



IOTG Russia

# OpenVINO™

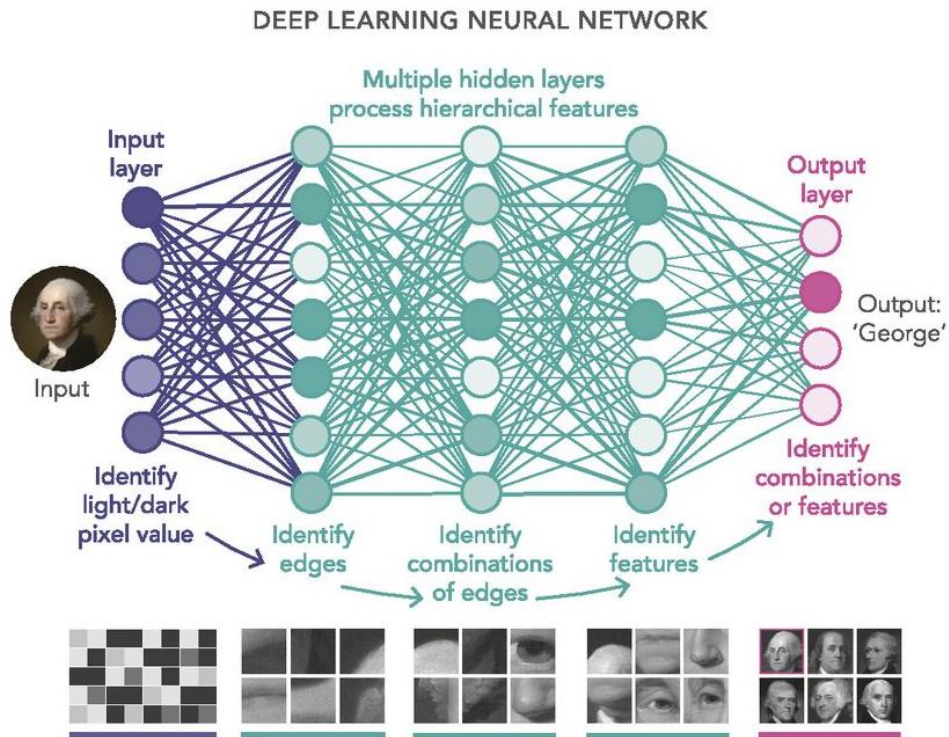
Visual Inference & Neural Network Optimization

Денис Орлов

# О чем сегодня пойдет речь?

- Краткое введение в нейронные сети
- Основы OpenVINO (Model Optimizer, Inference Engine)
- Поддерживаемые устройства
- Оптимизация с помощью OpenVINO
- Дополнительные компоненты OpenVINO
- Способы распространения OpenVINO
- Дополнительные материалы

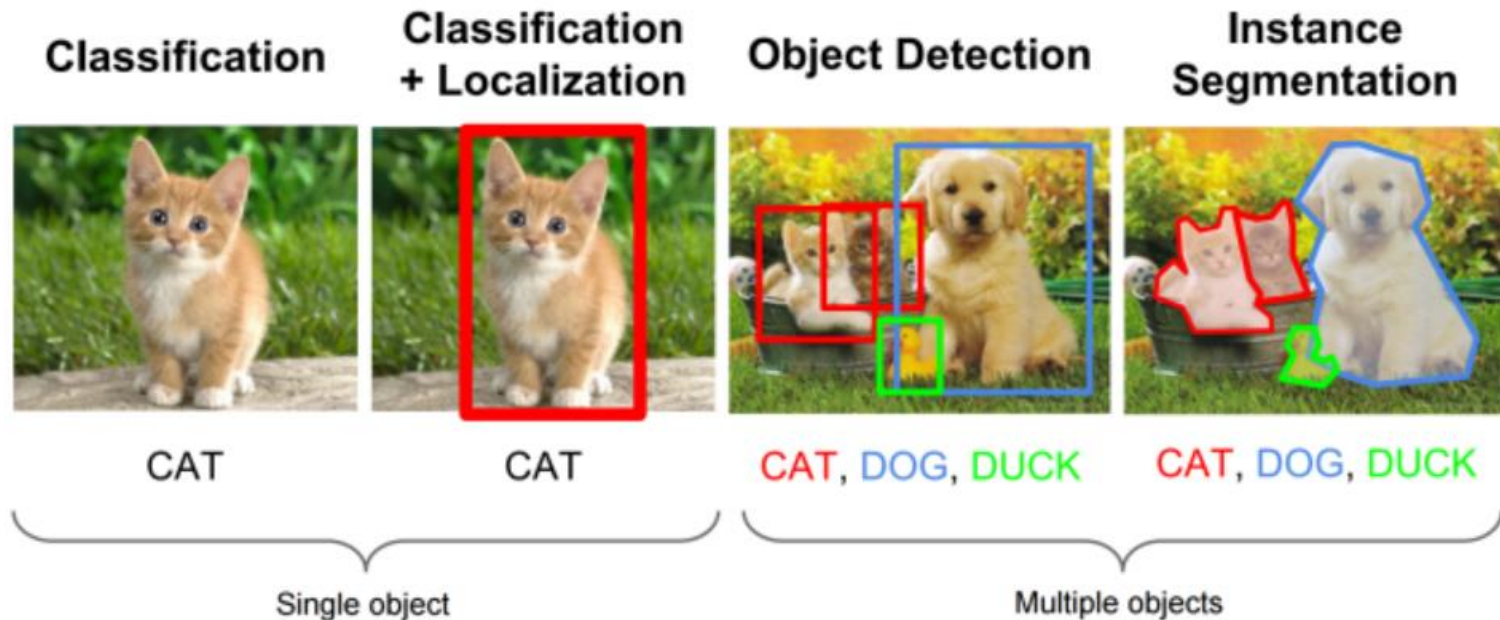
# Краткое введение в нейронные сети



M. Mitchell Waldrop PNAS 2019;116:4:1074-1077

# Методы deep learning для компьютерного зрения

- Классификация объектов



<https://medium.com/analytics-vidhya/yolov3-real-time-object-detection-54e69037b6d0>

