

Использование нейросетевого метода для решения задач сегментации изображений

М. Н. Б. Муаль

ФГАОУ ВО «Российский университет дружбы народов», г. Москва, Российская Федерация
Адрес: 117198, Российская Федерация, г. Москва, ул. Миклухо-Маклая, д. 6
bmouale@mail.ru

Аннотация

Сегментация изображений играет важную роль при выявлении различных заболеваний и патологий посредством обработки медицинских изображений. За прошедшие годы был разработан целый ряд традиционных подходов такие как бинарный пороговый метод (метод Оцу), метод водоразделов, метод кластеризации с помощью K -средних к выполнению сегментации изображений, использующих специфичные для предметной области знания для эффективного решения проблем сегментации в конкретных областях применения. Но, к сожалению, эти методы являлись не эффективными. На самом деле, суть сегментации изображений является классификацией содержащих каждый пиксель с аналогичными атрибутами. А цель классификации пиксель – это обработать его крупно, нечувствительно к шуму, мелко и применить средний звездочный фильтр то, что не позволяют сделать традиционные методы. Поэтому есть необходимость предложить новый метод, позволяющий устранять эти недостатки. Мы предлагаем метод, основанный на автокодировщике (специальная архитектура искусственных нейронных сетей, позволяющая применять обучение без учителя при использовании метода обратного распространения ошибки), позволяющий сжимать любое большое изображение на мелкое (с теми же свойствами как входные). Этот метод позволяет также сэкономить ресурсы, при этом, мы используем метод обработки изображение под названием «MaxPooling» (U-Net). В данной работе создана собственная метрика для отслеживания результата тренировки сети. В практической части сегментированы на классы изображения со стройки, а также попиксельно определяется местонахождения самолета на изображении.

Ключевые слова: сегментация, пиксель, автокодировщик, MaxPooling, U-Net, метрика, обнаружение объектов, искусственные нейронные сети, CNN-сети, полносверточная сеть

Конфликт интересов: автор заявляет об отсутствии конфликта интересов.

Для цитирования: Муаль М. Н. Б. Использование нейросетевого метода для решения задач сегментации изображений // Современные информационные технологии и ИТ-образование. 2022. Т. 18, № 4. С. 744-755. doi: <https://doi.org/10.25559/SITITO.18.202204.744-755>

© Муаль М. Н. Б., 2022



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



Using a Neural Network Method to Solve Image Segmentation Problems

M. N. B. Mouale

Peoples' Friendship University of Russia, Moscow, Russian Federation
Address: 6 Miklukho-Maklaya St., Moscow 117198, Russian Federation
bmouale@mail.ru

Abstract

Image segmentation plays an important role in detecting various diseases and pathologies through medical image processing. Over the years, a number of traditional approaches such as the binary threshold method (Otsu method), watershed method, and K-means clustering method have been developed to perform image segmentation, using domain-specific knowledge to efficiently solve segmentation problems in specific applications. But unfortunately, these methods were not effective. In fact, the essence of image segmentation is the classification of containing each pixel with similar attributes. And the goal of classifying a pixel is to treat it coarse, insensitive to noise, fine, and a medium star filter. Something that traditional methods cannot do. Therefore, there is a need to propose a new method that allows eliminating these disadvantages. We propose a method based on autoencoder (a special architecture of artificial neural networks that allows the application of learning without a teacher using the method of back propagation of the error), which allows compressing any large image into a small one, (with the same properties as the input). This method also saves resources. In this case, we use an image processing method called "MaxPooling" (U-Net). In the work, we created our own metric to track the result of the network training. In the practical part, we tried segmenting into classes the images from the construction site, as well as pixel by pixel the locations of the aircraft in the images.

Keywords: segmentation, pixel, autoencoder, MaxPooling, U-Net, metrics, object detection, artificial neural networks, CNNs, full-width network

Conflict of interests: The author declares no conflict of interest.

For citation: Mouale M.N.B. Using a Neural Network Method to Solve Image Segmentation Problems. *Modern Information Technologies and IT-Education*. 2022;18(4):744-755. doi: <https://doi.org/10.25559/SITITO.18.202204.744-755>



Введение

В век нанотехнологий работа нейронных сетей актуальна как никогда. К числу ключевых направлений, которые в последнее время привлекли к себе большое внимание, относятся сегментация изображений, а также локализация и обнаружение объектов. Сегментация – это выделение объектов в исходных данных. Как отдельный класс сегментация изображений заключается в подсвечивании объектов на изображении, другими словами, можно сказать, что сегментация изображений – это попиксельная классификация, где каждому пикселю объекта присваивается определенный класс. Цель сегментации изображения состоит в том, чтобы изменить представление изображения во что-то более значимое и более простое для анализа. Сегментация изображения обычно используется для поиска объектов и границы (линии, кривые и т. д.) на изображениях. Она также находит применение в таких отраслях, как авиастроение, машиностроение и многие другие, где оно может быть использовано для выявления всевозможных дефектов, таких как трещины или другие механические разрушения в оборудовании [1].

Изображения ночного неба могут сегментироваться для обнаружения неизвестных галактик, звезд и планет. Обнаружение и локализация объектов широко используется в местах, требующих постоянного автоматизированного наблюдения за текущей активностью, например в супермаркетах, магазинах, на промышленных предприятиях и т.п. Кроме того, данную технологию можно применять для подсчета количества объектов и людей в определенном месте, а также в системах управления дорожным движением [2, 3].

Статья начинается с обсуждения традиционных методов сегментации изображений, чтобы мы могли оценить, насколько нейронные сети отличаются от традиционных аналогов. После этого мы рассмотрим архитектуры нейронных сетей для решения задач сегментации. Далее, мы создадим свою собственную архитектуру, запустим и оценим точность на проверочной выборке. Наконец, завершаем наше обсуждение рядом экспериментов относительно того, каким образом можно попиксельно определить местонахождение самолета на изображении.

