

Научная статья

УДК 004.032.26

URL: <https://trudymai.ru/published.php?ID=185106>

EDN: <https://www.elibrary.ru/MMJVDE>

## МЕТОДИКА ПОДБОРА ГИПЕРПАРАМЕТРОВ НЕЙРОСЕТЕВОЙ МОДЕЛИ В ЗАДАЧАХ ОПТИЧЕСКОЙ НАВИГАЦИИ

Андрей Сергеевич Дорошев<sup>1</sup>, Дмитрий Алексеевич Шеломанов<sup>2</sup>✉

<sup>1</sup>Московское опытно-конструкторское бюро «Марс»,

Москва, Россия

<sup>2</sup>Московский авиационный институт (национальный исследовательский университет), Москва, Россия

<sup>1</sup>[doroshevas@mai.ru](mailto:doroshevas@mai.ru)

<sup>2</sup>[shelomanovda@mai.ru](mailto:shelomanovda@mai.ru)✉

**Аннотация.** В данной статье рассматривается задача автономного управления беспилотного летательного аппарата (БЛА) с использованием методов Simultaneous Localization and Mapping (SLAM). Одной из проблем данного метода при использовании оптического диапазона в качестве источника данных является оценка расстояния до объектов. Для определения расстояния предложена методика обучения нейросетевой модели, способной проводить оценку расстояния на основе монокулярного изображения с камеры с дополнительной оптимизацией гиперпараметров с целью достижения более качественного результата обучения модели. Использование предложенной методики позволило улучшить целевую

метрику обучения в рамках одинакового количества итераций подбора гиперпараметров модели. Дополнительно данная методика проверялась при использовании двух других архитектур нейронных сетей и наборов данных, показав положительные результаты что позволяет сделать предположение об эффективности данного метода и возможности его применения для улучшения точности обучаемых моделей.

**Ключевые слова:** нейронные сети, беспилотные летательные аппараты, гиперпараметры, оптическая навигация, интеллектуальные системы

**Для цитирования:** Дорошев А.С., Шеломанов Д.А. Методика подбора гиперпараметров нейросетевой модели в задачах оптической навигации // Труды МАИ. 2025. № 142. URL: <https://trudymai.ru/published.php?ID=185106>

Original article

## **METHODOLOGY FOR SELECTING HYPERPARAMETERS OF A NEURAL NETWORK MODEL IN OPTICAL NAVIGATION TASKS**

**Andrey S. Doroshev<sup>1</sup>, Dmitry A. Shelomanov<sup>2</sup>✉**

<sup>1</sup>Moscow Experimental Design Bureau "Mars",  
Moscow, Russia

<sup>2</sup>Moscow Aviation Institute (National Research University), Moscow, Russia

<sup>1</sup>[doroshevas@mai.ru](mailto:doroshevas@mai.ru)

<sup>2</sup>[shelomanovda@mai.ru](mailto:shelomanovda@mai.ru) ✉

**Abstract.** This article presents a technique for optimizing hyperparameters of a neural network model in tasks of autonomous optical navigation of unmanned aerial vehicles (UAVs) using Simultaneous Localization and Mapping (SLAM) methods. The problem of estimating the distance to objects based on monocular camera data is considered. To solve the problem, a hyperparameter selection technique is proposed, including a consistent narrowing of the hyperparameter search area and the use of Bayesian optimization.

The main stages of the methodology:

- 1) Defining the initial search area of hyperparameters;
- 2) Formation of the objective function and the direction of optimization.
- 3) Conducting a series of experiments;
- 4) Narrowing the search area for hyperparameters based on the analysis of the coefficient of variation of parameters;
- 5) Repeat training until the optimal result is achieved.

The application of the proposed methodology made it possible to increase the accuracy of model training while maintaining a fixed amount of trials for training experiments. To test the effectiveness, the architectures of encoder-decoder models with a pre-trained Mobilenet model and NYU Depth Dataset V2 data were used. The training was carried out using the target loss function, which includes metrics on pixel difference, structural similarity and image gradients.

The results showed that the proposed technique improved the prediction of the depth of monocular images, and was also tested on tasks with other architectures (MNIST,

ResNet18), confirming the universality of the approach. For example, for ResNet18, the accuracy on the test sample increased by 1.08%.

Optimization of hyperparameters has demonstrated effectiveness in improving the accuracy of models while maintaining a fixed trials amount, which makes the technique especially useful for tasks with limited computing resources.

**Keywords:** neural networks, unmanned aerial vehicles, hyperparameters, optical navigation, intelligent systems

**For citation:** Doroshev A.S., Shelomanov D.A. Methodology for selecting hyperparameters of a neural network model in optical navigation tasks. *Trudy MAI*. 2025. No. 142. (In Russ.).

