

**Описание технической архитектуры
программного обеспечения
«Система поиска Family КОЗ 1 Автономный поиск»**

Москва, 2025

СОДЕРЖАНИЕ

Термины и определения	3
1. Общие сведения	4
2. Описание архитектуры приложения	5
2.1. Описание функций и файлов	5
2.2. Диаграмма последовательностей	8
3. Описание архитектурного стиля	9

Термины и определения

- БВС – беспилотное воздушное судно;
НТИ – национальная технологическая инициатива;
ПО – программное обеспечение;

1. Общие сведения

Настоящий документ содержит описание технической архитектуры программного обеспечения «Система поиска Family КОЗ 1 Автономный поиск».

Исключительные права на программное обеспечение принадлежат Фонду НТИ (далее – Компания).

Настоящий документ подлежит размещению на официальном сайте Компании в сети Интернет по адресу: <https://nti.fund/about/activity/information.php> (далее – официальный сайт).

ПО «Система поиска Family КОЗ 1 Автономный поиск» предназначено для автоматизированного поиска людей в природной местности. ПО обеспечивает автоматическое распознавание людей на фотографиях, полученных после полета БВС.

ПО не имеет пользовательского графического интерфейса и представляет собой модуль на языке Python 3.10 для запуска на одноплатных компьютерах Jetson Orin Nano 8gb, допускающих размещение на борту БВС.

2. Описание архитектуры приложения

2.1. Описание функций и файлов

ПО «Система поиска Family КОЗ 1 Автономный поиск» состоит из следующих файлов и функций:

1. Файл «solution.py» - основной модуль программы. Содержит в себе необходимые функции для запуска процесса детектирования объектов поиска (живые люди, манекены).

В файле «solution.py» содержатся следующие основные функции:

–Функция «predict(images: Union[List[np.ndarray], np.ndarray]) -> List[List[dict]]» - основная функция программы. Функция производит инференс модели нейросети на одном или нескольких изображениях.

Входящие аргументы:

images (Union[List[np.ndarray], np.ndarray]): Список изображений или одно изображение.

Возвращаемый результат:

List[List[dict]]: Список списков словарей с результатами предикта

на найденных изображениях.

Пример выходных данных:

```
[  
  [  
    {  
      'xc': 0.5,  
      'yc': 0.5,  
      'w': 0.2,  
      'h': 0.3,  
      'label': 0,  
      'score': 0.95
```

```

        },
        ...
    ],
    ...
]

```

Функция «infer_image_bbox(image: np.ndarray) -> List[dict]» – вспомогательная функция для получения ограничивающих рамок объектов на изображении.

Входящие аргументы:

image (np.ndarray): Изображение, на котором будет производиться инференс.

Возвращаемый результат:

List[dict]: Список словарей с координатами ограничивающих рамок и оценками.

Пример выходных данных:

```

[
    {
        'xc': 0.5,
        'yc': 0.5,
        'w': 0.2,
        'h': 0.3,
        'label': 0,
        'score': 0.95
    },
    ...
]

```

–Функция «resize(image, new_shape=inference_shape)» –
вспомогательная функция для изменения размера изображения,
оптимизированная для устройства Jetson.

2. Файл «11m_tune_best.engine» - содержит обученные веса для модели «YOLOv11-m», сконвертированные в формат TensorRT для устройства Jetson Orin Nano 8gb.

2.2. Диаграмма последовательностей

Диаграмма последовательности программного обеспечения «Система поиска Family КОЗ 1 Автономный поиск» приведена на рисунке:



При выполнении программы объект «Пользователь» передает список с фотографиями, полученными с БВС, в качестве параметра объекту ПО «Система поиска Family КОЗ 1 Автономный поиск». После получения данных заказа от объекта «Пользователь» объект «Система поиска Family КОЗ 1 Автономный поиск» осуществляет детектирование объектов поиска (живые люди, манекены) для каждой фотографии из предоставленного каталога. После детектирования «Система поиска Family КОЗ 1 Автономный поиск» возвращает результаты поиска в виде списка.

3. Описание архитектурного стиля

Программное обеспечение «Система поиска Family КОЗ 1 Автономный поиск» разработано с использованием преимущественно функционального стиля программирования.

При подготовке программного обеспечения использовались библиотеки с открытым исходным кодом.

Программное обеспечение разработано на языке программирования Python 3.10 в среде разработки VS Code.

Для связи с разработчиками писать на почту ntifundsoft@nti.fund