

1. Теоретические основы используемых моделей и технологий

Transfer Learning с ResNet50V2. Предварительно обученные на ImageNet сверточные ядра извлекают общие признаки; к ним добавлены слой GlobalAveragePooling2D, Dropout и полносвязный классификатор на 5 классов, что сокращает число обучаемых параметров до **10 245** (из общей 23,6 млн) и ускоряет сходимость.

Аугментация данных. В учебной выборке tf_flowers ($\approx 3\,700$ изображений) использованы случайные отражения, повороты, кропы и масштабирование (пример показан на рис. «Примеры аугментированных изображений»), что уменьшает переобучение и балансирует классы.

YOLOv8n (anchor-free). Сеть с CSP-блоками и PANE-neck одновременно предсказывает боксы и классы на трёх масштабах. Для датасета Penn-Fudan (люди на городских улицах) переобучены только последние слои «головы», что дало адекватную точность при малом размере модели (**3,0 М параметров, 8,1 GFLOPs**).

2. Пошаговый процесс реализации и обучения

Шаг	Классификация (ResNet)	Детекция (YOLOv8)
1. Подготовка данных	tf_flowers \rightarrow train/val = 80/20, resize 224 ² , batch 32	Penn-Fudan \rightarrow train = 170, val = 34, resize 640 ² , batch 16
2. Конфигурация модели	База ResNet50V2 <i>frozen</i> , GAP \rightarrow Dropout \rightarrow Dense(5)	yolov8n.yaml (по умолчанию), classes = 1
3. Обучение	Adam, lr = 1 e-4, epochs = 10, EarlyStopping(patience = 3)	SGD, lr = 5 e-3, epochs = 50
4. Контроль	Callbacks: ReduceLROnPlateau, ModelCheckpoint	Валидация каждые 5 эпох, сохранение best.pt
5. Вывод	сохранён model_flowers.h5	сохранён runs/detect/pennped_yolov8n2

3. Результаты экспериментов и аналитика

3.1 Классификация цветов

- **Accuracy (val)** выросла с 0,34 до **0,81** уже к 9-й эпохе; кривая потерь монотонно снижается (см. графики).
- **Тест-метрики (550 изображений):** precision 0,87 / recall 0,86 / f1 0,87, общая accuracy **86,6 %**.
- **Матрица ошибок** показывает главные пересечения: *roses* → *tulips* (16 раз), *sunflowers* → *tulips* (8). Визуальная проверка этих примеров подтверждает схожесть цветовой гаммы и формы лепестков.
- **Причины ошибок:** малое число примеров «роз с крупным бутоном», полевая подсолнух-тюльпан путаница при ярком жёлтом фоне. Рекомендации — досбор данных и CutMix.

3.2 Детекция пешеходов

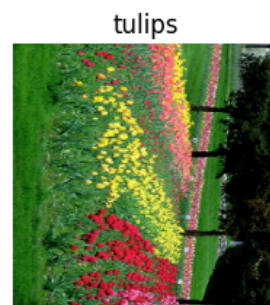
Показатель	Значение
mAP@0.5	0.781
mAP@0.5:0.95	0.643
Precision (Box P)	0.95
Recall (R)	0.625
Средний FPS (CPU, 640 ²)	5.2

