональный элемент равен:

$$I(\theta) = E\left(\frac{\partial}{\partial \theta} CE(x, y)\right)^2$$

$$L = \|F_{W_i, t}(\theta_i) - C\|^2 + \lambda \sum_i I(\theta_i)(\theta_i - \theta_i^*)^2$$

Хотя методы D-SVDD, JO и PANDA пытаются справиться с проблемой **catastrophic collapse** с помощью различных техник, таких как: архитектурные модификации, вспомогательные задачи и т.д., они полностью не избавляют от **catastrophic collapse**.

Одним из предложенных методов борьбы стала **средне-смещённая контрастная функция потерь** ([13])

4 Методы сравнения эмбеддингов

Эти методы используют нейронные сети в качестве бэкбона для извлечения векторных представлений (эмбеддингов), которые описывают всё входное изображение, либо какую-то его часть. Роль коэффициента аномальности в подобных методах играет расстояние между объектами, построенными на основе полученного представлением входного изображения и векторов, построенных по обучающей выборке нормальных изображений.

В случае описания целого изображения одним вектором, каждое нормальное изображение x_i ассоциируется с вектором $F_\theta(x_i) \in \mathcal{V}$, где F_θ - предобученная свёрточная нейронная сеть.

Для задачи локализации векторы, построенные на целом изображении, не подойдут, поскольку интерпретация этого вектора в терминах частей изображения представляется сложной. Введём понятие патча (от англ. patch). Пусть размерность признакового пространства N раскладывается в произведение $W \times H$, где $W \in \mathbb{N}, H \in \mathbb{N}$.

Пусть наибольший свёрточный слой в F_θ имеет размерность $W \times H$, где $W \in \mathbb{N}, H \in \mathbb{N}$. Тогда любое из нормальных изображений $x_i, i = 1, \dots, n$ можно разделить на $w \times h$ непересекающихся связанных патчей одинакового размера, где $1 \leq w \leq W, 1 \leq h \leq H$, w делит W и h делит H . Каждый патч p_{jk} из образованной сетки $(j, k) \in \{1, \dots, w\} \times \{1, \dots, h\}$ может быть ассоциирован с вектором a_{jk} . Такой вектор будем называть активационным вектором для данного патча p_{jk} . При $w = W, h = H$ патч является отдельным пикселем изображения.

4.1 SPADE

В [15] был предложен **метод семантического пирамидального обнаружения аномалий (SPADE)**, в котором был использован алгоритм K ближайших соседей (k-NN) на представлениях, полученных из слоёв предобученного бэкбона F_θ . Для входного изображения x считается представление $F_\theta(x)$, к нему ищутся K ближайших представлений нормальных изображений по евклидовой метрике. Множество этих векторов обозначим через $N_K(F_\theta, x)$. Величина

$$d(x) = \frac{1}{K} \sum_{a \in N_K(F_\theta, x)} \|a - F_\theta(x)\|^2$$

объявляется коэффициентом аномальности для входного изображения x . Изображение объявляется аномальным, если $d(x) > \tau$, где $\tau > 0$ - некий пороговый гипер-параметр.

Для локализации аномалий для каждого патча (или пикселя) p_{jk} входного изображения x предлагается построить по K ближайшими к x нормальными изображениями множество G из активационных векторов, ассоциированных с соответствующими патчами в ближайших изображениях.

Тогда коэффициент аномальности $d(x, p)$ для пикселя p_{jk} во входном изображении x определяется как среднее расстояние между a_{jk} и κ ближайшими к нему векторами из множества G . Множество ближайших векторов обозначим через $N_\kappa(F_\theta, x, p)$.

$$d(x, p) = \frac{1}{\kappa} \sum_{a \in N_\kappa(F_\theta, x, p)} \|a - a_{jk}\|^2$$

Аналогично, пиксель объявляется аномальным, если $d(x, p) > \tau$, где $\tau > 0$ - некий пороговый гипер-параметр.

