DOI 10.37882/2223–2966.2022.11.22

ИЗМЕНЕНИЕ РАЗМЕРОВ ИЗОБРАЖЕНИЙ ДЛЯ УЛУЧШЕНИЯ КЛАСТЕРИЗАЦИИ

RESIZING IMAGES TO IMPROVE CLUSTERING

M. Markeev

Summary. The purpose of this article is to investigate the dependence of the accuracy of neural networks (image clustering) on the size and proportions of images at the input of the neural network itself. Modern neural networks are used for image recognition, and they do it with great accuracy, sometimes even more accurately than humans. The problem is that the images themselves are not perfect. To improve the quality of recognition, the same image is recognized in different scales, rotations and mirroring. The work of this technique was tested on the partitioning of images into 2 clusters "cats" and "dogs". Studies have shown that the best results are obtained by zooming in the image by 30% at a height of 286–346 pixels and a width of 272–383 pixels for the convolution neural network, which was trained on a size of 224 x 224. The results may be different on different data sets, so calibration is required in each case.

Keywords: neural networks, clustering, image resizing, artificial intelligence, Keras, TensorFlow.

Маркеев Максим Валерьевич
Независимый исследователь
Нижегородская область
Нижний Новгород
mmarkeev@gmail.com

Аннотация. Целью данной статьи является проведение исследований зависимости точности работы нейросети (кластеризация изображений) от размеров и пропорций изображений на входе самой нейросети. Современные нейросети используются для распознавания изображений, при этом делают это с большой точностью, иногда даже более точно, чем люди. Проблема заключается в не идеальности самих изображений. Для улучшения качества распознавания производится распознавание одного и того же изображения в различных масштабах, поворотах и зеркальном отображения. Работа этой методики была апробирована на разбиения изображений на 2 кластера «кошки» и «собаки». Проведенные исследования показали, что наилучшие результаты получаются при увеличении изображения на 30% при высоте 286–346 пикселей и ширине 272–383 пикселя для сверточной нейросети, которая была обучена на размерах 224 x 224. Результаты могут быть разными на разных наборах данных, поэтому в каждом случае необходима калибровка.

Ключевые слова: нейронные сети, кластеризация, изменение размеров изображений, искусственный интеллект, Keras, TensorFlow.

Введение

Современные нейросети показывают результаты распознавания изображений лучше, чем эксперты-люди. [1–2] Однако качество распознавания все равно далеко от идеала и может быть улучшено. В данной статье предлагаются методики повышения качества распознавания изображений.

Основная проблема заключается в не идеальности самих данных (изображений), а не в нейросетях [3–4]. На некоторых изображениях распознаваемый объект может быть очень маленьким (например, занимать всего 10% от изображений), на других наоборот, слишком крупный и не помещаться целиком, на третьих, объект может быть растянут по вертикали или горизонтали или сдвинут в любую сторону, на некоторых может быть несколько объектов, а на некоторых изображениях искомого объекта может и вообще не быть. Эти проблемы приводят к тому, что обученные сверточные нейронные сети не могут найти искомые признаки на изображении и, как следствие, неверное его классифицируют [5].

Обычно самым лучшим решением данной проблемы будет правильная подготовка данных. Вручную, либо специальными сетями детекторами объектов, сначала на изображении находятся объекты и координаты прямоугольников, в которые они попадают. Затем обнаруженные в прямоугольниках объекты уже подаются на распознавание обученной сверточной нейронной сети. Но если такое распознавание объектов недоступно, то существуют другие методы повышений качества распознавания изображений. Именно они рассматриваются в данной статье.

Основная идея заключается в том, что на вход сверточной нейросети мы подаем не оригинальное изображение, а модифицированное изображение, например, большего размера, чем тот на который обучена нейросеть [6–7]. Зачастую качество распознавания меняется.

