Intel OpenVINO™ Toolkit: анализ производительности выполнения генеративно-сопоставительных нейронных сетей

В.В. Зунин, А.Ю. Романов

Национальный исследовательский университет «Высшая школа экономики», г. Москва,
vzunin@hse.ru, a.romanov@hse.ru

Аннотация — В работе проводится обзор современного состояния реализации нейронных сетей, их способов выполнения, а также инструментария Intel® OpenVINO™ Toolkit для выполнения нейронных сетей на различных аппаратных платформах компании Intel (CPU, GPU, Neural Compute Stick 2). Описан выбор различных генеративно-сопоставительных нейронных сетей и наборов данных для их тестирования, а также дано описание эксперимента на аппаратной платформе. Дается анализ производительности и себестоимости выполнения генеративно-сопоставительных нейронных сетей с использованием инструментария OpenVINO™ Toolkit в обычном режиме, а также с использованием Multi-Device Plugin и Heterogeneous Plugin на несколько подключенных устройствах.

Ключевые слова — Intel, OpenVINO, GAN, нейронные сети.

I. ВВЕДЕНИЕ

Искусственные нейронные сети приобрели широкое распространение для решения множества различных задач. Они применяются как для распознавания изображений, так и для генерации фотореалистичных изображений. Последняя область применения использует генеративно-сопоставительные нейронные сети, которые были впервые описаны в статье [1].

Генеративно-сопоставительная нейронная сеть (GAN) [2] является алгоритмом машинного обучения, который использует комбинацию из двух нейронных сетей. Одна нейронная сеть генерирует образцы, которые другая нейронная сеть (дискриминативная) должна отличить от подлинных. Таким образом две нейронные сети со временем обучаются, и генеративная нейронная сеть может генерировать образцы похожие на примеры, использованные для обучения дискриминативной нейронной сети. Так как образцы могут быть различными, то GAN могут использоваться как для генерации реалистичных лиц, пейзажей или других изображений, похожих на фотографии нашего мира, так и для улучшения качества испорченных фотографий. Таким образом генеративно-сопоставительные нейронные сети имеют широкую область применения и являются перспективным направлением для исследований. При этом для выполнения GAN, как и любых других нейронных сетей, необходимо обоснование выбора

аппаратной платформы для обеспечения заданных критериев производительности.

Различные аппаратные платформы могут использовать различные способы запуска нейронной сети, поэтому разработчику требуется разработать программное обеспечение для каждой платформы, провести анализ производительности выполнения нейронной сети на устройствах и выбрать подходящую. Использование Intel OpenVINO™ Toolkit позволяет провести тестирования на различном оборудовании компании Intel без необходимости создания дополнительных программ, что уменьшает необходимое разработчику для тестирования время. В работе представлены результаты тестирования различных генеративно-сопоставительных нейронных сетей, а также продемонстрированы варианты выбора наиболее выгодных комбинаций использования устройств из представленных.

II. INTEL OPENVINO TOOLKIT

В 2018 году компания Intel представила комплексный программный пакет Intel OpenVINO™ Toolkit [3-4], основанный на сверточных нейронных сетях. Основным компонентом OpenVINO Toolkit является Deep Learning Deployment Toolkit. Он включает в себя: Deep Learning Model Optimizer (MO); Deep Learning Inference Engine (IE); вспомогательные программные инструменты; демонстративные примеры.

Model Optimizer выполняет анализ обученных моделей нейронных сетей поддерживаемого формата (TensorFlow, Caffe, ONNX и др.), настраивает и преобразует их для оптимального выполнения на целевых устройствах. Таким образом генерируется промежуточное представление (Intermediate Representation, IR), которое сохраняется в специальный формат для последующего использования при запуске.

Inference Engine выполняет инференс модели в промежуточном представлении на заданном устройстве (CPU, Integrated Graphics, Movidius Neural Compute Stick, Neural Compute Stick 2 [5], Vision Accelerator Design with Intel® Movidius™ Vision Processing Unit (VPU), FPGA). При наличии на заданном устройстве нескольких поддерживаемых вычислительных устройств можно реализовать автоматическое разделение выполнения нейронной

сети между устройствами (Heterogeneous Plugin) или запуск одной сети на нескольких устройствах параллельно (Multi-Device Plugin). На рис. 1 представлена схема стандартного процесса развертывания обученной модели с использованием OpenVINO.