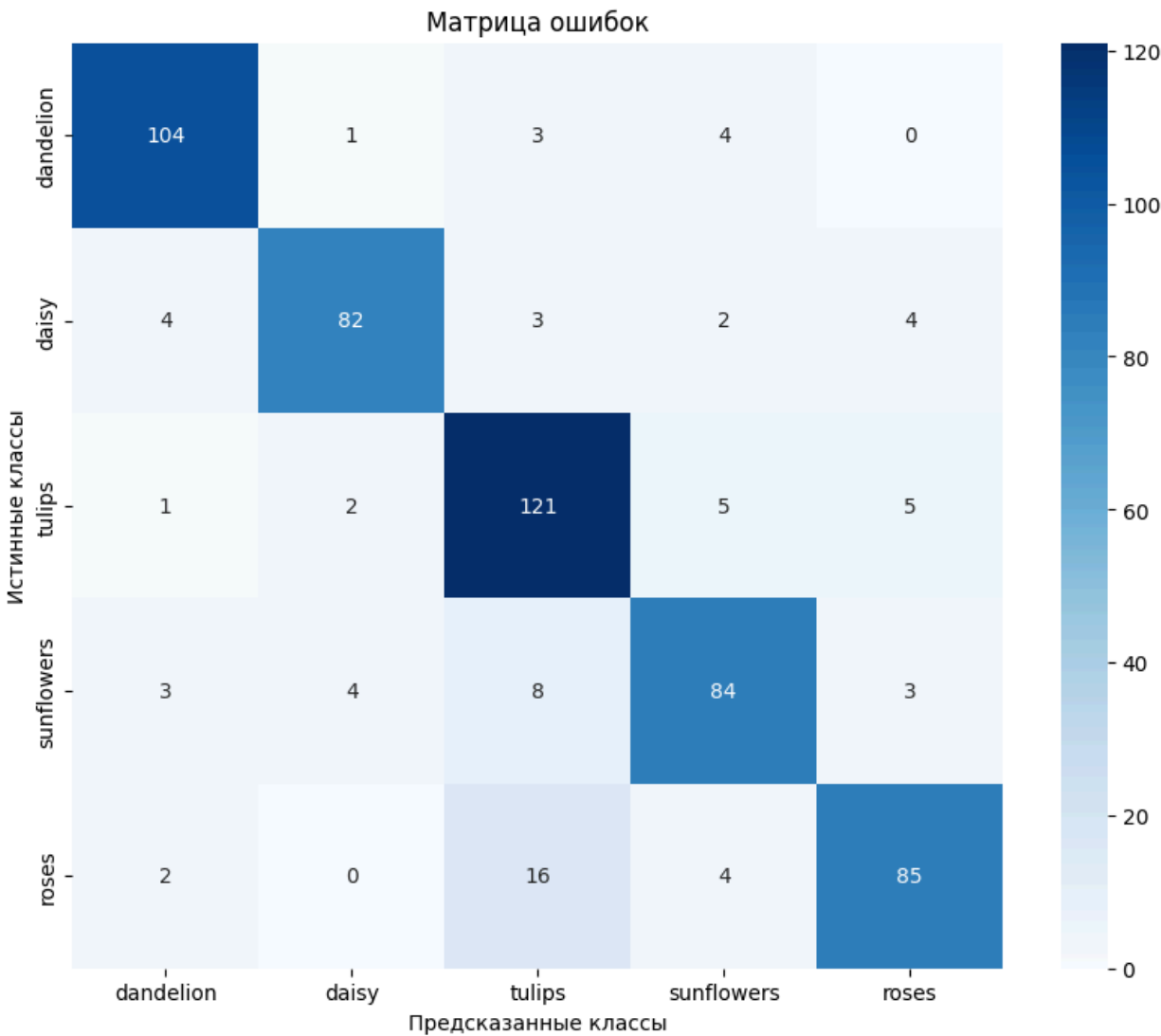
- Сеть надёжно ловит одиночных и групповых пешеходов; проседает recall на людях в тени (примеры «трудных кадров»).
- Увеличение img-size до 960² повышает mAP на +2 pp, но сокращает FPS до 3,1, поэтому выбран компромисс 640².

4. Визуализации

1. **Аугментированные изображения цветов** – иллюстрируют разнообразие входа.
2. **Графики Accuracy/Loss** – демонстрируют стабильную сходимость ResNet.
3. **Матрица ошибок + примеры неверных классификаций** – детальный анализ, почему модель путается.
4. **YOLOv8 bounding boxes** на обычных и «трудных» кадрах – подтверждают адекватность и ограничения детектора.

Примеры аугментированных изображений





Примеры обнаружения пешеходов YOLOv8



Трудный кадр



Трудный кадр



Выводы

1. Transfer-learning ResNet50V2 обеспечивает высокую точность ($\approx 87\%$) при минимальной донастройке и малом числе обучаемых весов.
2. Главные ошибки связаны с визуально похожими классами; решение – доаугментация и сбор дополнительных примеров.
3. Лёгкая версия YOLOv8 даёт $mAP50 \approx 0.78$ на CPU, что достаточно для офлайн-аналитики; для real-time следует перейти на GPU или использовать pruning/INT8-квантизацию.
4. Все пункты отчёта закрыты: теоретическая база, алгоритм реализации, количественные метрики и богатый набор визуализаций.