Глобально все эти методы называются дословно «Увеличение времени тестирования», в русскоязычной среде скорее всего встретиться термин «**аугментация**» или Test Time Augmentation (TTA) [8]. Мы подаем на распознавание картинку не 1 раз, а несколько раз в разных

Таблица 1. Необходимое разрешение для изображений на вход нейросети

Модель	Разрешение изображения
EfficientNetB0	224
EfficientNetB1	240
EfficientNetB2	260
EfficientNetB3	300
EfficientNetB4	380
EfficientNetB5	456
EfficientNetB6	528
EfficientNetB7	600

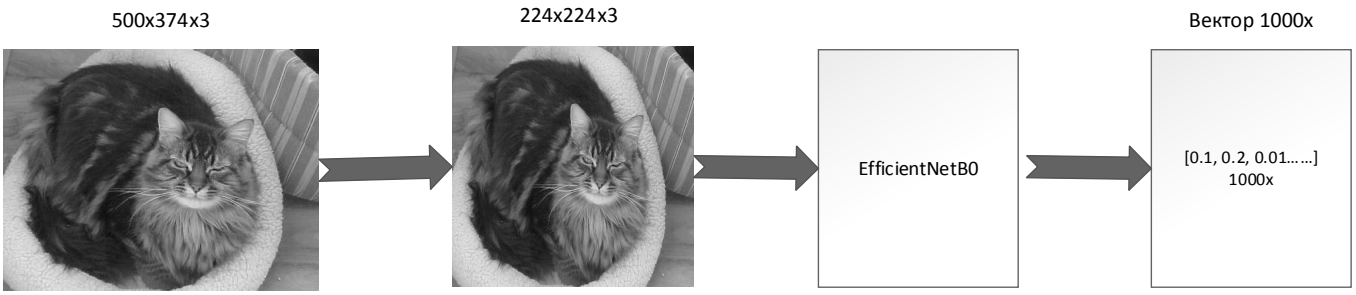

Рис. 1. Пример работы нейросети EfficientNetB0

модификациях (больше размер, меньше размер, поворот, зеркальное отображение, сдвиги, фильтры и т.д.). Это и есть аугментация. Эта техника позволяет увеличивать количество изображений для обучения нейросети и очень часто применяется. А также, ее можно применять и для распознавания изображений (режим инференс). Например, распознать оригинал и зеркальное отображение, а результат распознавания усреднить. В этом случае придется дважды распознавать изображение, что увеличивает вычислительную нагрузку, но качество практически всегда будет выше. Здесь стоит оговориться, что все эти модификации не должны изменить суть объекта. Например, если распознаются рукописные цифры, то не стоит делать поворот на 180 градусов, иначе число 6 превратиться в число 9 и изображение будет неверно интерпретировано нейросетью.

Задача исследования понять, как лучше изменять начальные размеры изображений и соотношение сторон для получения лучшего результата кластеризации.

В исследование для классической задачи классификации кошек и собак применяется датасет из 4005 изображений кошек и 4000 собак, а также, использован язык программирования Python и библиотеки ИИ TensorFlow с оболочкой Keras от компании Google.

Общие принципы работы нейросети

Нейросеть получает на вход набор данных (вектора, матрицы (изображения) или токены (слова)), обрабатывает их и передает на выход (выходы) результат. В этом исследовании используется нейросеть EfficientNet [9], которая обучена на датасете ImageNet [10] (в котором 1000 различных классов картинок) получает на вход изображение в формате 3-й мерной матрицы: Высота x Ширина x 3 Канала (RGB), а на выходе получится вектор размером 1000, который содержит вероятности для каждой из 1000 категорий:

Среди этих категорий названия животных, а для некоторых есть породы, а также вещей, овощей и фруктов.

Чем выше модификация EfficientNet — тем выше точность предсказаний на датасете ImageNet, но и требуется больше ресурсов и большее разрешения изображения.

Модель EfficientNetB0 содержит около 5 млн. параметров и обучена на изображениях 224x224, а EfficientNetB7 около 70 млн. параметров и обучена на изображениях 600x600. см. рисунок 3.