Рис. 1. Схема развертывания обученной модели с использованием OpenVINO [6]

Представленный инструментарий является новым, но при этом вызвал большой интерес у исследователей. Так, в работе [7] проведен анализ ускорения выполнения нейронной сети на процессоре Intel для SSD_MobileNet_V2 [8-9] с использованием изображений, выбранных из набора данных COCO [10-11]. Результаты этого анализа показали, что для выбранной нейронной сети средний выигрыш с использованием инструментария OpenVINO составляет 126,45 раза. В то же время он не обеспечивает стабильного времени обработки различных видов изображений. Тем не менее увеличение производительности на 2 порядка за счет использования инструментария OpenVINO делает его привлекательным решением.

В другой работе [12] авторы выбрали 14 моделей нейронных сетей для классификации изображений и провели анализ производительности с использованием OpenVINO и выполнения на встроенном в процессор графическом ядре (GPU). Результаты эксперимента показали, что модели с точностью FP32 на встроенном в процессор графическом ядре выполняются в среднем в 1,5 раза быстрее, чем на самом 4-х ядерном процессоре Intel Xeon.

Таким образом, несмотря на то, что инструментарий новый, он обладает высоким потенциалом для ускорения работы нейронных сетей, что определяет актуальность исследований его использования для выполнения различных видов нейронных сетей, в том числе генеративно-состязательных.

III. ЭКСПЕРИМЕНТ

A. Выбор данных и аппаратных платформ

Для проведения тестирования генеративно-состязательных нейронных сетей были выбраны следующие нейронные сети [13]: GAN MNIST [14] (позволяет генерировать рукописные цифры из случайного шума), группа сетей Candy [15], Mosaic [15], Pointilism [15], Rain Princess [15], Udnie [15] (производят преобразование исходного изображения в новое с различным стилем

отображения), а также Super Resolution [16] (позволяет повысить разрешение входного изображения).

В каждой из представленных генеративно-состязательных нейронных сетей была выбрана та, которая отвечает за генерацию новых изображений и было проведено тестирование с использованием одной тысячи фотографий из базы данных COCO 2017 Test images [10] или генерируемого шума.

В качестве аппаратных платформ для реализации сетей были выбраны компьютеры с операционной системой Windows 10 (64 бит). Целевые аппаратные платформы, на которых производился запуск нейронных сетей:

- 1) CPU1: Intel® Core™ i9-9900KF;
- 2) CPU2: Intel® Core™ i7-7700HQ;
- 3) CPU3: Intel® Core™ i5-8250U;
- 4) GPU1: Intel® HD Graphics 630;
- 5) GPU2: Intel® UHD Graphics 620;
- 6) VPU: Neural Compute Stick 2.

GPU1 и GPU2 являются встроенными графическими процессорами в процессоры CPU2 и CPU3 соответственно.

B. Проведение эксперимента

Для анализа производительности была разработана программа, которая использует инструментарий OpenVINO версии 2020.3, а также представленные нейронные сети, их параметры и входные изображения для генерации исполняемых файлов, позволяющих запускать тестирование нейронных сетей на всех поддерживаемых и доступных аппаратных платформах с точностью вычисления FP32. Полученные с использованием программы исполняемые файлы позволяют произвести тестирование нейронной сети на любом компьютере с операционной системой, используемой для генерации файлов на всех доступных аппаратных платформах. Для обеспечения точности тестирования, компьютер, на котором запускается исполняемый файл, загружался с минимальным количеством фоновых процессов, влияющих на производительность выполнения нейронной сети.

В результате выполнения исполняемых файлов создаются файлы с результатами, в которых содержатся значения производительности выполнения нейронной сети для каждого доступного на компьютере и поддерживаемого инструментарием устройства как в одиночном режиме, так и с использованием Multi-Device Plugin, устанавливая единичный размер батча. Результат тестирования производительности выполнения нейронной сети является медианным значением количества обработанных кадров (изображений) в секунду (FPS).

