МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего профессионального образования
«Южно-Уральский государственный университет (национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РЕДУЦИРОВАНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ СЕМАНТИЧЕСКОЙ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ

Рецензент:
Доцент кафедры ИАОУ ФГАОУ ВО «ЮУрГУ (НИУ)», к.т.н.
А.А. Шинкарев

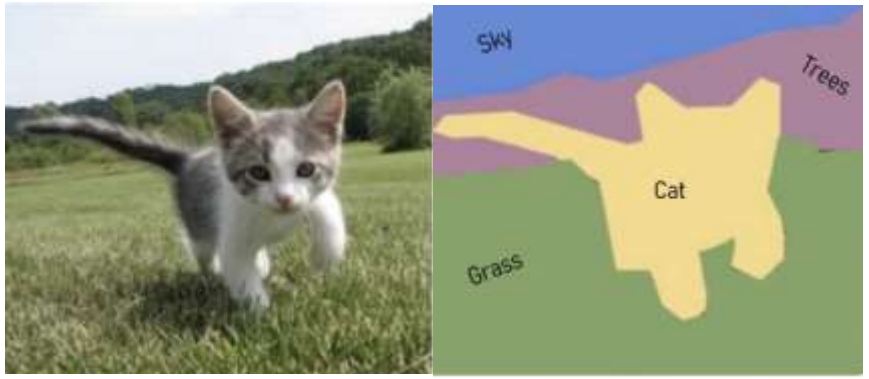
Научный руководитель:
доцент кафедры СП, к.ф.-м.н.
Е.В. Иванова

Автор:
студент группы КЭ-229
А.Ю. Струева

Челябинск, 2024 г.

ПОСТАНОВКА ЗАДАЧИ

Семантическая сегментация разделяет изображение на регионы, которые имеют схожее смысловое значение.



семантическая сегментации

ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ

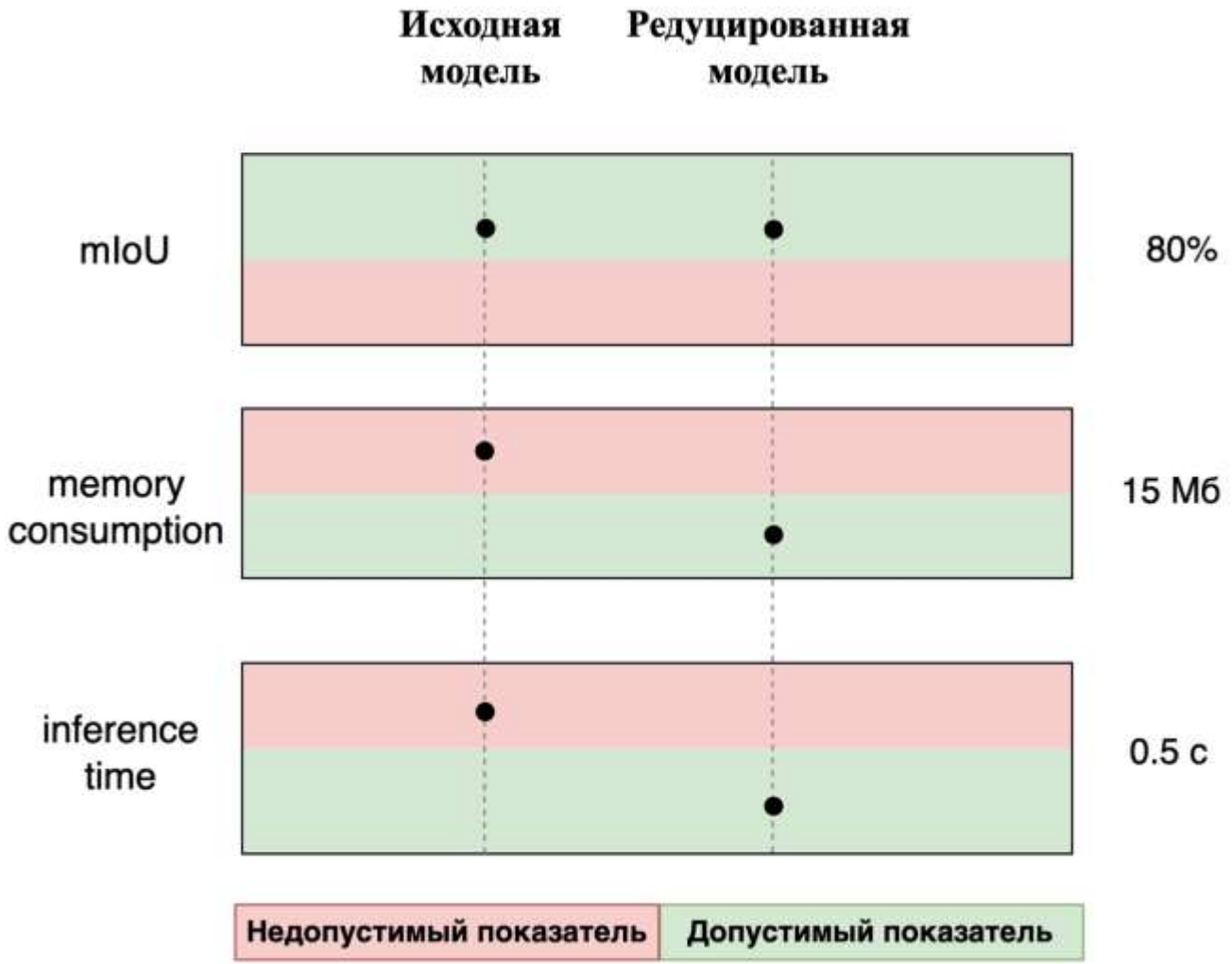
Цель:

