



<http://dx.doi.org/10.35596/1729-7648-2026-24-4-38-46>

УДК 681.5, 681.7, 621.7

РАЗРАБОТКА ПЛАТФОРМЕННО-ОРИЕНТИРОВАННОГО АЛГОРИТМА СИНТЕЗА ОПТИМАЛЬНОЙ НЕЙРОСЕТЕВОЙ АРХИТЕКТУРЫ ДЛЯ МОНИТОРИНГА ЛАЗЕРНОЙ ПОЛИРОВКИ КВАРЦЕВЫХ СТЕКОЛ

В. А. ПРОХОРЕНКО, В. С. СМОРОДИН, Ю. В. НИКИТЮК

Гомельский государственный университет имени Франциска Скорины (Гомель, Республика Беларусь)

Аннотация. Предложен алгоритм эволюционного синтеза архитектур нейронных сетей, ориентированный на решение задач мониторинга технологических процессов в условиях ограниченных вычислительных ресурсов и обучающих выборок в режиме реального времени. Алгоритм основан на модифицированном подходе нейроэволюции, в котором геном особи включает в себя граф крупных функциональных блоков и происходит отслеживание истории инноваций. Оценка особей популяции выполнялась многокритериальным алгоритмом NSGA-II на основе точности классификации и времени прямого прохода сети на целевом устройстве. Работоспособность алгоритма продемонстрирована на задаче классификации изображений с использованием набора данных CIFAR-10, а также на прикладной задаче классификации режимов работы при реализации технологической операции лазерной однолучевой полировки кварцевых стекол.

Ключевые слова: нейронные сети, нейроэволюция, поиск нейросетевых архитектур, лазерная обработка материалов, мониторинг технологических процессов.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Для цитирования. Прохоренко, В. А. Разработка платформенно-ориентированного алгоритма синтеза оптимальной нейросетевой архитектуры для мониторинга лазерной полировки кварцевых стекол / В. А. Прохоренко, В. С. Смородин, Ю. В. Никитюк // Доклады БГУИР. 2026. Т. 24, № 4. С. 38–46. <http://dx.doi.org/10.35596/1729-7648-2026-24-4-38-46>.

DEVELOPMENT OF A PLATFORM-ORIENTED ALGORITHM FOR SYNTHESIZING AN OPTIMAL NEURAL NETWORK ARCHITECTURE FOR MONITORING LASER POLISHING OF QUARTZ GLASS

VLADISLAV PROKHORENKO, VICTOR SMORODIN, YURI NIKITJUK

Francisk Skorina Gomel State University (Gomel, Republic of Belarus)

Abstract. An evolutionary synthesis algorithm for neural network architectures is proposed. It is designed to address real-time process monitoring tasks with limited computing resources and training data. The algorithm is based on a modified neuroevolution approach, in which the individual genome comprises a graph of large functional blocks and the history of innovations is tracked. The evaluation on population individuals was performed by the multi-criteria algorithm NSGA-II based on classification accuracy and the forward pass time network on the target device. The algorithm's performance is demonstrated on an image classification task using the CIFAR-10 dataset, as well as on the applied task of classifying operating modes during the single-beam laser polishing of quartz glass.

Keywords: neural networks, neuroevolution, search for neural network architectures, laser processing of materials, monitoring of technological processes.

Conflict of interests. The authors declare that there is no conflict of interests.

For citation. Prokhorenko V., Smorodin V., Nikitjuk Yu. (2026) Development of a Platform-Oriented Algorithm for Synthesizing an Optimal Neural Network Architecture for Monitoring Laser Polishing of Quartz Glass. *Doklady BGUIR*. 24 (4), 38–46. <http://dx.doi.org/10.35596/1729-7648-2026-24-4-38-46> (in Russian).

Введение

Автоматизированное проектирование нейросетевых архитектур – важная задача на современном этапе развития области машинного обучения. Экспертное конструирование архитектур нейронных сетей является сложным и требует специфического опыта работы с предметной областью. В связи с этим в настоящее время активно развиваются методы поиска нейросетевых архитектур (Neural Architecture Search, NAS), позволяющие сгенерировать нейронную сеть под конкретную техническую задачу. Среди методов NAS можно выделить в отдельную категорию эволюционные методы поиска архитектуры сети.

