OpenVINO™ Toolkit Overview

Alexander Nesterov

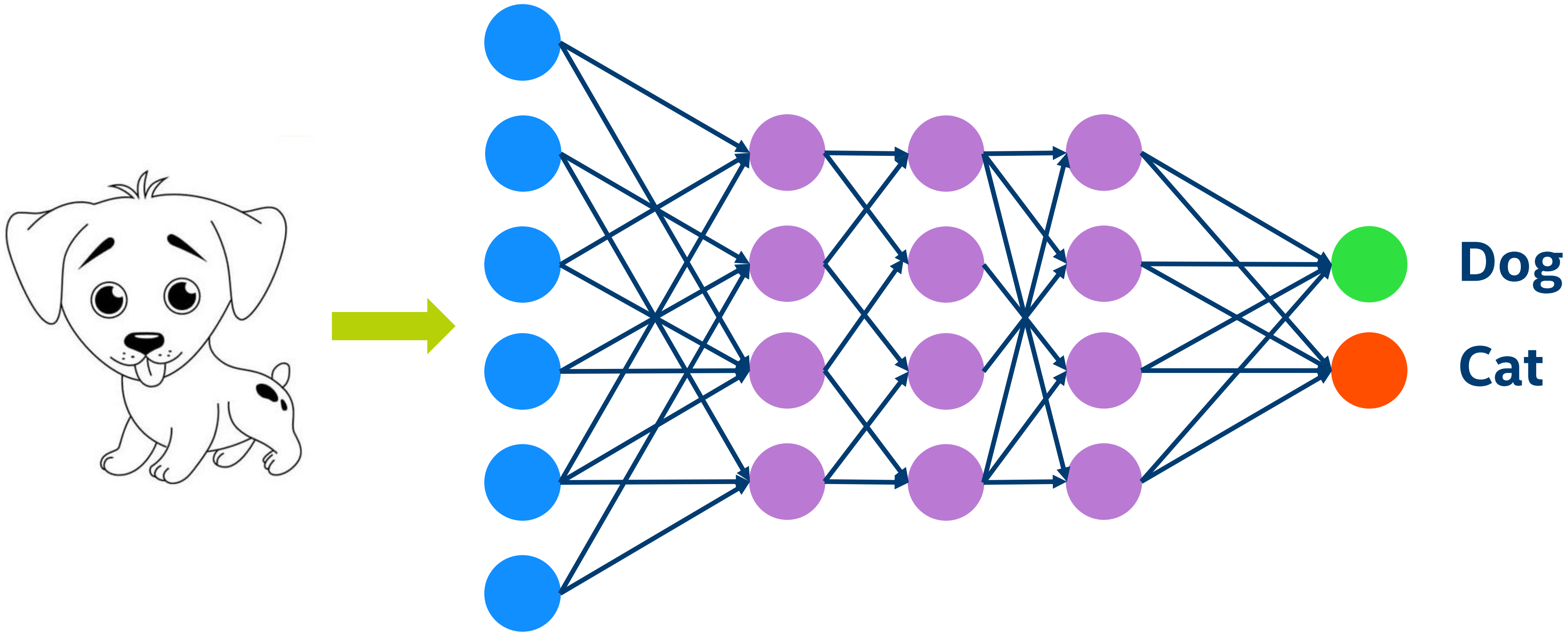
Internet of Things Group

What is OpenVINO Toolkit?

Different applications

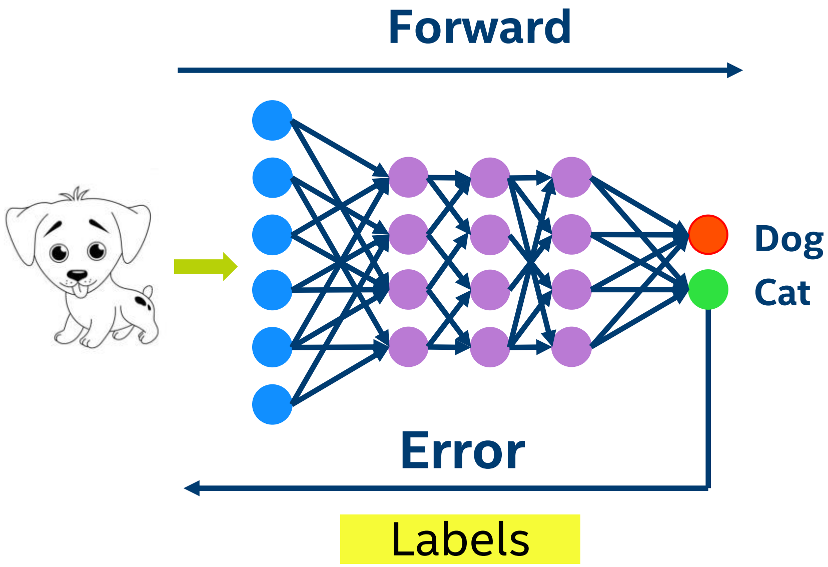
- Solutions for emulate human vision
  - ✓ Convolution neural networks
  - ✓ Traditional computer vision
- Supports heterogeneous execution
  - ✓ CPU, iGPU, FPGA, VPU
- Easy-to-use library of pre-optimized kernels

Deep neural network

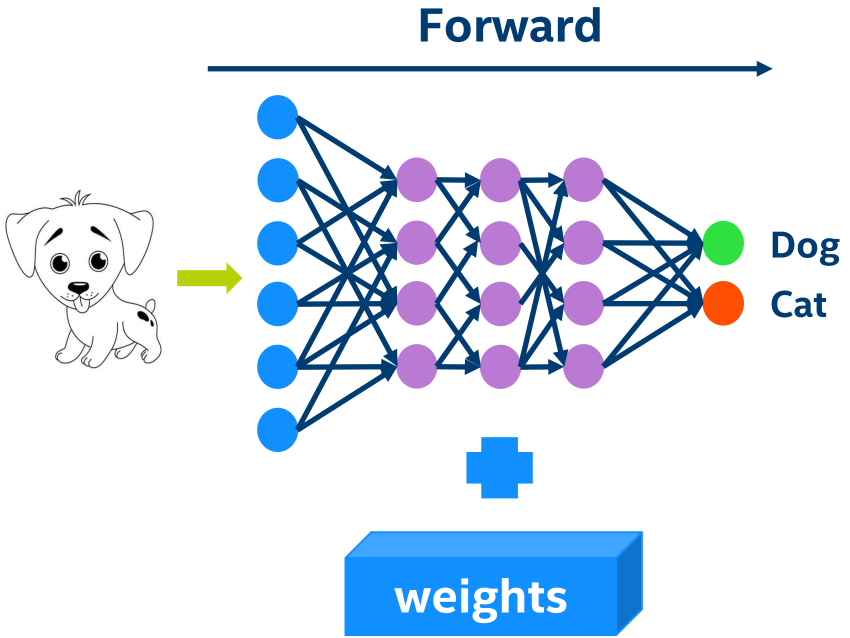


What can be done with DNN?

Learning

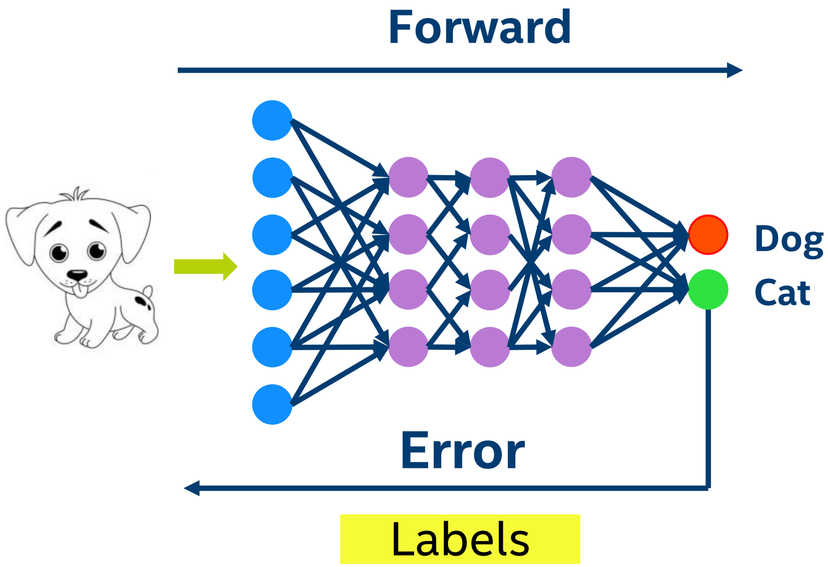


Inference



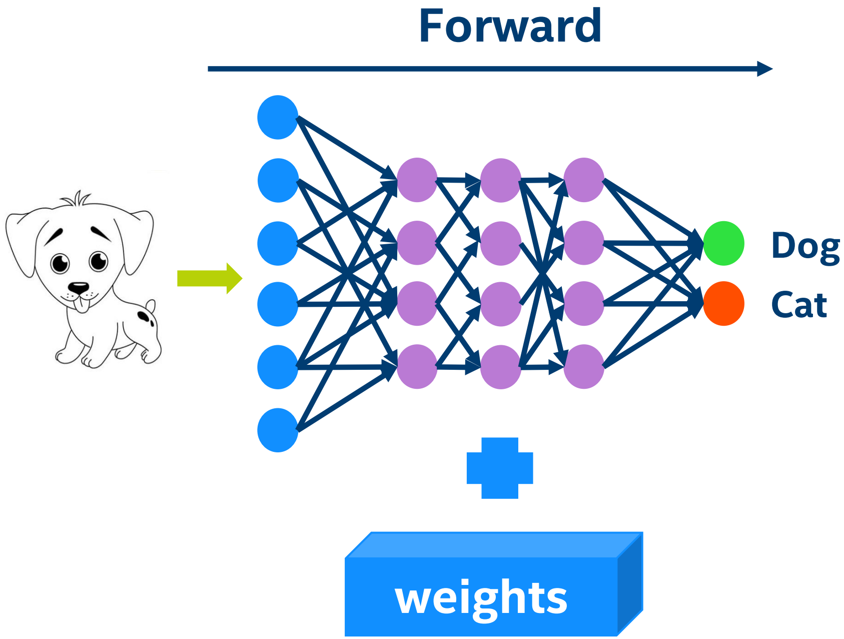
What can be done with DNN?