URL: <https://trudymai.ru/eng/published.php?ID=185106>

## **Автономная навигация и алгоритмы SLAM**

Для решения задачи планирования траектории устройства необходимо сначала определить его текущее местоположение и оценить окружающую среду. Управление движением зависит от точной или приближенной оценки положения в пространстве с известной ошибкой определения координат, что позволяет задать потенциальную область, где находится устройство.

Для обеспечения автономной навигации, обнаружения препятствий и предотвращения столкновений БЛА требуется постоянное обновление информации о положении и состоянии окружающей среды. Для достижения данной цели применяют спутниковые системы навигации, инерциальные навигационные системы, корреляционно экстремальные навигационные системы (КЭНС), системы с лазерным

дальномером Light Detection and Ranging (LIDAR) и системы с радиолокационным дальномером Radio Detection and Ranging (RADAR). Однако, малые БЛА накладывают существенные ограничения на применимую датчиковую аппаратуру [1], в виду этого становится невозможным использовать LIDAR и RADAR системы из-за ограничений по массогабаритным характеристикам. Спутниковые навигационные системы имеют низкую устойчивость сигналов в условиях переотражений, затрудненного обзора неба, что накладывает ограничение на их применение на определенных участках движения. Микромеханическая датчиковая аппаратура обладает высоким уровнем погрешности измерений, а КЭНС системы требуют задания минимальной высоты полета и априорной информации о местности, что ограничивает их применение на низких высотах полета. Таким образом, необходимо использовать метод способный функционировать в условиях неизвестной среды и низких высот, как в закрытых помещениях, так и открытых пространствах, с учетом ограничений и процессов миниатюризации БЛА.

Одним из новейших методов, предложенных для решения задачи определения собственного положения, является применение элементов компьютерного зрения [2, 3, 4]. Высокая производительность современных вычислительных систем позволяет осуществлять обработку видеопотока с целью извлечения данных о смещении видимых точек кадров с видеопотока камеры, что позволяет решать задачу оценки относительного движения объекта с последующим итеративным построением его траектории, иначе говоря, осуществлять визуальную одометрию объекта. Со временем данная задача привела к развитию более сложной проблемы —

одновременной локализации и картографии Simultaneous Localization and Mapping (SLAM), например, с использованием LIDAR. В задаче SLAM предполагается, что аппарат находится в среде, о которой у него нет предварительной информации. С использованием только собственных бортовой датчиковой аппаратуры объект должен сформировать карту местности и получить точную оценку своей траектории.

Алгоритмы VSLAM в отличие от алгоритмов SLAM, используют данные видеопотока с камер, позволяющую им получать более детальную информацию об окружающей среде, что делает применение оптических сенсоров одним из наиболее перспективных решений, применяемым в задачах SLAM.

Определение координат видимых точек кадра при использовании алгоритмов VSLAM сопровождается рядом проблем: особенности освещения и оптические характеристики камер требуют калибровки применяемого алгоритма под конкретную датчиковую аппаратуру и условия внешней среды. Использование монокулярной камеры увеличивает общую погрешность работы алгоритма. В виду данной особенности перспективностью обладают алгоритмы на основе пары цветного изображения камеры и соответствующего ему изображения глубины, или алгоритмы, основанные на стереокамерах. С их помощью ускоряется процесс вычисления координат отслеживаемых точек и инициализация алгоритма.

Однако, использование стереокамер требует дополнительных затрат при проектировании БЛА, а использование RGB-D камер ограничено навигацией внутри помещений в виду малой предельной дальности оценки глубины изображения. Но исследования, описанные в работе «Непрерывная эволюция Visual SLAM от

геометрии к глубокому обучению: проблемы и возможности» [5] показали, что применение нейросетевых моделей также способно решить задачу предсказания глубины кадра лишь по монокулярному изображению камеры. При использовании определенных методов обучения [6] нейронные сети будут устойчивы к изменению оптических характеристик камер и способны улучшить как разрешающую способность алгоритма, так и точность в сравнении с классическими VSLAM.

### **Задача предсказания глубины изображения**

Важной и актуальной проблемой в области компьютерного зрения и робототехники является задача предсказания глубины монокулярного изображения.

Особенно актуальна данная задача для роботизированных комплексов, ограниченных габаритными и аппаратными возможностями. Компактные нейросетевые модели с меньшим количеством параметров требуют меньше вычислительных ресурсов и энергозатрат при использовании, что особенно актуально для поддержания автономности устройств с ограниченными вычислительными ресурсами, работающих в режиме реального времени.

Монокулярная RGB камера не способна предоставлять данные о глубине кадра, в виду этого было решено разработать нейросетевую модель, которая способна определять глубину изображения по имеющемуся цветному изображению кадра, что позволяет превратить монокулярную RGB камеру в аналог RGBD камеры с возможностью отслеживания глубины.