Таблица 1

Результат тестирования для CPU1 и VPU

Нейронная сеть	Производительность в FPS при выбранном типе запуска						
	ONNX CPU1	CPU1	VPU	MULTI CPU1, VPU	MULTI VPU, CPU1	HETERO CPU1, VPU	HETERO VPU, CPU1
GAN MNIST	64,01	79,20	16,00	79,16	16,00	79,16	16,00
Candy	21,34	21,34	2,56	21,34	2,56	21,34	2,56
Mosaic	21,34	21,34	2,56	21,34	2,56	21,34	2,56
Udnie	21,34	21,34	2,56	21,34	2,56	21,34	2,56
Rain_Princess	21,34	21,34	2,56	21,34	2,56	21,34	2,56
Pointilism	21,34	21,34	2,56	21,34	2,56	21,34	2,56
Super Resolution	64,01	101,99	10,67	63,98	10,67	63,98	10,67

Таблица 2

Результаты тестирования для CPU2 и GPU1

Нейронная сеть	Производительность в FPS при выбранном типе запуска						
	ONNX CPU2	CPU2	GPU1	MULTI CPU2, GPU1	MULTI GPU1, CPU2	HETERO CPU2, GPU1	HETERO GPU1, CPU2
GAN MNIST	501,41	504,00	100,28	502,19	100,27	501,35	100,45
Candy	8,09	6,87	13,19	6,39	12,85	4,82	12,53
Mosaic	5,51	4,23	12,23	3,71	12,08	4,84	12,23
Udnie	7,54	6,55	12,70	6,55	12,69	7,71	12,70
Rain_Princess	7,54	6,60	12,69	6,55	12,69	7,71	12,69
Pointilism	7,48	6,51	12,69	6,55	12,69	7,71	12,69
Super Resolution	29,49	45,57	50,12	45,29	47,75	47,75	50,05

Таблица 3

Результаты тестирования для CPU3 и GPU2

Нейронная сеть	Производительность в FPS при выбранном типе запуска						
	ONNX CPU3	CPU3	GPU2	MULTI CPU3, GPU2	MULTI GPU2, CPU3	HETERO CPU3, GPU2	HETERO GPU2, CPU3
GAN MNIST	501,29	811,43	110,97	501,47	100,64	501,41	100,82
Candy	6,77	6,23	12,85	6,23	12,85	7,01	12,85
Mosaic	6,77	6,27	12,85	6,23	12,85	7,01	12,85
Udnie	6,73	6,27	12,85	6,26	12,85	7,06	12,85
Rain_Princess	6,77	6,23	12,85	6,23	12,85	7,06	12,85
Pointilism	6,77	6,23	12,85	6,23	12,85	7,06	12,85
Super Resolution	27,10	45,57	50,13	32,47	47,68	43,59	50,13

Результаты тестирования для CPU4

Нейронная сеть	Производительность в FPS при выбранном типе запуска	
	ONNX CPU4	CPU4
GAN MNIST	500,16	500,04
Candy	9,52	5,49
Mosaic	8,26	5,26
Udnie	9,01	5,38
Rain_Princess	8,47	5,35
Pointilism	8,85	5,38
Super Resolution	26,67	43,83

Рис. 2. Гистограммы распределения производительности нейронных сетей на различных аппаратных платформах

С. Результаты экспериментов

В табл. 1–4 приведены полученные результаты тестирования производительности выполнения различных нейронных сетей на выбранных аппаратных платформах.

1) Сравнение производительности выполнения нейронных сетей на различных аппаратных платформах

Для каждой нейронной сети было проведено сравнение производительности выполнения (рис. 2). Гистограммы построены для каждой аппаратной платформы и представлены в логарифмическом масштабе, т. к. он удобен для представления больших диапазонов величин.

Обобщая и анализируя полученные данные, тестируемые нейронные сети можно разделить на два основных класса.

В первом классе нейронных сетей содержатся Candy, Mosaic, Udnie, Rain_Princess и Pointilism. Производительность выполнения этих нейронных сетей на центральном процессоре с использованием инструментария OpenVINO уступает запуску с использованием стандартной библиотеки (на 11,5 %), но сравнивая производительность выполнения на встроенном графическом процессоре с использованием инструментария OpenVINO среднее увеличение производительности достигает 84 %.

Рассматривая возможность выполнения нейронной сети на нескольких устройствах с использованием

Multi-Device Plugin, при сравнении производительности выполнения с запуском нейронной сети с использованием инструментария на одиночном устройстве, используемом в плагине в качестве основного, производительность относительно использования OpenVINO с теми же устройствами отдельно не увеличивается, а снижается на 0–12 %. С другой стороны, для Heterogeneous Plugin результаты не однозначны. Показатели колеблются как в зависимости от использованного центрального и графического процессоров, так и от тестируемой нейронной сети.