В оригинальной статье, эксперименты проводились на Wide ResNet-50-2 [32], предобученном на ImageNet, в качестве бэкбона. Векторные представления извлекались, из 3 первых свёрточных слоёв (56×56 , 28×28 , 14×14) бэкбона.

Из минусов SPADE следует выделить, что алгоритм k-NN требует запоминать в память всю обучающую выборку и, к тому же, его вычислительная сложность линейно зависит от размера этой выборки и экспоненциально от размерности входных данных, что может помешать в использовании SPADE в промышленных приложениях.

4.2 PaDiM

Метод **патчевого моделирования распределений (PaDiM)**, предложенный в [16] схож со SPADE в том, что тоже генерирует представления патчей для локализации аномалий. Однако, свойство нормальности в PaDiM описывается с помощью семейства гауссовских распределений.

Пусть наибольший свёрточный слой в бэкбоне F_θ имеет размерность $w_{max} \times h_{max}$. Тогда все изображения можно разбить в сетку $(j, k) \in \{1, \dots, w_{max}\} \times \{1, \dots, h_{max}\}$

По всем нормальным изображениям из обучающей выборки x_i , $i = 1, \dots, n$ считается активационный вектор, из которых затем строится множество эмбедингов для каждого из патчей p_{jk} :

$$A_{jk} = \{a_{jk}^i, i = 1, \dots, n\}, \\ (j, k) \in \{1, \dots, w_{max}\} \times \{1, \dots, h_{max}\}$$

Предположим, что множества A_{jk} сгенерированы многомерным гауссовским распределением $\mathcal{N}(\mu_{jk}, \Sigma_{jk})$, где μ_{jk} - эмпирическое среднее множества A_{jk} , а Σ_{jk} - эмпирическая ковариационная матрица, задаваемая следующей формулой

$$\Sigma_{jk} = \frac{1}{n-1} \sum_{i=1}^n (a_{jk}^i - \mu_{jk})(a_{jk}^i - \mu_{jk})^\top + \epsilon E,$$

где последнее слагаемое выполняет роль регуляризации для того, чтобы матрица Σ_{jk} была обратима. Таким образом, каждому патчу ставится в соответствие многомерное гауссовское распределение.

Для вычисления попатчевого коэффициента аномальности предлагается использовать расстояние Махаланобиса [17]:

$$d_M(a_{jk}) = \sqrt{(a_{jk} - \mu_{jk})^\top \Sigma_{jk}^{-1} (a_{jk} - \mu_{jk})},$$

Таким образом, матрица $D_M = (d_M(a_{jk}))_{1 \leq j \leq w_{max}, 1 \leq k \leq h_{max}}$ формирует карту аномальности, в которой высокие значения обозначают высокую аномальность соответствующей зоны на входном изображении.

Авторы PaDiM предложили алгоритм случайного выбора признаков для снижения вычислительной сложности и необходимой памяти. Для обнаружения аномалий этот метод оказывается более качественным и быстрым, чем метод главных компонент (PCA).

PaDiM, в отличие от SPADE не использует k-NN, что отбрасывает проблему масштабированности, о который было сказано выше, поскольку больше не требуется рассчитывать и сортировать расстояния до эталонных векторов, чтобы посчитать коэффициент аномальности.

В оригинальной статье, авторы проверили PaDiM на бэкбонах: ResNet18 [22], Wide ResNet-50-2 [32] и EfficientNet-B5 [36], все из которых были предобучены на датасете ImageNet. Векторные представления в случае ResNet извлекались, как и в SPADE, из 3 первых слоёв бэкбона. Для EfficientNet использовались 7-ой, 20-ый и 26-ой слои. На момент публикации оригинальной статьи, PaDiM показал SOTA результаты в задаче обнаружения и локализации аномалий.

4.3 PatchCore

[19]

5 Реконструкционные методы

5.1 CFLOW-AD

6 Влияние инициализации слоёв бэкбона

В этой секции представлены результаты исследования влияния инициализации слоёв бэкбона на качество и скорость работы SOTA методов сравнения эмбеддингов, описанных в предыдущем разделе.