Обозначения:

$I(i, j)$ – интенсивность пикселя;

$I(i, j)$ – интенсивность сегментированного изображения;

T – пороговое значение;

k – значение, которое может принимать интенсивность пикселя;

σ^2 – дисперсия изображения;

σ_w^2 – дисперсия внутри класса;

σ_b^2 – дисперсия между классами;

ω_1 – вероятность оказаться (принадлежать) в 1 классе;

σ_1^2 – дисперсия первого класса;

ω_2 – вероятность оказаться во 2 классе;

σ_2^2 – дисперсия второго класса;

$p(k)$ – вероятность каждого уровня интенсивности пикселя;

$hist$ – значение гистограммы;

N_i – общее количество пикселей в изображении;

$\sigma_1^2(T)$ – дисперсия первого класса;

$\sigma_2^2(T)$ – дисперсия второго класса;

N_1 – вектор от 0 до $T-1$;

N_2 – вектор от T до 255;

$Moу_1$ – среднее значение класса 1;

$Moу_2$ – среднее значение класса 2;

$S_1, S_2, S_3, \dots, S_k$ – кластеры или классы;

$C_1, C_2, C_3, \dots, C_k$ – центроиды;

L – сумма внутрикластерных расстояний;

$J(A, B)$ – метрика;

$A \cap B$ – площадь пересечения;

$A \cup B$ – площадь объединения;

DSC – перекрытие сегментированных объемов;

ASD – среднее симметричное поверхностное расстояние;

Def – собственная метрика;

y_{true} – значение ошибки, которое должно быть;

y_{pred} – значение ошибки, которое возвращает модель;

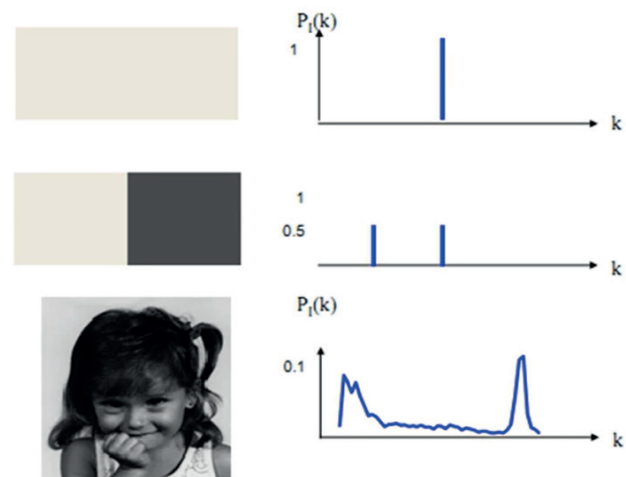
N – количество элементов (классов).

Традиционные методы сегментации изображений

Для решения задач сегментации изображений существуют различные традиционные методы, рассмотрим некоторые из них.

2.1. Пороговый метод, основанный на гистограммах интенсивности пикселей

Гистограмма – это графическое представление суммы пикселей изображений (Рис. 1). Это график, представляющий распределение интенсивности каждого пикселя в изображении. Надо отметить, что два разных изображения могут иметь одинаковую гистограмму. Гистограмма изображения позволяет оценить количество и разнообразие оттенков изображения, а также общий уровень яркости изображения [4].



Р и с. 1. Гистограмма изображений

F i g. 1. Image histogram

Пороговый метод – это метод обработки, который позволяет выделить на изображениях важную информацию. Входное



изображение является изображением в градациях серого, а изображение, полученное в результате этой обработки, является черно-белым. Эта обработка требует настройки параметра: порога T . Порог – это общее среднее значение пикселей в изображении.

Часто в изображении имеются только две значимые области, представляющие интерес: объект и фон. В подобных сценариях гистограмма интенсивности пикселей представляет распределение вероятности, являющееся бимодальным, т.е. таким, которое характеризуется наличием двух пиков интенсивности пикселей.

Если изображение представлено функцией $I(i, j)$ а пороговое значение t выбрано на основании гистограммы интенсивности пикселей, то новое сегментированное изображение $I'(x, y)$ можно представить в виде:

$$I'(i, j) = \frac{(I(i, j) - \text{Min}(I)) \times 255}{\text{Max}(I) - \text{Min}(I)} \quad (1)$$

Существуют фиксированный, глобальный и локальный методы определения порогового значения T .

Для фиксированного порога в качестве параметра функции $I(x, y)$ берется фиксированный порог T . Пороговое значение T берется случайным образом. Проходим по всем пикселям нашего изображения, если пиксель больше этого порога, то берем значение 1, иначе берем значение 0.

$$\begin{cases} I'(i, j) = 0, \text{если } (I(i, j) > T), \\ I'(i, j) = 1, \text{если } (I(i, j) \leq T) \end{cases} \quad (2)$$

Для глобального порогового метода мы используем функцию фиксированного порогового значения, но здесь мы вычисляем фиксированное пороговое значение. Для вычисления порога мы складываем все пиксели изображения и делим все на количество пикселей исходного изображения. Нужно заметить, что если среднее пороговое значение слишком низкое, то мы не сможем получить правильные контуры изображения.

В случае локального метода мы также вычисляем среднее значение, но это среднее значение является локальным для текущего пикселя. Мы передаем в качестве параметра окрестность, которая используется для определения того, какие пиксели изображения мы должны использовать для расчета среднего значения. При окрестности 1 среднее значение рассчитывается по 8 точкам, окружающим текущий пиксель.

Лучшее пороговое значение – это фиксированное пороговое значение. Недостатком этого метода является выбор значения порога n , потому что это делается случайным образом.

2.2. Метод Оцу

Метод Оцу используется для автоматического определения порога по форме гистограммы изображения. Следовательно, этот метод требует предварительного расчета гистограммы изображения. Алгоритм предполагает, что изображение, которое необходимо преобразовать в бинарную форму, содержит только два класса (объекты и фон). Затем итеративный алгоритм вычисляет оптимальный порог T , который разделяет

эти два класса так, чтобы дисперсия внутри класса была минимальной, а дисперсия между классами максимальной¹ [5].

Дисперсия внутри класса:

$$\sigma_w^2 = \omega_1(T) \times \sigma_1^2(T) + \omega_2(T) \times \sigma_2^2(T) \quad (3)$$

Дисперсия между классами:

$$\sigma_y^2 = \sigma^2 - \sigma_w^2 \quad (4)$$

Расчет вероятности класса 1 и 2: для расчета вероятности попадания в класс 1 или 2 по порогу T достаточно просуммировать вероятности уровня интенсивности каждого пикселя.

$$\omega_1 = \sum_{k=1}^T p(k) \quad (5)$$

$$\omega_2 = \sum_{k=T+1}^{256} p(k) \quad (6)$$

Расчет вероятности интенсивности каждого пикселя

Расчет гистограммы: для расчета гистограммы необходимо просмотреть изображение целиком и подсчитать количество пикселей для уровня интенсивности каждого пикселя (niveau de gris)

$$\text{hist} = \sum_{i=1}^N \sum_{j=1}^M (I(i, j) == k) \quad (7)$$

Расчет вероятности для каждого уровня интенсивности пикселя

Вероятность каждого уровня интенсивности пикселей вычисляется путем деления числа пикселей, присутствующих для каждого уровня интенсивности пикселей, на общее количество пикселей в изображении.