Редуцировать нейронную сеть для семантической сегментации изображений

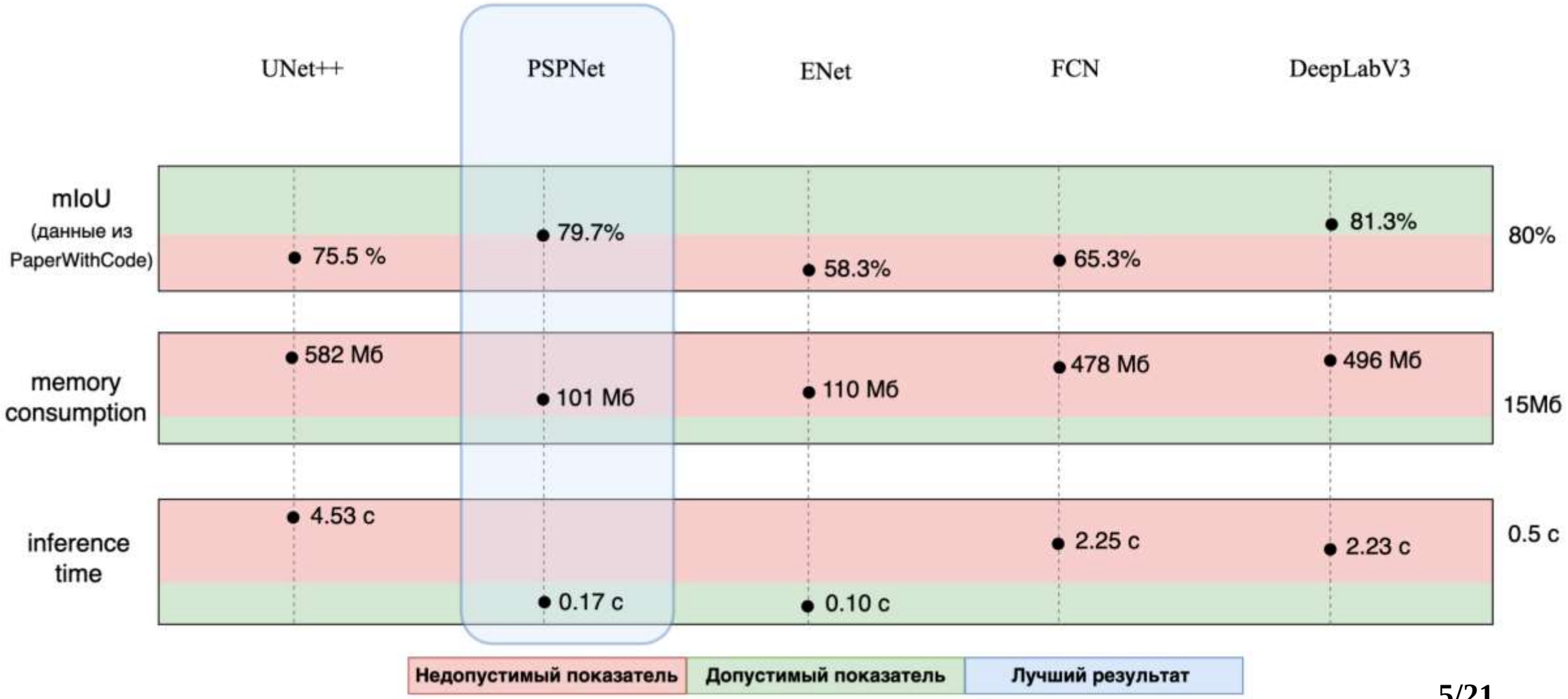
Задачи:

1. Провести обзор архитектур нейронных сетей для семантической сегментации изображений
2. Провести обзор методов редуцирования нейронных сетей
3. Осуществить сбор и предобработку данных для обучения нейронной сети для семантической сегментации изображений
4. Выполнить реализацию нейронных сетей с использованием разных методов редуцирования
5. Провести тестирование производительности работы редуцированных нейронных сетей