Learning



Inference

OpenVINO™



OpenVINO™

Model Optimizer

Deep Learning Workbench

Inference Engine

Samples

OpenVINO™

Model Optimizer

Deep Learning Workbench

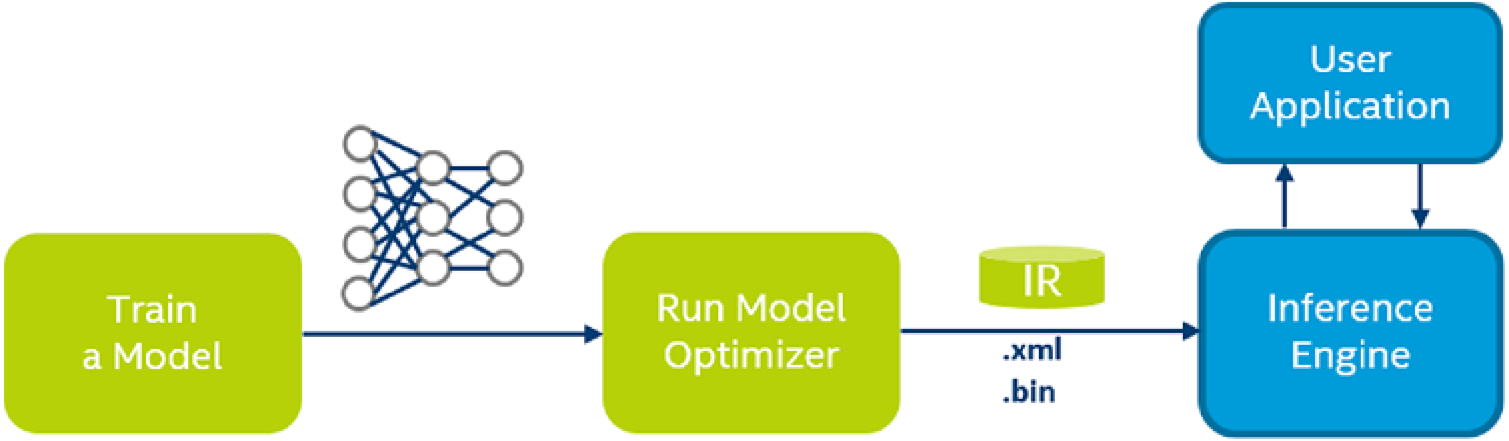
Inference Engine

Samples

https://github.com/openvinotoolkit/openvino

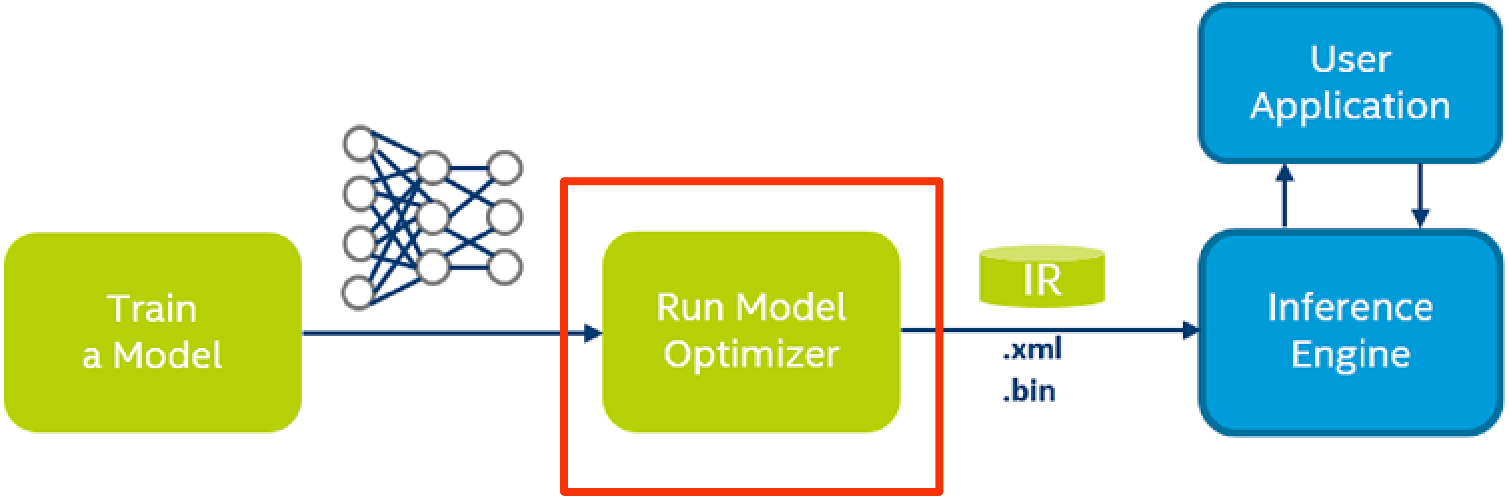
Model Optimizer

OpenVINO™

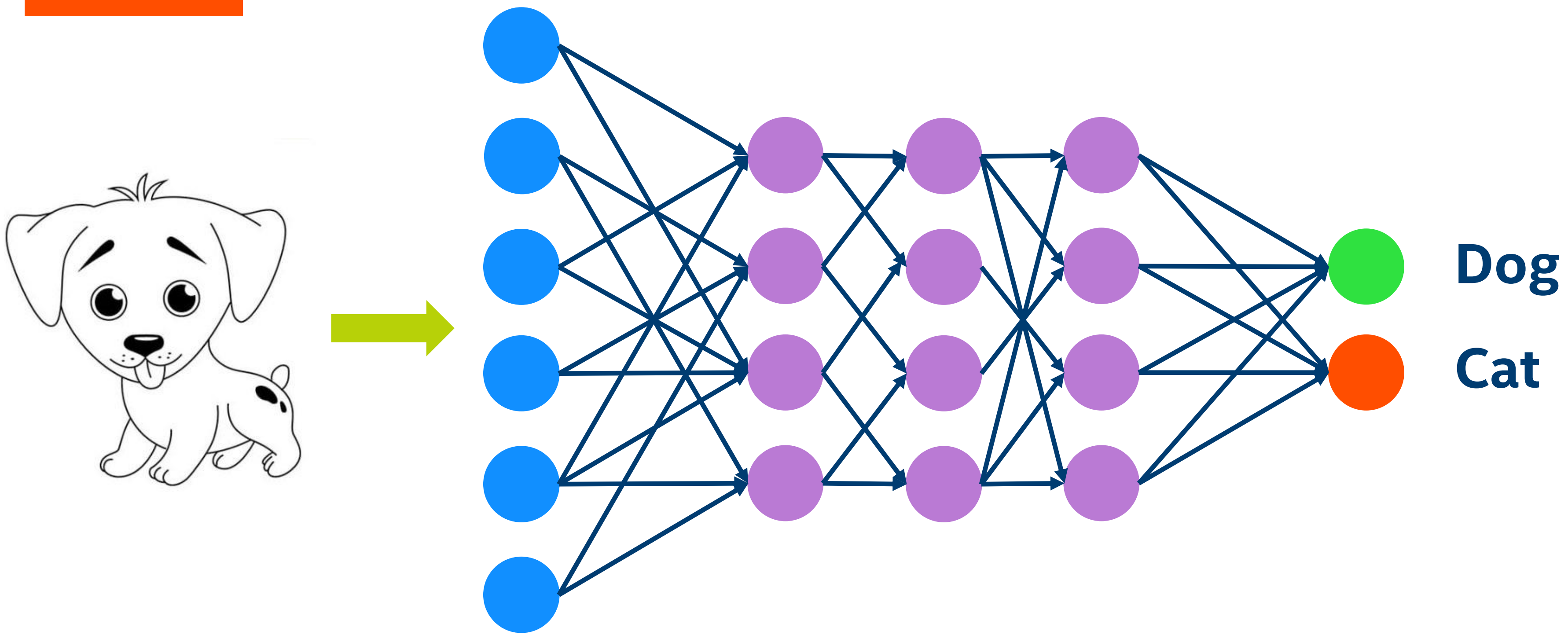


Model Optimizer

OpenVINO™



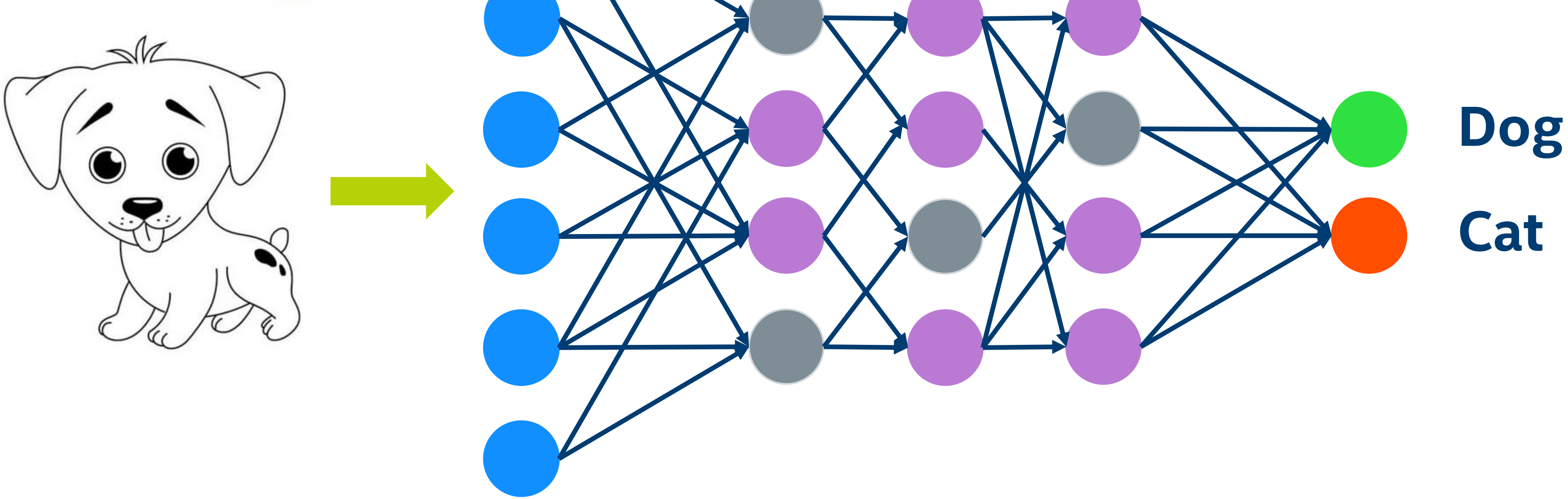
Model Optimizer



Model Optimizer

OpenVINO™

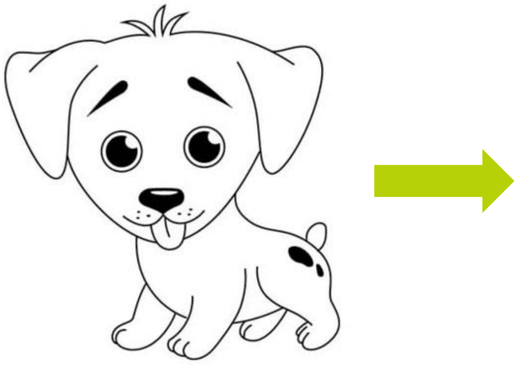
Search equivalent vertices

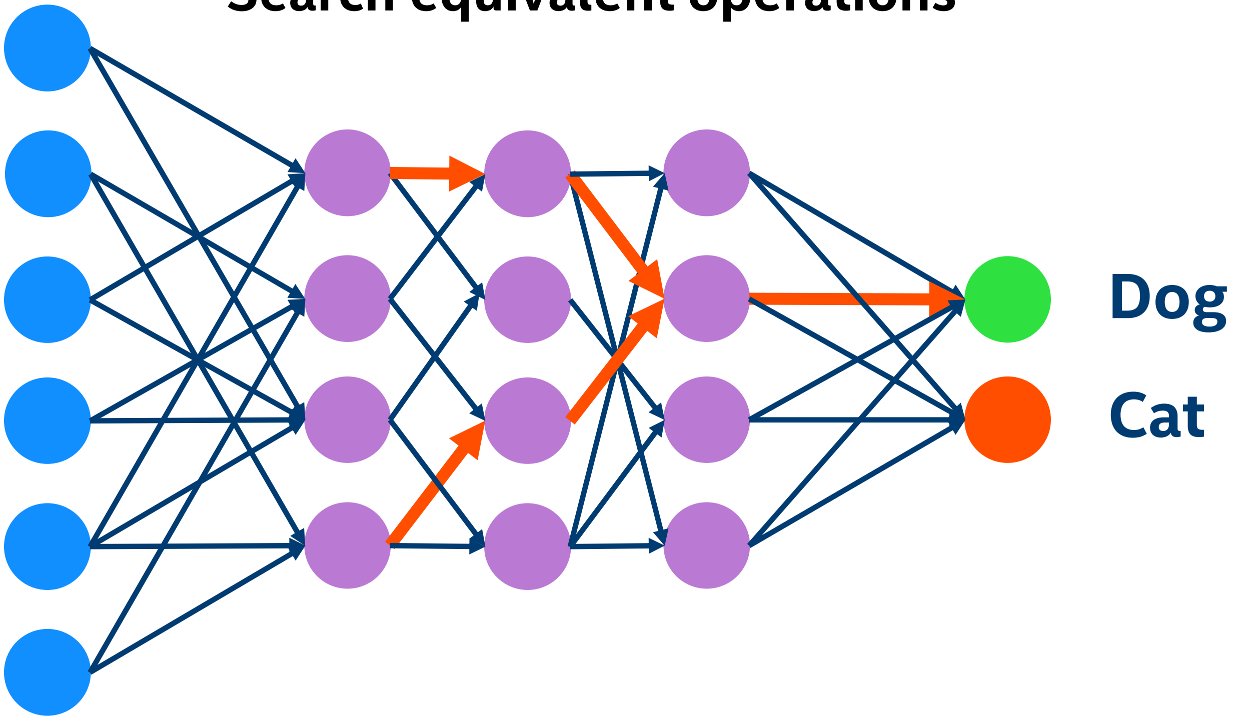