$$p(k) = \frac{\text{hist}(k)}{N_i} \quad (8)$$

Расчет дисперсии каждого класса :

$$\sigma_1^2(T) = \frac{\sum_{i=1}^T (N_1(i) - M_0 y_1(T))^2 \times P(i)}{\omega_1} \quad (9)$$

$$\sigma_2^2(T) = \frac{\sum_{i=T+1}^{256} (N_2(i) - M_0 y_2(T))^2 \times P(i)}{\omega_2} \quad (10)$$

Вычисление среднего значения каждого класса: среднее значение каждого класса вычисляется путем суммирования вектора N , который умножается на вероятность каждого уровня интенсивности пикселей. Затем все делится на вероятность класса.

$$\text{Moy}_1(T) = \frac{\sum_{i=1}^T N_1(i) \times P(i)}{\omega_1(T)} \quad (11)$$

$$\text{Moy}_2(T) = \frac{\sum_{i=T+1}^{256} N_2(i) \times P(i)}{\omega_2(T)} \quad (12)$$

¹ Morse B. S. Lecture 4: Thresholding [Электронный ресурс] // Brigham Young University, 2000. URL: https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf (дата обращения: 19.10.2022).



2.3. Сегментация изображений методом кластеризации с помощью K -средних

Широко известный алгоритм K -средних также может быть использован для сегментации изображений. « K » в названии – это параметр алгоритма, определяющий количество кластеров. Алгоритм формирует кластеры, каждый из которых представляется центром кластера, который называется центроидом и выбирается на основе признаков входных данных. В случае сегментации изображений такими признаками обычно служат интенсивность пикселя и три его пространственные координаты: горизонтальная, вертикальная и цветовой канал [6]. Поэтому вектор входных признаков можно представить в виде $v \in \mathbb{R}^{4 \times 1}$, где $v = [I(x, y, z), x, y, z]^T$. (13)

В данном алгоритме используются такие меры расстояния, как L^2 или L^1 :

$$D(v^{(i)}, v^{(j)}) = \|v^{(i)} - v^{(j)}\| = \sqrt{(v^{(i)} - v^{(j)})^T (v^{(i)} - v^{(j)})}, \quad (14)$$

$$D(v^{(i)}, v^{(j)} / L^1) = \|v^{(i)} - v^{(j)}\|_1$$

Посмотрим принцип работы алгоритма:

- Шаг 1: начинаем с выбранных случайным образом K центроидов кластеров $C_1, C_2, C_3, \dots, C_k$ соответствующих K кластерам $S_1, S_2, S_3, \dots, S_k$
- Шаг 2: вычисляем для каждого пикселя расстояния его вектора признаков $v^{(i)}$ до центроидов кластеров и относим объекты в ближайшую центральную группу. Повторяем этот процесс для всех векторов признаков пикселей до тех пор, пока каждый из пикселей не будет приписан к одному из K кластеров.
- Шаг 3: вычислить центр тяжести или центроиды каждой группы. Определив центроиды новых кластеров для всех пикселей, вычисляем их заново путем усреднения векторов признаков пикселей в каждом кластере:

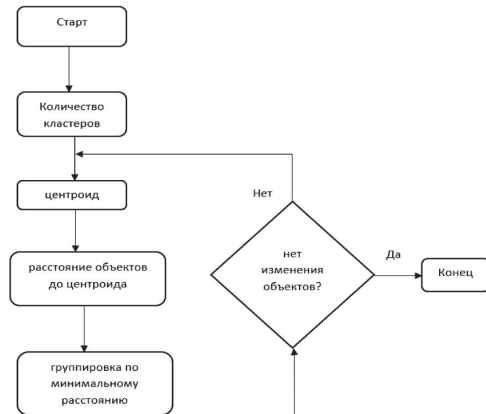
$$C_j = \sum_{v^{(i)} \in S_j} v^{(i)}$$

Повторяем шаги 2 и 3 в течение нескольких итераций до тех пор, пока центроиды не перестанут изменяться. Посредством этого итеративного процесса мы уменьшаем сумму внутрикластерных расстояний, имеющую следующий вид:

$$L = \sum_{j=1}^k \sum_{v^{(i)} \in S_j} \|v^{(i)} - C_j\| \quad (15)$$

Недостатки алгоритма

- Количество кластеров должно быть зафиксировано на старте;
- Результат зависит от первоначальной жеребьевки центроидов кластеров, которые называются *центроидами*;
- Кластеры строятся по отношению к несуществующим объектам (центры).



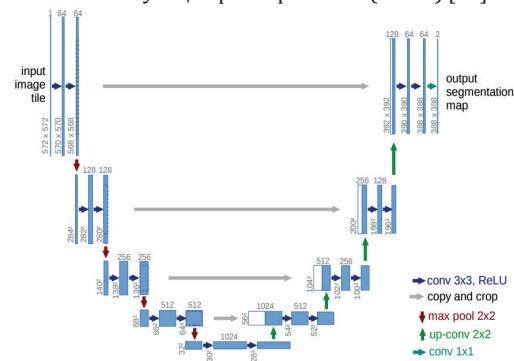
Р и с. 2. Блок схемы алгоритма K -средних
F i g. 2. Flow diagram of the K -means algorithm

2.4. Архитектуры нейронных сетей для решения задач сегментации

В последние годы большую популярность приобрела сегментация изображений с помощью сверточных нейронных сетей² [7, 8]. При решении задач сегментации изображений с помощью нейронных были разработаны различные архитектуры нейронных сетей [9, 10]. Рассмотрим некоторые из них.

U-NET

По своей структуре сеть схожа с автокодировщиком, в котором сеть сжимает данные в скрытое пространство, тем самым выявляя основные признаки, и потом восстанавливает изображение из скрытого пространства. Архитектура U-net представляет собой последовательность блоков, сначала уменьшая размерность изображения (блоки включают Pooling-слои), а потом увеличивая, предварительно объединив с выходами начальных блоков с соответствующей размерностью (Рис. 3) [11].