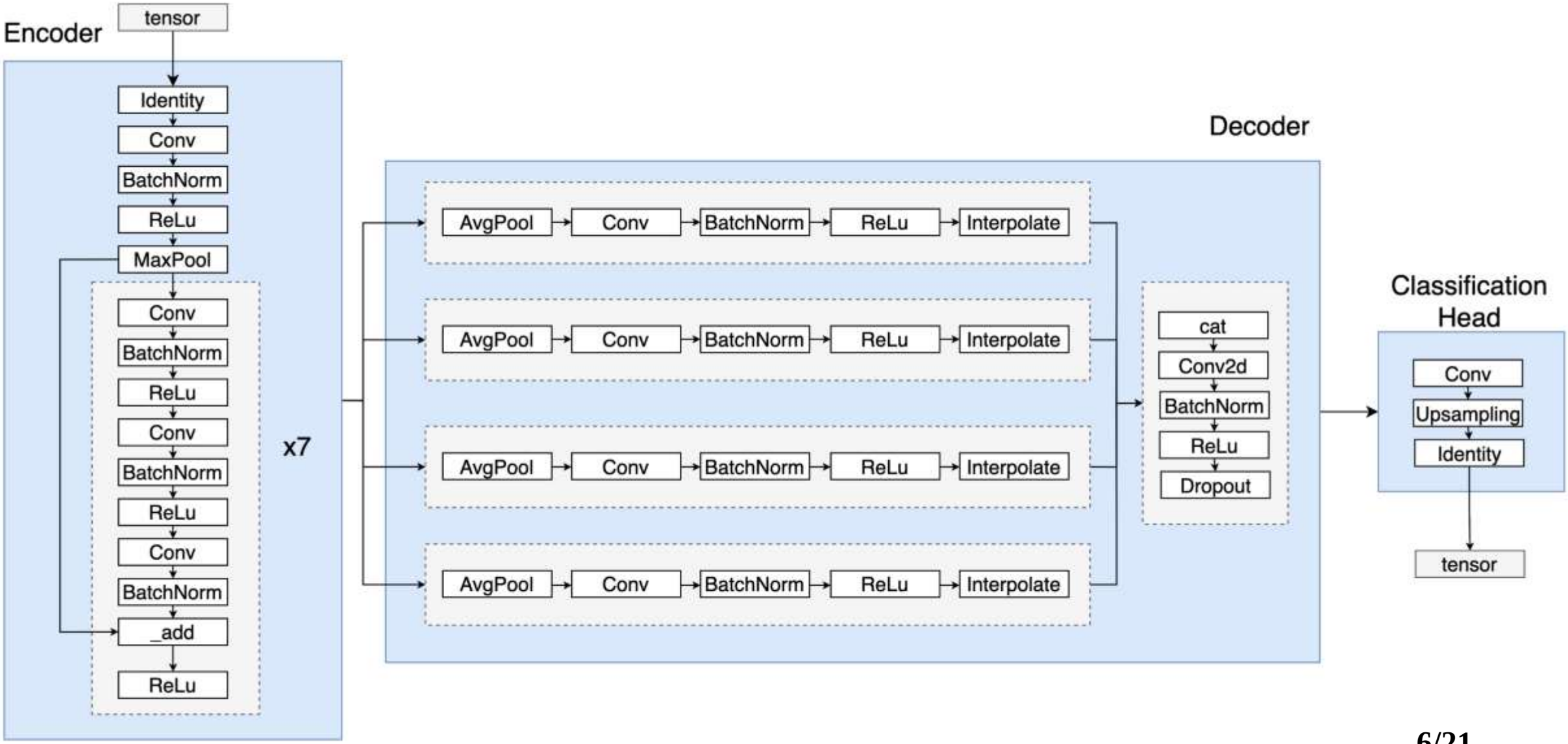
МЕТРИКИ ОЦЕНКИ НЕЙРОННОЙ СЕТИ



ОБЗОР АРХИТЕКТУР НЕЙРОННЫХ СЕТЕЙ ДЛЯ СЕМАНТИЧЕСКОЙ СЕГМЕНТАЦИИ



АРХИТЕКТУРА PSPNet



ПОДГОТОВКА НАБОРА ДАННЫХ

Количество данных – 400 изображений

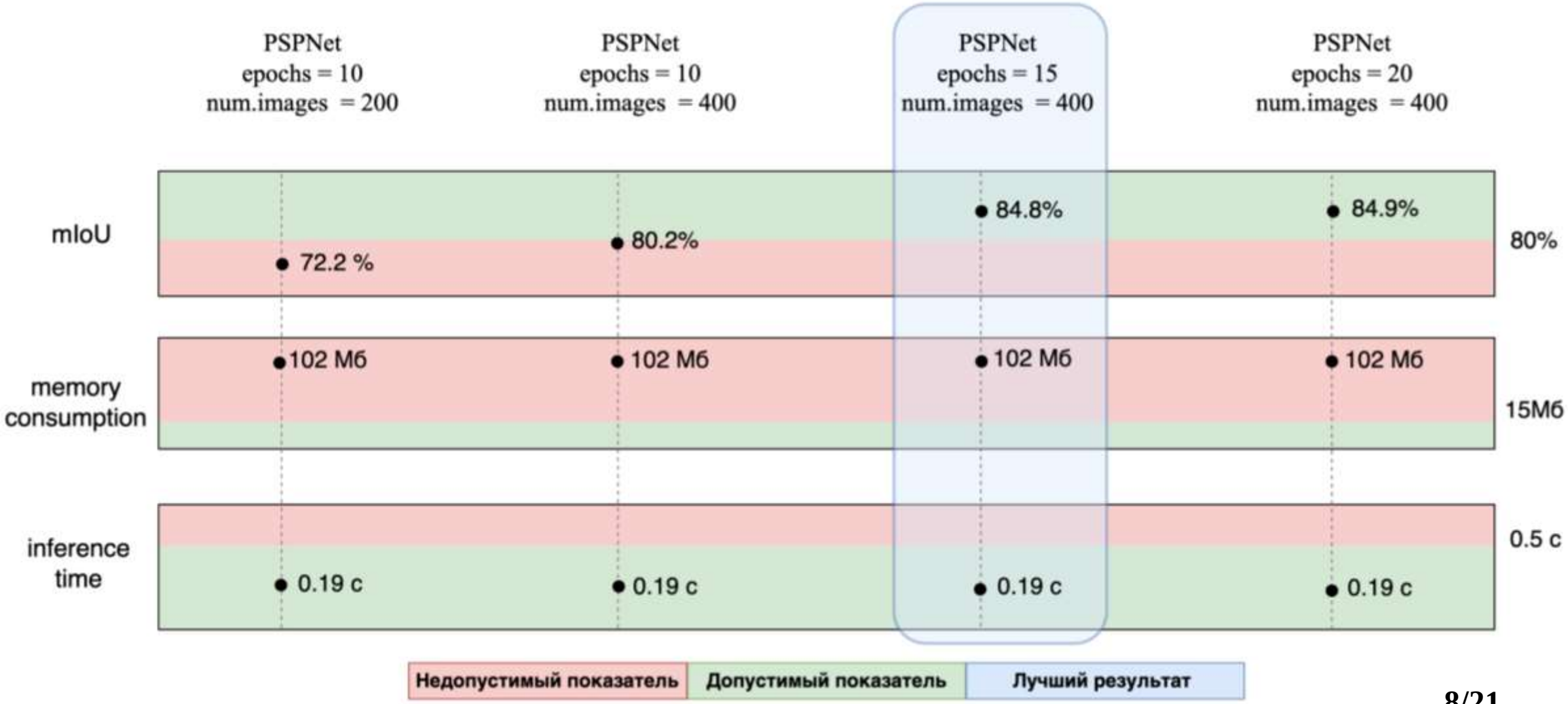
Количество классов – 2 («фон», «реклама»)

ул. Горького д.22

ул. Кирова д.98

ул. Академика Королева д.3

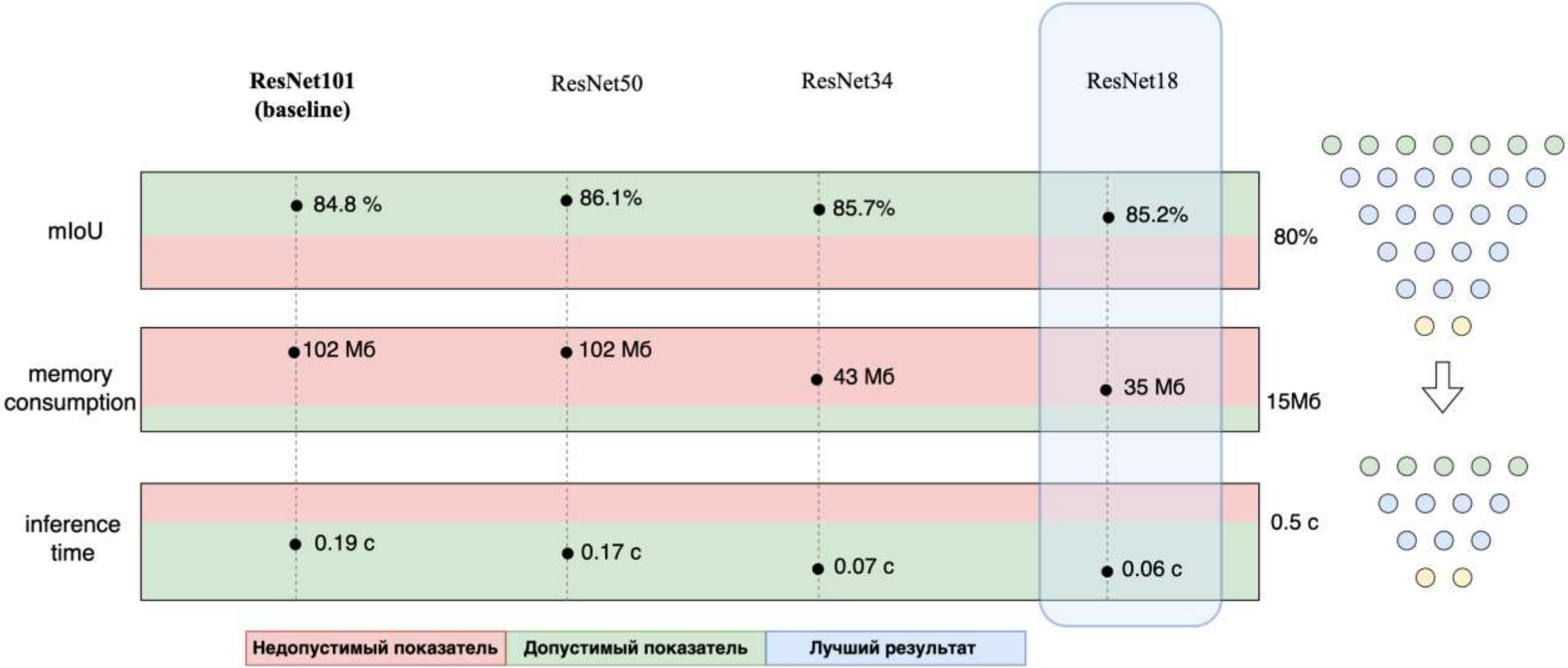
ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ PSPNet