# Методы deep learning для компьютерного зрения

- Семантическая сегментация



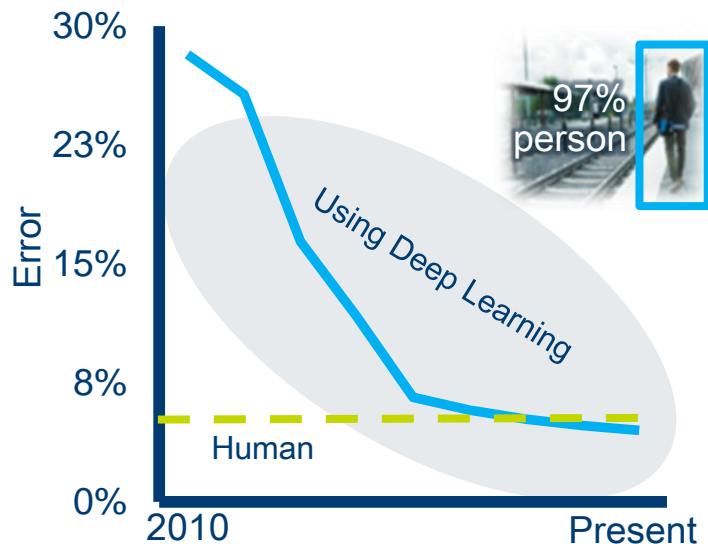
<https://mc.ai/introduction-to-semantic-image-segmentation/>

# Новые применения методов deep learning

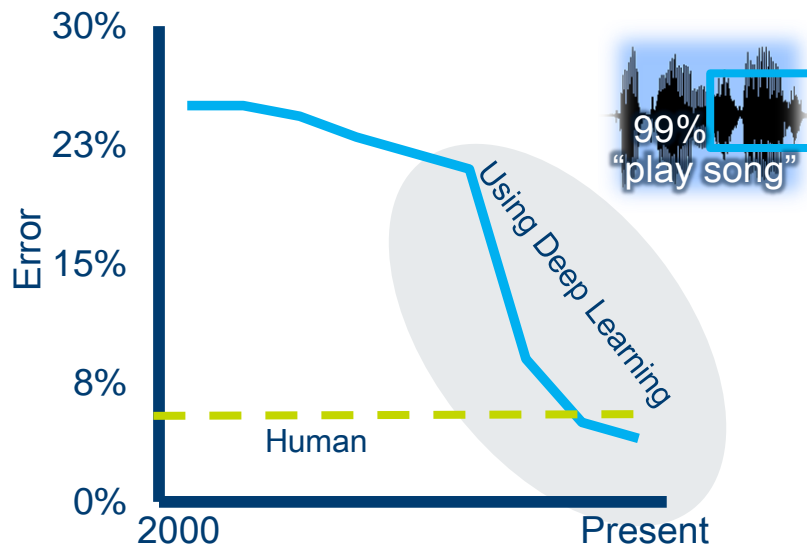
- Машинный перевод
- Распознавание голоса
- Устранение шумов и отражений в звуке
- Классификация звука
- Классификация текста
- Анализ тональности текста (sentiment analysis)
- Идентификация говорящего
- Генерация голоса
- Рекомендательные системы
- ...