Model Optimizer

OpenVINO™

Search equivalent operations

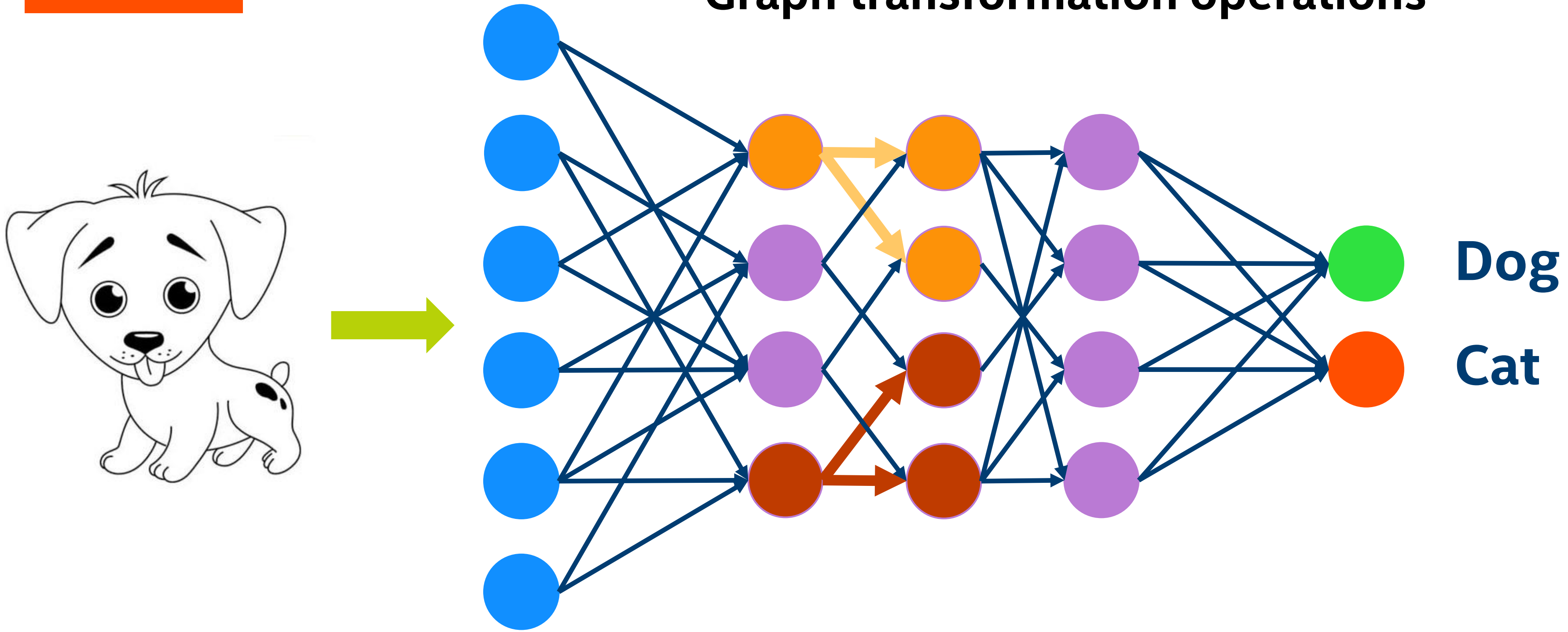




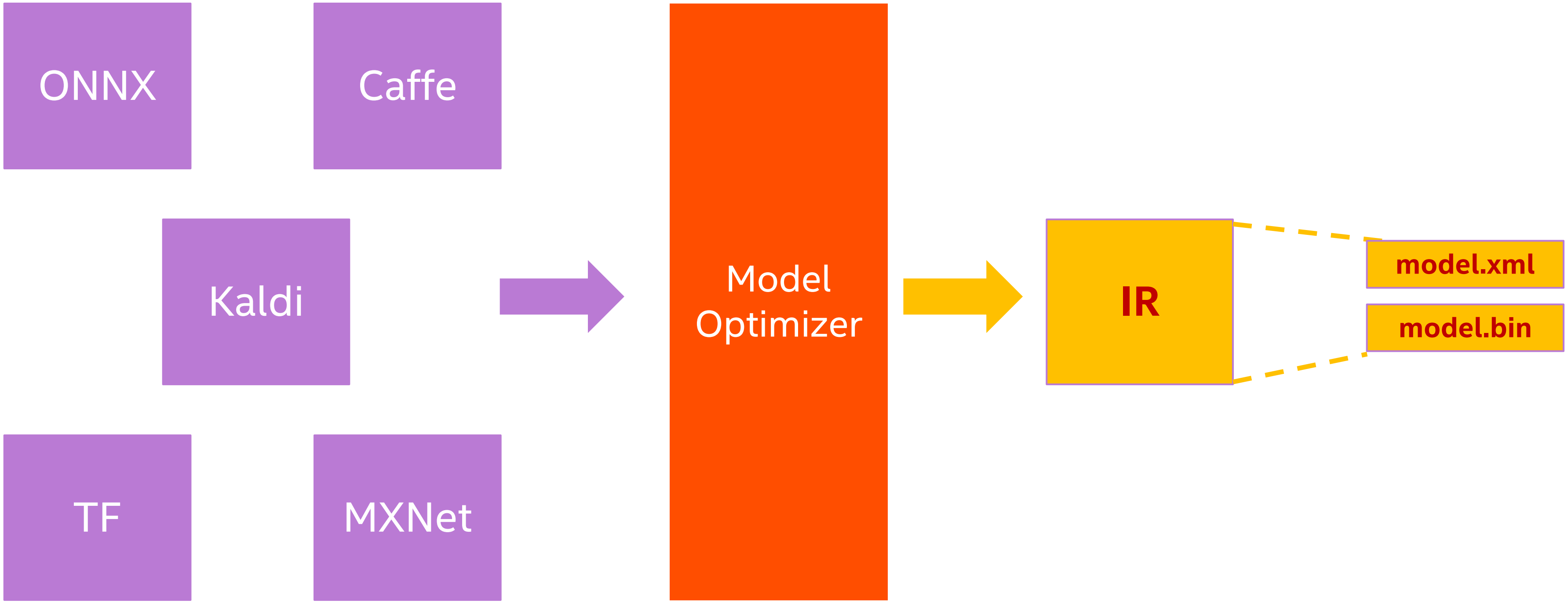
Model Optimizer

OpenVINO™

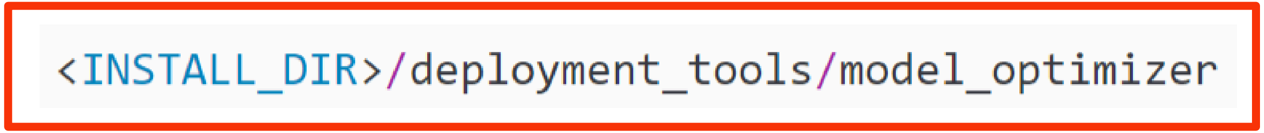
Graph transformation operations



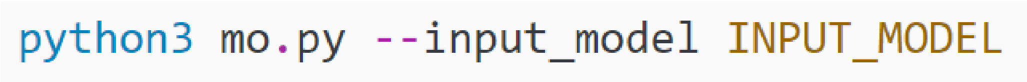
OpenVINO™


ONNX
Caffe
Kaldi
TF
MXNet
Model Optimizer
IR
model.xml
model.bin

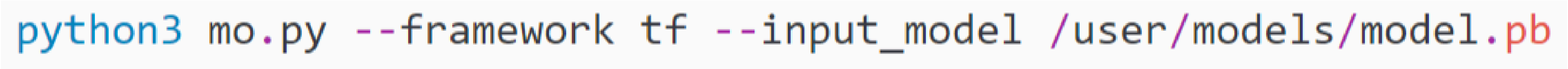
Model Optimizer



```
<INSTALL_DIR>/deployment_tools/model_optimizer
```



```
python3 mo.py --input_model INPUT_MODEL
```



```
python3 mo.py --framework tf --input_model /user/models/model.pb
```

