



Московский государственный технический
университет имени Н.Э. Баумана
Кафедра «Системы автоматизированного
проектирования»



Анализ распознавания изображения с использованием FPGA – чип

Студент: Караф С.М

Научный руководитель: Витюков Ф. А.



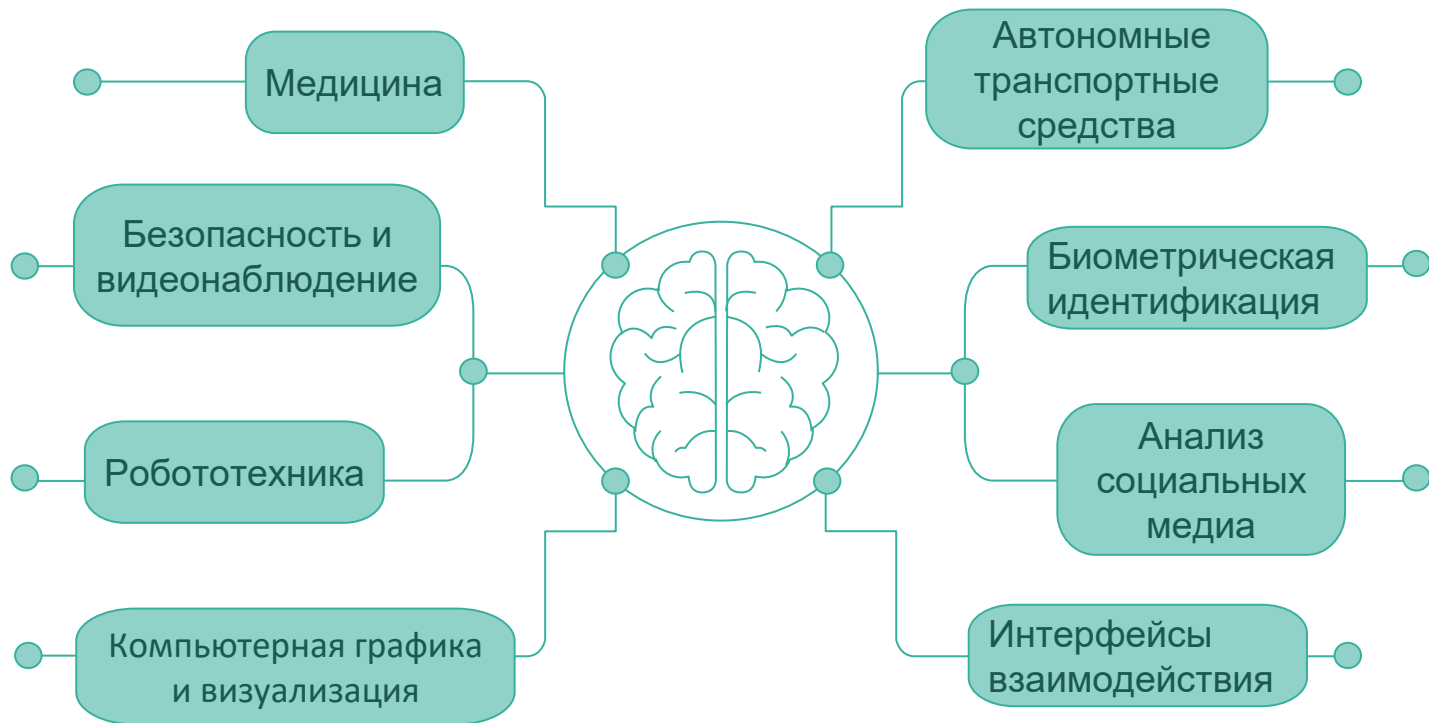
Москва, 2023

Содержание:

- Введение и актуальность
- Постановка задачи
- Основная теория
- Описание датасета
- Обучение нейронной сети
- Результаты нейронной сети с учителем и без учителя
- Результаты сравнения с существующей нейронной сетью на PyTorch
- Конвертация модели на ONNX
- Проблемы с импортом модели в Quartus
- Аппаратная реализация на FPGA
- Результаты написанной модели и аппаратной реализации
- Вывод и дальнейшие исследования



Введение и актуальность



Постановка задачи

Создание датасета:

- Подготовка и сбор изображений.
- Разметка изображения.
- Разделение датасета на обучающую и тестовую выборки.

Обучение модели и сравнение с существующими:

- Обучение сверточной нейронной сети.
- Оценка точности модели на тестовой выборке.
- Сравнение результатов нашей модели с уже существующими моделями распознавания изображений.