# Прогресс в области глубокого обучения

## Image Recognition

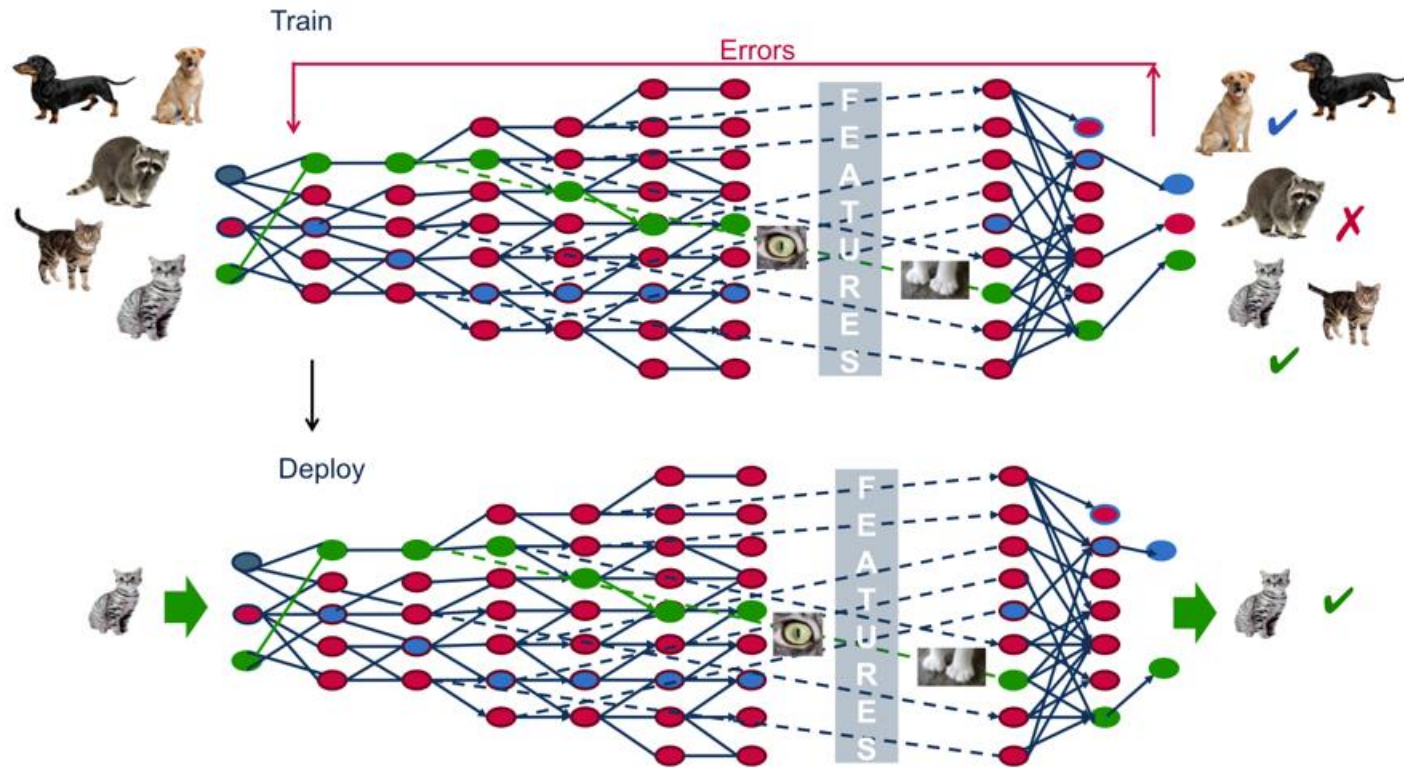


## Speech Recognition



Source: ILSVRC ImageNet winning entry classification error rate each year 2010-2016 (Left), <https://www.microsoft.com/en-us/research/blog/microsoft-researchers-achieve-new-conversational-speech-recognition-milestone/> (Right)  
Source: <https://www.mckinsey.com/featured-insights/artificial-intelligence/notes-from-the-ai-frontier-applications-and-value-of-deep-learning>

# Тренировка vs Запуск («Инференс»)



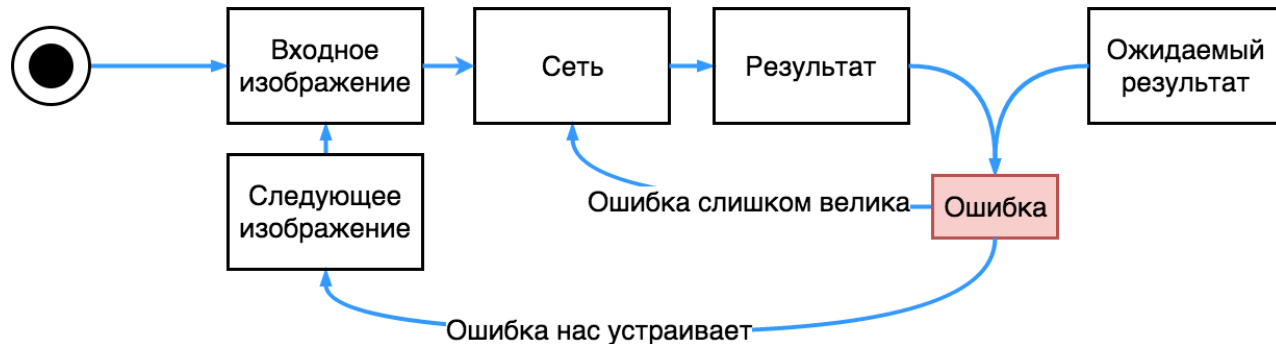
<https://www.slideshare.net/caroljmcDonald/demystifying-ai-machine-learning-and-deep-learning>



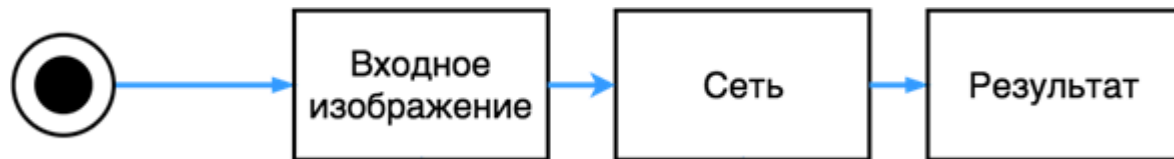
# Тренировка vs Запуск («Инференс»)

## Тренировка требует:

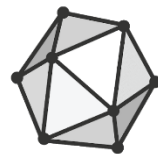
- больших объёмов данных
- времени (дни, недели)
- значительных вычислительных ресурсов



**Инференс** – запуск натренированной сети как готовой программы



# Популярные фреймворки и инструменты



ONNX

 PyTorch





TensorFlow



Keras

Caffe



KALDI

# Основы OpenVINO

OFFLINE



Trained Models

Caffe\*

TensorFlow\*

MxNet\*

ONNX\*

Pytorch\*, Caffe2\* & more

Kaldi\*

Model Optimizer

IR



IR = Intermediate Representation format

OpenVINO

Inference Engine

Infer

CPU Plugin

GPU Plugin

FPGA Plugin

Myriad Plugin for Intel NCS & NCS

HDDL Plugin for VAD\*

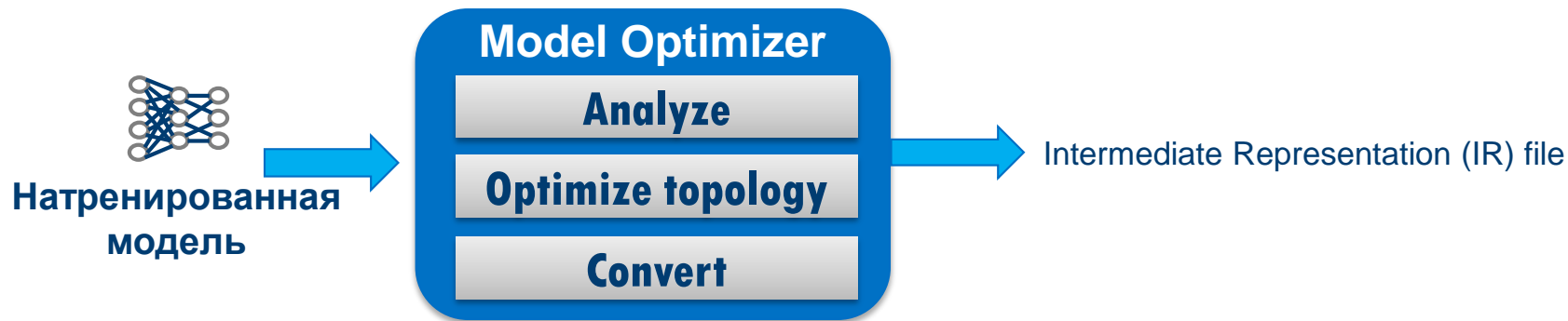
GNA Plugin



GPU = Intel CPU with integrated GPU/Intel® Processor Graphics, Intel® NCS = Intel® Neural Compute Stick (VPU)