OpenVINO™

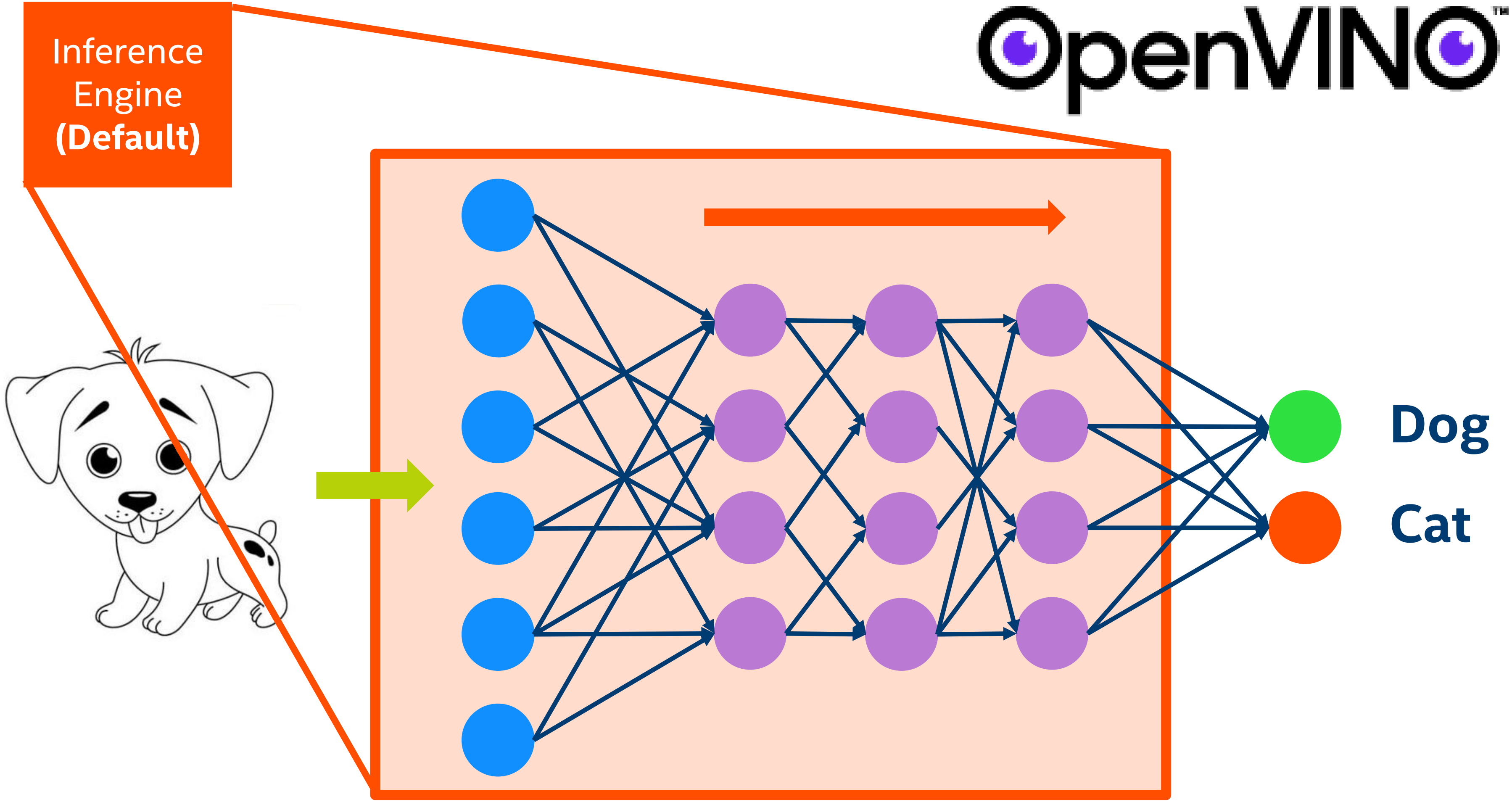
Model Optimizer

Deep Learning Workbench

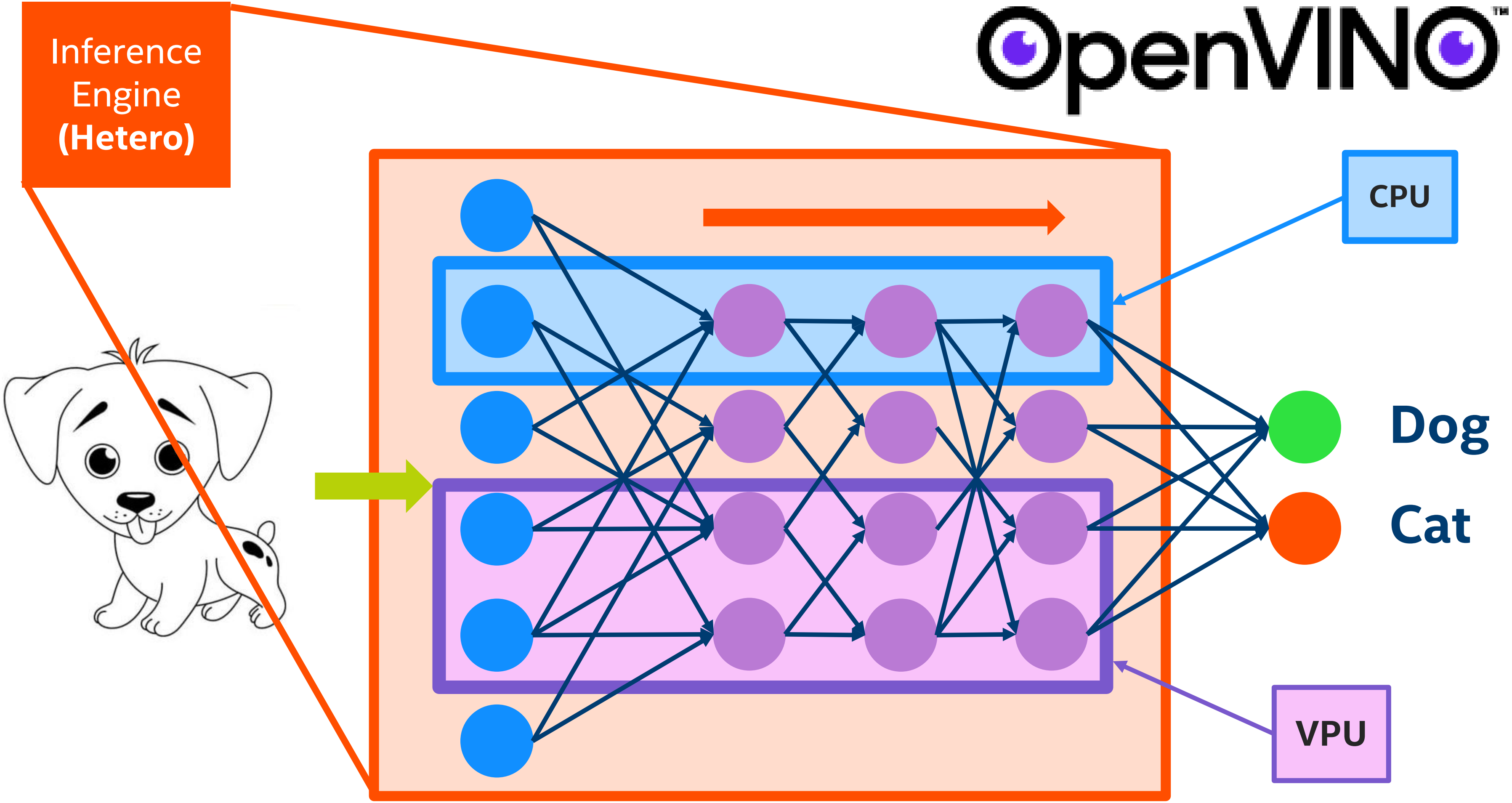
Inference Engine

Samples

OpenVINO™

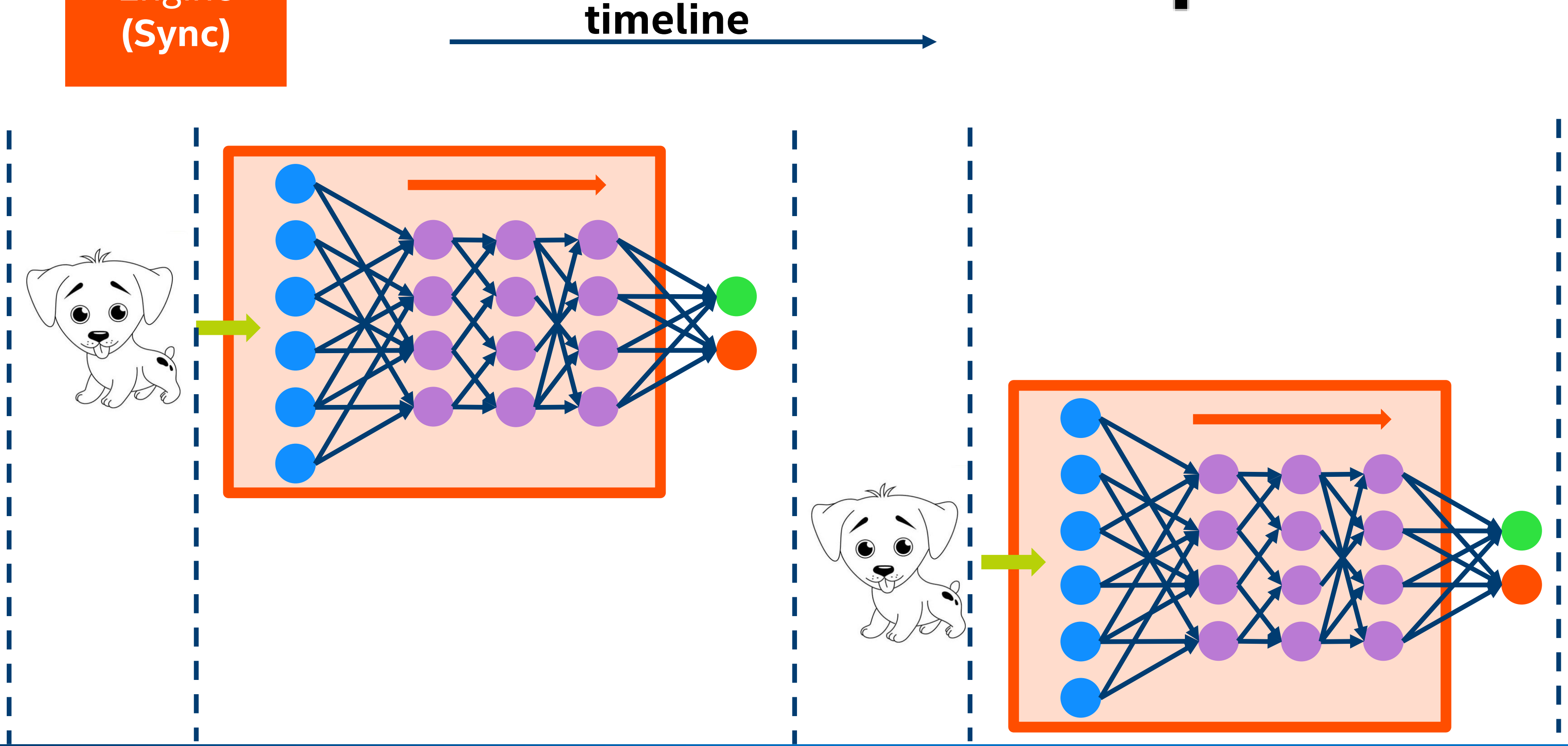


OpenVINO™



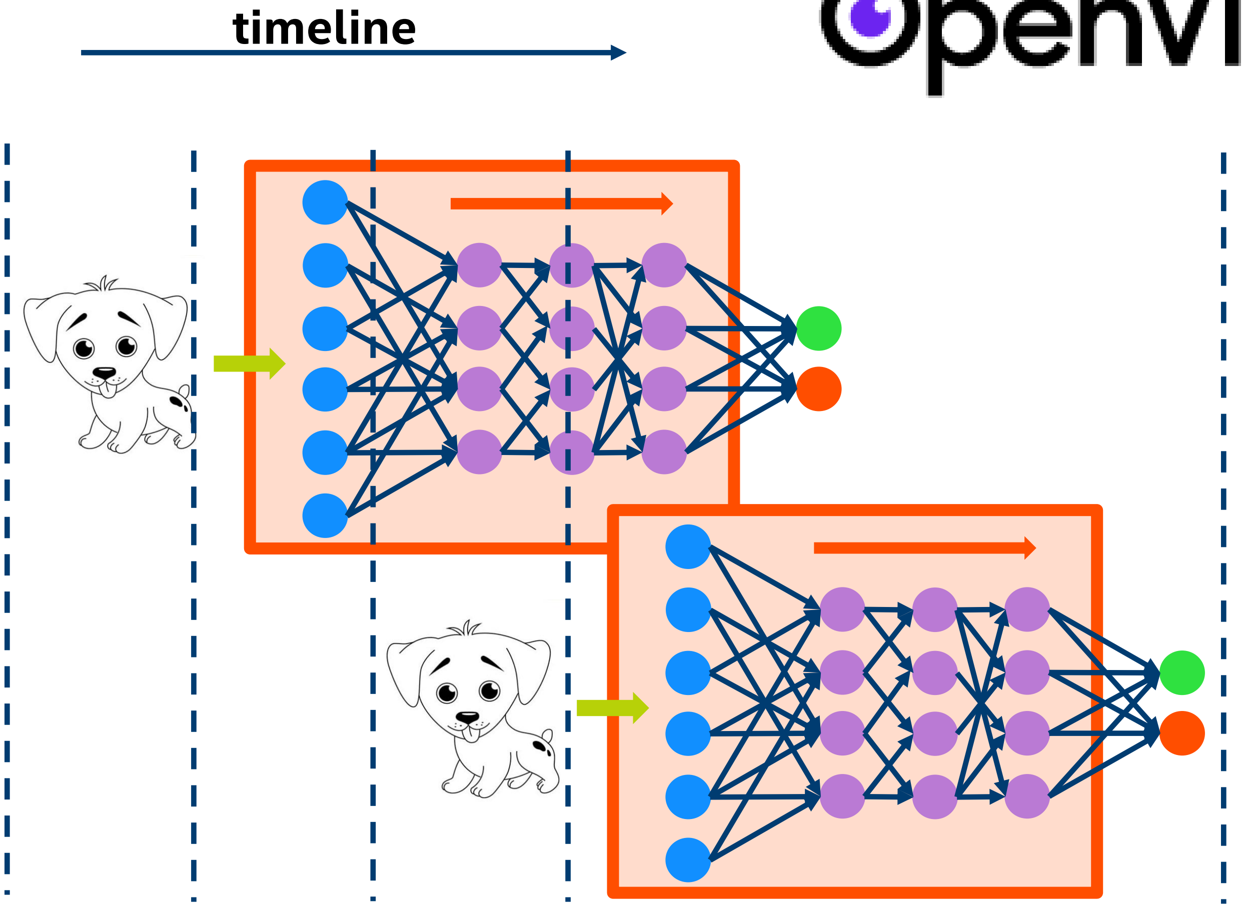
Inference Engine **(Sync)**

OpenVINO™