При мониторинге технологических процессов лазерной обработки материалов, в частности лазерной полировки кварцевых стекол, по одномодальному видеосигналу при наличии источника поляризованного света можно визуально оценить степень полировки за пятном лазерного излучения в режиме реального времени по наличию ряда характерных визуальных признаков. Выявление и своевременная коррекция режимов выполнения технологической операции, в которых происходит недостаточная полировка, либо наоборот, перегрев области обработки, – важная научно-техническая задача, решение которой позволит повысить качество выпускаемой продукции и снизить уровень брака. Использование нейронных сетей в технической системе в режиме реального времени предъявляет жесткие требования к скорости и точности классификации режимов обработки стекол. В этих условиях особую значимость приобретает задача синтеза аппаратно-эффективной нейросетевой архитектуры, позволяющая обеспечить баланс между качеством распознавания и скоростью обработки сенсорных данных.

Обзор существующих подходов

В истории развития методов эволюции нейронных сетей можно выделить алгоритм NEAT [1], в котором происходит совместная эволюция весов и архитектуры сети, а также используется механизм отметок инноваций для возможности реализации скрещивания особей популяции с частично совпадающей историей. Идеи NEAT были обобщены на глубокие нейросетевые модели в алгоритмах DeepNEAT и CoDeepNEAT [2], где пространство эволюционного поиска включает в себя композиции блоков глубоких сетей. Принципиальными преимуществами такого подхода являются расширение пространства поиска и возможность открытия нетривиальных нейросетевых архитектур. Однако значительный его недостаток – высокая вычислительная сложность, так как оценка членов популяции требует их обучения. А это приводит к тому, что время поиска оптимальной архитектуры может занимать сотни GPU-часов [2].

Следует отметить, что эволюционный поиск является лишь одним из направлений NAS. В тот же период времени, что и CoDeepNEAT, разрабатывались подходы, основанные на обучении с подкреплением [3]. Согласно [3], рекуррентная сеть на основе методов обучения с подкреплением обучалась генерировать такие нейросетевые архитектуры, которые максимизировали точность в рассматриваемой задаче классификации. Однако, несмотря на высокое качество открытых архитектур, практическая реализация данных методов также требовала сотни GPU-часов поиска. В обзоре предметной области [4] подчеркивается, что именно ускорение процедуры оценки архитектур-кандидатов в методах NAS является важнейшей задачей исследований.

В последующих работах по тематике предлагался ряд подходов к снижению стоимости этой процедуры за счет использования вместо полного обучения нейронных сетей приближенных оценок их качества. К числу таких подходов можно отнести неполное обучение (low-fidelity training) и применение различных стратегий раннего выхода (early exit), позволяющие не исключать неэффективные архитектуры без полного цикла обучения [4, 5]. Так, в [5] предложены эволюционные NAS с механизмами раннего выхода, что позволило радикально сократить время поиска по сравнению с рядом более ранних NAS-алгоритмов. Можно также выделить использование приближенных оценок качества нейросетевых архитектур, в том числе предсказание времени прямого прохода.

Отдельно следует отметить такое направление современных исследований, как платформенно-ориентированные алгоритмы NAS, в которых архитектура оптимизируется не только по качеству классификации, но и по выполнению на целевом устройстве. Например, в [6] предложено выполнять оценку архитектуры не через оценку числа операций с плавающей точкой, а через непосред-

ственно либо приближенно измеренное время прямого прохода. Такой подход нашел применение при разработке компактных сверточных нейросетей для мобильных устройств с ограниченными вычислительными ресурсами. В последние годы для мобильных устройств был вручную спроектирован ряд производительных архитектур, таких как MobileNetV2, MobileNetV3 [7, 8], с архитектурными макроблоками, которые можно использовать для формирования пространства поиска в алгоритмах NAS.

Формирование пространства поиска нейросетевых архитектур для алгоритма

В исследованиях в качестве минимального структурного элемента, который в процессе мутации может быть добавлен в архитектуру, использовали макроблок сверточной архитектуры. Также были выбраны макроблоки по сверточным нейросетям последних лет:

- MBConv [8] – эффективный сверточный блок с остаточной связью;
- Fused-MBConv [9] – является развитием MBConv, где часть операций объединяется для повышения скорости обучения и прямого прохода;
- Dynamic Convolution [10], где повышение точности достигается за счет адаптивного взвешивания фильтров;
- ShuffleBlock, использованный в сети ShuffleNetV2 [11] архитектуры, оптимизированной для практической аппаратной эффективности;
- ConvNeXt [12], отличающийся использованием операции нормализации слоя (LayerNorm);
- UIB (Universal Inverted Bottleneck), использованный в MobileNetV4, который является дальнейшим развитием мобильных инвертированных архитектур [13].