\*VAD = Intel® Vision Accelerator Design Products (HDDL-R)

# Model Optimizer

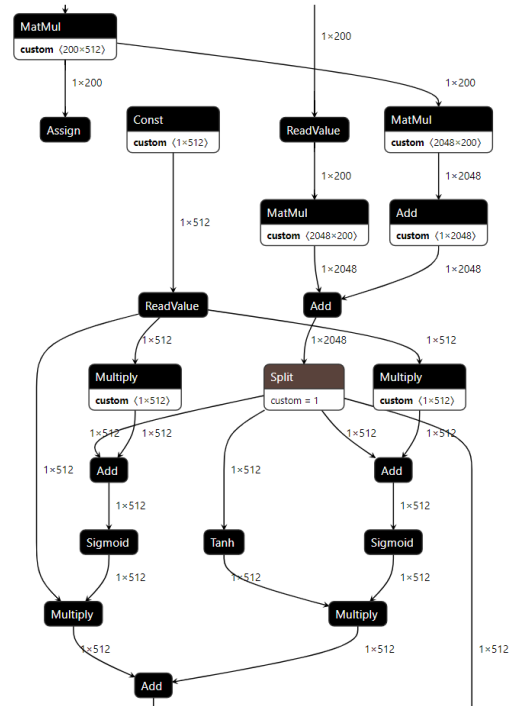
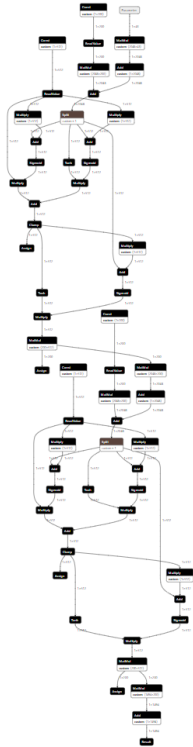


Поддерживаемые фреймворки: Caffe, TensorFlow, MXNet, Kaldi; формат ONNX (Pytorch, Caffe2 и другие, использующие ONNX).

Intermediate Representation (IR) состоит из:

- Xml file (описание топологии)
- Bin file (веса)
- *Альтернативный способ описания модели – с помощью API*

# Пример нейронной сети



# Как выглядит модель в формате IR?

```
<?xml version="1.0" ?>
<net name="nsnet2-20ms-baseline" version="10">
  <layers>
    <layer id="0" name="input" type="Parameter" version="opset1">
      <data element_type="f32" shape="1,100,161"/>
      <output>
        <port id="0" precision="FP32">
          <dim>1</dim>
          <dim>100</dim>
          <dim>161</dim>
        </port>
      </output>
    </layer>
    <layer id="1" name="MatMul_0/1_port_transpose1025_const" type="Const" version="opset1">
      <data element_type="f32" offset="0" shape="400,161" size="257600"/>
      <output>
        <port id="1" precision="FP32">
          <dim>400</dim>
          <dim>161</dim>
        </port>
      </output>
    </layer>
    <layer id="2" name="MatMul_0" type="MatMul" version="opset1">
      <data transpose_a="False" transpose_b="True"/>
      <input>
        <port id="0">
          <dim>1</dim>
          <dim>100</dim>
          <dim>161</dim>
```

[ ... ]

# Как выглядит модель в формате IR?

```
<edges>
  <edge from-layer="0" from-port="0" to-layer="2" to-port="0"/>
  <edge from-layer="1" from-port="1" to-layer="2" to-port="1"/>
  <edge from-layer="2" from-port="2" to-layer="4" to-port="0"/>
  <edge from-layer="3" from-port="1" to-layer="4" to-port="1"/>
  <edge from-layer="4" from-port="2" to-layer="5" to-port="0"/>
  <edge from-layer="5" from-port="1" to-layer="7" to-port="0"/>
  <edge from-layer="6" from-port="1" to-layer="7" to-port="1"/>
  <edge from-layer="7" from-port="2" to-layer="9" to-port="0"/>
  <edge from-layer="8" from-port="1" to-layer="9" to-port="1"/>
  <edge from-layer="9" from-port="3" to-layer="11" to-port="0"/>
  <edge from-layer="10" from-port="1" to-layer="11" to-port="1"/>
  <edge from-layer="11" from-port="2" to-layer="12" to-port="0"/>
  <edge from-layer="9" from-port="2" to-layer="14" to-port="0"/>
  <edge from-layer="13" from-port="1" to-layer="14" to-port="1"/>
  <edge from-layer="14" from-port="2" to-layer="16" to-port="0"/>
  <edge from-layer="15" from-port="1" to-layer="16" to-port="1"/>
  <edge from-layer="16" from-port="2" to-layer="18" to-port="0"/>
  <edge from-layer="17" from-port="1" to-layer="18" to-port="1"/>
  <edge from-layer="18" from-port="3" to-layer="20" to-port="0"/>
  <edge from-layer="19" from-port="1" to-layer="20" to-port="1"/>
  <edge from-layer="20" from-port="2" to-layer="21" to-port="0"/>
  <edge from-layer="18" from-port="2" to-layer="23" to-port="0"/>
  <edge from-layer="22" from-port="1" to-layer="23" to-port="1"/>
  <edge from-layer="23" from-port="2" to-layer="25" to-port="0"/>
  <edge from-layer="24" from-port="1" to-layer="25" to-port="1"/>
  <edge from-layer="25" from-port="2" to-layer="27" to-port="0"/>
  <edge from-layer="26" from-port="1" to-layer="27" to-port="1"/>
</edges>
```

# OpenVINO Inference Engine

- библиотека на C++ (Python / C), позволяющая приложению:
  - прочитать модель из файла (IR) или создать с помощью API
  - загрузить модель в модуль, работающий с конкретным устройством
  - отправить данные для обработки (картинка, текст, звук, ...)
  - получить результаты обработки (вероятности, координаты, ...)