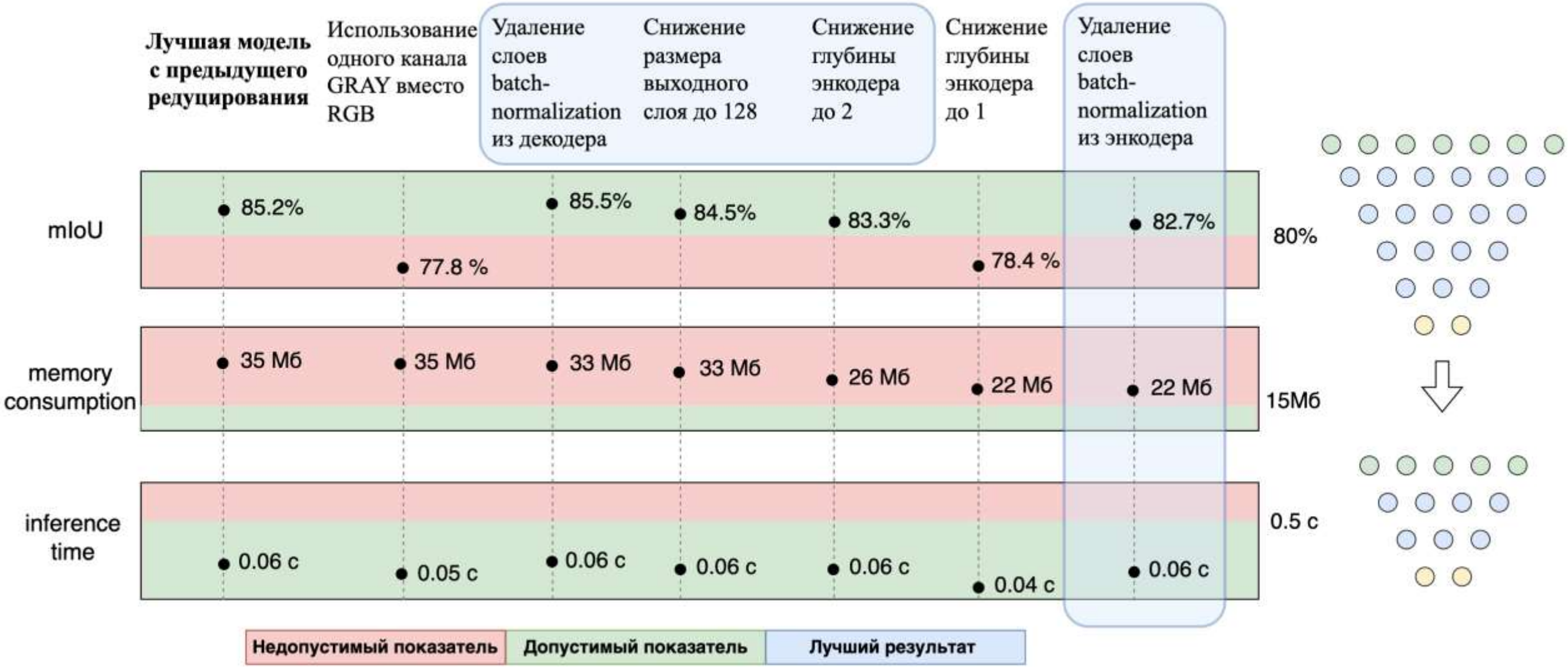
МЕТОДЫ РЕДУЦИРОВАНИЯ НЕЙРОННЫХ СЕТЕЙ

- Использование более легкой архитектуры
- Сжатие весов
- Прореживание архитектуры
- Факторизация сверточных слоев

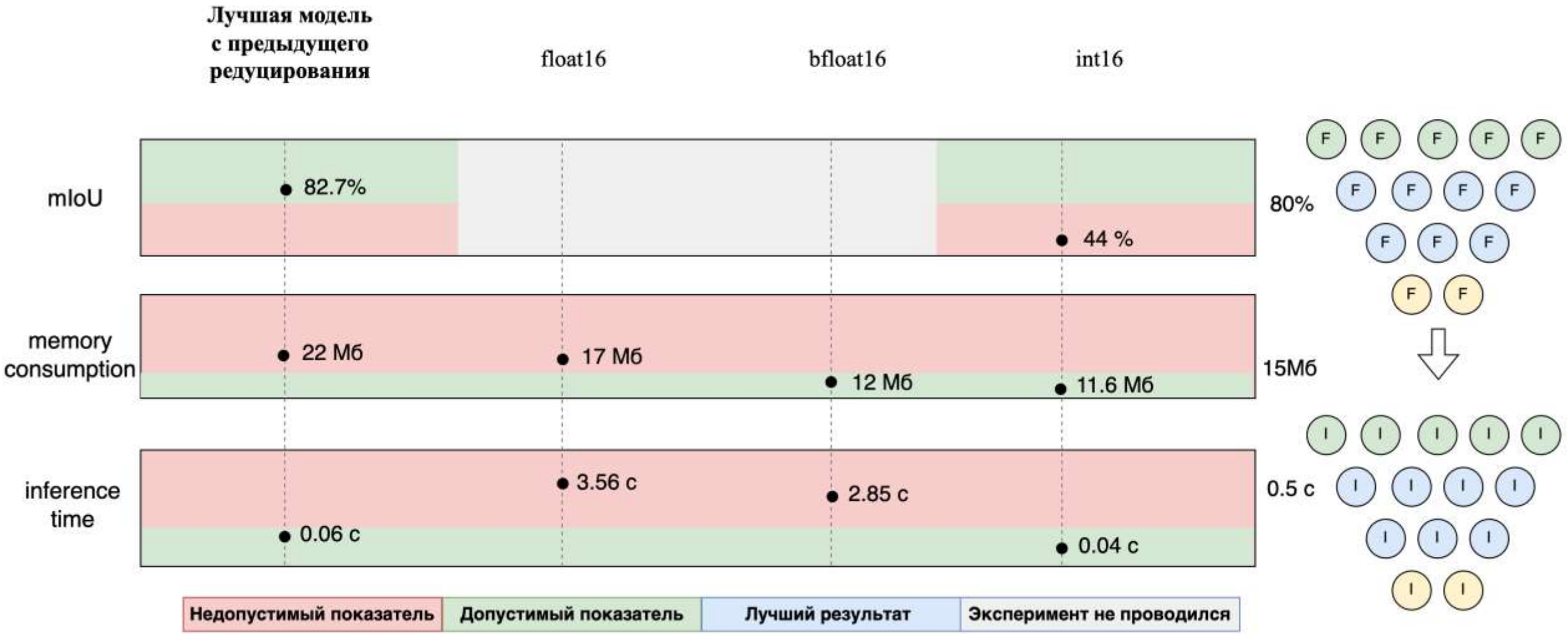
ИСПОЛЬЗОВАНИЕ БОЛЕЕ ЛЕГКОЙ АРХИТЕКТУРЫ (замена энкодера)