Р и с. 3. Архитектура U-net [12, С. 235]

F i g. 3. U-net Architecture [12, P. 235]

² Сикорский О. С. Обзор сверточных нейронных сетей для задачи классификации изображений // Новые информационные технологии в автоматизированных системах. 2017. № 20. С. 37-42. URL: <https://www.elibrary.ru/item.asp?id=29109627> (дата обращения: 19.10.2022); Hizem W. Capteur Intelligent pour la Reconnaissance de Visage: Thèse présentée de pour l'obtention du diplôme de Docteur De L'institut National Des Telecommunications. Paris : Télécommunications et l'Université Pierre et Marie Curie – Paris, 2009. 111 p. URL: http://biometrics.it-sudparis.eu/downloads/publications/hizem_PHD.pdf (дата обращения: 19.10.2022); Fiche C. Repousser les limites de l'identification faciale en contexte de vidéo-surveillance: Thèse présentée de pour l'obtention du diplôme de Docteur. Spécialité: Signal – Images – Parole – Télécoms. Grenoble, France : Université de Grenoble, 2012. 187 p. URL: <https://tel.archives-ouvertes.fr/tel-00767214/document> (дата обращения: 19.10.2022); Van Wambeke M. Reconnaissance et suivi de visages et implémentation en robotique temps-réel: en vue de l'obtention du diplôme de Master Ingénieur Civil En Génie Biomédical. Louvain-la-Neuve, Belgique : Université Catholique de Louvain, 2009-2010. 87 p. URL: <https://fdocuments.fr/document/reconnaissance-et-suivi-de-visages-et-impl-ementation-reddotcommission-gbio.html> (дата обращения: 19.10.2022); Phung Van Doanh. Reconnaissance de visages en utilisant le descripteur POEM pattern of oriented Edge Magnitudes, Stage. Vietnam, Hanoi : Institut de la Francophonie pour l'informatique, 2010.



Плюсы U-net

- Вычислительно эффективна;
- Обучается на небольшом датасете;
- Изначально – для биомедицинских изображений.

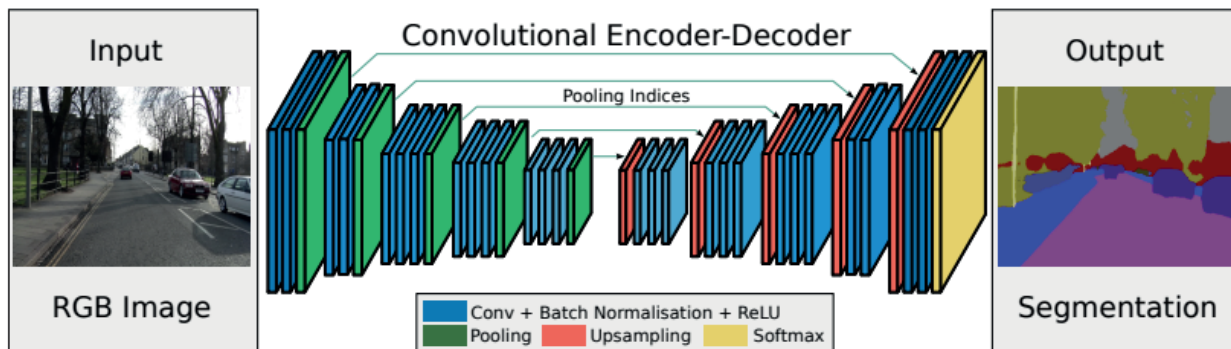
Минусы U-net

- Сложнее с многоклассовой сегментацией. Полносверточная сеть (*fully convolutional network*) – FPN and PSPNet более приспособлены;

- Проблема границ.

SEGNET

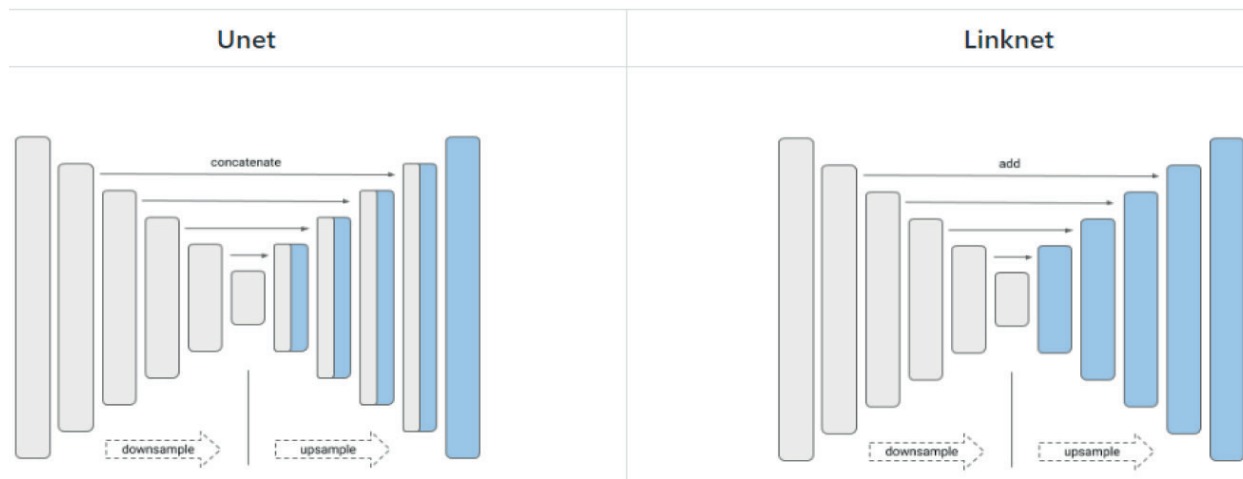
SegNet – это сверточный автокодировщик, последний слой которого – слой попиксельной классификации. Архитектура состоит из сверточных блоков, сначала уменьшая размерность изображения (блоки включают Pooling-слои), а потом увеличивая до исходной размерности (Рис. 4) [13].



Р и с. 4. Архитектура SegNet [14, С. 2484]
 F i g. 4. An illustration of the SegNet architecture [14, P. 2484]

LinkNet

По своей структуре LinkNet – это Unet, в которой слои объединения (*concatenate*) заменены на слои добавления (Рис. 5) [8], [15].