Во втором классе нейронных сетей, GAN MNIST и Super Resolution, производительность выполнения нейронных сетей с использованием инструментария OpenVINO не ухудшается. Но при этом использование

Multi-Device Plugin и Heterogeneous Plugin не увеличивает производительность.

Таким образом, для одних нейронных сетей использование инструментария OpenVINO целесообразно для увеличения показателей производительности, а для других нейронных сетей прирост производительности не наблюдается.

2) Сравнение показателей себестоимости выполнения различных нейронных сетей на различном аппаратном обеспечении

В дополнении к стандартному сравнению производительности был проведен анализ показателей себестоимости (FPS/\$), характеризующий выгоду от использования выбранного устройства (табл. 5–8).

Таблица 5

Сравнение себестоимости для CPU1 и VPU

Нейронная сеть	Тип запуска						
	ONNX CPU1	CPU1	VPU	MULTI CPU1, VPU	MULTI VPU, CPU1	HETERO CPU1, VPU	HETERO VPU, CPU1
GAN MNIST	0,0	23,7	67,7	7,6	67,7	7,6	67,7
Candy	0,0	0,0	-19,5	-13,0	-19,5	-13,0	-19,5
Mosaic	0,0	0,0	-19,5	-13,0	-19,5	-13,0	-19,5
Udnie	0,0	0,0	-19,5	-13,0	-19,5	-13,0	-19,5
Rain_Princess	0,0	0,0	-19,5	-13,0	-19,5	-13,0	-19,5
Pointilism	0,0	0,0	-19,5	-13,0	-19,5	-13,0	-19,5
Super Resolution	0,0	38,7	11,8	-13,0	11,8	-13,0	11,8

Таблица 6

Сравнение себестоимости для CPU2 и GPU1

Нейронная сеть	Тип запуска						
	ONNX CPU2	CPU2	GPU1	MULTI CPU2, GPU1	MULTI GPU1, CPU2	HETERO CPU2, GPU1	HETERO GPU1, CPU2
GAN MNIST	0,0	0,5	-80,0	0,2	-80,0	0,0	-80,0
Candy	0,0	-15,1	63,2	-21,0	59,0	-40,4	55,0
Mosaic	0,0	-23,2	122,0	-32,6	119,3	-12,1	122,0
Udnie	0,0	-13,1	68,4	-13,1	68,4	2,3	68,4
Rain_Princess	0,0	-12,5	68,4	-13,1	68,3	2,3	68,3
Pointilism	0,0	-13,0	69,6	-12,4	69,6	3,1	69,6
Super Resolution	0,0	54,5	70,0	53,6	61,9	61,9	69,7

Таблица 7

Сравнение себестоимости для CPU3 и GPU2

Нейронная сеть	Тип запуска						
	ONNX CPU3	CPU3	GPU2	MULTI CPU3, GPU2	MULTI GPU2, CPU3	HETERO CPU3, GPU2	HETERO GPU2, CPU3
GAN MNIST	0,0	61,9	-77,9	0,0	-79,9	0,0	-79,9
Candy	0,0	-8,1	89,6	-8,1	89,6	3,5	89,7
Mosaic	0,0	-7,5	89,7	-8,1	89,7	3,5	89,7
Udnie	0,0	-6,9	90,9	-7,0	90,9	4,9	91,0
Rain_Princess	0,0	-8,1	89,7	-8,1	89,7	4,2	89,7
Pointilism	0,0	-8,1	89,7	-8,1	89,7	4,2	89,7
Super Resolution	0,0	68,2	85,0	19,8	75,9	60,9	85,0

Таблица 8

Сравнение себестоимости для CPU4

Нейронная сеть	Тип запуска	
	ONNX CPU4	CPU4
GAN MNIST	0,0	0,0
Candy	0,0	-42,3
Mosaic	0,0	-36,3
Udnie	0,0	-40,3
Rain_Princess	0,0	-36,9
Pointilism	0,0	-39,2
Super Resolution	0,0	64,3

Рассматривая показатели себестоимости VPU, стоит отметить, что для *GAN MNIST* и *Super Resolution* показатели увеличиваются, что отражает целесообразность использования данного устройства в ситуациях, когда требования к производительности меньше, чем к стоимости. Для остальных нейронных сетей использование VPU нецелесообразно, т.к. показатели стоимости уменьшаются.