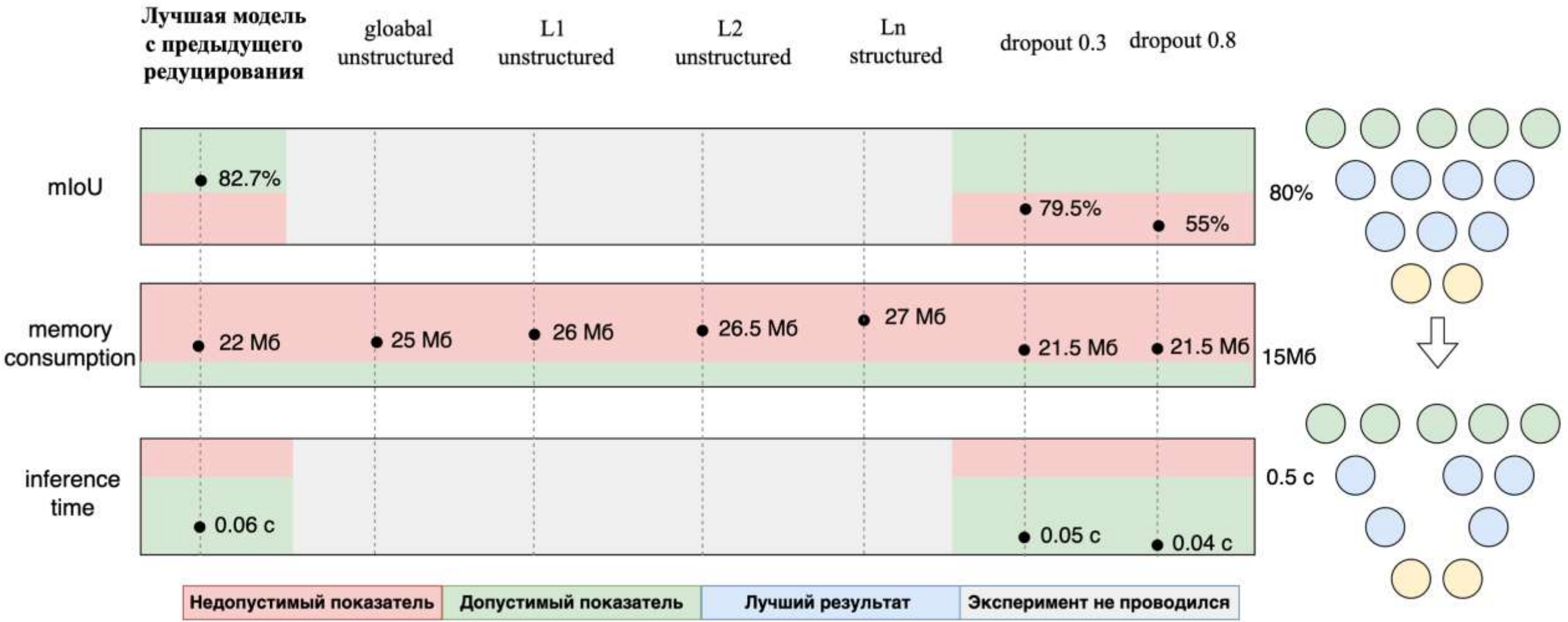
ИСПОЛЬЗОВАНИЕ БОЛЕЕ ЛЕГКОЙ АРХИТЕКТУРЫ (упрощения в модели)