Р и с. 5. Архитектура LinkNet³
 F i g. 5. Architecture LinkNet

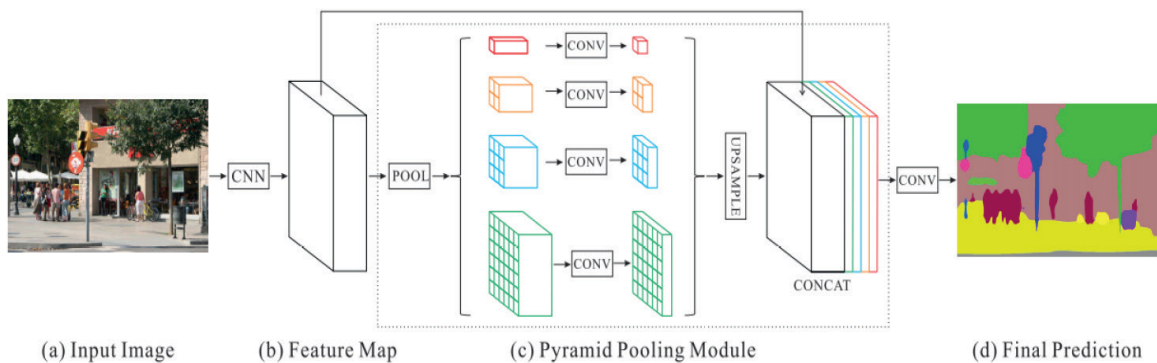
PSPNet

Архитектура PSPNet построена на модели pooling пирамид, состоит из блока нескольких сверточных слоев (Рис. 6) [16]. Карта признаков на выходе данного блока проходит через pooling пирамиду. То есть карта признаков проходит через не-

сколько Pooling-слоев с разны ядром. После чего каждый слой в отдельности проходит через сверточные слои и возвращается соответствующая размерность соответствующего слоя (Upsampling, Conv2DTranspose). Далее все слои объединяются (*concatenate*) и подаются на выходной сверточный слой.

³ Iakubovskii P. Segmentation Models [Электронный ресурс] // GitHub repository. GitHub, 2019. URL: https://github.com/qubvel/segmentation_models (дата обращения: 19.10.2022).





Р и с. 6. Архитектура PSPNet [17, С. 6233]

F i g. 6. Architecture PSPNet [17, P. 6233]

Создание собственной метрики для отслеживания результата тренировки сети (пример: сегментация на классы изображения со стройки)

Мы имеем изображение и нам нужно его подать, чтобы получить на выходе того же изображение. Но если есть, например, во входном изображении «человек», то мы будем каждый пиксель входного изображения анализировать и предсказывать во входном изображении: есть ли человек или нет (Рис. 7).



Р и с. 7. Последовательность шагов сегментации изображений

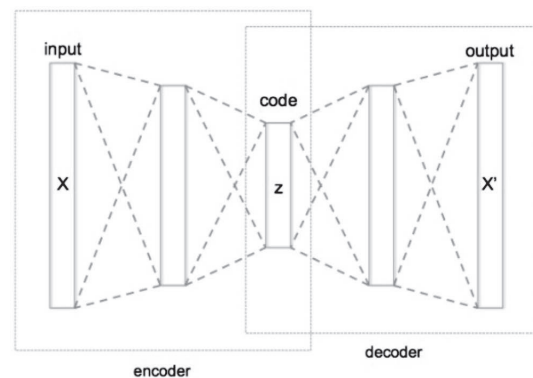
F i g. 7. Sequence of Image Segmentation Steps

Предсказание делается по классу, т.е. задача сегментации – это задача классификации пикселей в изображении. Но нужно классифицировать пиксели, имеющие какие-то разнообразия. Классифицируя пиксели, мы хотим обработать их крупно, мелко, и средним звездочным фильтром. Традиционная функция не позволяет это сделать. Новый метод (модифицирование функции потерь), позволяющий эффективно решить эту задачу, основан на автокодировщике.

Автокодировщик (*autoencoder*) – специальная архитектура

искусственных нейронных сетей, позволяющая применять обучение без учителя, при использовании метода обратного распространения ошибки.

Цель обучения автокодировщиков: найти внутреннюю структуру в данных. Применение автокодировщиков: улучшение качества изображения (увеличение размера, шумоподавление и т. д.), генерация новых данных по заданному образцу (например, раскрашивание черно-белых картинок в цветные), поиск выбросов. Простейшая архитектура автокодировщика – сеть прямого распространения без обратных связей, наиболее схожая с перцептроном и содержащая входной слой, промежуточный слой и выходной слой. В отличие от перцептрона, выходной слой автокодировщика должен содержать столько же нейронов, сколько и входной слой⁴ (Рис. 8) [18-20].

Р и с. 8. Архитектура автокодировщика⁵

F i g. 8. Autoencoder architecture

Автокодировщик состоит из двух частей: энкодера (кодирует выборку X в свое внутреннее представление Z) и декодера (восстанавливает исходную выборку X'). Таким образом, автокодировщик пытается совместить восстановленную версию каждого объекта выборки с исходным объектом.

Рассмотрим пример работы автоэнкодера на восстановлении

⁴ Goodfellow I., Bengio Y., Courville A. Chapter 14: Autoencoders [Электронный ресурс] // Deep Learning. MIT Press, 2016. P. 499-523. URL: <https://www.deeplearningbook.org/contents/autoencoders.html> (дата обращения: 19.10.2022); Khandelwal R. Deep Learning – Different Types of Autoencoders [Электронный ресурс] // DataDrivenInvestor. December 2, 2018. URL: <https://medium.datadriveninvestor.com/deep-learning-different-types-of-autoencoders-41d4fa5f7570> (дата обращения: 19.10.2022).

⁵ by Chervinskii – Own work, CC-BY-SA-4.0 [Электронный ресурс]. URL: https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png (дата обращения: 19.10.2022).



изображений из библиотеки mnist. На вход сеть принимает объект X (цифра 4) (Рис. 9).

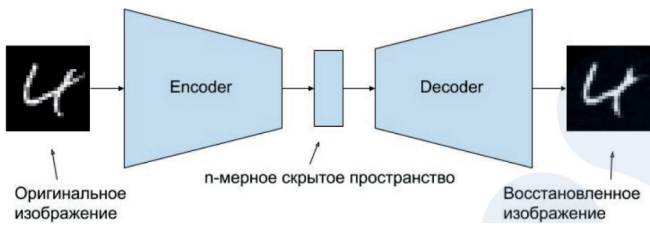


Рис. 9. Пример работы автоэнкодера на восстановлении изображений
Fig. 9. An example of how an autoencoder works on image recovery

Наша сеть кодирует этот объект в скрытое состояние. Затем по скрытому состоянию восстанавливается реконструкция объекта X, которая должна быть похожа на X. Как мы видим, восстановленное изображение (справа) стало более размытым. Объясняется это тем, что мы стараемся сохранить в скрытом представлении только наиболее важные признаки объекта, поэтому объект восстанавливается с потерями.

3.1. Проведение практического примера сегментации на классы изображения со стройки

В практической части, мы собирали данные (несколько фотографий со стройки) для подачи в модель, визуализировали выборку с несколькими выделенными классами на картинке. Подготовили разные варианты модели U-net (одна из архитектур для сегментации). Создавали свою метрику для отслеживания результата тренировки сети. А также попиксельно определили местонахождение самолета. Мы задаём изображения на вход нейронной сети и на выходе получим классификацию каждого пикселя, т.е. определим в каком классе каждый пиксель относится.