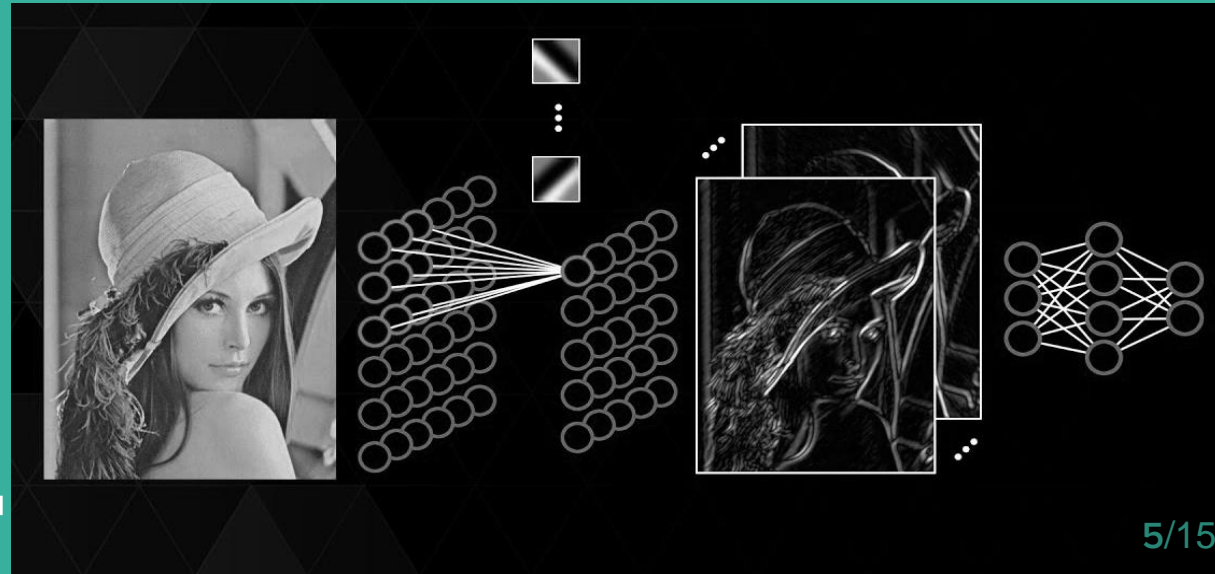
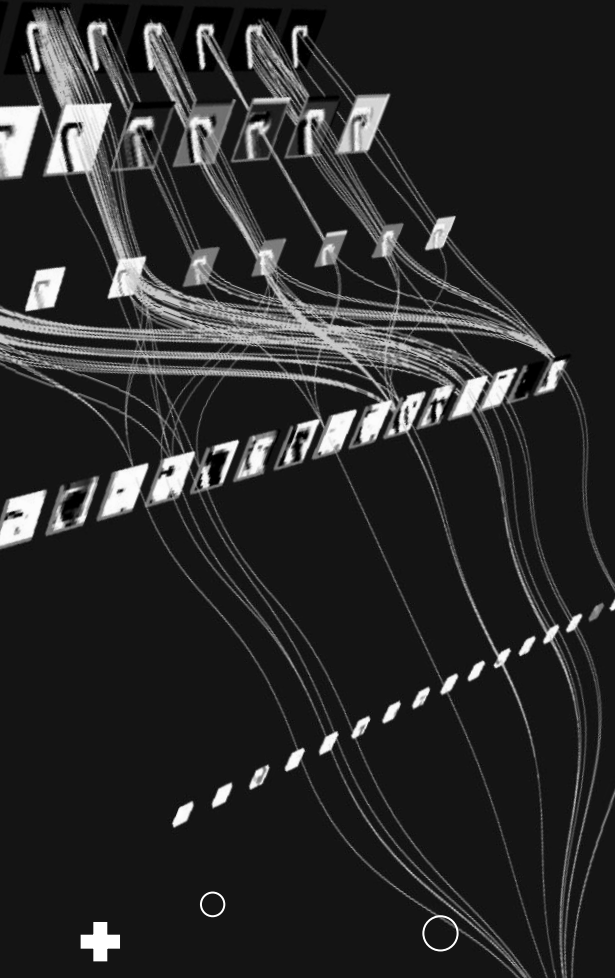
Симуляция модели на плате FPGA:

- Конвертация обученной модели в формат ONNX для совместимости с платой FPGA.
- Симуляция работы модели на плате FPGA

Анализ данных и результатов

Convolutional Neural Network (CNN)