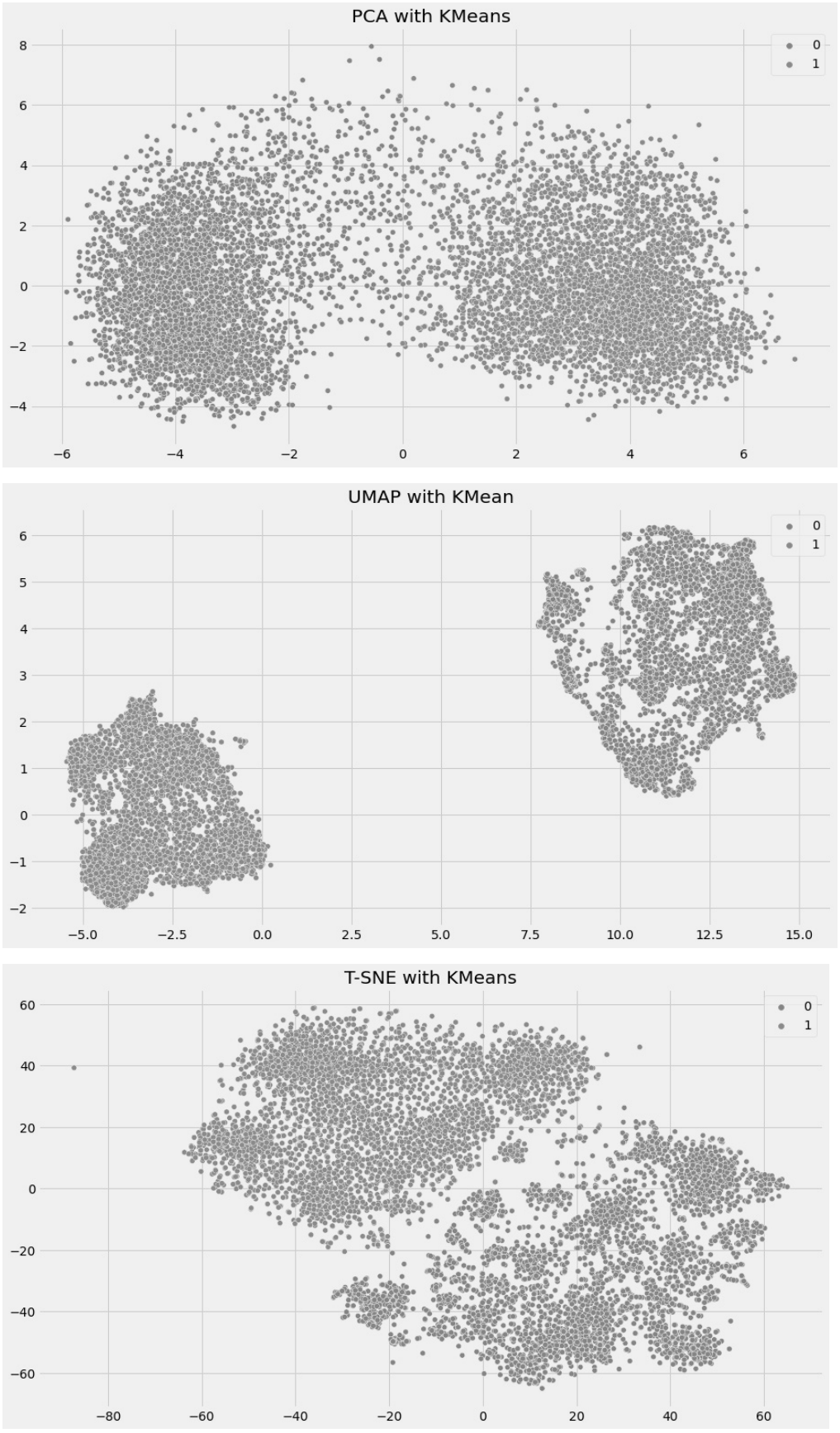

Рис. 2. Алгоритмы снижения размерности PCA, UMAP и T-SNE с разделением данных на 2 кластера