Метрики качества

Использование метрики предназначено для минимизации модели сегментации. Общая формула определение метрики:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{16}$$

Точность сегментации оценивается с помощью двух показателей:

1. Dice Similarity Coefficient (DSC) (перекрывание сегментированных объемов);

$$DSC = \frac{2|S_{pred} \cap S_{ref}|}{|S_{pred}| + |S_{ref}|} \tag{17}$$

Average Symmetric Surface Distance (ASD) Среднее симметричное поверхностное расстояние

$$ASD = \frac{1}{N} \left(\sum_{x \in B_{pred}} d(x, B_{ref}) + \sum_{x \in B_{ref}} d(x, B_{pred}) \right) \tag{18}$$

Создание собственной функции метрики

$$Def = \frac{2(\sum k(y_{true} \times y_{pred}) + 1)}{(\sum k(y_{true}) + \sum k(y_{pred}) + 1)} \tag{19}$$

Результаты обучения нейронной сети

Для проверки эффективности работы созданной собственной метрики Def (dice_coef), мы создавали собственную архитектуру U-net и провели эксперименты для различных batch_size и различными активационными функциями. На вход подаем изображение со стройки и на выход получаем Сегментированные изображения. На рисунках 10 и 11 представлены оригинальные изображения и Сегментированные изображения соответственно, а также результаты распознавания линейной сегментирующей сети на рисунке 12 [21-25].

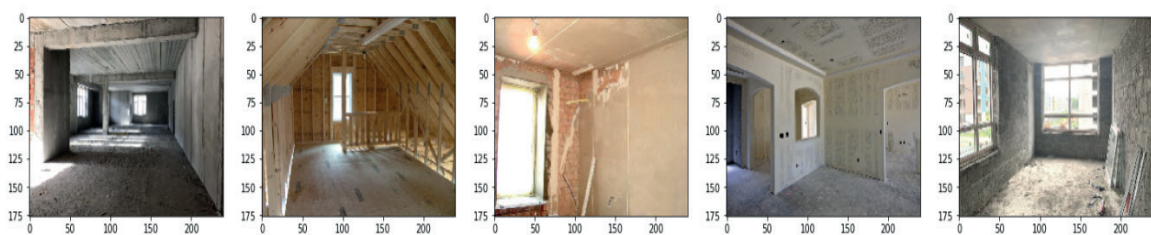


Рис. 10. Оригинальные изображения
Fig. 10. The original images

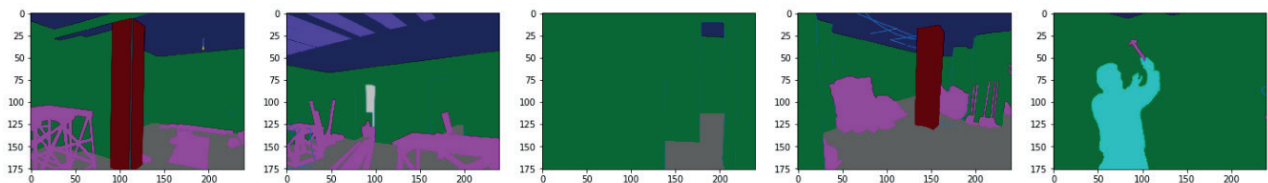
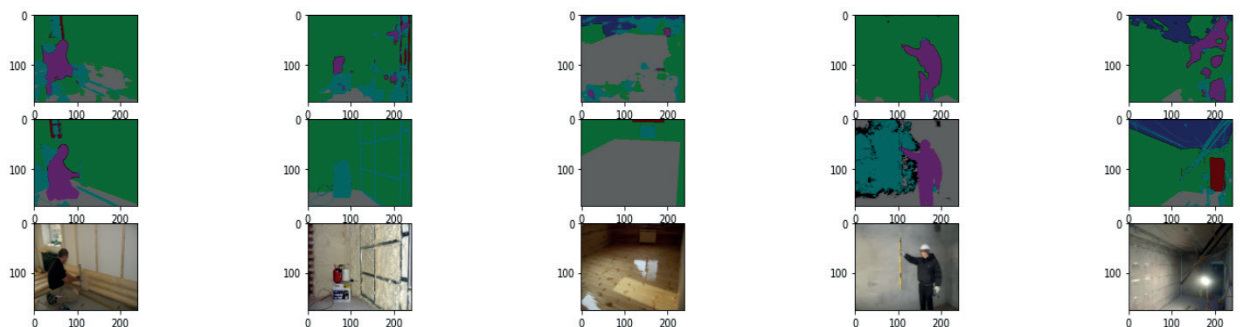


Рис. 11. Сегментированные изображения
Fig. 11. The segmented images





Р и с. 12. Результаты распознавания линейной сегментирующей сети

F i g. 12. Recognition results of a linear segmenting network

Результаты обучения сетей с различными batch_size и различными активационными функциями представлены в Таблице 1.

Т а б л и ц а 1. Результаты обучения сетей с различными batch_size и различными активационными функциями

T a b l e 1. Results of training networks with different batch_size and different activation functions

Название сети	Значения batch_size	Активационные функции	метрики	Точности Распознания
Линейная сегментирующая сеть	32	softmax	dice_coef	0.4415
U-net	32	softmax	dice_coef	0.7284
Упрощённая U-net	32	softmax	dice_coef	0.6034
	8		dice_coef:	0.6366
	128		dice_coef	0.5648
	32	elu	dice_coef	0.6460
		tanh	dice_coef	0.6730
		linear	dice_coef	0.6732
Расширенная U-net	16	softmax	dice_coef	0.7734
U-Net simple training	32	softmax	iou_score:	0.4259
U-Net fine tuning	32	softmax	dice_coef	0.8977