Для оценки качества в задаче обнаружения аномалий будет использована метрика площади под ROC-кривой (ROC-AUC), равная соотношению между долей верно классифицированных как аномальные объектов от общего количества и долей ошибочно классифицированных как аномальные объектов от общего количества объектов. Для оценки качества в задаче локализации аномалий будет использована попиксельная метрика площади под ROC-кривой (pROC-AUC), равная соотношению между долей верно классифицированных как аномальные пикселей от общего количества пикселей и долей ошибочно классифицированных как аномальные пикселей от общего количества пикселей.

Всё обучение и экспериментальные оценки проводятся на конвенционально признанном датасете MVTeс AD [18], созданном специально для тестирования алгоритмов решения задачи обнаружения и локализации аномалии в задаче компьютерного зрения. Датасет содержит примерно 250 изображений, каждое из которых принадлежит одному из 15 классов. Для каждого класса, помимо нормальных изображений для обучения, авторы датасета для каждого из классов предоставили изображения с различными дефектами для тестирования качества обнаружения и локализации. 15 классов состоят из 5 текстурных: ; и 10 объектных классов: . Изображения в MVTeс AD имеют размерность от 700x700 до 1024x1024.

В качестве предобработки изображений, как и в оригинальных статьях PaDiM [16] и SPADE [15] будет использован следующий набор действий: приведение разрешения всех изображений к 256 на 256 и вырез центральной части размера 224 на 224 из полученного изображения.

Для тестирования бэкбонов под PaDiM в наших экспериментах мы будем использовать, представленный в [16], метод R_d снижения размерности через случайный выбор $d \in \mathbb{N}$ измерений, то есть, для входного вектора $a = (a^1, \dots, a^D)$, где $D > d$ из a^1, \dots, a^D выбираются случайные d координат. Полученный $R_d(a) = (a^{i_1}, \dots, a^{i_d})$ используется в PaDiM для моделирования нормального распределения.

Эксперименты проводились на базе фреймворка PyTorch для языка Python 3. Для исполнения псевдослучайного кода во всех экспериментах использовался случайный сид 1024. Все расчёты и замеры производительности производились на ЦП Intel i5 11400F-2.60 ГГц.

В каждой из последующих подразделов будут даны: краткое описание глубокой свёр-

точной нейронной сети, подсеть которой будет использована как опорная сетевая функция для PaDiM и SPADE, и результаты проведённых экспериментов. Полученные цифры с детализацией по каждому классу и изученной модели будут продемонстрированы в приложении.

6.1 AlexNet

AlexNet consists of 5 convolution layers, 3 max-pooling layers and 2 fully-connected layers. Одной из первых архитектур, достигших хороших результатов на датасете ImageNet была Alexnet [20]. Сеть состоит из 5 свёрточных слоёв, 3 слоёв подвыборки максимумом (max-pooling) и 2 полносвязных слоёв. В этом разделе будут представлены результаты экспериментов с инициализацией слоёв опорной сетевой функции всеми пятью последовательными свёрточными слоями сети Alexnet, предобученной на ImageNet. К полученному активационному вектору ($\dim a = 256$) применяется алгоритм случайного снижения размерности Rd с параметром $d = 100$. Подробные результаты экспериментов представлены в 2.

6.2 VGG

Более глубокая и тяжёлая архитектура VGG [21] активно использовала свёртки 3x3 вместо 5x5 и 7x7. Для реализации опорной сетевой функции архитектурой VGG, как и в AlexNet будет использован выход всей свёрточной части предобученной сети. Результаты экспериментов представлены в 3.

Для исследования влияния инициализации опорной сетевой функции вариациями VGG будут использованы 4 модели без батч-нормализации: 11-слойная, 13-слойная, 16-слойная и 19-слойная; а так же 4 модели с аналогичным числом слоёв с батч-нормализацией. Активационный вектор реализуется выходом всей свёрточной структуры VGG. Для всех рассматриваемых моделей размерность активационного вектора равно 512. Применяется алгоритм Rd с параметром $d = 120$. Полученные в результате экспериментов метрики качества и производительности по классам MVТес представлены в .