В отличие от ряда предложенных в последние годы алгоритмов NAS, где ускорение процедуры поиска достигается за счет сужения пространства поиска, задания шаблона сети, для которого перебираются различные варианты операций или используются градиентные методы поиска [4, 14–16], в проводимых исследованиях авторы статьи осуществляли эволюционный поиск архитектуры на основе схемы кодирования геномов, скрещиваний, мутаций и отслеживания инноваций из классического алгоритма NEAT [1]. NEAT за счет использования разнородных современных блоков потенциально способен открывать нетривиальные архитектуры более высокого уровня организации, тогда как сокращение времени поиска обеспечивается за счет оценки пригодности членов генома по укороченной процедуре обучения. Используемые в предлагаемом алгоритме блоки приведены на рис. 1.

Алгоритм эволюционного синтеза оптимальной нейросетевой архитектуры

Пусть $Population = \{G_1, G_2, \dots, G_N\}$ – популяция из N особей, где G_i – геном члена популяции. $Arch(G)$ – нейросетевая архитектура, соответствующая геному G . Пусть геном G кодируется ориентированным ациклическим графом $g = (V, E)$, где каждая вершина $v \in V$ представляет собой входы сети $Input$, выходной блок сети F_{OUT} , содержащий выходной слой сети, либо один вычислительный блок сети Blk_i из множества выбираемых пользователем алгоритма перед началом его использования нейросетевых блоков $Blocks = \{Blk_1, Blk_2, \dots, Blk_B\}$.

Цель алгоритма – генерация нейросетевой архитектуры $Arch(G^*)$, оптимальной по совокупности численных критериев $C = \{C_1, C_2, \dots, C_R\}$, вычисляемых процедурами оценки нейросетевых архитектур $Eval(Arch(G))$ и $Eval_{short}(Arch(G))$, которые определяются до начала использования алгоритма.

В алгоритме использовались следующие наборы мутаций:

- добавление связи между узлами $v_1, v_2 \in V$ графа $g = (V, E)$;
- удаление (отключение) связи между узлами $v_1, v_2 \in V$ графа $g = (V, E)$, если это не нарушает связности графа и есть подмножество ребер $E^* \in E$, представляющее собой путь из v_{Input} в v_{FOUT} ;
- добавление вершины в граф: выбирается случайное ребро графа $e(v_1, v_2) \in E$, оно деактивируется, создается со случайно выбранным из множества $Blocks$ вершина v_{new} , добавляется два ребра $e(v_1, v_{new})$ и $e(v_{new}, v_2)$;
- удаление случайной вершины v графа $g = (V, E)$ и ребер, связанных с этой вершиной;