При выполнении с использованием встроенного графического процессора у нейронных сетей *Candy*, *Mosaic*, *Udnie*, *Rain_Princess*, *Pointilism* и *Super Resolution* показатели себестоимости выше по сравнению со стандартным выполнением. Учитывая, что центральный процессор и графический процессор находятся в одном чипе, то в ситуациях с выполнением нескольких нейронных сетей параллельно целесообразность использования подобных устройств вместе с инструментарием OpenVINO возрастает еще больше.

IV. ЗАКЛЮЧЕНИЕ

Таким образом, в данной работе произведено тестирование семи генеративно-состязательных

нейронных сетей с использованием инструментария OpenVINO. Полученные результаты показывают, что использование инструментария OpenVINO для выполнения нейронных сетей на встроенных графических ядрах для протестированных нейронных сетей, за исключением *GAN MNIST*, увеличивает производительность (до 122 %) и увеличивает себестоимость (до 90 %). При этом нейронная сеть *GAN MNIST*, наоборот, увеличивает производительность при применении инструментария на идентичном устройстве, а при использовании встроенного графического ядра снижает производительность (до 80 %) и себестоимость (до 80 %). Применение же Neural Compute Stick 2 демонстрирует иные результаты. Его использование уменьшает производительность (до 75 %), но при этом увеличивает показатель себестоимости (до 67 %). При этом показано, что использование одновременного выполнения нейронных сетей на нескольких устройствах Intel с использованием Multi-Device Plugin и Heterogeneous Plugin не увеличивает производительность и не улучшает себестоимость выполнения нейронных сетей.

- [1] Illingworth W.T. Beginners guide to neural networks // IEEE Aerospace and Electronic Systems Magazine. 1989. V. 4, № 9. P. 44–49.
- [2] Prabhat, Nishant, Dinesh K. V. Comparative Analysis of Deep Convolutional Generative Adversarial Network and Conditional Generative Adversarial Network using Hand Written Digits // IEEE Proceedings of the International Conference on Intelligent Computing and Control Systems. 2020. P. 1072–1075.
- [3] Deep Learning & Vision Tools. OpenVINO. URL: <https://software.intel.com/content/www/us/en/develop/tools/openvino-toolkit.html> (дата обращения: 25.05.2021).
- [4] Castro-Zunti R.D., Yépez J., Ko S.B. License plate segmentation and recognition system using deep learning and OpenVINO // IEEE IET Intelligent Transport Systems. 2020. V. 14, № 2. P. 119–126.
- [5] Aleksandrova O., Bashkov Y. Face recognition systems based on Neural Compute Stick 2, CPU, GPU comparison // 2nd IEEE International Conference on Advanced Trends in Information Theory (ATIT). 2020. P. 104–107.
- [6] Model Optimizer Developer Guide. URL: https://docs.openvino toolkit.org/latest/openvino_docs_MO_DG_Deep_Learning_Model_Optimizer_DevGuide.html (дата обращения: 25.05.2021).
- [7] Andriyanov N.A. Analysis of the Acceleration of Neural Networks Inference on Intel Processors Based on OpenVINO Toolkit // IEEE Systems of Signal Synchronization, Generating and Processing in Telecommunications. 2020. P. 1–5.
- [8] Zhang F. et al. An expression recognition method on robots based on mobilenet V2-SSD // IEEE 6th International Conference on Systems and Informatics, ICSAI. 2019. P. 118–122.
- [9] Solovyev R.A. et al. Real-Time Recognition of Handwritten Digits in FPGA Based on Neural Network with Fixed Point Calculations // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2019. Вып 4. С. 38–42.
- [10] COCO Dataset URL: <https://cocodataset.org/#download> (дата обращения: 25.05.2021).
- [11] Lin T. Y. et al. Microsoft COCO: Common Objects in Context // Lecture Notes in Computer Science. Springer Verlag, 2014. V. 8693, № 5. P. 740–755.
- [12] Jin Z., Finkel H. Analyzing Deep Learning Model Inferences for Image Classification using OpenVINO // IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). 2020. P. 908–911.
- [13] Васильев Е.П., Рустамов А.З. OpenVINO GAN DEMOS repository URL: <https://github.com/itlab-vision/OpenVINO-GAN-DEMO> (дата обращения: 25.05.2021).
- [14] Тьюриал: создание простой GAN на Python с библиотекой Keras. URL: <https://neurohive.io/ru/tutorial/simple-gan-python-keras/> (дата обращения: 12.05.2021).
- [15] Johnson J., Alahi A., Fei-Fei L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution // European Conference on Computer Vision. 2016. P. 694–711.
- [16] Shi W. et al. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 1874–1883.