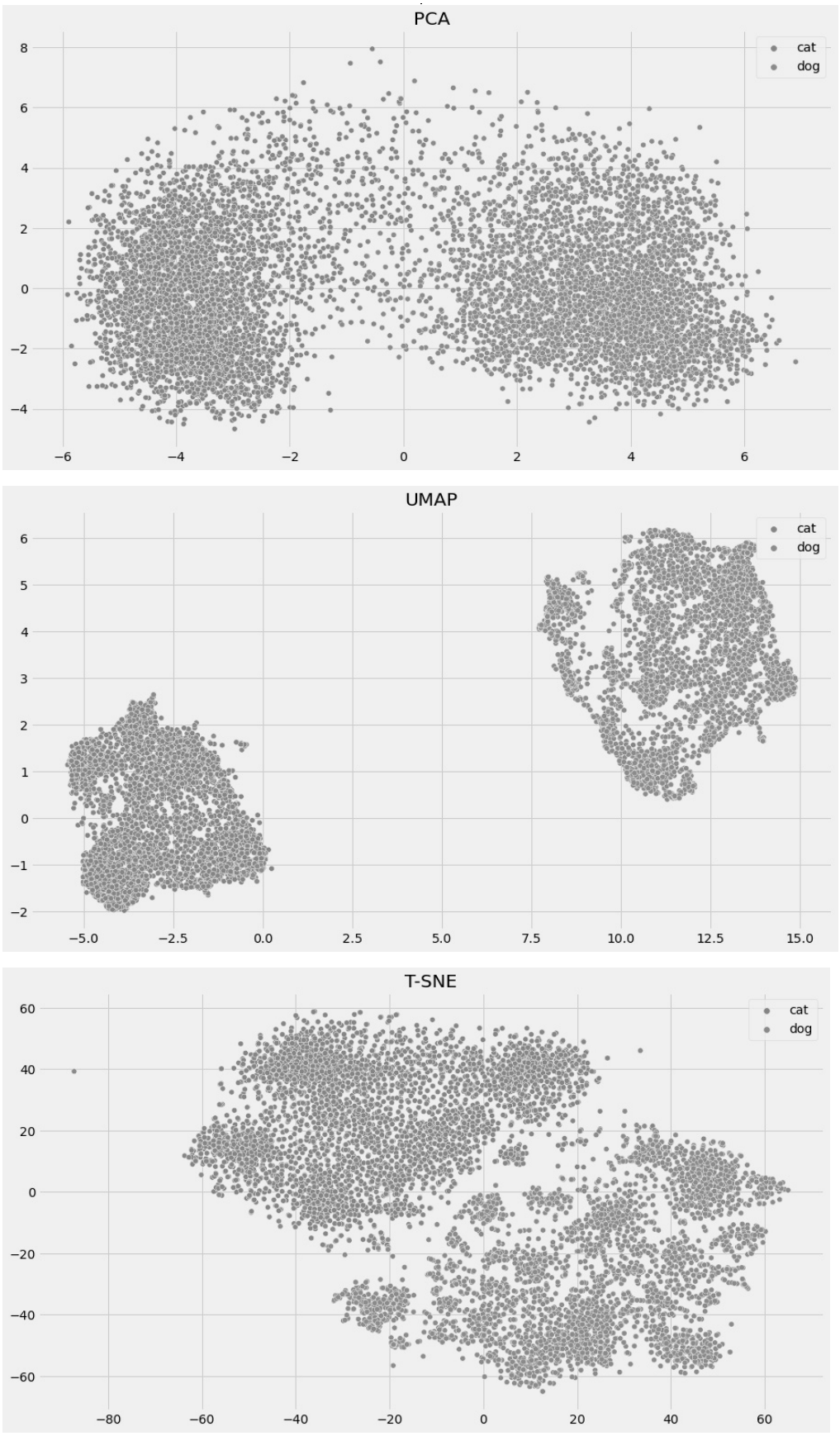

Рис. 3. Результат работы алгоритмов PCA, UMAP и T-SNE кластеризации изображений

Таблица 2. Результаты работы нейросети при разных входных значениях высоты и ширины изображений