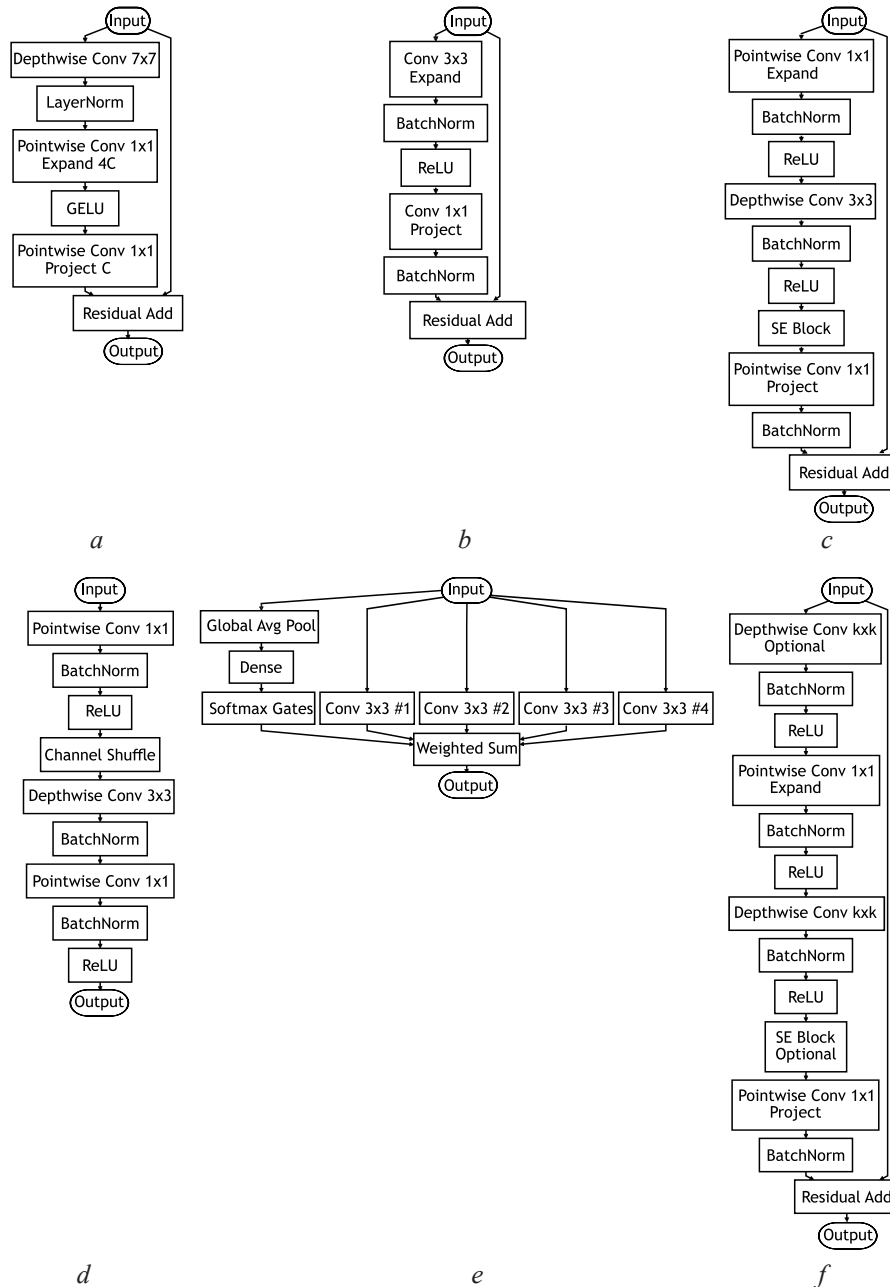


Рис. 1. Схемы использованных в алгоритме архитектурных блоков: *a* – ConvNeXt; *b* – Fused-MBConv; *c* – MBConv; *d* – ShuffleBlock; *e* – Dynamic Convolution; *f* – UIB

Fig. 1. Schemes of the architectural blocks used in the algorithm: *a* – ConvNeXt; *b* – Fused-MBConv; *c* – MBConv; *d* – ShuffleBlock; *e* – Dynamic Convolution; *f* – UIB

– случайные изменения параметров блока Blk_i , соответствующего случайной вершине v графа $g = (V, E)$.

Вероятности появления мутаций задаются множеством $P_{mut} = \{P_1, P_2, P_3, P_4, P_5\}$ и определяются пользователем алгоритма до начала его использования.

Для обеспечения корректного сопоставления структурных мутаций между различными геномами в алгоритме применяется глобальный трекер инноваций (*InnovationTracker*), который хранит множества $\{innov_{node}^i\}$ и $\{innov_{conn}^j\}$ уникально нумерованных инноваций, связанных с вершинами графа и ребрами соответственно. В алгоритме используется процедура отбора $SelectForCrossover(Population_k)$ членов популяции текущего поколения $\{G_i^{sel}\} \in Population_k$, $i = 1, 2, \dots, N_S$, $N_S < N$, для скрещивания, которая определяется пользователем алгоритма перед началом его применения.

В алгоритме используется процедура скрещивания $G_{new} = Crossover(G_1, G_2)$, реализованная, как в NEAT [1], но на уровне крупных блоков, а не отдельных нейронов. Алгоритм выполнялся по следующим этапам.

1. Пользователь алгоритма определяет набор нейросетевых блоков *Blocks*, выходной блок сети F_{OUT} , состав численных критериев оценки членов популяции $\{C_1, C_2, \dots, C_R\}$, определяет вычислительные процедуры $Eval(Arch(G))$ и $Eval_{short}(Arch(G))$ полной и короткой оценок члена популяции, количество особей популяции N , максимальное число поколений T_{max} , функцию ранней остановки эволюции $EarlyExit(C(G))$, принимающую значения $\{True, False\}$, множество вероятностей возникновения мутаций $P_{mut} = \{P_1, P_2, P_3, P_4, P_5\}$. Пользователь выбирает стратегию многокритериального отбора и определяет процедуру отбора членов популяции $SelectForCrossover(Population_k)$, число членов популяции N и максимальное число поколений K .

2. Инициализация популяции N геномами.

3. Оценка всех членов начальной популяции с помощью процедуры $C(G_i) = Eval_{short}(Arch(G_i))$.

4. Начало основного цикла алгоритма.

5. Применение процедуры отбора членов популяции для скрещивания популяции $SelectForCrossover(Population_k)$, реализующей выбранную пользователем стратегию многокритериального отбора. На ее основе определяется множество геномов, допущенных к скрещиванию.

6. Формирование нового поколения $Population_{k+1}$ на основе геномов, отобранных на шаге 5. Новые члены популяции последовательно создаются с использованием операторов скрещивания $G_{new} = Crossover(G_1, G_2)$ и набора мутаций до тех пор, пока размер новой популяции не достигнет N . Конкретный способ выбора родительских пар, включения элитных особей, замещения предыдущего поколения и формирования итогового состава $Population_{k+1}$ определяется выбранной пользователем стратегией многокритериального отбора.