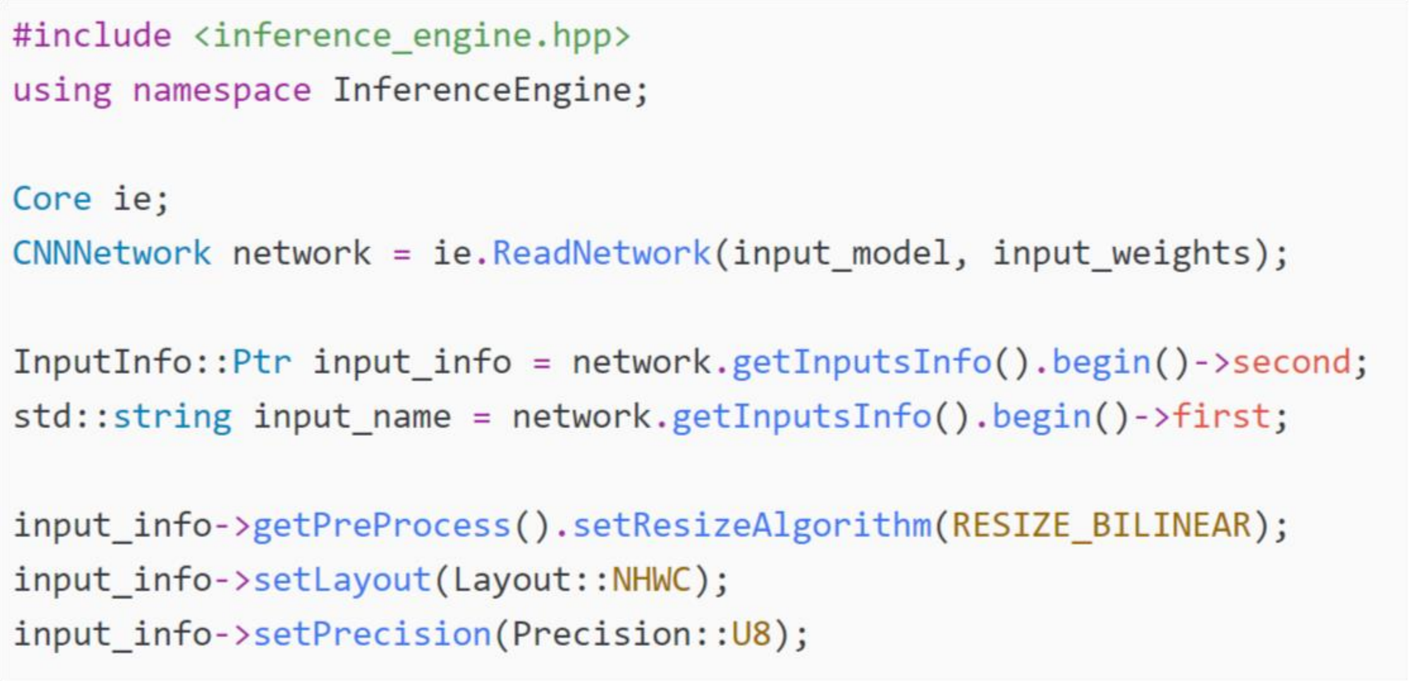


Inference Engine **(Async)**

OpenVINO™

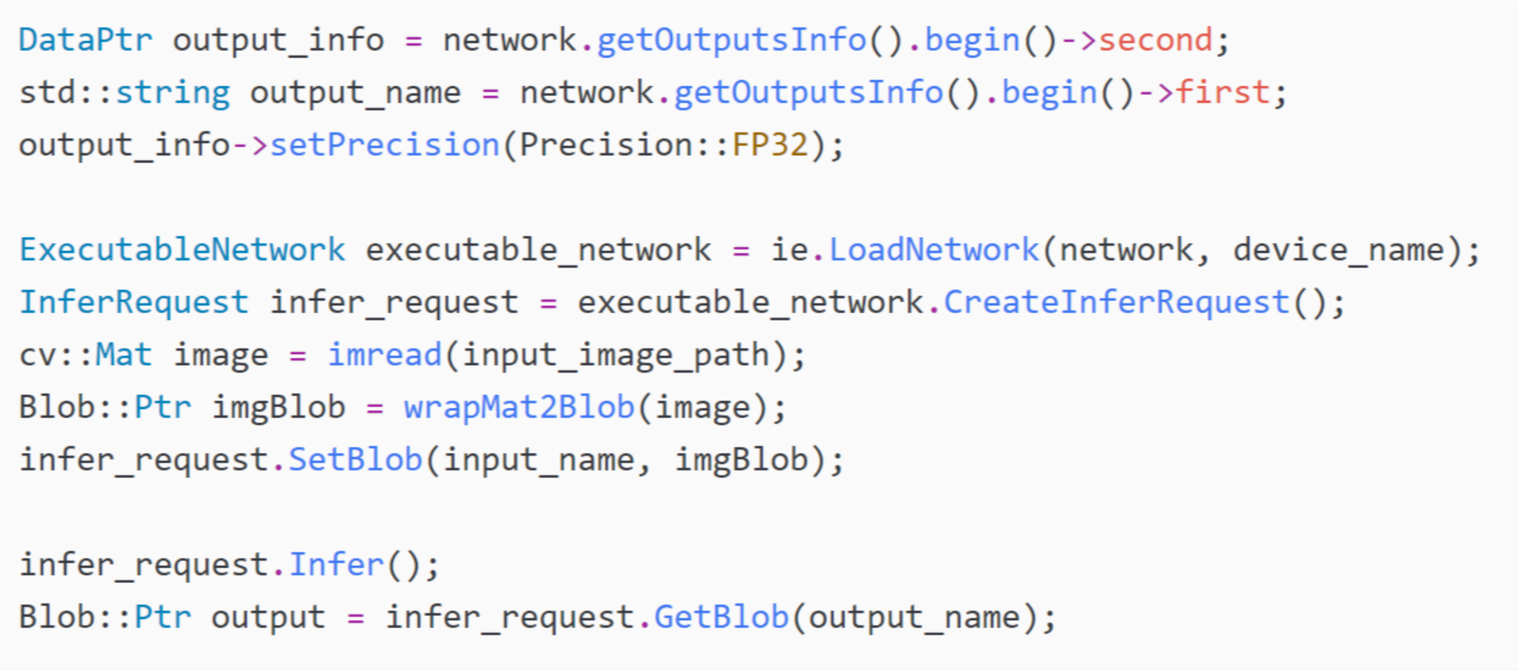


Inference Engine (C++)



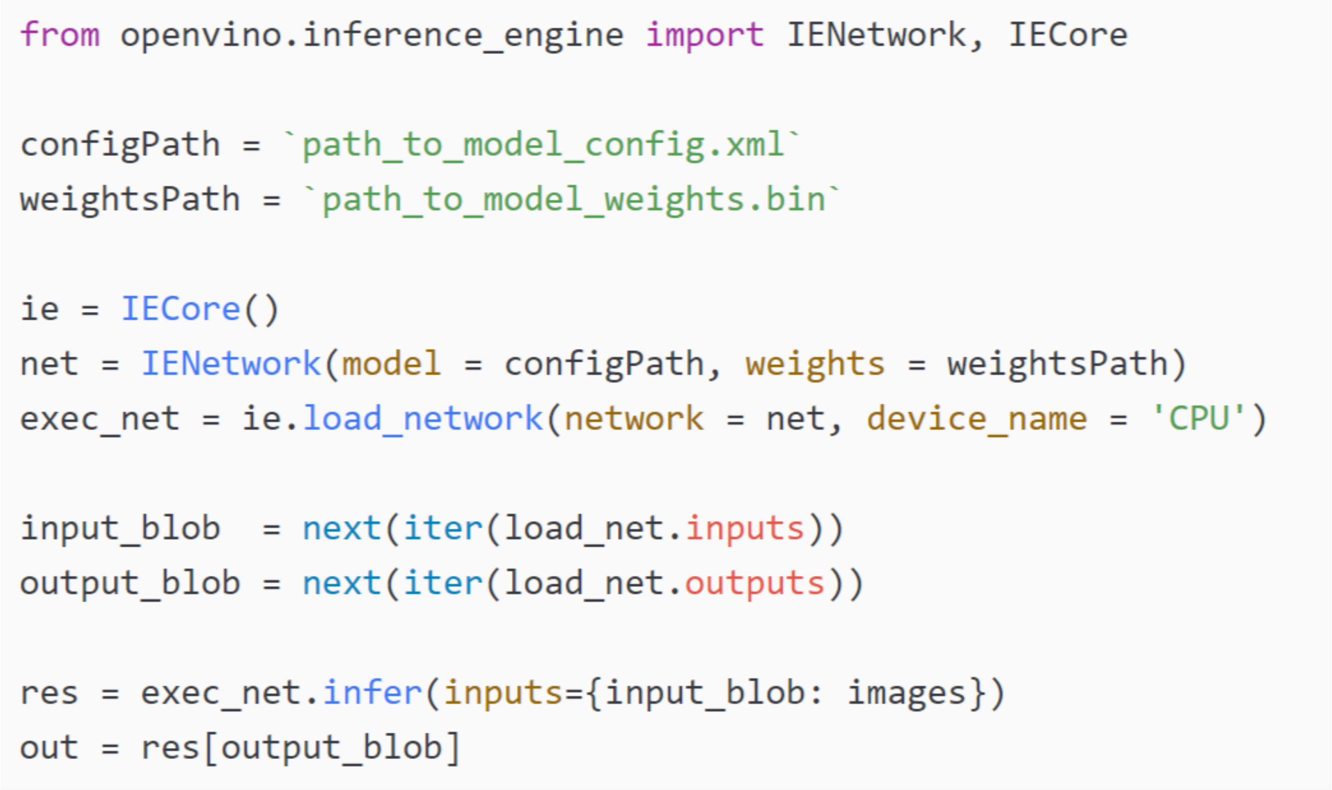
```cpp
#include <inference_engine.hpp>
using namespace InferenceEngine;

Core ie;
CNNNetwork network = ie.ReadNetwork(input_model, input_weights);

InputInfo::Ptr input_info = network.getInputsInfo().begin()->second;
std::string input_name = network.getInputsInfo().begin()->first;

input_info->getPreProcess().setResizeAlgorithm(RESIZE_BILINEAR);
input_info->setLayout(Layout::NHWC);
input_info->setPrecision(Precision::U8);
```