Intel OpenVINO™ Toolkit: Performance Analysis of Generative Adversarial Neural Networks

V.V. Zunin, A.Yu. Romanov

HSE University, Moscow

vzunin@hse.ru, a.romanov@hse.ru

Abstract — The paper provides an overview of the current state of the implementation of neural networks and their methods of execution. A detailed description of the main components of the Intel® OpenVINO™ Toolkit for executing neural networks on various Intel hardware platforms (CPU, GPU, Neural Compute Stick 2) is considered. The works in which this toolkit is used are investigated and the conclusions made in them are presented. Before carrying out the experiment, the selection of various generative adversarial neural networks, data sets for testing them, and the process of launching neural networks on selected hardware platforms (four experimental stands with different hardware equipment) and obtaining results are described. Based on the results of the experiments, the obtained data are presented based on two metrics: performance in FPS and the cost of using the tested hardware platform to execute a neural network. Based on the data obtained, the analysis of the performance metrics of

the previously selected generative adversarial neural networks is carried out without the use of tools, using the OpenVINO™ Toolkit in the usual mode, as well as using the Multi-Device Plugin and Heterogeneous Plugin on several connected devices.

Keywords — Intel, OpenVINO, GAN, neural networks.

REFERENCES

- [1] Illingworth W.T. Beginners guide to neural networks // IEEE Aerospace and Electronic Systems Magazine. 1989. V. 4, № 9. P. 44–49.
- [2] Prabhat, Nishant, Dinesh K. V. Comparative Analysis of Deep Convolutional Generative Adversarial Network and Conditional Generative Adversarial Network using Hand Written Digits // IEEE Proceedings of the International Conference on Intelligent Computing and Control Systems. 2020. P. 1072–1075.

- [3] Deep Learning & Vision Tools. OpenVINO. URL: <https://software.intel.com/content/www/us/en/develop/tools/openvino-toolkit.html> (access date: 25.05.2021).
- [4] Castro-Zunti R.D., Yépez J., Ko S.B. License plate segmentation and recognition system using deep learning and OpenVINO // IEEE IET Intelligent Transport Systems. 2020. V. 14, № 2. P. 119–126.
- [5] Aleksandrova O., Bashkov Y. Face recognition systems based on Neural Compute Stick 2, CPU, GPU comparison // 2nd IEEE International Conference on Advanced Trends in Information Theory (ATIT). 2020. P. 104–107.
- [6] Model Optimizer Developer Guide. URL: https://docs.openvino toolkit.org/latest/openvino_docs_MO_DG_Deep_Learning_Model_Optimizer_DevGuide.html (access date: 25.05.2021).
- [7] Andriyanov N.A. Analysis of the Acceleration of Neural Networks Inference on Intel Processors Based on OpenVINO Toolkit // IEEE Systems of Signal Synchronization, Generating and Processing in Telecommunications. 2020. P. 1–5.
- [8] Zhang F. et al. An expression recognition method on robots based on mobilenet V2-SSD // IEEE 6th International Conference on Systems and Informatics, ICSAI. 2019. P. 118–122.
- [9] Solovyev R.A. et al. Real-Time Recognition of Handwritten Digits in FPGA Based on Neural Network with Fixed Point Calculations // Problems of Perspective Micro- and Nanoelectronic Systems Development. 2019. Issue 4. P. 39–42.
- [10] COCO Dataset URL: <https://cocodataset.org/#download> (access date: 25.05.2021).
- [11] Lin T. Y. et al. Microsoft COCO: Common Objects in Context // Lecture Notes in Computer Science. Springer Verlag, 2014. V. 8693, № 5. P. 740–755.
- [12] Jin Z., Finkel H. Analyzing Deep Learning Model Inferences for Image Classification using OpenVINO // IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). 2020. P. 908–911.
- [13] Vasiliev E.P., Rustamov A.Z. OpenVINO GAN DEMOS repository URL: <https://github.com/itlab-vision/OpenVINO-GAN-DEMO> (access date: 25.05.2021)
- [14] Tutorial: Creating a Simple GAN in Python with Keras Library. URL: <https://neurohive.io/ru/tutorial/simple-gan-python-keras/> (access date: 12.05.2021).
- [15] Johnson J., Alahi A., Fei-Fei L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution // European Conference on Computer Vision. 2016. P. 694–711.
- [16] Shi W. et al. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 1874–1883.