7. Оценка новых членов популяции процедурой $Eval_{short}(Arch(G))$. При этом проверяется условие $EarlyExit(C(G)) = True$, при его выполнении цикл прекращается и осуществляется переход на шаг 9. Лучшим геномом считается тот, который вызвал раннюю остановку цикла эволюционного поиска.

8. Если было осуществлено число шагов цикла $k = K$, то алгоритм останавливается. Лучшим геномом считается тот, который был возвращен первым процедурой $SelectForCrossover(Population_k)$. Иначе выполняется возвращение к шагу 5.

9. Для лучшего генома выполняется процедура $Eval(Arch(G))$ полного обучения и вычисляются финальные метрики оценки качества нейросети применительно к решаемой задаче. Итоговая нейросетевая модель сохраняется.

Результаты применения алгоритма

Предложенный алгоритм применялся к задачам классификации изображений известного набора данных CIFAR-10 [17] и классификации режимов реализации технологической операции лазерной полировки кварцевых стекол [18]. При решении задачи синтеза оптимальной нейросетевой архитектуры для распознавания изображений зоны обработки в случае реализации технологической операции лазерной полировки кварцевых стекол и задачи распознавания изображений CIFAR-10 в качестве численных критериев оценки нейросетевой архитектуры выбирались точность на валидационном множестве val_acc по результатам реализации процедуры неполного обучения нейросети на выборках исходных данных, содержащих размеченные изображения, а также средняя скорость прямого прохода сети на целевом устройстве dev_lat . Под работой в режиме реального времени в данной статье понимается режим потоковой обработки изображений зоны лазерной обработки, при котором время прямого прохода нейросетевой модели на целевом устройстве меньше допустимого интервала обновления данных мониторинга.

Процедура полной оценки включала в себя обучение на протяжении 15 эпох на исходном наборе данных изображений. Процедура раннего выхода состояла в проверке значения val_acc на превышение величины 0,62 в случае задачи классификации CIFAR-10 и 0,7 – в случае классификации изображений лазерной обработки. Вероятности мутаций были заданы как $\{0,4, 0, 0,5, 0, 0,4\}$. Использовалась стратегия многокритериального отбора членов популяции из известного

алгоритма NSGA-II [19]. Программная реализация выполнялась на языке Python с помощью библиотек Keras и Tensorflow. По результатам применения алгоритма найдены нейросетевые архитектуры, которые позволили распознавать изображения из набора данных CIFAR-10 с точностью не менее 0,72 и режимы лазерной полировки кварцевых стекол с точностью не менее 0,98 (по результатам кросс-валидации) на целевом устройстве с видеокартой NVIDIA GTX 1070 со средним временем обработки изображения 8–10 мс.

В задаче классификации изображений известного набора данных CIFAR-10 с цветными (три канала) изображениями разрешением 32×32 10 000 изображений использовались в процессе короткого обучения для валидации, 50 000 – для обучения сетей. На рис. 2, *a* показана архитектура сети, найденная алгоритмом на втором поколении его реализации, на рис. 2, *b* представлен график распределения геномов, полученных в ходе двух поколений работы алгоритма, в пространстве критериев точности и времени прямого прохода сети на устройстве при решении задачи классификации изображений из набора данных CIFAR-10.