**Главная идея: единый API для разных устройств, выпускаемых Intel**  
(оставляя возможность «тонкой настройки» для конкретных устройств)



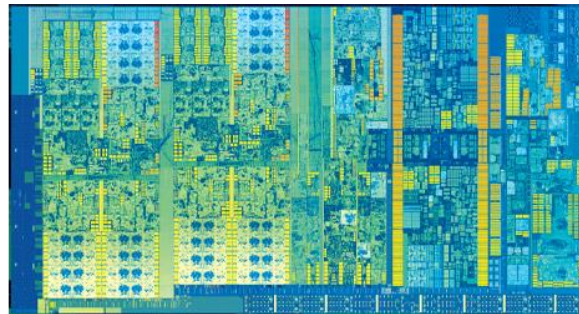
# Пример кода, использующего Inference Engine

```
// Базовый объект Inference Engine
Core ie;
// Чтение сети из файла IR (intermediate representation)
CNNNetwork network = ie.ReadNetwork(input_model);
// Определение имен входов и выходов
std::string input_name = network.getInputsInfo().begin()->first;
std::string output_name = network.getOutputsInfo().begin()->first;
// Загрузка модели в плагин
ExecutableNetwork executable_network = ie.LoadNetwork(network, device_name);
// Создание infer request'a
InferRequest infer_request = executable_network.CreateInferRequest();
// Задание входных данных
infer_request.SetBlob(input_name, imgBlob);
// Инференс
infer_request.Infer();
// Чтение выходных данных
Blob::Ptr output = infer_request.GetBlob(output_name);
```

# Поддерживаемые устройства



Процессоры (CPU)



Графические карты (GPU)



Field-programmable gate array  
(FPGA)



Процессоры машинного зрения  
(VPU)