6.3 ResNet

Впервые представленные в [22] глубокие нейронные сети с остаточными (residual) блоками, которые помогали бороться с затуханием градиента при обучении путём добавления линейной функции (остатка) в аргумент функции активации через прокидывания связей в обход свёрток. В экспериментах из оригинальных статей [15] и [16] в качестве бэкбона использовалась только самая лёгкая из предложенных в [22] residual сетей, 18-слойная ResNet-18. В этом разделе описаны результаты экспериментов с остальными, более тяжеловесными, вариантами сетей, представленными в оригинальной [22] (ResNet-

34, ResNet-50, ResNet-101 и ResNet-152), в инициализации слоёв опорной сетевой функции.

Точно так же, как в экспериментах оригинальной статьи PaDiM, для извлечения векторных представлений, то есть, в качестве опорной сетевой функции будет использоваться конкатенация выходов первых трёх слоёв ResNet-сетей, предобученных на ImageNet. Для снижения размерности 18- и 34-слойных сетей, для которых размерность активационного вектора равен 448, использовался параметр $d = 100$, а для 50-, 101- и 152-слойных сетей с $\dim a = 1792$, $d = 550$. Эти же значения d показали наилучший результат в [16]. Развёрнутые цифры по исследуемым метрикам представлены в 4.

6.4 SqueezeNet

Маленькие свёрточные нейронные сети, такие как SqueezeNet [23], с лёгкой архитектурой выгодны для промышленного использования, так как занимают мало места на носителе, требуют минимального времени для загрузки на устройство, а так же сильно быстрее в эксплуатации. Архитектуре SqueezeNet удалось достичь результатов уровня AlexNet при этом имея размер массива параметров меньше 0.5 мегабайт, в то время как размер AlexNet составляет примерно 240 Мб. Такой эффект был достигнут с помощью замены фильтров 3×3 на 1×1 , уменьшением входных каналов в фильтры 3×3 . **еще** В ряде промышленных приложениях для задачи обнаружения аномалий, например, в области беспилотных автомобилей, где требуется быстро получать обновления с сервера на устройство, или для программируемых пользователем вентильных матриц (FPGA), размер бэкбона может иметь сильные ограничения, в связи с чем, исследование инициализации бэкбона легковесными архитектурами представляется интересным и важным с практической точки зрения. В этой работе будут рассмотрены самые передовые и популярные семейства легковесных архитектур.

В этом разделе описаны результаты экспериментов с инициализацией опорной сетевой функции слоями SqueezeNet версий 1.0 и 1.1 для алгоритма PaDiM. Поскольку дизайн архитектуры ограничивает размерность свёрточных слоёв и глубину сети, для извлечения активационных векторов использовалась вся свёрточная структура SqueezeNet с дополнительным снижением размерности путём случайного выбора признаков с параметром $d = 100$.

Результаты представлены в таблице 2.

6.5 DenseNet

Архитектура DenseNet, или компактносвязанная свёрточная нейронная сеть, была описана в [24]. Продолжая идею ResNet о том, что остаточные связи позволяют обучать более глубокие и точные модели, в DenseNet были введены компактносвязанные блоки, которые соединяют каждый слой с каждым другим слоем. Главное отличие от ResNet,

состоит в том, что перед выходы слоя передачей в следующий слой конкатенируются, а не суммируются, как в ResNet.

В этом разделе представлены результаты инициализации опорной сетевой функции с помощью конкатенации выходов первых трёх компактносвязанных блоков DenseNet 121, 169, 161, 201 для алгоритма PaDiM. Первые два вектора для операции реализуются 6-ым и 12-ым компактносвязанными слоями, соответственно. Третий вектор для 121 реализуется 24-ым слоем, для 169 – 32-ым, для 161 – 35-ым и для 201 – 48-ым. Все 4 сети были предобучены на ImageNet. Развёрнутые результаты тестов представлены в таблице 5.

6.6 Inception

До описания в [25] сетей архитектуры Inception, большинство подходов к улучшению качества свёрточных сетей заключались в добавлении слоёв, что делало сеть глубже и тяжелее. Идея Inception состоит в использовании в свёрточных слоях нескольких фильтров разной размерности, что увеличивало сеть вширь, а не вглубь.