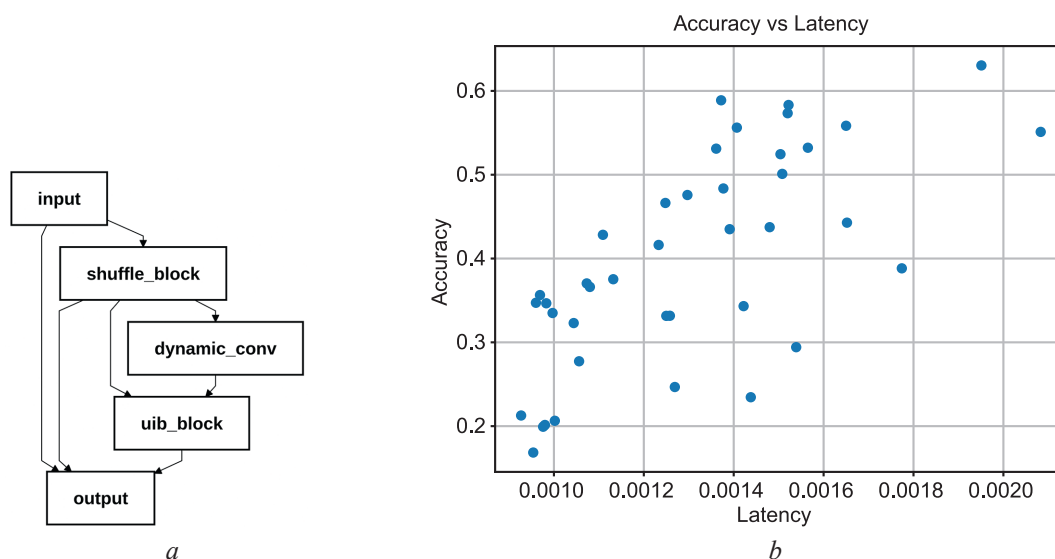


Рис. 2. Задача классификации изображений набора данных CIFAR-10:

a – схема архитектуры нейронной сети, найденной алгоритмом на втором поколении;

b – график распределения геномов, полученных в ходе двух поколений работы алгоритма

Fig. 2. Image classification problem of the CIFAR-10 dataset: *a* – diagram of the neural network architecture found by the algorithm in the second generation; *b* – plot of the distribution of genomes obtained during two generations of the algorithm's operation

В табл. 1 приведены значения метрик оценки и параметров нейросети, полностью обученной на наборе данных CIFAR-10, после полного цикла обучения на протяжении 15 эпох. Число членов популяции – 20, максимальное число поколений – 5. Обучение сетей производилось оптимизатором Adam со значением скорости обучения $\alpha = 1e - 3$ на графическом ускорителе NVIDIA T4. Время, затраченное на поиск архитектуры, составило 48 мин.

Таблица 1. Значения метрик оценки и параметров нейросети, обученной на наборе данных CIFAR-10

Table 1. Values of evaluation metrics and parameters of the neural network trained on the CIFAR-10 dataset

Метрика		Число параметров	Время прямого прохода, мс
Top-1 Accuracy	Top-5 Accuracy		
0,7292	0,9795	77 198	2,0

В задаче классификации изображений, снятых камерой во время реализации технологической операции лазерной полировки кварцевых стекол, рассматривались три класса: нормальная полировка, недополировка и переполіровка. Объем исходной выборки составил 10 332 изображения (3709 изображений нормальной полировки, 4231 изображение недополировки, 2392 изображения переполіровки). Примеры исходных изображений, использованных при обучении нейросетей распознаванию режимов реализации технологической операции лазерной полировки кварцевых стекол, приведены на рис. 3. Исходные изображения имели разрешение 384×384 (три

канала), перед обработкой сетью разрешение было уменьшено до 32×32 , изображения приводились к черно-белому виду (один канал). На каждой итерации данные разделялись на обучающую и тестовую подвыборки в отношении 80/20 с сохранением долей классов.

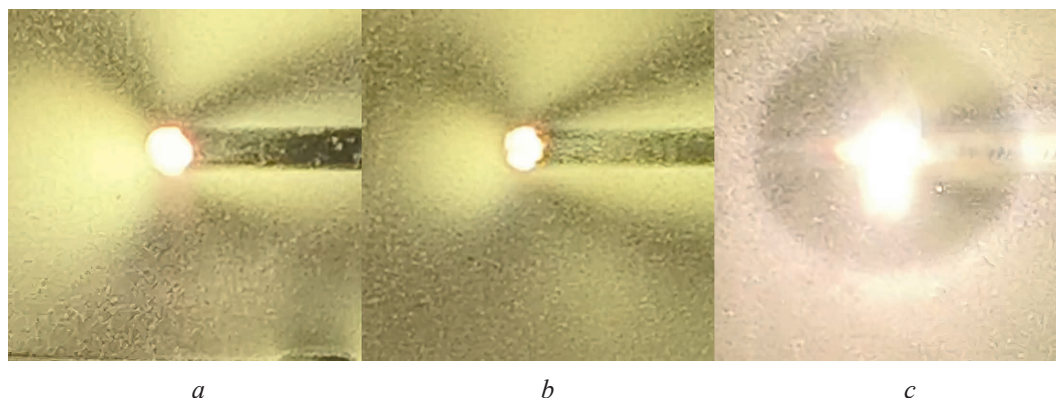


Рис. 3. Примеры изображений, использованных при обучении нейросетей распознаванию режимов лазерной полировки кварцевых стекол:

a – нормальная полировка; *b* – недополировка; *c* – переполіровка (расплавление)

Fig. 3. Examples of images used to train neural networks to recognize laser polishing modes for quartz glass:

a – normal polishing; *b* – underpolishing; *c* – overpolishing (melting)

На рис. 4 показана архитектура нейронной сети, найденная алгоритмом за пять поколений его реализации в задаче классификации изображений зоны лазерной обработки при реализации технологической операции лазерной полировки кварцевых стекол (приведены отслеживаемые инновации для соединений). При этом условием раннего выхода было достижение точности валидации по результатам сокращенного цикла обучения 0,70, но оно не было активировано.