Высота	Ширина	Размер	KMeans	PCA	UMAP	t-SNE	Среднее
202	202	90%	0.9818	0.9808	0.9885	0.9878	0.984725
213	213	95%	0.9829	0.9809	0.9893	0.9874	0.985125
224	224	100%	0.9861	0.9845	0.991	0.9886	0.98755
235	235	105%	0.988	0.987	0.9913	0.99	0.989075
247	247	110%	0.9881	0.9855	0.991	0.9883	0.988225
259	259	115%	0.9884	0.9875	0.9928	0.9904	0.989775
272	272	120%	0.9889	0.9873	0.9918	0.9909	0.989725
286	286	125%	0.9891	0.987	0.9939	0.9935	0.990875
300	300	130%	0.9896	0.9879	0.9929	0.9913	0.990425
315	315	135%	0.9898	0.9873	0.9938	0.9928	0.990925
331	331	140%	0.9891	0.988	0.9913	0.9909	0.989825
348	348	145%	0.9893	0.9866	0.9933	0.9898	0.98975
365	365	150%	0.9888	0.9874	0.9915	0.9888	0.989125
383	383	155%	0.9881	0.9864	0.991	0.982	0.986875

Кластеризация изображений

Предлагаемая нейросеть обучена распознавать 1000 классов, но перед ней поставлена задача отделять собак от кошек (всего 2 класса). Здесь понадобятся «Эмбеддинги» (от английского слова Embedding, в русском языке иногда встречается термин «Вложения»).

В отличие от стандартной модели работы здесь нам не нужен выходной слой. Вместо него мы получили внутренний слой (Эмбеддинг) на выходе.

Эмбеддинг — это то, как модель представляет себе суть изображения (сжатое изображение), записанное в виде вектора. У разных моделей размер эмбеддинга может быть разным, а также можно брать не последний выходной слой(N), а более глубокий, например N-1, но обычно используется именно последний слой перед выходным.

Пропустив все 8005 изображений размера 224 x 244 через нейросеть EfficientNetB0 и взяв предпоследний выход (эмбеддинг), получается матрица размером: 8005 x 1280 т.е. для каждого изображения вектор длинной 1280. Далее нужно разделить вектора на 2 кластера — это и будут 2 наших класса (кошки и собаки). Для этого воспользуемся Методом Ближайшего Соседа [11] (KMeans). Он позволяет разделить данные на нужное количество кластеров (2 в нашем случае). В дополнении к этому воспользуемся алгоритмами снижения размерности векторов, такими как PCA [12], UMAP [13] и t-SNE [14] для снижения размерности вектора изображения до 2-х, а далее к полученным векторам опять применим Метод Ближайшего Соседа и сравним качество работы. Снижение размерности до 2-х позволит наглядно посмотреть на графике как тот или иной алгоритм разделяет данные на кластеры. Для метрики используем простую точность (определяется как Количество правильных предсказаний поделить на общее количество предсказаний).

После применения метода кластеризации KMeans получим (см. рис. 2).

На рисунке 2 видно, что метод KMeans разметил данные на 2 кластера (0 и 1).

Также невооруженным взглядом заметно, что все используемые методы выделяют 2 выраженных кластера. В методе UMAP кластеры наиболее ярко выражены.

Следует отметить, что для алгоритма кластеризации важно разделить данные на заданное количество кластеров, при этом какой номер получит какой кластер может отличатся. Следовательно, если наша точность получиться сильно меньше 0.5 (50%), то мы переворачиваем метки кластеров местами (т.к. у нас в датасете примерно равное количество собак и кошек (4005 изображений кошек и 4000 собак), то даже если модель вообще не работает, а случайно угадывает, то точность будет около 0.5).

Визуализируем правильность работы моделей (рис. 3).

Визуально заметно, что модели работают, также видно, что есть ошибки. Сравним точность:

KMeans 0.9861, PCA 0.9845, UMAP 0.9910, T-SNE0.9886

Точность методов от 0.9845 до 0.9910. Максимальная точность у метода UMAP.