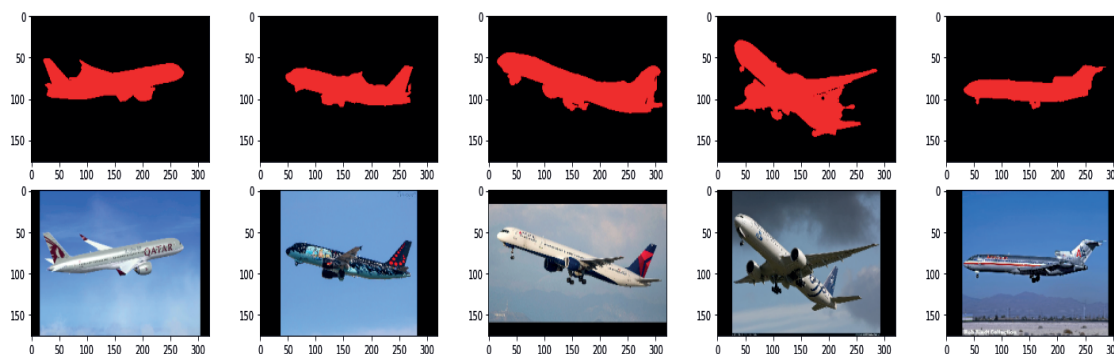


Выводы

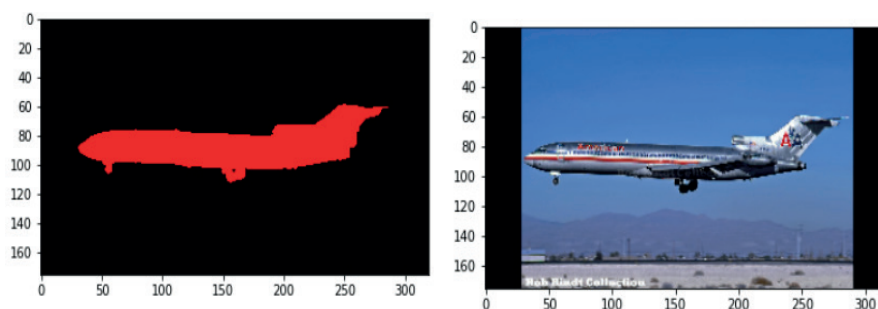
Наилучший результат показала модель U-Net, предобученная на датасете «ImageNet» [26]. Результат после дообучения dice_coef : 0.8977, что превысило результаты других моделей как минимум на 12%. Для обучения данной сети предварительно был запущен код загрузки наборов данных с размером $256 \times 256 \times 3$. Далее код обновлен под размер данных $176 \times 240 \times 3$. В целом хороший результат дала расширенная U-Net и чуть слабее модель U-Net. В аутсайдерах линейная модель, даже на 100 эпохах результат не улучшился. Худший показатель получен у модели U-Net simple training, где была изменена метрика

dice_coef и $\text{loss} \Rightarrow \text{categorical_crossentropy}$ на $\text{loss} = \text{bce_jaccard_loss}$, $\text{metrics} = [\text{iou_score}]$. Увеличение batch_size до 128 скорее ухудшило качество работы сети. А уменьшение batch_size до 8 дало незначительное улучшение качества сегментации. Лучшие результаты у функций активации «linear» и «tanh». Как ни странно «softmax» прекрасно отработал.

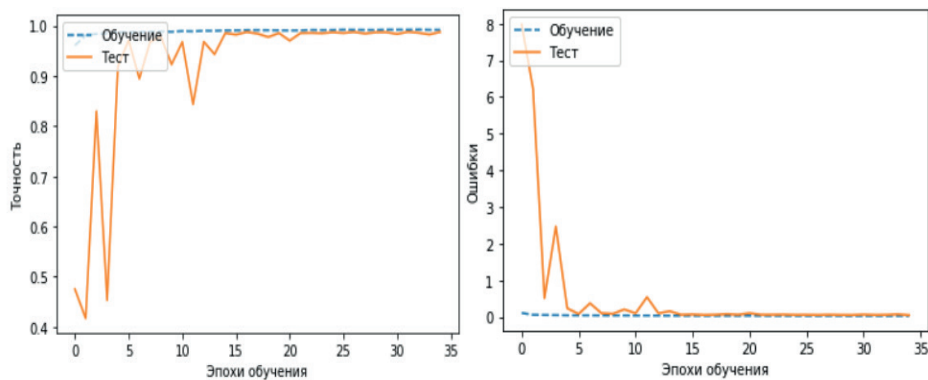
3.2. Попиксельное определение местонахождение самолета
Мы написали PSPNet (архитектуры нейронных сетей) для сегментации на базе самолетов, добились максимально возможной точности как показано на рисунке 13.



Р и с. 13. Результаты попиксельного определения местонахождения самолета
F i g. 13. Results of pixel-by-pixel aircraft positioning



Р и с. 14. Результат наложения маски на изображение самолета
F i g. 14. The result of masking aircraft images



Р и с. 15. Точность нейронной сети на обучающих и тестовых данных
F i g. 15. Neural network accuracy on training and test data



Вывод

Фаза обучения заняла около 5 минут в нашем случае с ПК, оснащенным процессором Intel Core i5 2,6 ГГц с 8 ГБ памяти и графической картой Intel NVIDIA GEFORCE 940MX CORE i5 с 64 МБ ОЗУ. После завершения на 35 эпохи обучения и построения нашей сети мы получили точность 98,39% на нашем тестовом наборе (они могут различаться в зависимости от используемого оборудования) как показаны графики на рисунки 15 и 16.

Заключение

В работе был предложен улучшенный метод сегментации изображений, основанный на автокодировщике (специальной архитектуре искусственных нейронных сетей, позволяющей применять обучение без учителя при использовании метода обратного распространения ошибки), позволяющий сжимать

любое большое изображение на мелкое (с теми же свойствами, как входное). Этот метод позволяет также сэкономить ресурсы. При этом, мы используем метода обработки изображения под названием «MaxPooling». В практической части работы, мы написали разные варианты модели U-net (одна из архитектур для сегментации), создавали метрику для отслеживания результата тренировки сети, а также попиксельно определили местонахождение самолета. Мы провели эксперименты для проверки эффективности и точности работы собственной метрики dice_coef. Полученные результаты показывают, что нейронная модель U-Net работает – это эффективная модель сегментации изображений, которая позволяет получить хорошие и точные результаты сегментации любых изображений, но у нее есть недостатки, такие как проблема определения границы и сложность с многоклассовой сегментацией, на которых нужно обратить внимание при реализации этой модели.