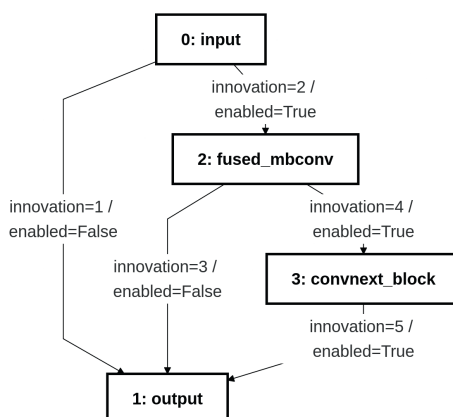


Рис. 4. Архитектура нейронной сети, найденной алгоритмом за пять поколений

Fig. 4. The architecture of the neural network found by the algorithm in five generations

В табл. 2 приведены значения метрик оценки и параметров лучшей архитектуры нейронной сети для распознавания режимов реализации технологической операции лазерной полировки кварцевых стекол после пятикратной кросс-валидации с полным циклом обучения (15 эпох). Число членов популяции – 25, максимальное число поколений – 5. Обучение сетей производилось оптимизатором Adam со значением скорости обучения $\alpha = 1e - 3$ на графическом ускорителе NVIDIA GTX1070. Время, затраченное на поиск архитектуры, составило 50 мин.

Таблица 2. Метрики оценки и параметры лучшей архитектуры нейронной сети

Table 2. Evaluation metrics and parameters for the best neural network architecture

Метрика				Число параметров	Время прямого прохода, мс
Accuracy	Average Recall	Average Precision	Average F1		
0,9830	0,9844	0,9841	0,9841	12 619	8–10

Заключение

1. Представлен платформенно-ориентированный алгоритм эволюционного синтеза оптимальной нейросетевой архитектуры, который может применяться при разработке систем мониторинга процессов лазерной обработки в режиме реального времени. Результаты проведенных экспериментов подтверждают работоспособность и практическую применимость данного алгоритма для задач классификации изображений в условиях ограниченных вычислительных ресурсов и потоковой обработки изображений.

2. Показано, что использование нейроэволюционного поиска с процедурой многокритериального отбора архитектур-кандидатов по точности и времени прямого прохода, а также с механизмами ускоренной оценки кандидатов позволяет находить компактные и эффективные архитектуры как для задачи классификации известного набора данных CIFAR-10, так и для прикладной задачи классификации изображений зоны лазерной обработки при осуществлении мониторинга реализации технологической операции лазерной полировки кварцевых стекол. При этом вопросы гарантированного жесткого реального времени на уровне промышленного контроллера и операционной системы требуют дальнейших исследований.

3. Полученные результаты свидетельствуют о перспективности предложенного подхода для построения интеллектуальных систем технологического мониторинга, функционирующих непосредственно на целевых аппаратных платформах, и могут быть применены в качестве компонентов интеллектуальных компьютерных систем для реализации алгоритмов построения обратных связей по управлению технологическими операциями автоматизированных технологических систем.