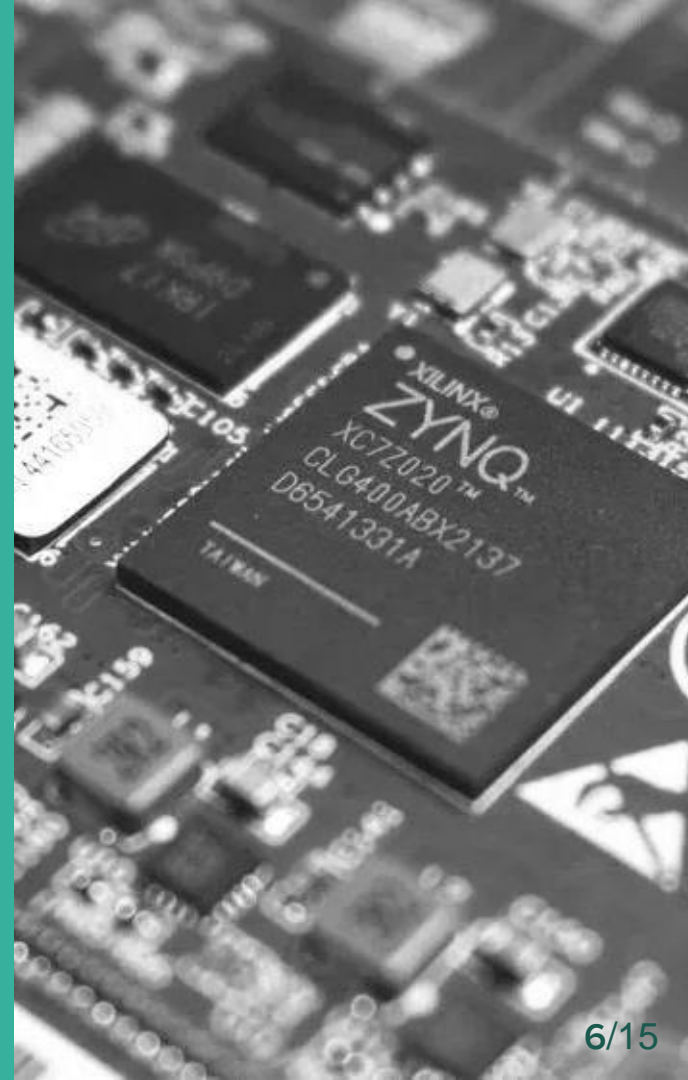
Она использует сверточные слои для извлечения признаков, пулинг для уменьшения размерности и полносвязные слои для классификации или анализа. CNN автоматически изучает иерархические признаки на разных уровнях, что делает ее мощным инструментом для обработки изображений.



FPGA (Field-Programmable Gate Array)

FPGA представляет собой полевой интегральный схемы, позволяющий пользователю настраивать его для выполнения различных функций.

- Гибкость: FPGA можно перепрограммировать для выполнения различных задач без необходимости разработки новой интегральной схемы.
- Высокая производительность: FPGA обеспечивает высокую скорость обработки данных и параллельную обработку.
- Низкая задержка: FPGA имеет очень низкую задержку при выполнении операций, что делает его подходящим для реализации в реальном времени.



Dataset

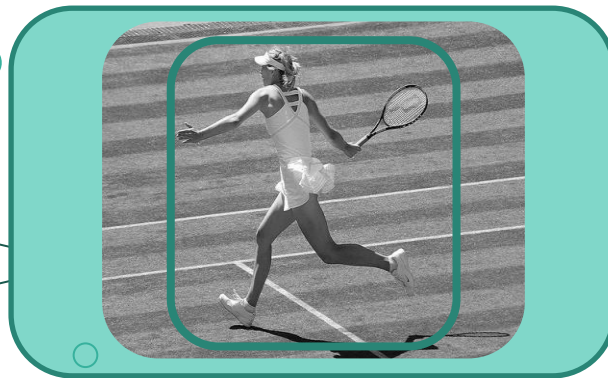
Сбор фотографий



Создание JSON

```
"annotations": [  
  {  
    "image_id": "Уникальный идентификатор изображения",  
    "class": "Класс объекта (например, 'person')",  
    "type": "Тип объекта (например, 'person')",  
    "bounding_box": {  
      "x": "Координата X левого верхнего угла ограничивающей рамки",  
      "y": "Координата Y левого верхнего угла ограничивающей рамки",  
      "width": "Ширина ограничивающей рамки",  
      "height": "Высота ограничивающей рамки"  
    },  
  },  
]
```