В этом разделе представлены результаты экспериментов со слоями GoogLeNet [25] и Inception v3 [26] в роли опорной сетевой функции. Для реализации опорной функции была выбрана конкатенация выходов первых 3-ёх inception блоков. Для GoogLeNet с активационным вектором размерности 1248 использовался алгоритм случайного снижения размерности Rd с параметром $d = 550$, для Inception v3 с $\dim a = 832$ параметр d рассматривался равным 220. Развёрнутые результаты экспериментов представлены в ????

6.7 ShuffleNet

Ещё одна легковесная архитектура под названием ShuffleNet, представленная в [27], для уменьшения размера моделей использует простую операцию перемешивания каналов, позволяющую отказаться от слоёв с группированными свёртками.

В этом разделе представлены результаты экспериментов с двумя нейросетями архитектуры ShuffleNet в качестве опорной сетевой функции. Для тестов были взяты 2 реализации архитектуры с 0.5x выходными каналами и 1.0x выходными каналами. Активационный вектор строится из конкатенации выходов последних свёрточных слоёв первых 3-х блоков архитектуры ShuffleNet. Для первой модели размерность активационного вектора равна 336, для второго – 812. Для снижения размерности применялся Rd с параметром d равным 100 и 220, соответственно. Полные результаты экспериментов представлены в таблице 6.

6.8 MobileNet

Как уже было упомянуто выше, эксперименты с легковесными архитектурами представляют особый интерес для ряда индустриальных приложений, которые в том числе могут дополнительно требовать использования мобильного устройства для работы. Подобная архитектура для мобильных устройств была предложена в [28]. Особенностью архитектуры стали легковесные глубокие свёрточные блоки $1 \times 3 \rightarrow 3 \times 1$ вместо классических 3×3 и особая функция активации, аппроксимирующая сигмоиду.

В этом разделе описаны результаты экспериментов с MobileNetV2 и двумя вариантами MobileNetV3, введённых в [29], отличающимися от MobileNetV2 наличием остаточных блоков, по аналогии с архитектурой ResNet. Для извлечения финального активационного вектора производится конкатенация активационных векторов первых трёх каскадов свёрток. Снижение размерности к полученным, и так низкоразмерным, активационным векторам ($\dim a = 64$) не применялось. Полные результаты тестов представлены в таблице 7.

6.9 ResNeXt

В качестве развития идей ResNet и GoogLeNet в [30] были представлены свёрточные остаточные нейронные сети типа ResNeXt, однако, вместо строго последовательной структуры из свёрточных блоков, использованной в [22], в ResNeXt, дополнительно, используется структура параллельных маленьких свёрточных блоков. Эта идея позволила повысить качество распознавания остаточных сетей без увеличения числа параметров и глубины нейронной структуры.

В этом разделе описаны результаты экспериментов с ResNeXt-50 $32 \times 4d$ и ResNeXt-101 $32 \times 8d$, представленных в оригинальной статье ResNeXt [30]. Как и в случае с классическими ResNet, для ResNeXt опорная сетевая функция представляет собой конкатенация выходов первых трёх остаточных слоёв. Для обеих моделей имеем $\dim a = 1792$. Опять же, к активационным векторам обеих моделей применялось снижение размерности алгоритмом R_{50} . Развёрнутые полученные результаты представлены в таблице 9.

6.10 SE-сети

[31]

6.11 Wide ResNet

Идея увеличивать структуру свёрточных нейронных сетей с остаточными блоками вширь, а не вглубь, была представлена в [32]. Реализация идея получила название Wide ResNet. Авторы оригинальной архитектуры показали, что Wide ResNet-50-2 показывает результаты на ImageNet лучше, чем глубокая ResNet-152.