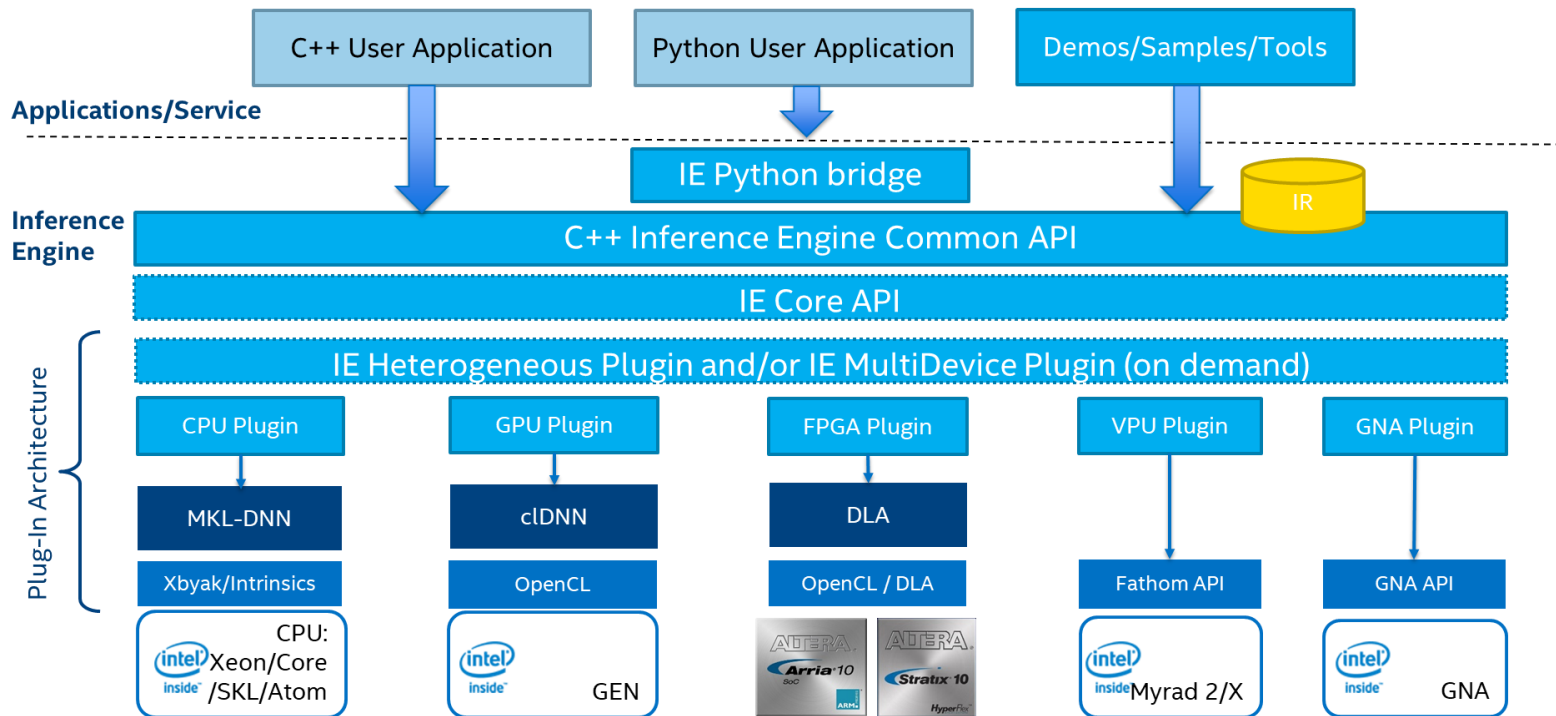
# Поддерживаемые устройства

## Gaussian & Neural Accelerator (GNA)

- маломощный сопроцессор для обработки звука

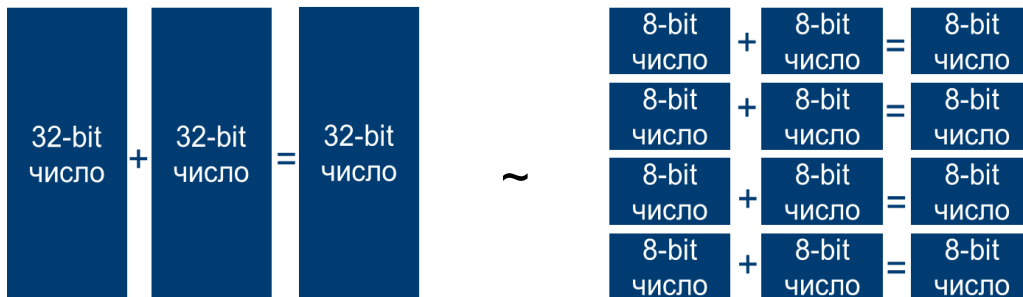


# Программный стек при использовании Inference Engine



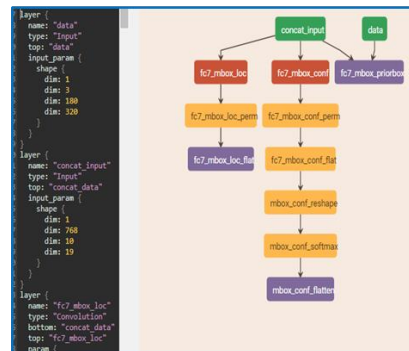
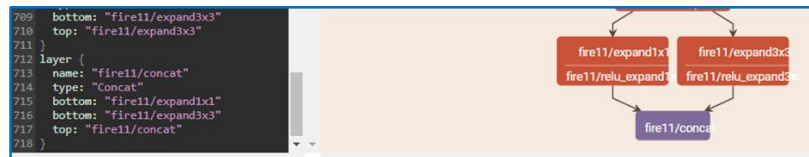
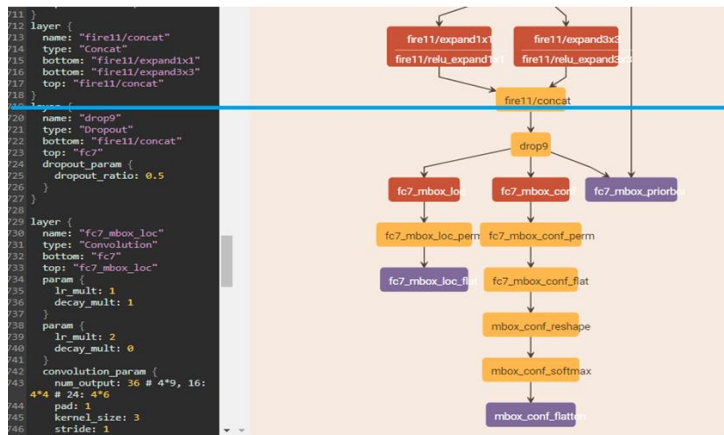
# Оптимизация с помощью Inference Engine

- Оптимальное использование аппаратных особенностей
- Объединение нескольких операций в одну (fusing)
- Пакетная обработка данных (несколько картинок обрабатываются одновременно)
- «Стримы» (несколько экземпляров сети запускаются одновременно)
- Использование вычислений с меньшей разрядностью



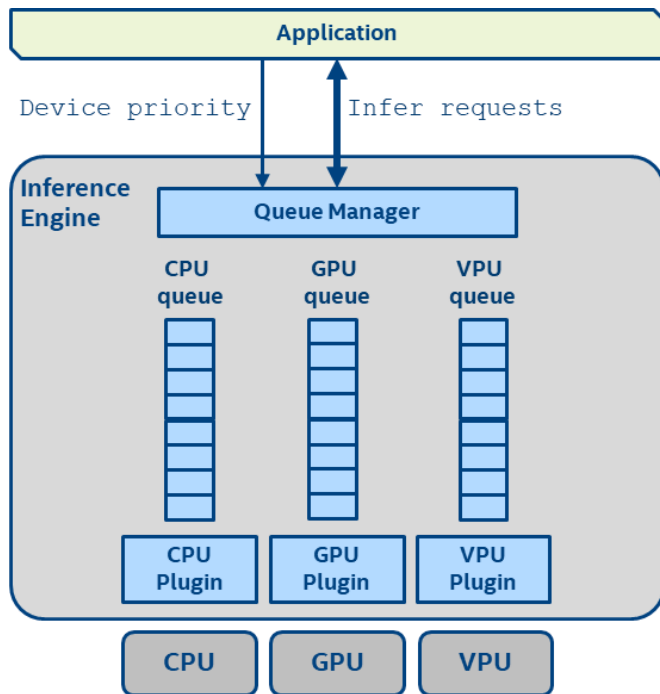
# Гетерогенный режим

Не поддерживаемые слои отправляются на другое устройство (fallback)