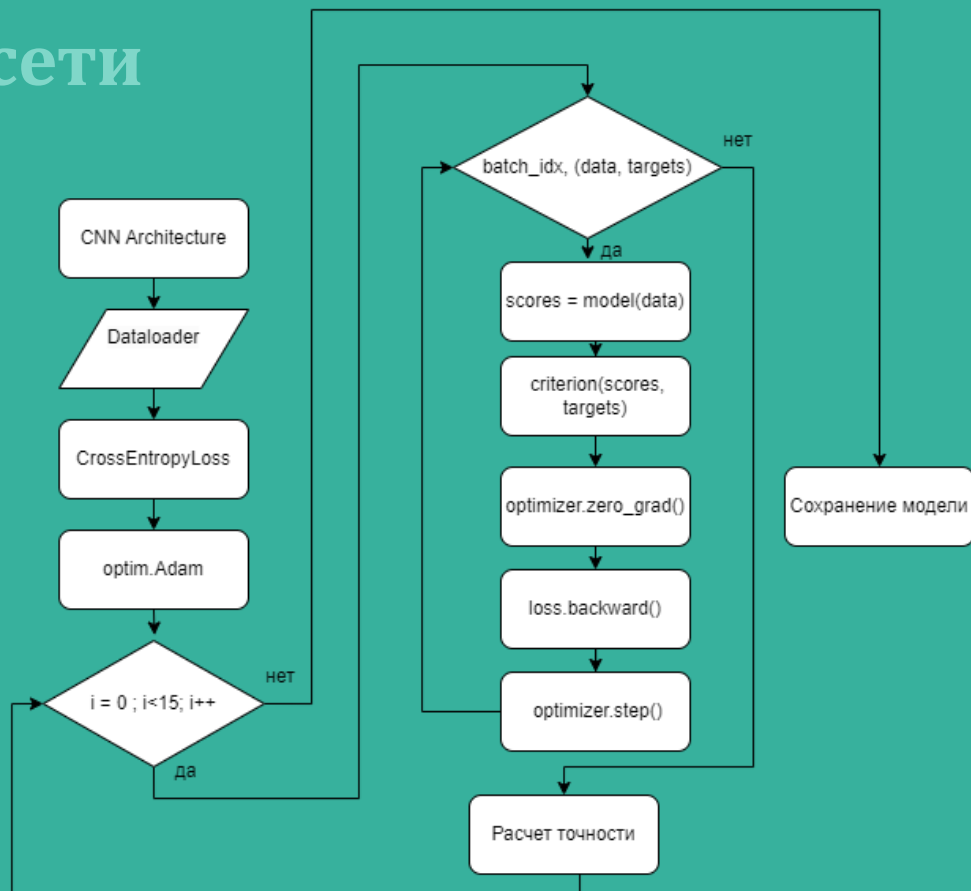
Разметка



Обучение нейронной сети

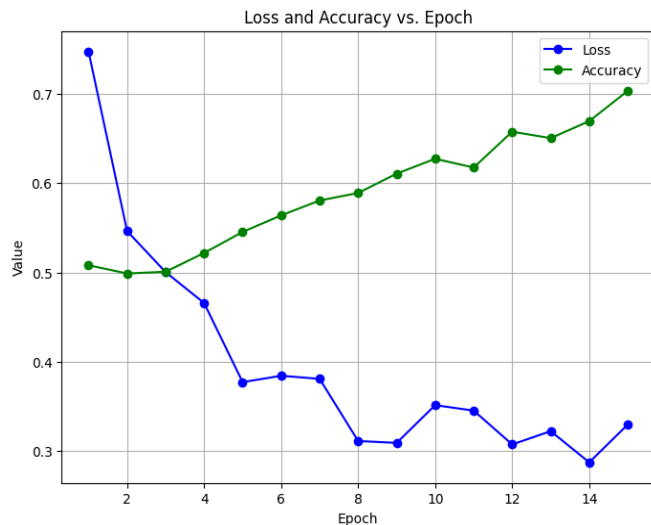
Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 16, 480, 640]	448
MaxPool2d-2	[-1, 16, 240, 320]	0
Conv2d-3	[-1, 32, 240, 320]	4,640
MaxPool2d-4	[-1, 32, 120, 160]	0
Linear-5	[-1, 256]	655,616
Linear-6	[-1, 2]	514
Total params: 661,218		
Trainable params: 661,218		
Non-trainable params: 0		

Архитектура сверточной нейронной сети

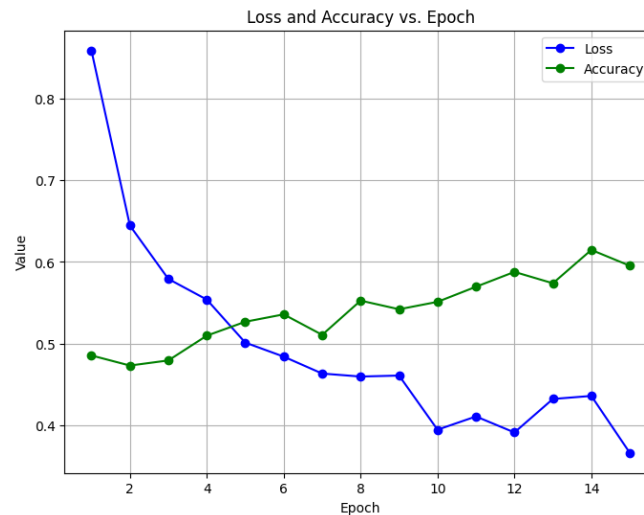


Блок схема обучение нейронной сети

Результаты нейронной сети с учителем и без учителя

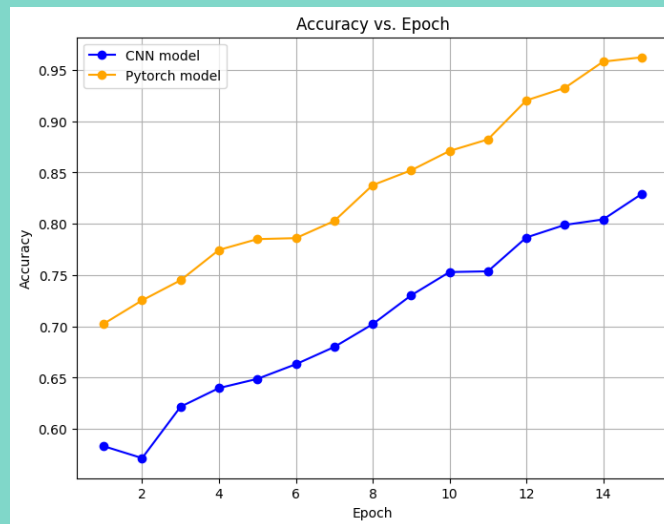
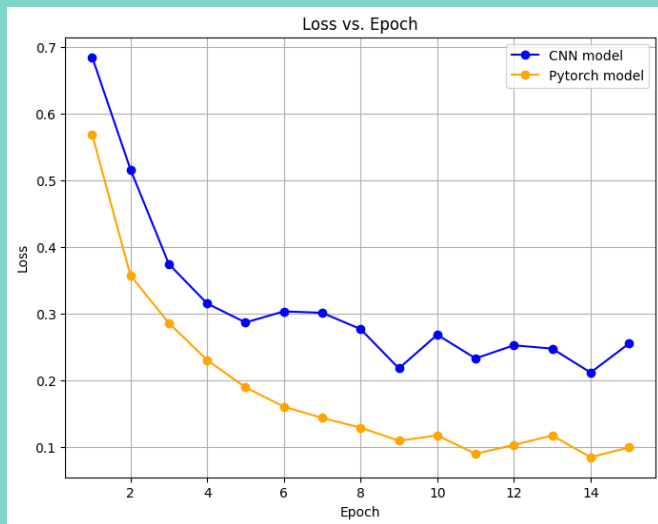