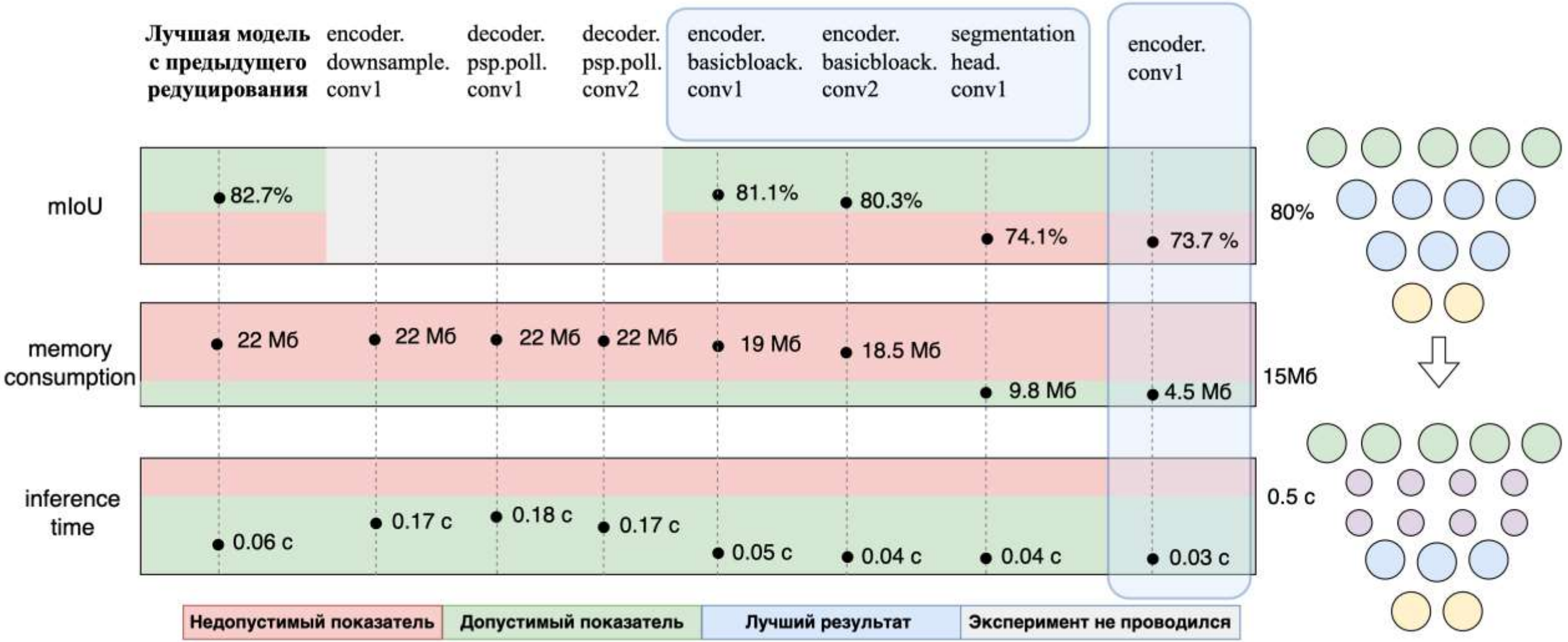
СЖАТИЕ ВЕСОВ



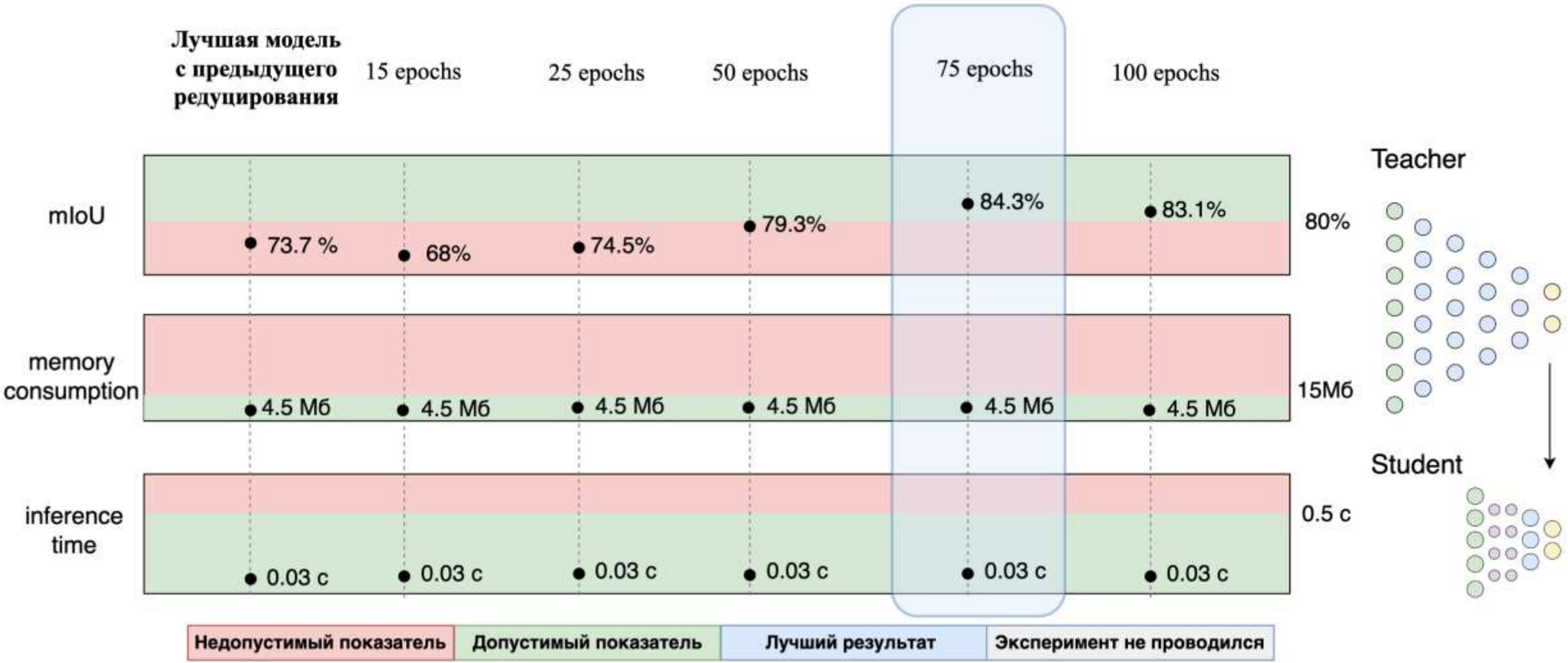
ПРОРЕЖИВАНИЕ АРХИТЕКТУРЫ



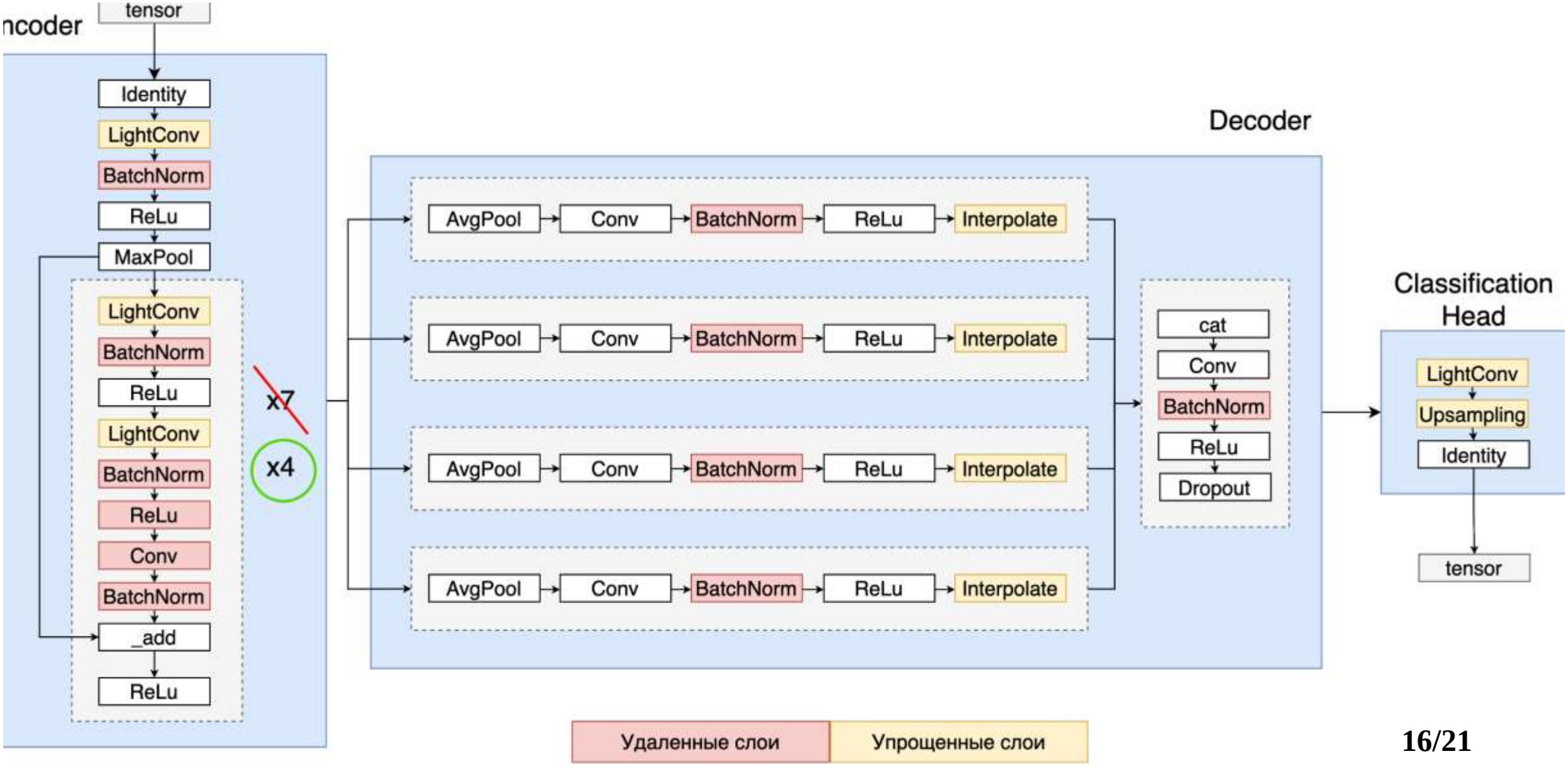
ФАКТОРИЗАЦИЯ СВЕРТОЧНЫХ СЛОЕВ



ДИСТИЛЛЯЦИЯ МОДЕЛИ