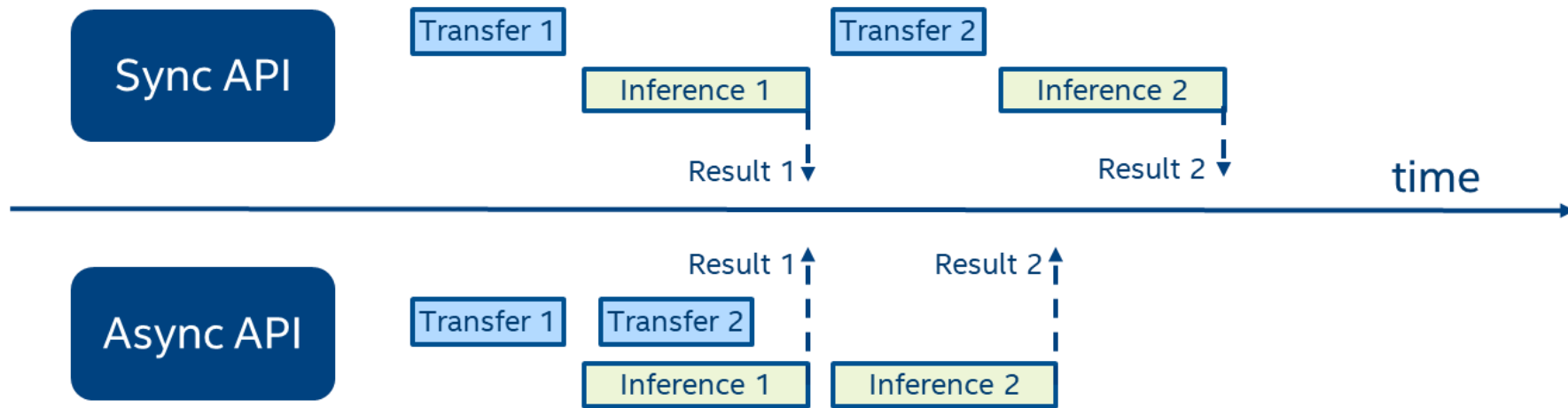
# «Multi-device» режим

Задачи могут автоматически распределяться между несколькими устройствами



# Синхронный и асинхронный режим

- Синхронный режим: выполнение блокируется до исполнения
- Асинхронный режим: выполнение продолжается; окончание отслеживается с помощью механизма callback





# Дополнительные средства OpenVINO



## [NEW] Post-training Optimization

- Reduce model size into low precision data types, such as INT8
- Reduces model size while also improving latency



## Model Analyzer

- Provides theoretical data on models: computational complexity (flops), number of neurons, memory consumption



## Benchmark App

- Measure performance (throughput, latency) of a model
- Get performance metrics per layer and overall basis



## Deployment Manager

- Generate an optimal, minimized runtime package for deployment
- Deploy with smaller footprint compared to development package



## Accuracy Checker

- Check for accuracy of the model (original and after conversion) to IR file using a known data set

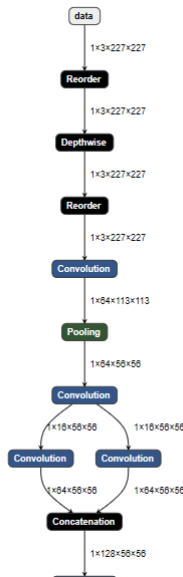


## Model Downloader

- Provides an easy way of accessing a number of public models as well as a set of pre-trained Intel models

# Deep Learning Workbench

- Конвертация сетей в IR
- Визуализация и профилировка сетей
- Подбор оптимальных параметров запуска
- Измерение точности сетей
- Работа с Open Model Zoo



CONFIGURATION

PROJECTS

V2.45

Projects

New

#	Model	Dataset	Target	Start Time	Latency	FPS	Accuracy	Status	Action
1	A. MobileNet (FP32)	ImagenetTest1	CPU	26/03/19, 13:10	973ms	80FPS	N/A		

Selected Model: 1A MobileNet - Baseline - FP32 • ImagenetTest1 • CPU

Profile

Optimize

Select Inference Type

Parallel Infers

Use Ranges

Min (1-8)

Max (1-512)

Step (1-511)

Batch (1-500)

Min (1-16)

Max (1-512)

Step (1-511)

Execute

Inference Results

Throughput (FPS)

Latency (ms)

Max Latency 1500

Inference History

Clear All

Export Data

#	Start Time	Infers	Batch	Status	Action	Compare
A Baseline	26/03/19, 13:10	1	1		</>	<input type="checkbox"/>
B	26/03/19, 14:42	2	4		</>	<input type="checkbox"/>
C	26/03/19, 15:20	1-10, 1	1-10, 1		</>	<input type="checkbox"/>

inference

Select All

Model Performance Summary

Execution Time by Layer Group

26% Convolution

16% Fully Connected

16% Pooling

16% Norm

14% Softmax

7% Reflu

5% Other

Mean Inference Time (ms)

125,000 ms

Layer Details

IOTG Russia

26

# OpenVINO samples

OpenVINO поставляется вместе с примерами, демонстрирующими использование OpenVINO для различных задач:

- классификация
- обнаружение объектов
- автоматическое распознавание голоса
- оценка performance для конкретных моделей
- ...

# OpenVINO Open Model Zoo



## Computer Vision

[Object detection](#)

[Object recognition](#)

[Reidentification](#)

[Semantic segmentation](#)

[Instance segmentation](#)

[Human pose estimation](#)

[Image processing](#)



## Audio, Speech, Language

[Text detection](#)

[Text recognition](#)



## Recommender

[Action recognition](#)



## Other

*(Data Generation,  
Reinforcement Learning)*

[Compression models](#)

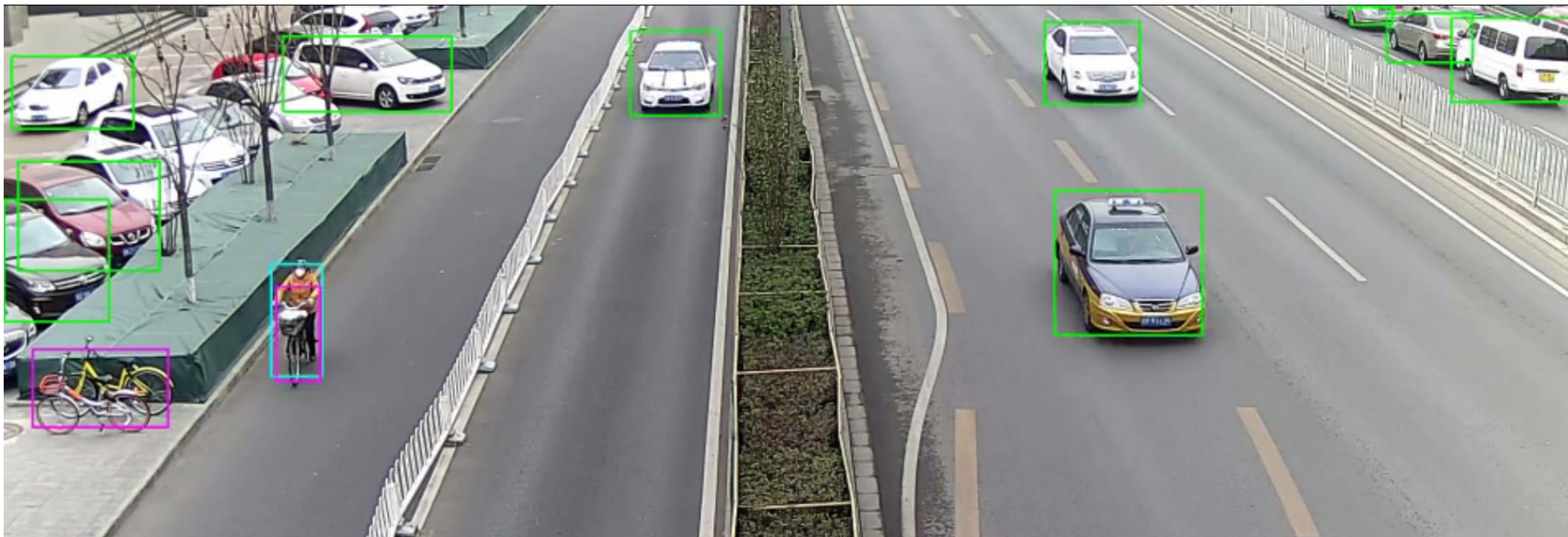
[Image retrieval](#)