Обучение без учителя



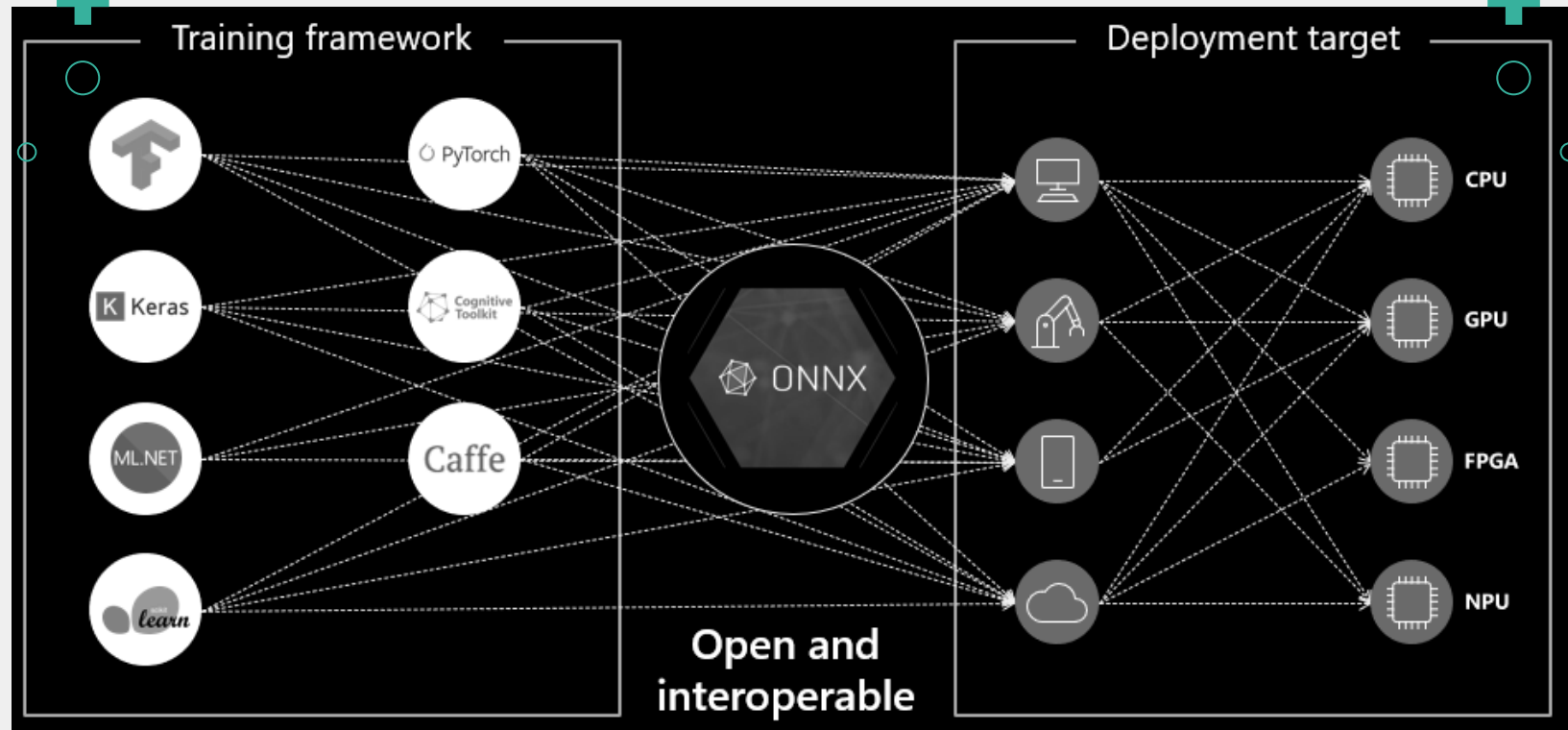
Обучение с учителем

Сравнение существующей нейронной сети на PyTorch

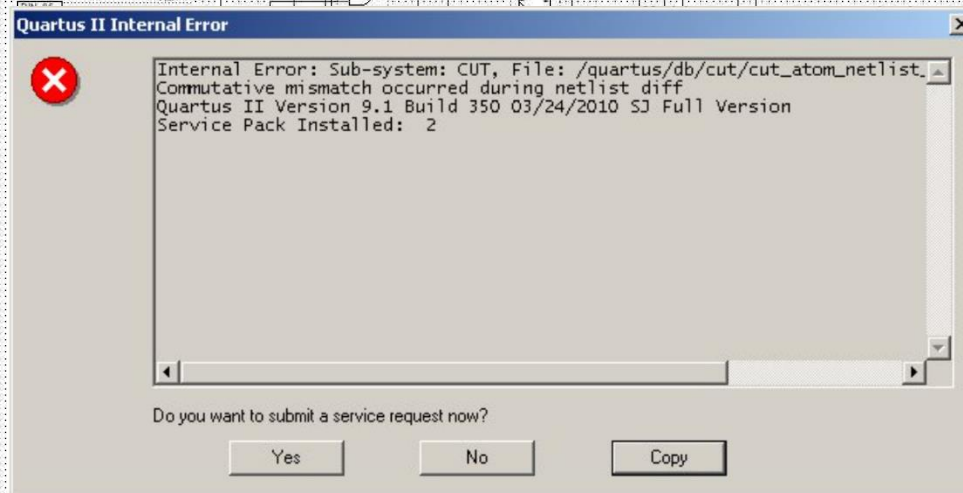


Сравнение обучения написанной модели с аналоговым от PyTorch.

Конвертация модели на ONNX

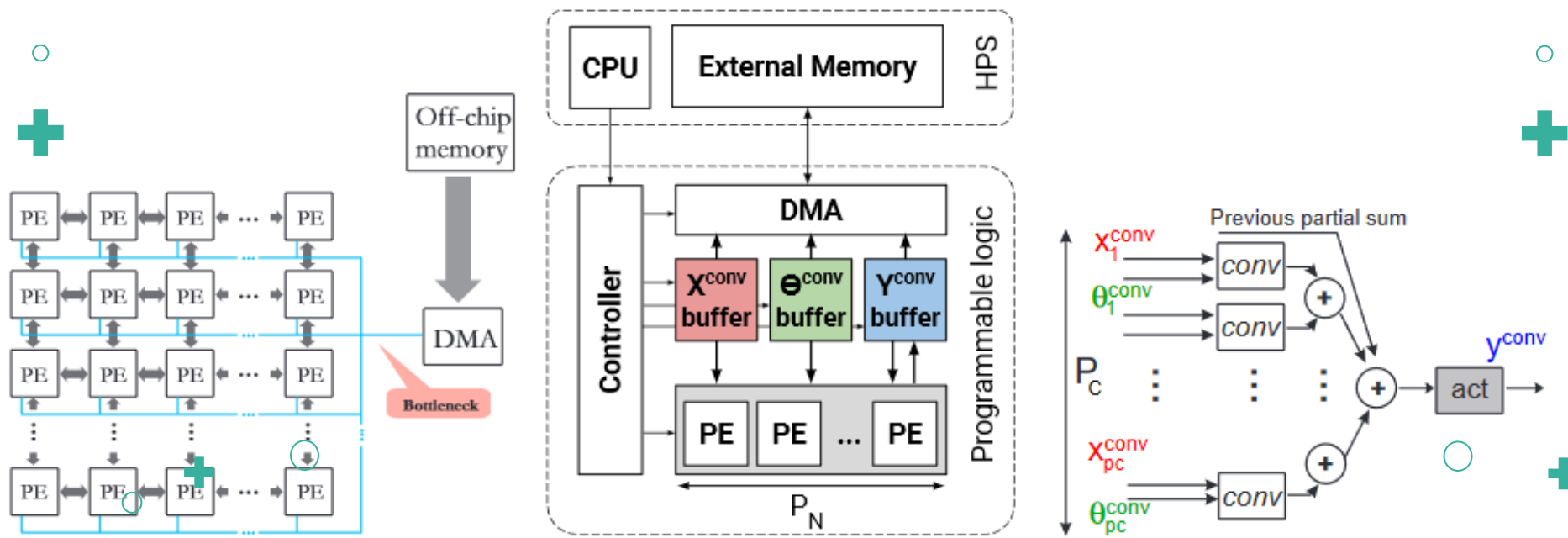


Проблемы с импортом модели в Quartus



Непреодолимая ошибка импортировки.

Аппаратная реализация на TPGA



Результаты написанной модели и аппаратной реализации



	Обученная моделью	FPGA
Время, сек	27 801	11 891
Точность, %	86,6	83,8

Вывод

- Обучение нейронной сети с учителем приводит к повышению точности распознавания изображений.
- Обученная модель показала хорошие результаты на COCO датасете, но требует значительного времени для проверки всех изображений.
- Аппаратная реализация на FPGA сократила время обработки в два раза, но точность немного снизилась.
- Дальнейшие исследования могут включать оптимизацию аппаратной реализации, поиск решений для проблемы импорта модели в Quartus, расширение датасета и исследование других платформ и фреймворков.

В целом, использование FPGA-платы в анализе распознавания изображений обладает потенциалом, и дальнейшие исследования могут улучшить точность и эффективность таких систем.