Список литературы / References

1. Stanley K. O., Miikkulainen R. (2002) Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation*. 10 (2), 99–127.
2. Miikkulainen R., Liang J., Meyerson E., Rawal A., Fink D., Francon O. (2017) Evolving Deep Neural Networks. *arXiv preprint arXiv:1703.00548*.
3. Zoph B., Le Q. V. (2016) Neural Architecture Search with Reinforcement Learning. *arXiv preprint arXiv:1611.01578*.
4. Elsken T., Metzen J. H., Hutter F. (2019) Neural Architecture Search: A Survey. *Journal of Machine Learning Research*. 20 (55), 1–21.
5. Termritthikun C., Jamtsho Y., Ieamsaard J., Muneesawang P., Lee I. (2021) EEEA-Net: An Early Exit Evolutionary Neural Architecture Search. *arXiv preprint arXiv:2108.06156*.
6. Tan M., Chen B., Pang R., Vasudevan V., Sandler M., Howard A., et al. (2019) MnasNet: Platform-Aware Neural Architecture Search for Mobile. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
7. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C. (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
8. Howard A., Sandler M., Chu G., Chen L., Chen B., Tan M., et al. (2019) Searching for MobileNetV3. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
9. Tan M., Le Q. V. (2021) EfficientNetV2: Smaller Models and Faster Training. *Proceedings of the 38th International Conference on Machine Learning (ICML)*.
10. Chen Y., Dai X., Liu M., Chen D., Yuan L., Liu Z. (2020) Dynamic Convolution: Attention over Convolution Kernels. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
11. Ma N., Zhang X., Zheng H.-T., Sun J. (2018) ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. *Proceedings of the European Conference on Computer Vision (ECCV)*.
12. Liu Z., Mao H., Wu C.-Y., Feichtenhofer C., Darrell T., Xie S. (2022) A ConvNet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
13. Qin D., Lechner C., Delakis M., Fornoni M., Luo S., Yang F., et al. (2024) MobileNetV4: Universal Models for the Mobile Ecosystem. *arXiv preprint arXiv:2404.10518*.
14. Dong X., Hu Y., Yang Y., Tian Y. (2021) Differentiable Architecture Search with Directional Derivatives. *Advances in Neural Information Processing Systems (NeurIPS)*.
15. Chen T., Xu Y., Wang S., Li H. (2020) Progressive Supernet Optimization for Efficient Neural Architecture Search. *Proceedings of the 37th International Conference on Machine Learning (ICML)*.
16. Cai H., Zhu L., Han S. (2020) Once for All: Train One Network and Specialize it for Efficient Deployment. *International Conference on Learning Representations (ICLR)*.

17. Krizhevsky A. (2009) *Learning Multiple Layers of Features from Tiny Images*.
18. Emelyanov V. A., Shershnev E. B., Nikityuk Yu. V., Sokolov S. I., Aushev I. Yu. (2022) Optimization of Laser Polishing of Quartz Glass Using the Method of Computational Experiment Design. *Problems of Physics, Mathematics and Engineering*. 51 (2), 26–30. https://doi.org/10.54341/20778708_2022_2_51_26 (in Russian).
19. Deb K., Pratap A., Agarwal S., Meyarivan T. (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. 6 (2), 182–197.

Поступила 07.04.2026

Принята в печать 05.05.2026

Received: 7 April 2026

Accepted: 5 May 2026

Вклад авторов

Прохоренко В. А. разработал и реализовал платформенно-ориентированный алгоритм эволюционного синтеза оптимальной нейросетевой архитектуры для мониторинга технологических операций лазерной обработки в режиме реального времени, подготовил рукопись статьи.

Сморodin В. С. провел консультацию по теоретической части и интерпретации результатов, сформулировал и уточнил исследовательские задачи и цели.

Никитюк Ю. В. разработал требования технологического регламента осуществления мониторинга и управления технологической операцией лазерной обработки материалов, предоставил набор размеченных изображений зоны лазерной обработки.

Authors' contribution

Prokhorenko V. developed and implemented a platform-oriented algorithm for the evolutionary synthesis of an optimal neural network architecture for monitoring laser processing operations in real time, prepared the manuscript to the article.

Smorodin V. provided consultation on the theoretical part and interpretation of the results, formulated and clarified the research objectives and goals.

Nikitjuk Yu. developed requirements for technological regulations for monitoring and managing the technological operation of laser processing of materials, and provided a set of marked images of the laser processing zone.

Сведения об авторах

Прохоренко В. А., ст. преп. каф. математических проблем управления и информатики, Гомельский государственный университет имени Франциска Скорины (ГГУ ФС)

Сморodin В. С., д-р техн. наук, проф., зав. каф. математических проблем управления и информатики, ГГУ ФС

Никитюк Ю. В., канд. физ.-мат. наук, доц., проректор по учебной работе, ГГУ ФС

Адрес для корреспонденции

246028, Республика Беларусь,
Гомель, ул. Кирова, 119
Гомельский государственный университет
имени Франциска Скорины
Тел.: +375 232 51-03-04
E-mail: shnysct@gmail.com
Прохоренко Владислав Александрович

Information about the authors

Prokhorenko V., Senior Lecturer at the Department of Mathematical Problems of Control and Computer Science, Francisk Skorina Gomel State University (FS GSU)

Smorodin V., Dr. Sci. (Tech.), Professor, Head of the Department of Mathematical Problems of Control and Computer Science, FS GSU

Nikitjuk Yu., Cand. Sci. (Phys. and Math.), Associate Professor, Vice-Rector for Academic Affairs, FS GSU

Address for correspondence

246028, Republic of Belarus,
Gomel, Kirova St., 119
Francisk Skorina
Gomel State University
Tel.: +375 232 51-03-04
E-mail: shnysct@gmail.com
Prokhorenko Vladislav