Таблица 3. 10 лучших результатов точности при разных значениях высоты и ширины

Высота	Ширина	Средняя Точность
331	383	0.990900
315	315	0.990900
300	365	0.990725
348	315	0.990725
259	315	0.990650
348	331	0.990625
300	286	0.990575
300	331	0.990525
300	315	0.990525
286	286	0.990500

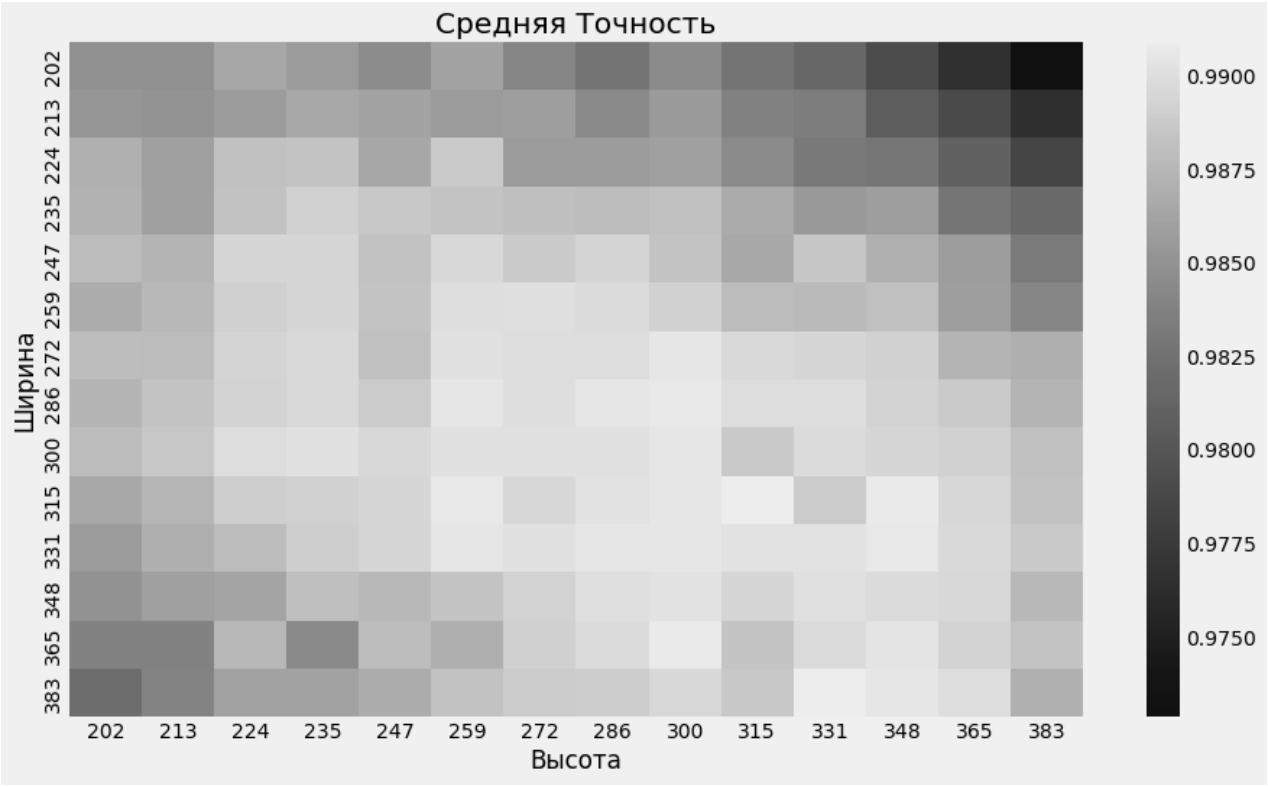

Рис. 4. Средняя точность работы моделей при разных значениях высоты и ширины изображений

Обычно при снижении размерности какая-то часть информации неизбежно теряется, поэтому точность после метода PCA всегда будет чуть хуже, чем KMeans без снижения размерности, однако благодаря этому снижению мы смогли визуализировать результат.

Изменение размеров изображений для улучшения точности работы модели

Модель EfficientNetB0 обучена на изображения 224 x 224. Этот размер мы использовали в примере выше.