Fonts & colors used

This presentation has been made using the following fonts:

Figtree

(<https://fonts.google.com/specimen/Figtree>)

#191919

#ffffff

#37b19d

#91d2c8

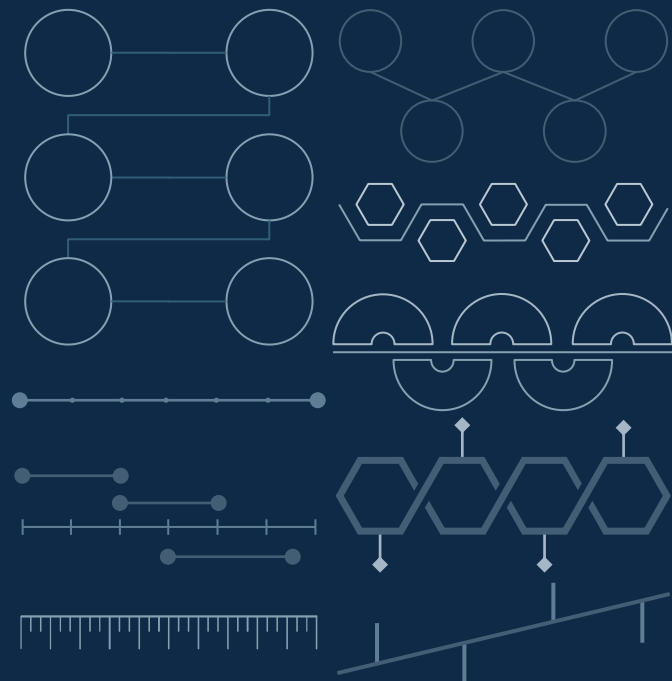
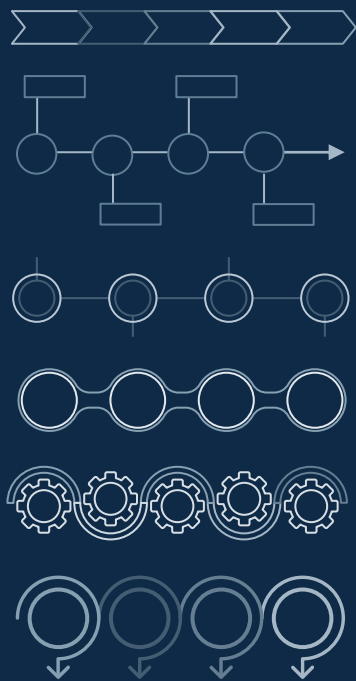
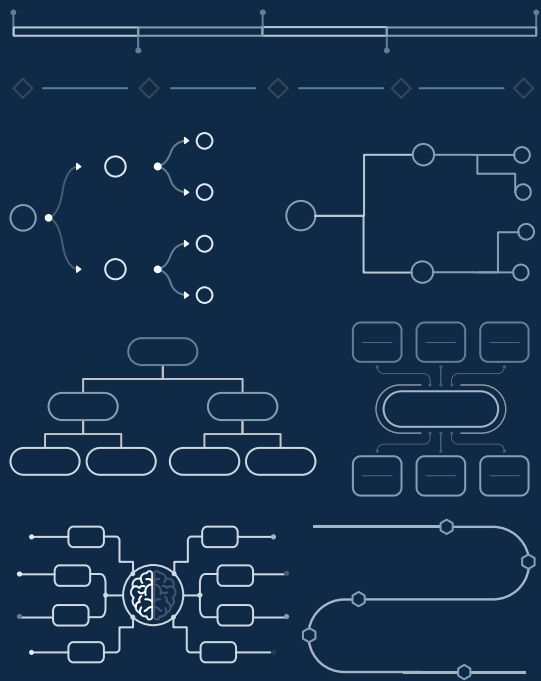
#efefef

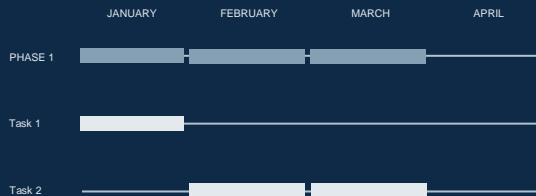
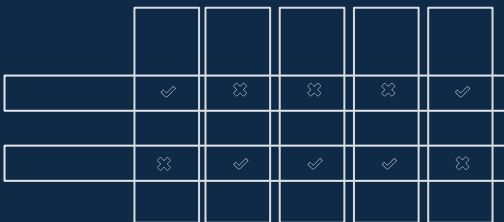
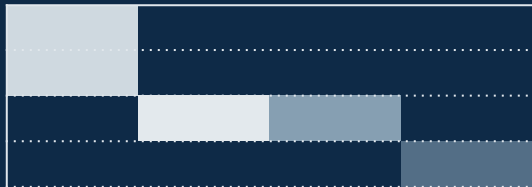
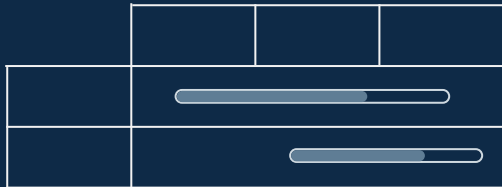
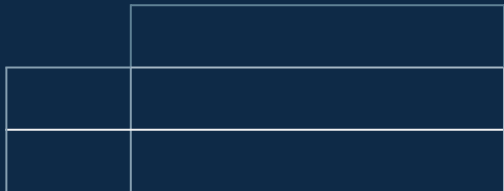
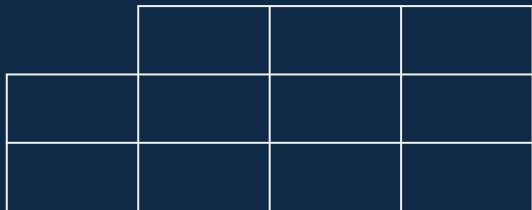
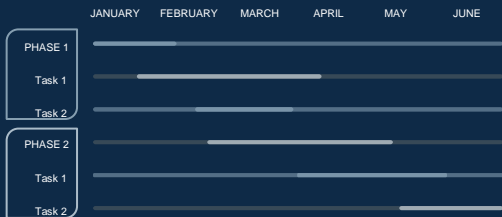
Use our editable graphic resources...