В процессе работы VSLAM производится инициализация системы координат положения ЛА и углов его ориентации. Имея данные текущего положения БЛА,

данные глубины отслеживаемых ключевых точек производится реконструкция облака точек. Для этого используется изображение карты глубины, представленное в виде  $M$  на  $N$  матрицы, коэффициент коррекции глубины, параметры калибровки камеры, координаты ключевых точек на изображении и задается предел «рабочих» значений глубины для задания верхнего и нижнего пределов соответственно. Коэффициент коррекции глубины предоставляется разработчиками RGB-D.

В процессе реализации модели проводились эксперименты с применением фреймворка Optuna. Одним из наиболее важных этапов при обучении нейросетевых моделей является подбор гиперпараметров. Гиперпараметры – это параметры, которые задаются заранее и контролируют процесс обучения модели, влияя на ее структуру, стратегию обучения и другие факторы, определяющие ее качество.

Работа [7] показала, что за счет варьирования гиперпараметров обучения становится возможным достичь более высокой точности обучения нейросетевой модели. Однако, авторы данных работ не описывают процесс построения и выбора области поиска гиперпараметров.

Авторы [8] показали возможность сокращения области поиска архитектур моделей за счет осуществления оценки результатов обучения на определенном этапе. С каждым последующим сужением области поиска архитектуры ошибка обученной нейросетевой модели снижалась, как и площадь распределения вероятностей ошибок.

В статье [9] авторы показали возможность ускорения обучения за счет применения подхода, названного "отсечением", для сокращения области поиска гиперпараметров в рамках последовательной оптимизации на основе фреймворка

(SMBO). В данной статье была введена метрика "потенциала" для областей пространства гиперпараметров, основанная на прогнозируемом результате на новых наборах данных. Используя результаты, полученные в ходе экспериментов с другими наборами данных, авторы показали возможность выявлять и отбрасывать области пространства поиска гиперпараметров, которые с малой вероятностью дадут улучшение в качестве обучения модели.

### **Предлагаемая методика**

Данная статья является идейным продолжением рассмотренных выше работ [7, 8, 9], но имеет ключевые отличия в методах оптимизации гиперпараметров за счет применения Байесовской оптимизации и предложенной методике по подбору гиперпараметров нейросетевых моделей.

Методика состоит из следующих этапов:

- 1) Выбор гиперпараметров для дальнейшего подбора и определение области допустимых значений гиперпараметров;
- 2) Задание целевой функции потерь и желаемого направления поиска (максимизация/минимизация целевой функции);
- 3) Осуществление первой группы экспериментов по обучению;
- 4) Сокращение области поиска гиперпараметров за счет удаления части гиперпараметров из области поиска и их замены на фиксированные значения;
- 5) Осуществление повторных экспериментов;

- 6) Остановка при отсутствии улучшений в точности модели, иначе повторение сокращения области поиска и повторное обучение модели.

Рассмотрим данные этапы более подробно:

В первом этапе необходимо задать начальную область поиска гиперпараметров.

Обратим внимание, что бесконечное увеличение области поиска гиперпараметров неизбежно влечет за собой увеличение требуемого времени для нахождения качественных результатов. Разделим гиперпараметры на два класса: влияющие на архитектуру модели и гиперпараметры влияющие на параметры обучения.

Параметрами, влияющими на архитектуру модели, являются:

- Количество слоев модели (глубина модели);
- Ширина слоев модели (количество нейронов на одном слое);
- Функции активации слоя;
- Степень прореживания (dropout rate) и др.

Параметрами, влияющими лишь на обучение модели, являются:

- Шаг обучения;
- Способ инициализации весов обучения;
- Оптимизатор;
- Функция потерь;
- Снижение шаг обучения (Learning Rate Decay) и др.

Совместное изменение как архитектуры модели, часто рассматриваемое в задачах AutoML (автоматическое машинное обучение), так и гиперпараметров обучения являются наиболее ресурсозатратным поиском. После выбора желаемых

гиперпараметров требуется задать область допустимых значений для их варьирования.

Далее необходимо определить целевую метрику модели. В задачах многокритериальной оптимизации происходит оценка нескольких целевых функций, например: количество нейронов модели и точность оценки класса валидационной выборки. Данные параметры могут иметь разные направления, например, минимизация размера модели при достижении максимальной точности. Также на данном этапе требуется выделить желаемый способ оптимизации: сеточные, случайные или байесовские методы.

После этого происходит осуществление первой группы экспериментов в желаемом количестве итераций подбора гиперпараметров (англ. trials) [10], которое зависит от используемого метода оптимизации и сложности обучаемой модели. Имея результаты первой группы обученных моделей, осуществляется сокращение области поиска гиперпараметров за счет выбора наименее варьирующихся параметров среди лучшей выборки моделей.

Для этого необходимо выбрать  $n$  лучших экспериментов по значению метрики качества и произвести числовое кодирование для элементов, не имеющих численное значение, иначе говоря, задать собственное числовое значение каждому уникальному элементу.

Таким образом, пусть  $R_{m \times n}$  — матрица результатов экспериментов размерностью  $m \times n$ , где  $m$  — количество экспериментов,  $n$  — количество

гиперпараметров. Каждый столбец соответствует одному гиперпараметру, а строки — различным испытаниям.