And more..

[https://github.com/openai/open\\_model\\_zoo](https://github.com/openai/open_model_zoo)

# Модели от Intel – Open Model Zoo (1)

**Open Model Zoo** – набор готовых бесплатных нейронных сетей, натренированных компанией Intel



**Модель:** person-vehicle-bike-detection-crossroad-1016

# Модели от Intel – Open Model Zoo (2)



Type: car  
Color: black

**Модель:** vehicle-attributes-recognition-barrier-0039

# Модели от Intel – Open Model Zoo (3)



**Модель:** person-reidentification-retail-0288

# Модели от Intel – Open Model Zoo (4)



**Модель:** semantic-segmentation-adas-0001



# Модели от Intel – Open Model Zoo (5)



**Модель:** instance-segmentation-security-0010

# Модели от Intel – Open Model Zoo (6)



**Модель:** text-detection-0004

# Модели от Intel – Open Model Zoo (7)

DRINKING EATING – 99.1%



**Модель:** driver-action-recognition-adas-0002-decoder

# Способы распространения OpenVINO

- Бинарный дистрибутив
- APT
- YUM
- Anaconda
- PyPI
- Docker Hub
- **Open source**

# OpenVINO на GitHub

The screenshot shows the GitHub interface for the `openvinotoolkit/openvino` repository. The page is titled "Pull requests · openvinotoolkit/openvino" and shows a list of pull requests. The repository has 128 watches, 1.7k stars, and 724 forks. The navigation bar includes links for Code, Issues (125), Pull requests (257), Actions, Projects, Wiki, Security, Insights, and Settings. The pull request list is filtered by "is:pr is:open" and shows 257 open pull requests. The list includes pull requests such as "ovino doc assets", "OneCore uap toolchain ninja", "OneCore toolchain", "[IE CLDNN] Cleanup cldnn source tree and README", and "Enable CPU and Interpreter Loop tests". Each pull request entry shows the title, status (Open, Draft, or Merged), author, and a "DO NOT MERGE" label. The "OneCore toolchain" pull request is highlighted with a blue border and shows categories like GPU, IE common, VPU, build, and nGraph, along with a platform of win32.

Filters:  Labels: 60 Milestones: 2 [New pull request](#)

257 Open ✓ 2,110 Closed

Author Label Projects Milestones Reviews Assignee Sort

- ☐ **ovino doc assets** ✓ **DO NOT MERGE** category: docs  
#3046 opened 1 hour ago by ntyukaev • Review required
- ☐ **OneCore uap toolchain ninja** ✓  
#3045 opened 4 hours ago by ilya-lavrenov • Draft
- ☐ **OneCore toolchain** ✗ category: GPU category: IE common category: VPU category: build category: nGraph  
platform: win32  
#3044 opened 4 hours ago by ilya-lavrenov • Review required 2021.2
- ☐ **[IE CLDNN] Cleanup cldnn source tree and README** ✓ category: GPU  
#3043 opened 5 hours ago by vladimir-paramuzov • Draft
- ☐ **Enable CPU and Interpreter Loop tests** ✗ **DO NOT MERGE** category: nGraph  
#3042 opened 5 hours ago by mbencer • Review required

<https://github.com/openvinotoolkit/openvino/compare/...> primitives to CPU plug-in from mkl-dnn fork ✗ category: CPU

# «Экосистема» OpenVINO

- Открытая архитектура, открытый исходный код
- Дополнительные средства:
  - NNCF (neural network compression framework) } Поддержка тренировки
  - Training extensions }
  - OpenVINO Model Server } Удаленное выполнение
  - DevCloud for the Edge }
  - DL Streamer (поддержка OpenVINO в gstreamer)

```
gst-launch-1.0 filesrc location=cut.mp4 ! decodebin ! videoconvert ! gvadetect  
model=face-detection-adas-0001.xml ! gvaclassify model=emotions-recognition-retail-  
0003.xml model-proc=emotions-recognition-retail-0003.json ! gvawatermark ! xvimagesink  
sync=false
```

# Дополнительные материалы

## Тренинги

- [Курсы по Deep Learning на Coursera](#)

## Книги

- [Николенко С.И., Кадури́н А. А. Глубокое обучение. Погружение в мир нейронных сетей](#)
- [Н.Будума, Н.Локашо. Основы глубокого обучения](#)

## Ресурсы в интернете

- [Документация по OpenVINO](#)
- [Papers with Code](#)

# We are hiring!

У нас много сложной и интересной работы!

JR0205104 – Deep Learning Software Engineering Intern (OpenVINO, GNA)

JR0152999 – Deep Learning Software Engineer (OpenVINO, GNA)

**Контакт:** [denis.orlov@intel.com](mailto:denis.orlov@intel.com)