В этом разделе описаны результаты экспериментов с самыми успешными представителями архитектуры Wide ResNet: Wide ResNet-50-2 и Wide ResNet-101-2. Для извлечения активационного вектора, по аналогии с обычными ResNet, были выбраны первые 3 слоя, выходы которых конкатенируются. К полученному активационному вектору размерности 1792, как и в оригинальной статье PaDiM [16], для снижения размерности в обоих исследуемых моделях применяется алгоритм Rd с параметром $d = 550$. Подробные полученные цифры по исследуемым метрикам представлены в таблице 8.

6.12 ResNeSt

Введённый в [33] блок разрозненного внимания (Split-Attention) позволяет применить механизм внимания на активационные векторы. Авторы [33] представили новую архитектуру ResNeSt, состоящую из блоков разрозненного внимания, выстроенных последовательно, как в структуре ResNet.

В этом разделе описаны результаты экспериментов с ResNeSt-50 и ResNeSt-101 для инициализации слоёв опорной сетевой функции. Как и в предыдущих случаях исследования остаточных сетей, для извлечения активационного вектора, здесь будет использована конкатенация первых трёх слоёв сети. К полученному вектору размерности 1792 применяется метод снижения размерности R_{550} . Полные результаты тестов описаны в таблице 9.

6.13 IBN-Net

[34]

Результаты представлены в таблице 10.

6.14 MnasNet

Для создания, представленной в [35], легковесной архитектуры MnasNet для мобильных устройств помимо использовалась концепция обучения с подкреплением, заключающаяся в максимизации функции награды.

В этом разделе представлены результаты экспериментов с двумя опорными сетевыми функциями для метода PaDiM, реализованные MnasNet-0.5 и MnasNet-1.0. Активационный вектор строится с помощью конкатенации выходов последних свёрточных слоёв первых трёх остаточных блоков моделей. Снижение размерности для получившихся векторов не применялось: $\dim d = 80$ и $\dim d = 144$ для MnasNet-0.5 и MnasNet-1.0, соответственно. Развёрнутые результаты экспериментов описаны в таблице 11.

6.15 EfficientNet

Авторы архитектуры EfficientNet в [36] предложили метод составного масштабирования, настраивающий в нужных пропорциях баланс между глубиной, шириной и раз-

Таблица 1: Размерности активационных векторов и параметры d для RegNet

| Модель | $\dim a$ | d | Модель | $\dim a$ | d |
|---------------|----------|-----|---------------|----------|-----|
| RegNetY_400MF | 360 | 100 | RegNetX_400MF | 256 | 100 |
| RegNetY_800MF | 528 | 150 | RegNetX_800MF | 480 | 100 |
| RegNetY_1.6GF | 504 | 150 | RegNetX_1.6GF | 648 | 150 |
| RegNetY_3.2GF | 864 | 220 | RegNetX_3.2GF | 720 | 200 |
| RegNetY_8GF | 1568 | 450 | RegNetX_8GF | 1040 | 300 |
| RegNetY_16GF | 1904 | 600 | RegNetX_16GF | 1664 | 500 |
| RegNetY_32GF | 2320 | 700 | RegNetX_32GF | 2352 | 700 |

решением. Метод реализуется решением оптимизационной задачи на метрики качества с ограничением по размеру модели. Путём решения такой задачи и варьированием гиперпараметров оптимизации, в оригинальной статье было предложено 8 моделей архитектуры EfficientNet.

В этом разделе демонстрируются результаты экспериментов с EfficientNet вариантов B0, ..., B7. Для извлечения активационного вектора, . Для вариантов B0, B1, B2, для которых активационный вектор имеет размерность 80, 80 и 88, соответственно, методы снижения размерности не применялись. Для B3, B4, B5, B6, B7 применялся алгоритм Rd с $d = 100$. Размерности активационных векторов: $\dim a_3 = 104$, $\dim a_4 = 112$, $\dim a_5 = 128$, $\dim a_6 = 144$, $\dim a_7 = 160$. Полученные при проведении экспериментов метрики показаны в.

6.16 RegNet

Семейство моделей RegNet, описанное в [37], представляет собой результат поиска в предложенном авторами пространстве гипер-параметров. Это пространство определяет множество возможных свёрточных нейросетей, каждая из которых обучается на ImageNet. В процессе анализа построенного пространства, авторы [37] приходят к семействам архитектур, каждое из которых оптимизировано под один из показателей: легковесность, эффективность и качество.