Inference Engine (C++)



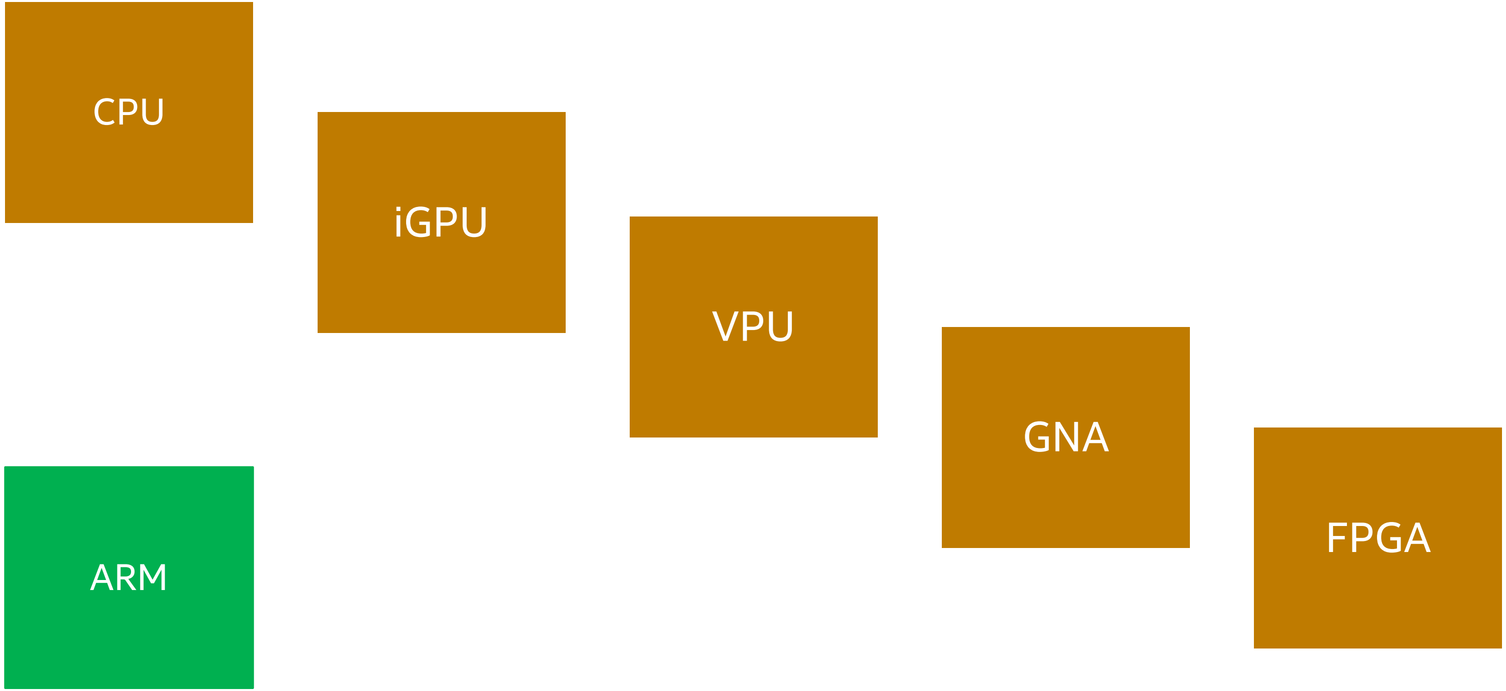
```
DataPtr output_info = network.getOutputsInfo().begin()->second;
std::string output_name = network.getOutputsInfo().begin()->first;
output_info->setPrecision(Precision::FP32);

ExecutableNetwork executable_network = ie.LoadNetwork(network, device_name);
InferRequest infer_request = executable_network.CreateInferRequest();
cv::Mat image = imread(input_image_path);
Blob::Ptr imgBlob = wrapMat2Blob(image);
infer_request.SetBlob(input_name, imgBlob);

infer_request.Infer();
Blob::Ptr output = infer_request.GetBlob(output_name);
```

Inference Engine **(Python)**



```
from openvino.inference_engine import IENetwork, IECore

configPath = `path_to_model_config.xml`
weightsPath = `path_to_model_weights.bin`

ie = IECore()
net = IENetwork(model = configPath, weights = weightsPath)
exec_net = ie.load_network(network = net, device_name = 'CPU')

input_blob  = next(iter(load_net.inputs))
output_blob = next(iter(load_net.outputs))

res = exec_net.infer(inputs={input_blob: images})
out = res[output_blob]
```