Выберем  $n$  лучших экспериментов по значению метрики качества на примере максимизирующего типа метрики:

$$R_{\text{лучш}} = \{r_j \in R \mid y_j \in n \max \{y_j\}\}; \quad (1)$$

Где:

$r_j$  — значения гиперпараметров для  $j$  — го эксперимента

$y_j$  — значение метрики качества для  $j$  — го эксперимента.

Затем, проходя по каждому столбцу гиперпараметров полученной матрицы  $R_{\text{лучш}}$  оценивается коэффициент вариации  $V$ .

$$V_j = \frac{\sigma_j}{\mu_j} - \text{коэффициент вариации} \quad (2)$$

Где:

$$\mu_j = \frac{1}{n} \sum_{i=1}^n x_i - \text{среднее значение} \quad (3)$$

$$\sigma_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} - \text{среднее квадратическое отклонение} \quad (4)$$

Где:

$n$  — количество элементов в матрице  $R_{\text{лучш}}$

$x_i$  — значение  $i$  — го гиперпараметра в матрице  $R_{\text{лучш}}$

$\bar{x}$  — среднее значение гиперпараметра в матрице  $R_{\text{лучш}}$

Далее сортируем коэффициент вариации в порядке возрастания для каждого гиперпараметра и выбираем  $k$  гиперпараметров с наименьшим коэффициентом

вариации. Данный выбор может быть расширен использованием порогового значения коэффициента вариации, который мы примем за допустимый предел разброса значений.

Далее необходимо произвести фиксирование значения полученных гиперпараметров, для этого необходимо произвести оценку как числовых, так и нечисловых параметров:

Для каждого числового параметра осуществляем оценку итогового значения по формуле межквартильного расстояния (IQR). Данная формула позволяет избежать включение выбросов в итоговые значения для оценки [11]. В рамках данной работы значения, выходящие за пределы диапазона от  $(-1,5 \cdot IQR)$  до  $(1,5 \cdot IQR)$ , рассматриваются как выбросы и подлежат исключению. Итоговая оценка значения гиперпараметра производится на основе вышеприведенной формулы среднего значения с учетом исключенных выбросов.

$$\hat{\rho}_j = \mu_j \quad (5)$$

Для каждого нечислового параметра производим частотную оценку итогового значения выбирая наиболее часто встречающееся значение гиперпараметра.

### **Реализация методики**

Имея данную методику, была обучена нейросетевая модель предсказания глубины. На основе статьи «Zero-shot Transfer learning by Combining Relative and Metric Depth» [12] было решено использовать архитектуру Энкодер – Декодер в виду малых вычислительных требований и, как следствие, высокой скорости обучения, что особенно важно при проведении множества экспериментов по обучению модели.

Особенностью данной архитектуры является использование пропуска соединений для передачи данных от слоев энкодера к декодеру. В качестве энкодера была выбрана предобученная модель Mobilenet без дополнительных модификаций, а декодер был реализован из блоков сверточных слоев, применяемых к билинейному объединению предыдущих слоев, связанных с слоем энкодера той же размерности.

В процессе обучения нейросетевой модели для определения глубины был использован набор данных NYU Depth Dataset V2. Этот набор является одним из наиболее популярных в задачах, связанных с оценкой глубины. NYU Depth Dataset V2 был создан с использованием Microsoft Kinect, позволяющий одновременно записывать цветные изображения и карты глубины. Датасет содержит более 464 различных сцен, охватывающих широкий спектр внутренних пространств, таких как жилые комнаты, офисы, кухни и коридоры. Датасет включает множество цветных изображений и соответствующих им карт глубины. Аппаратные возможности Microsoft Kinect привели к ограничению определяемой глубины до 10 метров и частичной отсутствию данных о глубине, отсутствие данных в свою очередь может быть устранено за счет предобработки изображения с целью заполнения отсутствующих пикселей. Цветные изображения и карты глубины имеют разрешение 640x480 пикселей.

Основываясь на работе «Высококачественная оценка глубины монокулярного изображения с применением трансферного обучения» [13] было сделано предположение о важности весовых коэффициентов для формирования итоговой

функции потерь при обучении модели, для достижения наибольшей точности и эффективности обучения.

Далее была реализована функция потерь, которая состояла из трех компонентов:

- $L_{\text{прямой разницы глубины}}$ ;
- $L_{SSIM}$ ;
- $L_{\text{градиентный}}$ .

$L_{\text{прямой разницы глубины}}$  представляет собой функцию потерь, основанную на средневзвешенной сумме попиксельной разницы глубин предсказанного и эталонного изображения.

$L_{SSIM}$  представляет собой функцию потерь, основанную на функции структурного сходства (SSIM) [14].