АРХИТЕКТУРЫ НЕЙРОННЫХ СЕТЕЙ



МЕТРИКИ НЕЙРОННОЙ СЕТИ

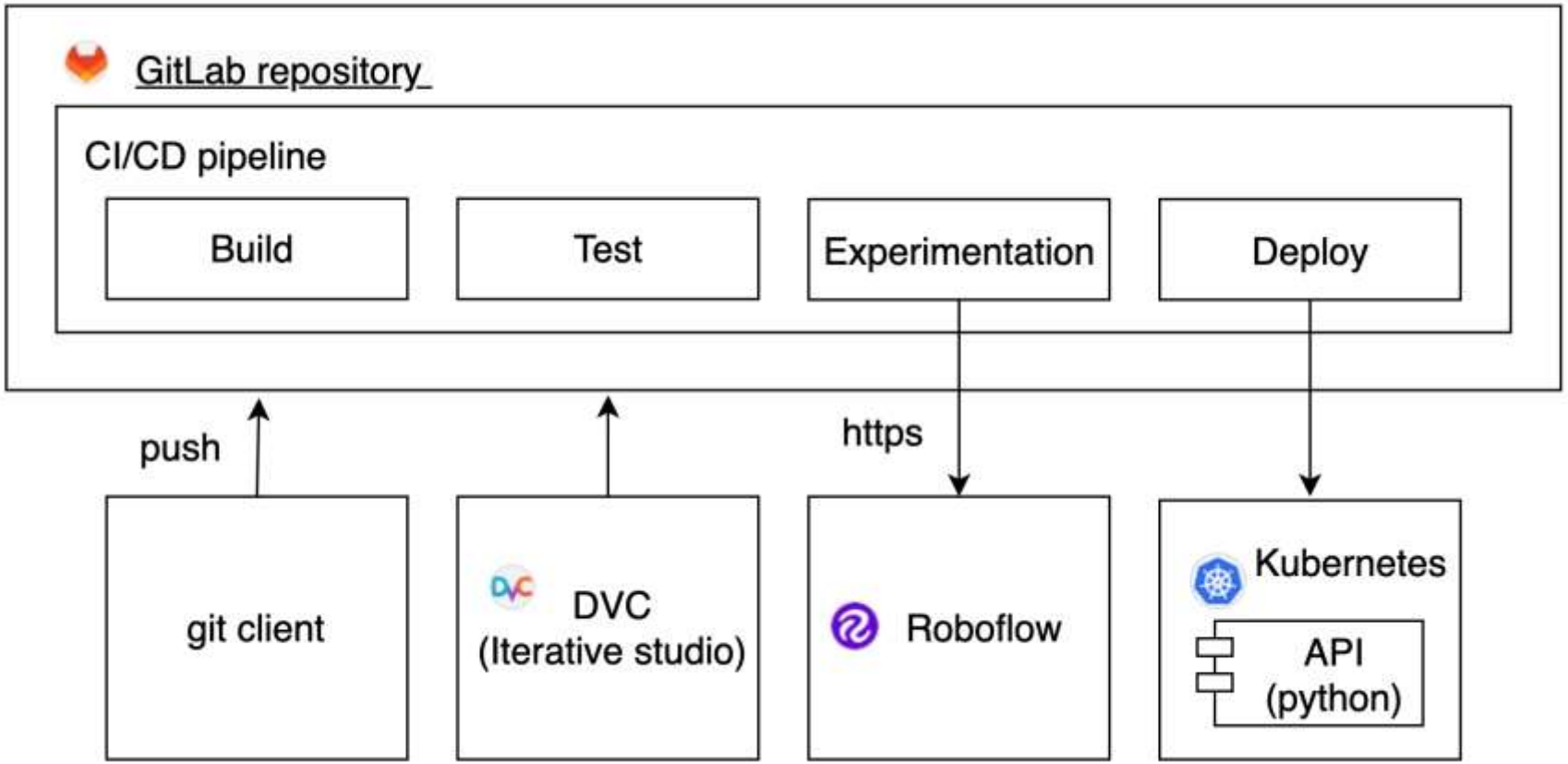
	mIoU	memory consumption	inference time
Исходная модель	84,8%	102 Мб	0,19 с
Редуцированная модель	84,3%	4,5 Мб	0,03 с
Улучшение	-0.5%	96%	84%