References

- [1] Girshick R., Donahue J., Darrell T., Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA: IEEE Computer Society; 2014. p. 580-587. doi: <https://doi.org/10.1109/CVPR.2014.81>
- [2] Panetto H., Cecil J. Information systems for enterprise integration, interoperability and networking: theory and applications. *Enterprise Information Systems*. 2013;7(1):1-6. doi: <https://doi.org/10.1080/17517575.2012.684802>
- [3] Poniszewska-Maranda A. Management of access control in information system based on role concept. *Scalable Computing: Practice and Experience*. 2011;12(1):35-49. Available at: <https://www.scpe.org/index.php/scpe/article/view/686> (accessed 19.10.2022).
- [4] Maturana D., Mery D., Soto Á. Face Recognition with Local Binary Patterns, Spatial Pyramid Histograms and Naive Bayes Nearest Neighbor Classification. In: 2009 International Conference of the Chilean Computer Science Society. Santiago, Chile: IEEE Computer Society; 2009. p. 125-132. doi: <https://doi.org/10.1109/SCCC.2009.21>
- [5] Otsu N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*. 1979;9(1):62-66. doi: <https://doi.org/10.1109/TSMC.1979.4310076>
- [6] Pattanayak S. Advanced Neural Networks. In: Pro Deep Learning with TensorFlow 2.0. Berkeley, CA: Apress; 2023. p. 511-634. doi: https://doi.org/10.1007/978-1-4842-8931-0_6
- [7] LeCun Y., Bengio Y. Convolutional networks for images, speech, and time series. In: The handbook of brain theory and neural networks. Cambridge, MA, USA: MIT Press; 1998. p. 255-258.
- [8] LeCun Y., Bottou L., Orr G.B., Müller K.R. Efficient BackProp. In: Orr G.B., Müller K.R. (eds.) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science. Vol. 1524. Berlin, Heidelberg: Springer; 1998. p. 9-50. doi: https://doi.org/10.1007/3-540-49430-8_2
- [9] Mouale M.N.B., Kozyrev D.V., Houankpo H.G.K., Nibasumba E. Development of a Neural Network Method in the Problem of Classification and Image Recognition. *Modern Information Technologies and IT-Education*. 2021;17(3):507-518. (In Russ., abstract in Eng.) doi: <https://doi.org/10.25559/SITITO.17.202103.507-518>
- [10] Mouale M.N.B. Using a Pre-Trained Neural Network (VGG 16) to Solve the Image Style Transfer Problem. *Modern Information Technologies and IT-Education*. 2022;18(2):241-248. (In Russ., abstract in Eng.) doi: <https://doi.org/10.25559/SITITO.18.202202.241-248>
- [11] Ciresan D.C., Gambardella L.M., Giusti A., Schmidhuber J. Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. In: Pereira F., Burges C.J., Bottou L., Weinberger K.Q. (eds.) Advances in Neural Information Processing Systems (NIPS 2012). Vol. 25. Curran Associates, Inc.; 2012. p. 1-9. Available at: https://proceedings.neurips.cc/paper_files/paper/2012/file/459a4ddcb586f24efd9395aa7662bc7c-Paper.pdf (accessed 19.10.2022).
- [12] Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science. Vol. 9351. Cham: Springer; 2015. p. 234-241. doi: https://doi.org/10.1007/978-3-319-24574-4_28
- [13] Long J., Shelhamer E., Darrell T. Fully convolutional networks for semantic segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA: IEEE Computer Society; 2015. p. 3431-3440. doi: <https://doi.org/10.1109/CVPR.2015.7298965>
- [14] Badrinarayanan V., Kendall A., Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017;39(12):2481-2495. doi: <https://doi.org/10.1109/TPAMI.2016.2644615>



- [15] Araújo R.L., Araújo F.H.D.d., Silva R.R.V.e. Automatic segmentation of melanoma skin cancer using transfer learning and fine-tuning. *Multimedia Systems*. 2022;28(4):1239-1250. doi: <https://doi.org/10.1007/s00530-021-00840-3>
- [16] He K., Zhang X., Ren S., Sun J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2015;37(9):1904-1916. doi: <https://doi.org/10.1109/TPAMI.2015.2389824>
- [17] Zhao H., Shi J., Qi X., Wang X., Jia J. Pyramid Scene Parsing Network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA: IEEE Computer Society; 2017. p. 6230-6239. doi: <https://doi.org/10.1109/CVPR.2017.660>
- [18] Vincent P., Larochelle H., Lajoie I., Bengio Y., Manzagol P.-A. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research*. 2010;11(110):3371-3408. Available at: <http://jmlr.org/papers/v11/vincent10a.html> (accessed 19.10.2022).
- [19] Rifai S., Vincent P., Muller X., Glorot X., Bengio Y. Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. In: Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML'11). Omnipress, Madison, WI, USA; 2011. p. 833-840. Available at: https://icml.cc/2011/papers/455_icmlpaper.pdf (accessed 19.10.2022).
- [20] Solovyeva E., Abdullah A. Dual Autoencoder Network with Separable Convolutional Layers for Denoising and Deblurring Images. *Journal of Imaging*. 2022;8(9):250. doi: <https://doi.org/10.3390/jimaging8090250>
- [21] Li S.Z., Jain A.K. Handbook of Face Recognition. New York, NY: Springer; 2005. 398 p. doi: <https://doi.org/10.1007/b138828>
- [22] Erhan D., Szegedy C., Toshev A., Anguelov D. Scalable Object Detection Using Deep Neural Networks. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA: IEEE Computer Society; 2014. p. 2155-2162. doi: <https://doi.org/10.1109/CVPR.2014.276>
- [23] He K., Zhang X., Ren S., Sun J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2015;37(9):1904-1916. doi: <https://doi.org/10.1109/TPAMI.2015.2389824>
- [24] Shan C., Gong S., McOwan P.W. Facial expression based on Local Binary Patterns: A comprehensive study. *Image and Vision Computing*. 2009;27(6):803-816. doi: <https://doi.org/10.1016/j.imavis.2008.08.005>
- [25] Arulogun O.T., Omidiora E.O., Olaniyi O.M., Ipadeola A.A. Development of Security System using Facial Recognition. *The Pacific Journal of Science and Technology*. 2008;9(2):377-385.
- [26] Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2017;60(6):84-90. doi: <https://doi.org/10.1145/3065386>

Поступила 19.10.2022; одобрена после рецензирования 30.11.2022; принята к публикации 06.12.2022.

Submitted 19.10.2022; approved after reviewing 30.11.2022; accepted for publication 06.12.2022.

Об авторе:

Муаль Мутуама Нда Бьенвеню, аспирант кафедры прикладной информатики и теории вероятностей факультета физико-математических и естественных наук, ФГАОУ ВО «Российский университет дружбы народов» (117198, Российская Федерация, г. Москва, ул. Миклухо-Маклая, д. 6), **ORCID:** <https://orcid.org/0000-0002-7230-5714>, bmouale@mail.ru

Автор прочитал и одобрил окончательный вариант рукописи.

About the author:

Moutouama N'dah B. Mouale, Postgraduate Student of the Department of Applied Probability and Informatics, Faculty of Science, Peoples' Friendship University of Russia (6 Miklukho-Maklaya St., Moscow 117198, Russian Federation), **ORCID:** <https://orcid.org/0000-0002-7230-5714>, bmouale@mail.ru

The author has read and approved the final manuscript.