$L_{\text{градиентный}}$  представляет собой функцию потерь, основанную на операторе Собеля [15]. Данный оператор позволяет выделять границы контуров объектов на изображении.

$$L_{\text{итог}} = L_{\text{прямой разницы глубины}} * \omega_{\text{глубины}} + L_{SSIM} * \omega_{SSIM} + L_{\text{градиентный}} * \omega_{\text{градиентный}}; \quad (6)$$

Где:

$\omega_{\text{глубины}}$  — множитель функции потерь  $L_{\text{прямой разницы глубины}}$ ;

$\omega_{SSIM}$  — множитель функции потерь  $L_{SSIM}$ ;

$\omega_{\text{градиентный}}$  — множитель функции потерь  $L_{\text{градиентный}}$ .

Данная функция потерь использовалась и для валидации модели, но весовые множители были заданы равными единице, обеспечило «непредвзятость» валидации модели, в случае использования оригинальных весовых коэффициентов автоматически бы достигалось минимальное значение за счет уменьшения численного значения функции потерь.

### Результаты обучения нейросетевой модели

Имея полученную функцию потерь и исходный набор данных для обучения нейронной сети были проведены эксперименты по обучению нейросетевой модели.

Согласно методике, была сформирована начальная область поиска гиперпараметров. Целевой функцией являлась минимизация функции потерь.

Область пространства анализируемых гиперпараметров состояла из элементов, представленных в таблице 1.

Таблица 1 - Область пространства анализируемых гиперпараметров первого эксперимента

| Гиперпараметр | Область оптимизируемых значений |
|---------------|---------------------------------|
| Оптимизатор   | RMSprop, SGD, Adam, AdamW       |
| $Lr$          | 0,000001 – 0,0001               |
| $SSIM_w$      | 0,01 – 0,5                      |
| $Depth_w$     | 0,5 - 1                         |
| $Edges_w$     | 0,5 – 1                         |

По результатам обучения первой группы моделей были отобраны модели с наименьшей целевой функцией потерь, отраженные в таблице 2.

Таблица 2– Результаты обучения первой группы моделей

| Значение<br>целевой<br>функции<br>потерь | Оптимизатор | $Lr,$<br>$10^{(-5)}$ | $SSIM_w,$<br>$10^{(-3)}$ | $Depth_w,$<br>$10^{(-3)}$ | $Edges_w,$<br>$10^{(-3)}$ |
|------------------------------------------|-------------|----------------------|--------------------------|---------------------------|---------------------------|
| 0,2212                                   | RMSprop     | 1,7429               | 29,7200                  | 718,6200                  | 994,0702                  |
| 0,2225                                   | AdamW       | 1,3143               | 53,6965                  | 784,2933                  | 964,1163                  |
| 0,2228                                   | AdamW       | 2,1009               | 37,8663                  | 693,7423                  | 985,7929                  |
| 0,2263                                   | RMSprop     | 1,7680               | 33,1005                  | 716,0840                  | 965,7040                  |
| 0,2278                                   | AdamW       | 1,9050               | 69,5786                  | 685,5762                  | 892,9161                  |
| 0,2279                                   | AdamW       | 1,2983               | 53,5390                  | 848,7503                  | 940,4304                  |
| 0,2280                                   | AdamW       | 1,8755               | 122,3192                 | 612,1548                  | 948,9161                  |

Используя представленную ранее методику, были оценены гиперпараметры первой группы моделей, согласно которой следующие параметры были установлены в фиксированное значение:

$$Edges_w = 0,9588$$

$$Depth_w = 0,7226$$

Вводя фиксированные значения гиперпараметров, было произведено обучение второй группы моделей, результаты обучения приведены в таблице 3.

Таблица 3 – Результаты обучения второй группы моделей

| Значение<br>целевой<br>функции<br>потерь | Оптимизатор | $Lr,$<br>$10^{(-5)}$ | $SSIM_w,$<br>$10^{(-3)}$ |
|------------------------------------------|-------------|----------------------|--------------------------|
| 0,1726                                   | RMSprop     | 1,3790               | 332,0803                 |

|        |         |        |          |
|--------|---------|--------|----------|
| 0,1830 | RMSprop | 1,0503 | 220,0725 |
| 0,1843 | RMSprop | 1,0002 | 158,5534 |
| 0,1844 | RMSprop | 1,1506 | 220,4828 |
| 0,1857 | RMSprop | 1,0976 | 328,3554 |
| 0,1908 | RMSprop | 1,8148 | 33,2630  |
| 0,1910 | RMSprop | 1,2265 | 237,8372 |

Для данной группы моделей фиксируемыми гиперпараметрами были приняты:

$$Lr = 0,000012$$

$$SSIM_w = 0,191362$$

Оптимизатор = RMSprop

После фиксирования данных гиперпараметров обучение модели не дало более качественный результат и обучение было завершено. Далее была проведена визуализация работы пары нейронных сетей. Для этого были выбраны лучшие модели среди первой и второй группы обучений, данные которых были отражены выше. Результаты обработки изображений обученных моделей представлены на рисунке 1.