В этом разделе будут описаны результаты экспериментов с 14 представителями семейства RegNet для инициализации слоёв опорной сетевой функции. Для каждого представителя семейства, активационный вектор строится как конкатенация последних слоёв 1-го, 2-го и 3-го блоков архитектуры RegNet. Получившиеся размерности активационных векторов и параметры d алгоритма снижения размерности Rd для каждой из моделей представлены в таблице 1. Полученные цифры по исследуемым моделям показаны в 12

6.17 ConvNeXt

Развитие архитектуры трансформеров, которые достигли SOTA результатов в области глубокой обработки естественного языка, в компьютерном зрении [38] Результаты представлены в таблице 13.

7 Заключение

Список литературы

- [1] N. Aronszajn. Theory of reproducing kernels. 1950.
- [2] Bernhard Scholkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, John C Platt. Support vector method for novelty detection. 2000.
- [3] David MJ Tax, Robert PW Duin. Support vector data description. Machine learning, 2004
- [4] Platt, J. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998
- [5] Pal, M., Foody, G. M. Feature selection for classification of hyperspectral data by SVM. IEEE Transactions on Geoscience and Remote Sensing.
- [6] Vempati, S., Vedaldi, A., Zisserman, A., Jawahar, C. Generalized RBF feature maps for Efficient Detection.
- [7] Lukas Ruff, Nico Gornitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert Vandermeulen, Alexander Binder, Emmanuel Muller, Marius Kloft. Deep one-class classification. 2018.
- [8] Kingma, D., Ba, J. Adam. A Method for Stochastic Optimization. 2014
- [9] Minyoung Huh, Pulkit Agrawal, Alexei A Efros. What makes ImageNet good for transfer learning?
- [10] Pramuditha Perera, Vishal M Patel. Learning deep features for one-class classification.
- [11] Tal Reiss, Niv Cohen, Liron Bergman, Yedid Hoshen. Panda—adapting pretrained features for anomaly detection.
- [12] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska. Overcoming catastrophic forgetting in neural networks. 2017
- [13] Tal Reiss, Yedid Hoshen. Mean-Shifted Contrastive Loss for Anomaly Detection.
- [14] Izhak Golan, Ran El-Yaniv. Deep anomaly detection using geometric transformations. 2018
- [15] N. Cohen, Y. Hoshen. Sub-image anomaly detection with deep pyramid correspondences. 2020
- [16] Thomas Defard, Aleksandr Setkov, Angelique Loesch, Romaric Audigier. PaDiM: a Patch Distribution Modeling Framework for Anomaly Detection and Localization.
- [17] P. Mahalanobis. On the generalized distance in statistics. 1936.

- [18] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger. Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection. 2019
- [19] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Scholkopf, Thomas Brox, Peter Gehler. Towards Total Recall in Industrial Anomaly Detection. 2021
- [20] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. 2014
- [21] Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2014
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. 2015
- [23] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. 2016
- [24] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. Densely Connected Convolutional Networks. 2016
- [25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. Going Deeper with Convolutions. 2014
- [26] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. 2015
- [27] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, Jian Sun. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. 2018
- [28] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018
- [29] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, Hartwig Adam. Searching for MobileNetV3. 2019
- [30] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. 2016
- [31] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu. Squeeze-and-Excitation Networks. 2017
- [32] Sergey Zagoruyko, Nikos Komodakis. Wide Residual Networks. 2016

- [33] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, Alexander Smola. ResNeSt: Split-Attention Networks. 2020
- [34] Xingang Pan, Ping Luo, Jianping Shi, Xiaoou Tang. Two at Once: Enhancing Learning and Generalization Capacities via IBN-Net. 2018
- [35] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, Quoc V. Le. MnasNet: Platform-Aware Neural Architecture Search for Mobile. 2018
- [36] Mingxing Tan, Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. 2019
- [37] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, Piotr Dollár. Designing Network Design Spaces. 2020
- [38] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, Saining Xie. A ConvNet for the 2020s. 2022