Однако в сверхточных нейронных сетях мы можем подавать на вход изображения любого размера (необязательно квадратной формы), качество работы моделей при этом меняется. В данном исследовании предполагается подавать на вход нейросети изображения разных размеров и оценить итоговую точность работы.

Размеры увеличиваются и уменьшаются с шагом 5%

Как видно из таблицы лучшие результаты достигаются при увеличении изображения на 30% как по высоте, так и по ширине, т.е. при размерах 300 x 300, хотя изначально модель EfficientNetB0 обучена на изображения 224 x 224.

Проведение дальнейших экспериментов с различными значениями высоты и ширины дал результат:

Оптимальные результаты получаются при высоте 286–348 и ширине 272–383, что довольно далеко от ожидаемых.

Лучшие результаты представлены в таблице 3.

Нейронные сети могут использоваться для кластеризации изображений. Чтобы повысить качество работы нейросети зачастую следует изменять размеры изображений, подаваемых на вход. Оптимальные размеры могут существенно отличаться от тех, на которых нейросеть была обучена. В данном исследовании для получения лучшего результата размеры изображений были увеличены примерно на 30%. Результаты могут быть разными для разных датасетов, поэтому для увеличения точности следует произвести подобную калибровку на конкретном датасете.

ЛИТЕРАТУРА

1. Лядова Е.Ф. Перспективные сервисы на основе технологий искусственного интеллекта и виртуальной реальности // Славянский форум. 2021. № 1 (31). С. 29–40
2. Мамадаев И.М., Минитаева А.М. Анализ способов распознавания достопримечательностей на фотографиях // Славянский форум. 2022. № 1 (35). С. 357–371.
3. Байгутлина И.А., Замятин П.А. Решение задач пространственного анализа с использованием нейропроцессоров российского производства //Славянский форум. 2022. № 1 (35). С. 301–313.
4. Барладян Б.Х., Шапиро Л.З., Маллачиев К.А., Хорошилов А.В., Солоделов Ю.А., Волобой А.Г., Галактионов В.А., Ковернинский И.В. Система визуализации для авиационной ОС реального времени JetOS. Труды ИСП РАН, том 32, вып. 1, 2020 г., стр. 57–70.
5. Сикорский О.С. Обзор свёрточных нейронных сетей для задачи классификации изображений //Новые информационные технологии в автоматизированных системах.— 2017.— № . 20.— С. 37–42.
6. Шайтура Н.С. Визуализации трехмерных сцен // Славянский форум.—2022.— № 3 (37) — с. 312–325.
7. Система генерации наборов изображений для задач компьютерного зрения на основе фотореалистичного рендеринга / В.В. Санжаров [и др.] // Препринты ИПМ им. М.В. Келдыша. 2020. № 80. 29 с.
8. Kandel I., Castelli M. Improving convolutional neural networks performance for image classification using test time augmentation: a case study using MURA dataset //Health information science and systems.— 2021.— Т. 9.— № . 1.— С. 1–22.
9. Tan M., Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks //International conference on machine learning.— PMLR, 2019.— С. 6105–6114.
10. Deng J. et al. Imagenet: A large-scale hierarchical image database //2009 IEEE conference on computer vision and pattern recognition.— Ieee, 2009.— С. 248–255.
11. Харитонов С.П. Метод" ближайшего соседа" для математической оценки распределения биологических объектов на плоскости и на линии //Вестник Нижегородского университета им. НИ Лобачевского. Серия: Биология.— 2005.— № . 1.— С. 213–221.
12. Yang J. et al. Two-dimensional PCA: a new approach to appearance-based face representation and recognition //IEEE transactions on pattern analysis and machine intelligence.— 2004.— Т. 26.— № . 1.— С. 131–137.
13. McInnes L., Healy J., Melville J. Umap: Uniform manifold approximation and projection for dimension reduction //arXiv preprint arXiv:1802.03426.— 2018.
14. Van der Maaten L., Hinton G. Visualizing data using t-SNE //Journal of machine learning research.— 2008.— Т. 9.— № . 11

Журнал «Современная наука: актуальные проблемы теории и практики»