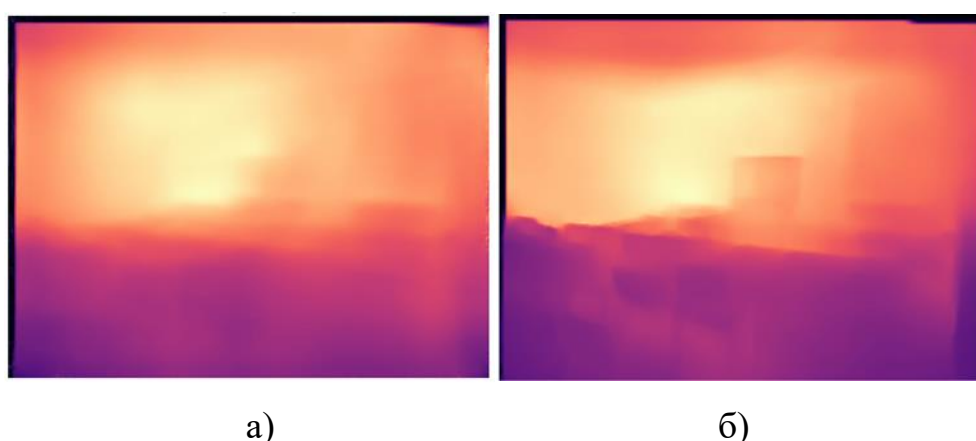


Рисунок 1 – Предсказанное изображение глубины первой (а) и второй модели (б)

Видно, что модель б) из второй группы, обученной после применения представленной методики, имеет значительное преимущество в качестве

изображения в сравнение с первой, что подтверждает улучшение её характеристик после использования данной методики.

Для подтверждения эффективности полученной методики на других задачах были проведены эксперименты по обучению моделей на основе статьи «Оптимизация гиперпараметров с удалением элементов нейронной сети» [16]. Целью являлось получение лучшего результата метрики в рамках одинакового количества итераций подбора гиперпараметров. Было проведено сравнение метрик для 200 моделей, обученных без использования предложенной методики и применение предложенной методики после 100 обученных моделей с дальнейшим обучением 100 новых моделей в усеченном пространстве поиска. Использовались две архитектуры моделей и два различных набора данных. Под моделью MNIST подразумевается полносвязная нейронная сеть, обученная на датасете MNIST. В модели ResNet18 использовалась одноименная архитектура и набор данных CIFAR10. Целевой метрикой являлась максимизация точности предсказания тестовой выборки. Результаты данных экспериментов приведены в таблице 4.

Таблица 4 – Результаты обучения групп моделей

| Модель   | Точность без применения методики | Точность с методикой | Улучшение     |
|----------|----------------------------------|----------------------|---------------|
| MNIST    | 98,00%                           | <b>98,18%</b>        | <b>+0,18%</b> |
| ResNet18 | 81,65%                           | <b>82,73%</b>        | <b>+1,08%</b> |

Как можно увидеть, применение данной методики обеспечивает улучшение метрик точности, что подтверждает её работоспособность.

## **Выводы**

Результатом данной работы является предложенная методика оптимизации гиперпараметров, применимая в задачах оптической автономной навигации в составе нейросетевой модели предсказания глубины. Методика позволяет обеспечить улучшение показателя метрик при одинаковом количестве итераций подбора гиперпараметров и была проверена на различных задачах обучения нейронных сетей. В перспективе данная методика способна обеспечить снижение затрат на разработку и обучение нейросетевых моделей при увеличении целевых характеристик, что положительно скажется на точности автономной навигации БЛА.

## **Список источников**

1. Forster C., Lynen S., Kneip L., Scaramuzza D. et al. Collaborative monocular slam with multiple micro aerial vehicles // 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2013. P. 3962-3970. DOI: [10.1109/IROS.2013.6696923](https://doi.org/10.1109/IROS.2013.6696923)
2. Blösch M., Weiss S., Scaramuzza D., Siegwart R. et al. Vision based MAV navigation in unknown and unstructured environments // 2010 IEEE International Conference on Robotics and Automation. IEEE, 2010. P. 21-28. DOI: [10.1109/ROBOT.2010.5509920](https://doi.org/10.1109/ROBOT.2010.5509920)
3. Олькина Д.С. Алгоритм семантической сегментации изображений для решения задачи позиционирования летательного аппарата на земной поверхности // Труды МАИ. 2023. № 130. URL: <https://trudymai.ru/published.php?ID=174617>. DOI: [10.34759/trd-2023-130-18](https://doi.org/10.34759/trd-2023-130-18)