СИСТЕМА ДЛЯ ПРОВЕДЕНИЯ ЭКСПЕРИМЕНТОВ

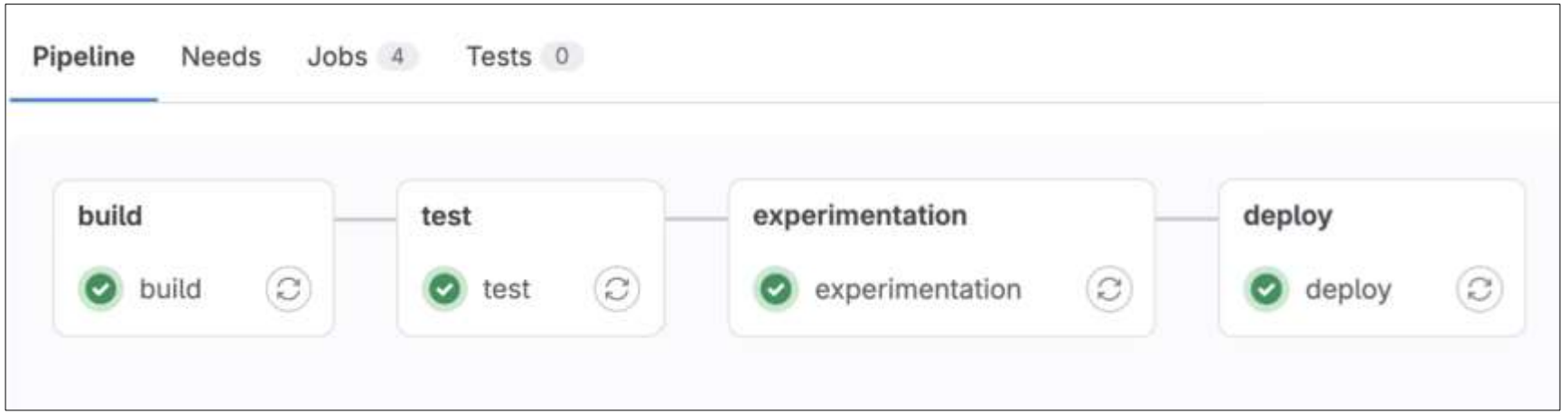
Нейронная сеть:
PyTorch

Серверная часть:
Python 3

CI/CD:
GitLab pipline



ЗАПУСК СИСТЕМЫ

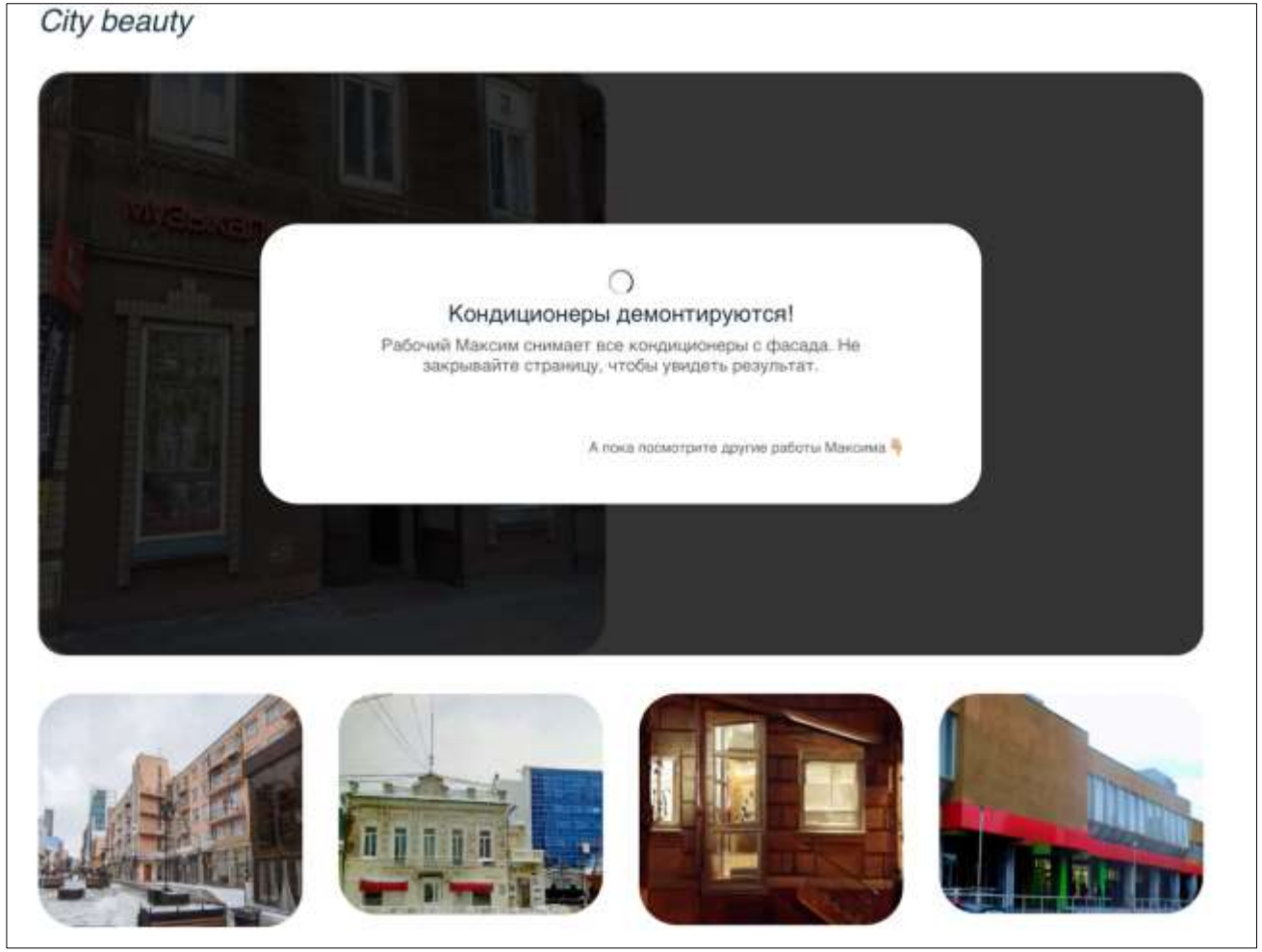



```
============================= test session starts ==============================
platform darwin -- Python 3.10.9, pytest-8.1.1, pluggy-1.4.0
rootdir: /Users/nastasy/reduction-of-the-semantic-segmenter
configfile: pyproject.toml
plugins: hydra-core-1.3.2
collected 34 items

tests/test_convetrors.py ...                                             [  8%]
tests/test_data_helper.py ....                                           [ 20%]
tests/test_dvc_params.py ..................                              [ 76%]
tests/test_trainloop.py ........                                         [100%]

============================== 34 passed in 5.28s ==============================
```

ПРИМЕНЕНИЕ

Разработан API
https://citybeauty.muxhomelab.ru
/segmentation-api

API можно использовать в собственном приложении.
Пример внедрения - веб-приложении для очистки фасадов зданий

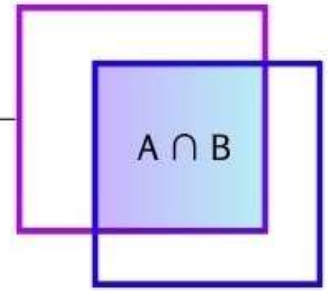


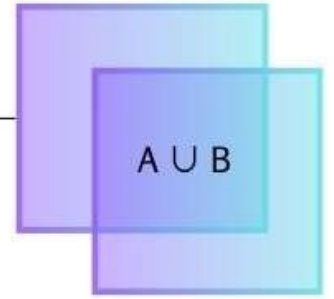
ОСНОВНЫЕ РЕЗУЛЬТАТЫ

1. Проведен обзор архитектур нейронных сетей для семантической сегментации изображений
2. Проведен обзор методов редуцирования нейронных сетей
3. Осуществлен сбор и предобработку данных для обучения нейронной сети для семантической сегментации изображений
4. Выполнена реализацию нейронных сетей с использованием разных методов редуцирования
5. Проведено тестирование производительности работы редуцированных нейронных сетей

МЕТРИКА mIoU
(ДОПОЛНИТЕЛЬНЫЙ СЛАЙД)

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$





ПРИМЕР ФАКТОРИЗАЦИИ (ДОПОЛНИТЕЛЬНЫЙ СЛАЙД)

```
conv1 = nn.Conv2d(3, 64, kernel_size=7, stride=2, padding=3)

conv1 _light = nn.Conv2d(3, 64, kernel_size=7, stride=4, padding=3)
```