Таблица 2: Полученные для Alexnet и SqueezeNet метрики ROC-AUC и pROC-AUC

| | | | |
|--------|---------|----------------|----------------|
| Бэкбон | Alexnet | SqueezeNet-1.0 | SqueezeNet-1.1 |
|--------|---------|----------------|----------------|

Таблица 3: Полученные для VGG метрики ROC-AUC и pROC-AUC

| | | | |
|--------|---------|-----|-----|
| Бэкбон | Alexnet | VGG | VGG |
|--------|---------|-----|-----|

Таблица 4: Полученные для ResNet метрики ROC-AUC и pROC-AUC

| | | | | | |
|--------|-----------|-----------|-----------|------------|------------|
| Бэкбон | ResNet-18 | ResNet-34 | ResNet-50 | ResNet-101 | ResNet-152 |
|--------|-----------|-----------|-----------|------------|------------|

Таблица 5: Полученные для DenseNet метрики ROC-AUC и pROC-AUC

| | | | | |
|--------|-------------|-------------|-------------|-------------|
| Бэкбон | densenet121 | densenet169 | densenet161 | densenet201 |
|--------|-------------|-------------|-------------|-------------|

Таблица 6: Полученные для ShuffleNet метрики ROC-AUC и pROC-AUC

| | | |
|--------|--------------------|--------------------|
| Бэкбон | shufflenet v2 x0 5 | shufflenet v2 x1 0 |
|--------|--------------------|--------------------|

Таблица 7: Полученные для MobileNet метрики ROC-AUC и pROC-AUC

| | | | |
|--------|-------------|-------------------|-------------------|
| Бэкбон | MobileNetV2 | MobileNetV3 Small | MobileNetV3 Large |
|--------|-------------|-------------------|-------------------|

Таблица 8: Полученные для Wide ResNet метрики ROC-AUC и pROC-AUC

| | | |
|--------|------------------|-------------------|
| Бэкбон | Wide ResNet-50-2 | Wide ResNet-101-2 |
|--------|------------------|-------------------|

| Бэкбон | PaDiM | |
|--------------------|---------|----------|
| | ROC-AUC | pROC-AUC |
| shufflenet v2 x0 5 | 1 | 1 |
| shufflenet v2 x1 0 | 1 | 1 |

Таблица 9: Полученные для ResNeSt и Wide ResNet метрики ROC-AUC и pROC-AUC

| | | | | |
|--------|------------|-------------|------------------|-------------------|
| Бэкбон | ResNeSt-50 | ResNeSt-101 | ResNeXt-50 32x4d | ResNeXt-101 32x8d |
|--------|------------|-------------|------------------|-------------------|

Таблица 10: Полученные для IBN-Net метрики ROC-AUC и pROC-AUC

| | | | | |
|--------|-----------------|-----------------|-------------------|---------------------|
| Бэкбон | ResNet-50-IBN-a | ResNet-10-IBN-a | ResNeXt-101-IBN-a | SE-ResNet-101-IBN-a |
|--------|-----------------|-----------------|-------------------|---------------------|

Таблица 11: Полученные для MnasNet и Inception метрики ROC-AUC и pROC-AUC

| | | | | |
|--------|-----------|--------------|-------------|-------------|
| Бэкбон | GoogLeNet | Inception v3 | MnasNet-0.5 | MnasNet-1.0 |
|--------|-----------|--------------|-------------|-------------|

Таблица 12: Полученные для RegNet метрики ROC-AUC и pROC-AUC

| | | | | |
|--------|---|---|---|---|
| Бэкбон | 1 | 2 | 3 | 4 |
|--------|---|---|---|---|

Таблица 13: Полученные для ConvNeXt метрики ROC-AUC и pROC-AUC

| Бэкбон | ConvNeXt tiny | ConvNeXt small | ConvNeXt base | ConvNeXt large |
|--------|---------------|----------------|---------------|----------------|
|--------|---------------|----------------|---------------|----------------|