4. Бурага А.В., Костюков В.М. Сравнительный анализ пассивных методов измерения дальности для малого беспилотного летательного аппарата // Труды МАИ. 2012. № 53. URL: <https://trudymai.ru/published.php?ID=29624>
5. Li R., Wang S., Gu D. Ongoing evolution of visual SLAM from geometry to deep learning: Challenges and opportunities // Cognitive Computation. 2018. V. 10, No 6. P. 875-889. DOI: [10.1007/s12559-018-9591-8](https://doi.org/10.1007/s12559-018-9591-8)
6. Wang W., Hu Y., Scherer S. TartanVO: A generalizable learning-based VO // Conference on Robot Learning. 2021. P. 1761-1772. DOI: [10.48550/arXiv.2011.00359](https://doi.org/10.48550/arXiv.2011.00359)
7. Van Rijn J. N., Hutter F. Hyperparameter importance across datasets // Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018. P. 2367-2376. DOI: [10.1145/3219819.3220058](https://doi.org/10.1145/3219819.3220058)
8. Radosavovic I., Kosaraju R.P., Girshick R., He K., Dollár P. et al. Designing network design spaces // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020. P. 10428-10436. DOI: [10.1109/CVPR42600.2020.01044](https://doi.org/10.1109/CVPR42600.2020.01044)
9. Wistuba M., Schilling N., Schmidt-Thieme L. Hyperparameter search space pruning—a new component for sequential model-based hyperparameter optimization // Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II 15. Springer International Publishing, 2015. P. 104-119. DOI: [10.1007/978-3-319-23525-7\\_7](https://doi.org/10.1007/978-3-319-23525-7_7)
10. Akiba T. et al. Optuna: A next-generation hyperparameter optimization framework // Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019. P. 2623-2631. DOI: [10.1145/3292500.3330701](https://doi.org/10.1145/3292500.3330701)

11. Makarov A., Namiot D. Overview of data cleaning methods for machine learning // International Journal of Open Information Technologies. 2023. V. 11, No. 10. P. 70-78.
12. Bhat S.F., Birkel R., Wofk D., Wonka P., Müller M. et al. Zoedepth: Zero-shot transfer by combining relative and metric depth // arXiv preprint arXiv:2302.12288. 2023. DOI: [10.48550/arXiv.2302.12288](https://doi.org/10.48550/arXiv.2302.12288)
13. Alhashim I. High quality monocular depth estimation via transfer learning // arXiv preprint arXiv:1812.11941. 2018. URL: <https://doi.org/10.48550/arXiv.1812.11941>
14. Wang Z., Bovik A.C., Sheikh H.R., Simoncelli E.P. et al. Image quality assessment: from error visibility to structural similarity // IEEE transactions on image processing. 2004. V. 13, No. 4. P. 600-612. DOI: [10.1109/TIP.2003.819861](https://doi.org/10.1109/TIP.2003.819861)
15. Kanopoulos N., Vasanthavada N., Baker R.L. Design of an image edge detection filter using the Sobel operator // IEEE Journal of solid-state circuits. 1988. V. 23, No. 2. P. 358-367. URL: <https://doi.org/10.1109/4.996>
16. Lee K., Yim J. Hyperparameter optimization with neural network pruning // arXiv preprint arXiv:2205.08695. 2022. DOI: [10.48550/arXiv.2205.08695](https://doi.org/10.48550/arXiv.2205.08695)
17. Engel J., Koltun V., Cremers D. Direct sparse odometry // IEEE transactions on pattern analysis and machine intelligence. 2017. V. 40, No. 3. P. 611-625. DOI: [10.1109/TPAMI.2017.2658577](https://doi.org/10.1109/TPAMI.2017.2658577)
18. Mokssit S. et al. Deep learning techniques for visual slam: A survey // IEEE Access. 2023. V. 11, P. 20026-20050. DOI: [10.1109/ACCESS.2023.3249661](https://doi.org/10.1109/ACCESS.2023.3249661)

19. Bischl B. et al. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2023. V. 13, No. 2. P. 1484. DOI: [10.1002/widm.1484](https://doi.org/10.1002/widm.1484)
20. Yang L., Shami A. On hyperparameter optimization of machine learning algorithms: Theory and practice // Neurocomputing. 2020. V. 415, P. 295-316. DOI: [10.1016/j.neucom.2020.07.061](https://doi.org/10.1016/j.neucom.2020.07.061)
21. Bartz E. et al. Experimental investigation and evaluation of model-based hyperparameter optimization // arXiv preprint arXiv:2107.08761. 2021. DOI: [10.48550/arXiv.2107.08761](https://doi.org/10.48550/arXiv.2107.08761)

## References

1. Forster C., Lynen S., Kneip L., Scaramuzza D. et al. Collaborative monocular slam with multiple micro aerial vehicles. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013. P. 3962-3970. DOI: [10.1109/IROS.2013.6696923](https://doi.org/10.1109/IROS.2013.6696923)
2. Blösch M., Weiss S., Scaramuzza D., Siegwart R. et al. Vision based MAV navigation in unknown and unstructured environments. *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010. P. 21-28. DOI: [10.1109/ROBOT.2010.5509920](https://doi.org/10.1109/ROBOT.2010.5509920)
3. Ol'kina D.S. Algorithm of semantic image segmentation for solving the problem of positioning an aircraft on the Earth's surface. *Trudy MAI*. 2023. No. 130. (In Russ.). URL: <https://trudymai.ru/eng/published.php?ID=174617>. DOI: [10.34759/trd-2023-130-18](https://doi.org/10.34759/trd-2023-130-18)