You can easily **resize** these resources without losing quality. To **change the color**, just ungroup the resource and click on the object you want to change. Then, click on the paint bucket and select the color you want. Group the resource again when you're done. You can also look for more **infographics** on [Slidesgo](#).

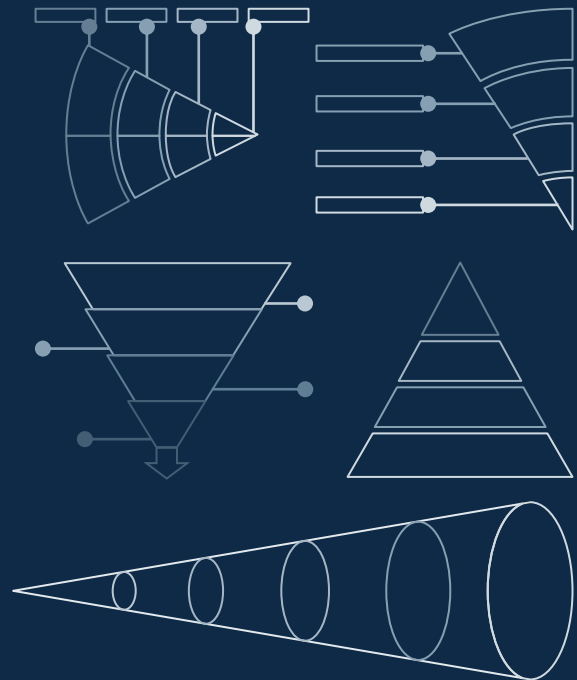
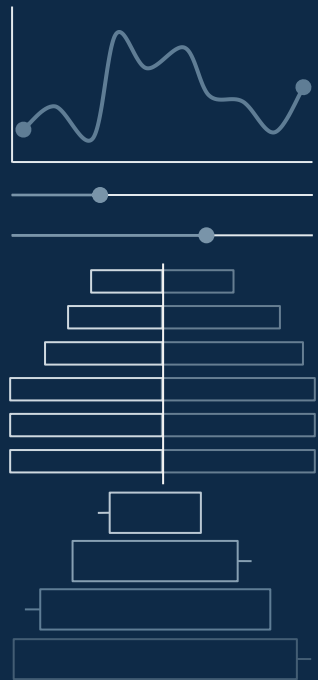
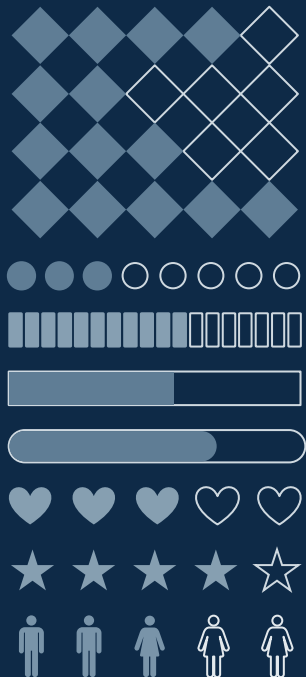
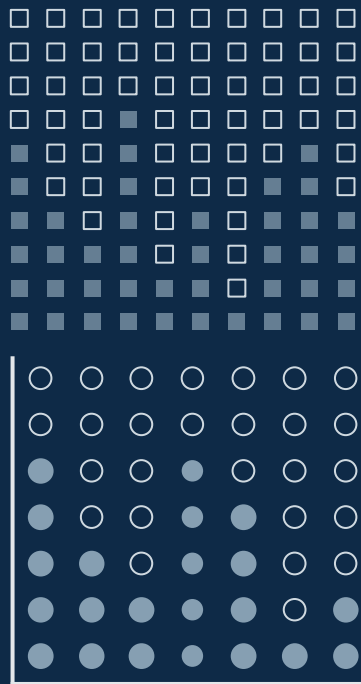












...and our sets of editable icons

You can **resize** these icons without losing quality.

You can **change the stroke and fill color**; just select the icon and click on the **paint bucket/pen**.

In Google Slides, you can also use **Flaticon's extension**, allowing you to customize and add even more icons.



Educational Icons



Medical Icons



Business Icons



Teamwork Icons



Help & Support Icons



Avatar Icons



Creative Process Icons



Performing Arts Icons



Nature Icons



SEO & Marketing Icons