4. Buraga A.V., Kostyukov V.M. Comparison of range estimation methods for small unmanned aerial vehicle. *Trudy MAI*. 2012. No. 53. (In Russ.). URL: <https://trudymai.ru/eng/published.php?ID=29624>
5. Li R., Wang S., Gu D. Ongoing evolution of visual SLAM from geometry to deep learning: Challenges and opportunities. *Cognitive Computation*. 2018. V. 10, No 6. P. 875-889. DOI: [10.1007/s12559-018-9591-8](https://doi.org/10.1007/s12559-018-9591-8)
6. Wang W., Hu Y., Scherer S. TartanVO: A generalizable learning-based VO. *Conference on Robot Learning*. 2021. P. 1761-1772. DOI: [10.48550/arXiv.2011.00359](https://doi.org/10.48550/arXiv.2011.00359)
7. Van Rijn J. N., Hutter F. Hyperparameter importance across datasets. *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018. P. 2367-2376. DOI: [10.1145/3219819.3220058](https://doi.org/10.1145/3219819.3220058)
8. Radosavovic I., Kosaraju R.P., Girshick R., He K., Dollár P. et al. Designing network design spaces. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020. P. 10428-10436. DOI: [10.1109/CVPR42600.2020.01044](https://doi.org/10.1109/CVPR42600.2020.01044)
9. Wistuba M., Schilling N., Schmidt-Thieme L. Hyperparameter search space pruning—a new component for sequential model-based hyperparameter optimization. *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II* 15. Springer International Publishing, 2015. P. 104-119. DOI: [10.1007/978-3-319-23525-7\\_7](https://doi.org/10.1007/978-3-319-23525-7_7)
10. Akiba T. et al. Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019. P. 2623-2631. DOI: [10.1145/3292500.3330701](https://doi.org/10.1145/3292500.3330701)

11. Makarov A., Namiot D. Overview of data cleaning methods for machine learning. *International Journal of Open Information Technologies*. 2023. V. 11, No. 10. P. 70-78.
12. Bhat S.F., Birkel R., Wofk D., Wonka P., Müller M. et al. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*. 2023. DOI: [10.48550/arXiv.2302.12288](https://doi.org/10.48550/arXiv.2302.12288)
13. Alhashim I. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*. 2018. URL: <https://doi.org/10.48550/arXiv.1812.11941>
14. Wang Z., Bovik A.C., Sheikh H.R., Simoncelli E.P. et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*. 2004. V. 13, No. 4. P. 600-612. DOI: [10.1109/TIP.2003.819861](https://doi.org/10.1109/TIP.2003.819861)
15. Kanopoulos N., Vasanthavada N., Baker R.L. Design of an image edge detection filter using the Sobel operator. *IEEE Journal of solid-state circuits*. 1988. V. 23, No. 2. P. 358-367. URL: <https://doi.org/10.1109/4.996>
16. Lee K., Yim J. Hyperparameter optimization with neural network pruning. *arXiv preprint arXiv:2205.08695*. 2022. DOI: [10.48550/arXiv.2205.08695](https://doi.org/10.48550/arXiv.2205.08695)
17. Engel J., Koltun V., Cremers D. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*. 2017. V. 40, No. 3. P. 611-625. DOI: [10.1109/TPAMI.2017.2658577](https://doi.org/10.1109/TPAMI.2017.2658577)
18. Mokssit S. et al. Deep learning techniques for visual slam: A survey. *IEEE Access*. 2023. V. 11, P. 20026-20050. DOI: [10.1109/ACCESS.2023.3249661](https://doi.org/10.1109/ACCESS.2023.3249661)

19. Bischl B. et al. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 2023. V. 13, No. 2. P. 1484. DOI: [10.1002/widm.1484](https://doi.org/10.1002/widm.1484)
20. Yang L., Shami A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*. 2020. V. 415, P. 295-316. DOI: [10.1016/j.neucom.2020.07.061](https://doi.org/10.1016/j.neucom.2020.07.061)
21. Bartz E. et al. Experimental investigation and evaluation of model-based hyperparameter optimization. *arXiv preprint arXiv:2107.08761*. 2021. DOI: [10.48550/arXiv.2107.08761](https://doi.org/10.48550/arXiv.2107.08761)

Статья поступила в редакцию 13.01.2025

Одобрена после рецензирования 28.01.2025

Принята к публикации 25.06.2025

The article was submitted on 13.01.2025; approved after reviewing on 28.01.2025; accepted for publication on 25.06.2025