OpenVINO™


CPU

iGPU

VPU

GNA

FPGA

ARM

https://github.com/openvinotoolkit/openvino_contrib

OpenVINO™

Model Optimizer

Deep Learning Workbench

Inference Engine

Samples

OpenVINO™

Model Optimizer

Deep Learning Workbench

Inference Engine

Samples

https://docs.openvinotoolkit.org

OpenVINO™

Accuracy Checker Utility

Benchmark App

Cross Check Tool

Post-training Optimization Tool

OpenVINO™

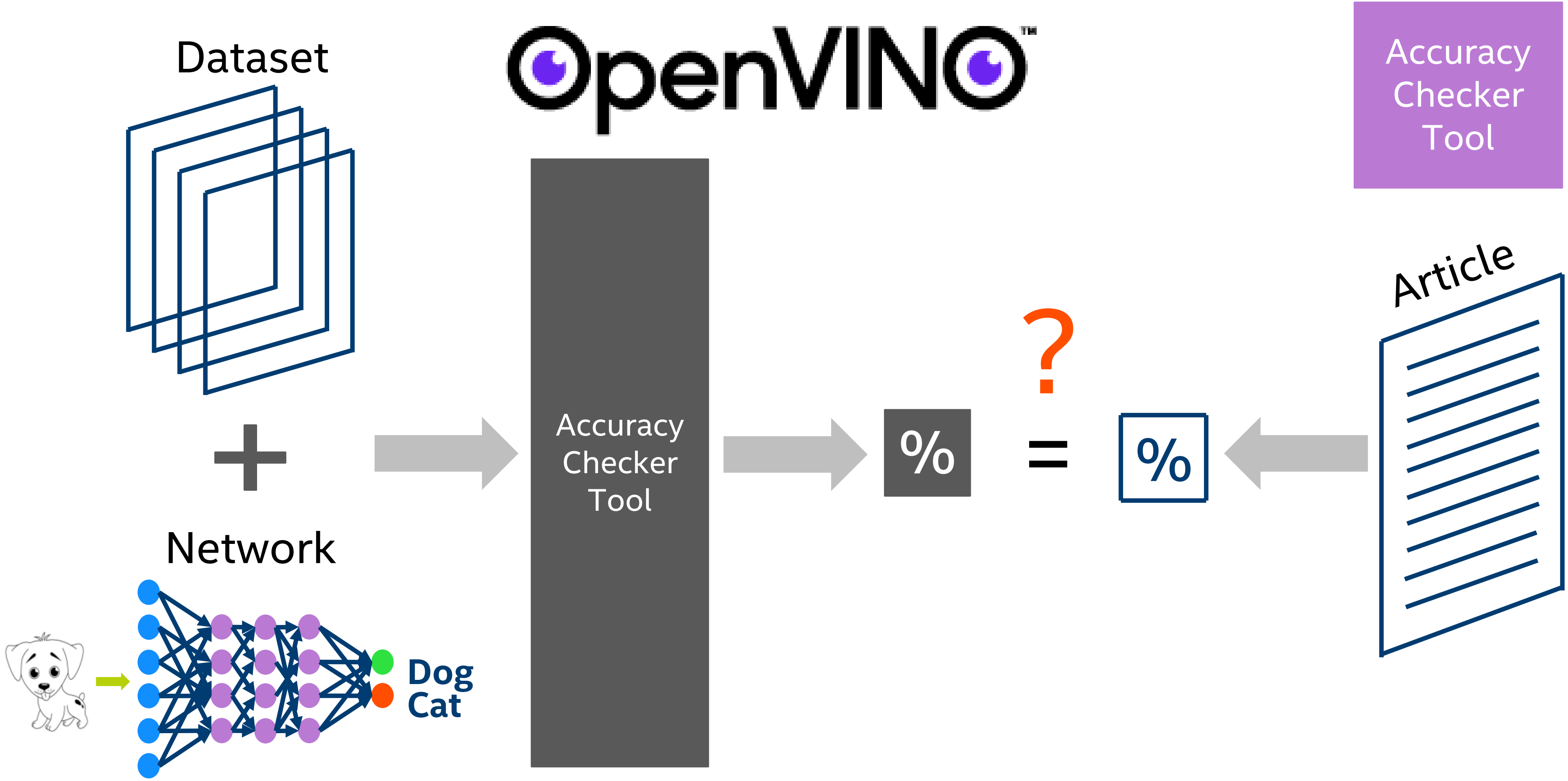
Accuracy Checker Utility

Benchmark App

Cross Check Tool

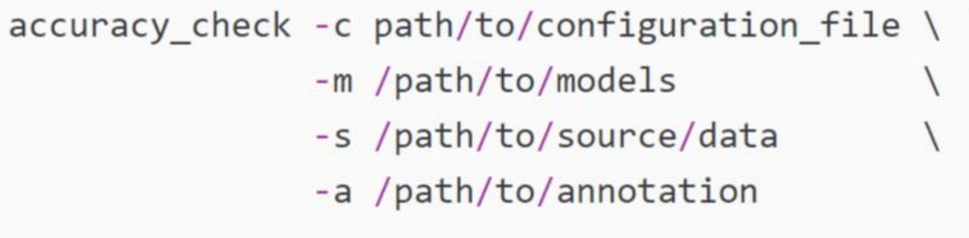
Post-training Optimization Tool

Accuracy Checker Tool

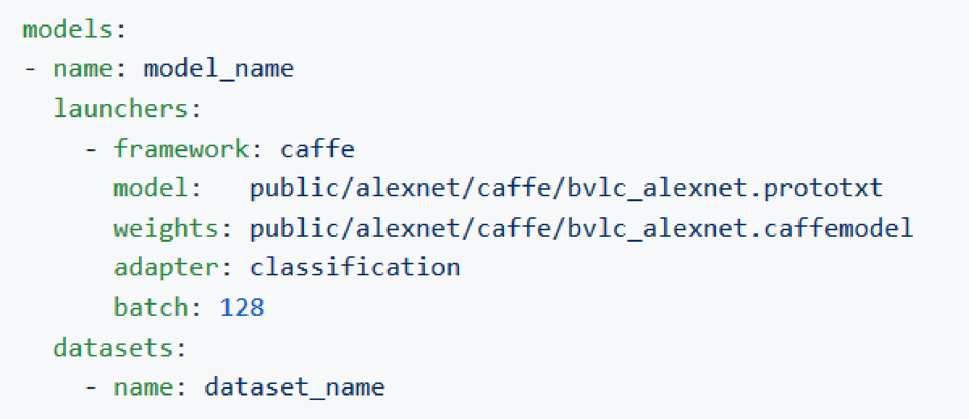


OpenVINO™

Accuracy Checker Tool



```
accuracy_check -c path/to/configuration_file \
               -m /path/to/models            \
               -s /path/to/source/data       \
               -a /path/to/annotation
```



```
models:
- name: model_name
  launchers:
    - framework: caffe
      model:   public/alexnet/caffe/bvlc_alexnet.prototxt
      weights: public/alexnet/caffe/bvlc_alexnet.caffemodel
      adapter: classification
      batch: 128
  datasets:
    - name: dataset_name
```

OpenVINO™

Accuracy Checker Utility

Benchmark App

Cross Check Tool

Post-training Optimization Tool

OpenVINO™

Accuracy Checker Utility

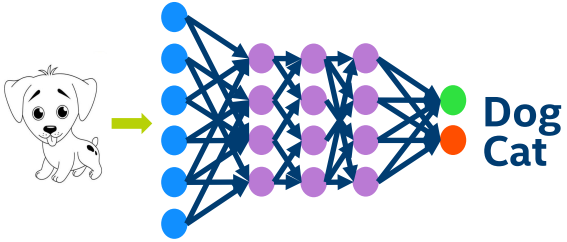
Benchmark App

Cross Check Tool

Post-training Optimization Tool

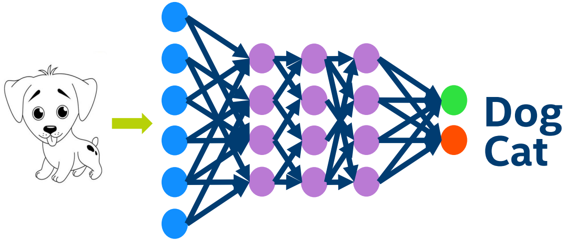
OpenVINO™

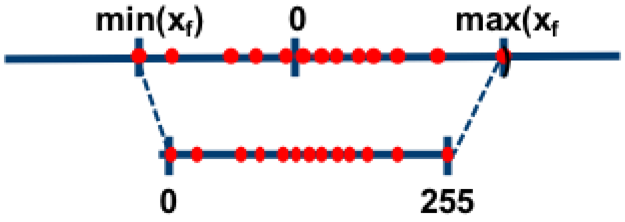
FP32 / FP16



Post-training Optimization Tool

INT8





OpenCV

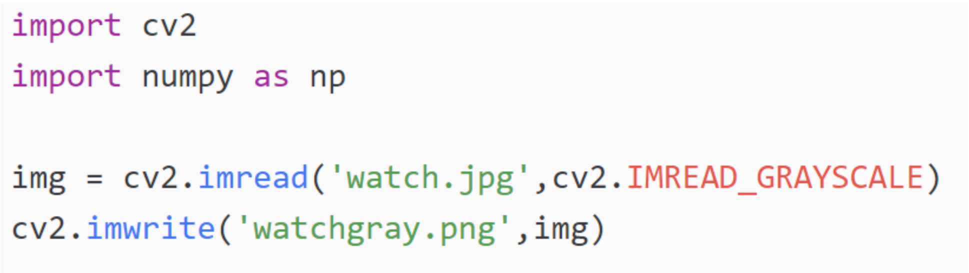
OpenVINO™

- C/C++
- Python
- Java
- Matlab
- JavaScript

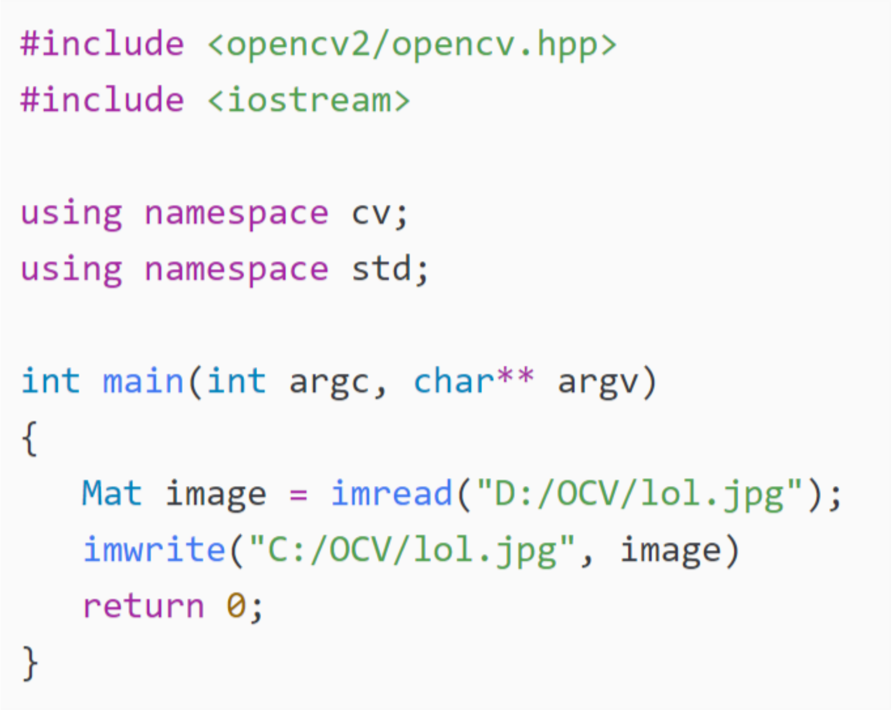
core	imgproc	imgcodecs	videoio
highgui	video	calib3d	features2d
objdetect	dnn	ml	flann
photo	stitching	gapi	tracking

OpenCV

OpenVINO™



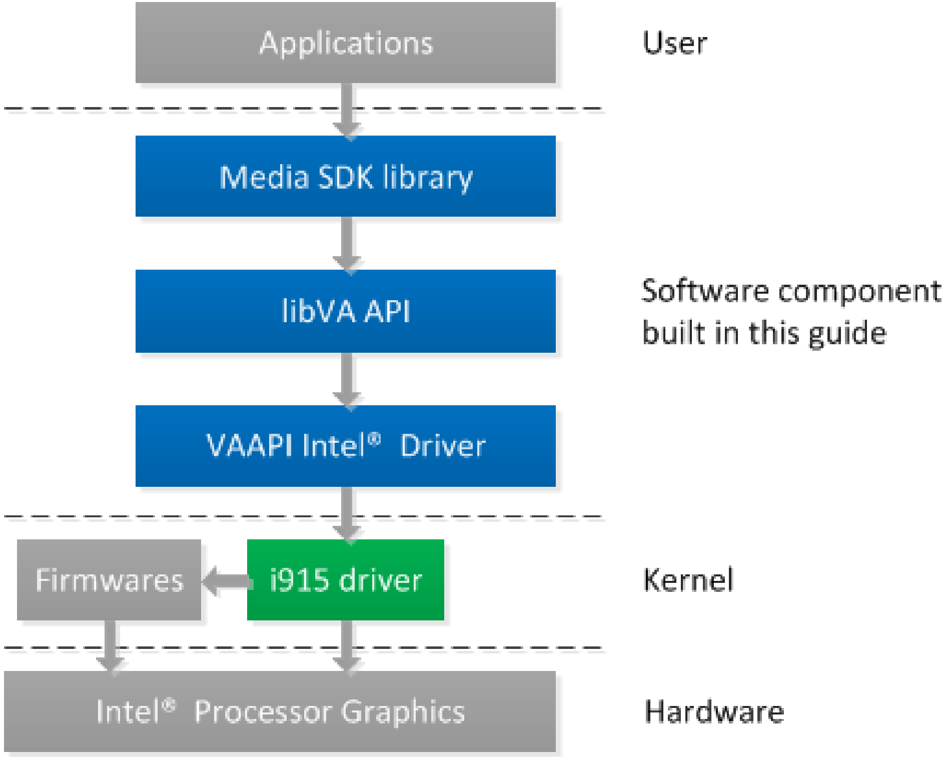
```python
import cv2
import numpy as np

img = cv2.imread('watch.jpg',cv2.IMREAD_GRAYSCALE)
cv2.imwrite('watchgray.png',img)
```



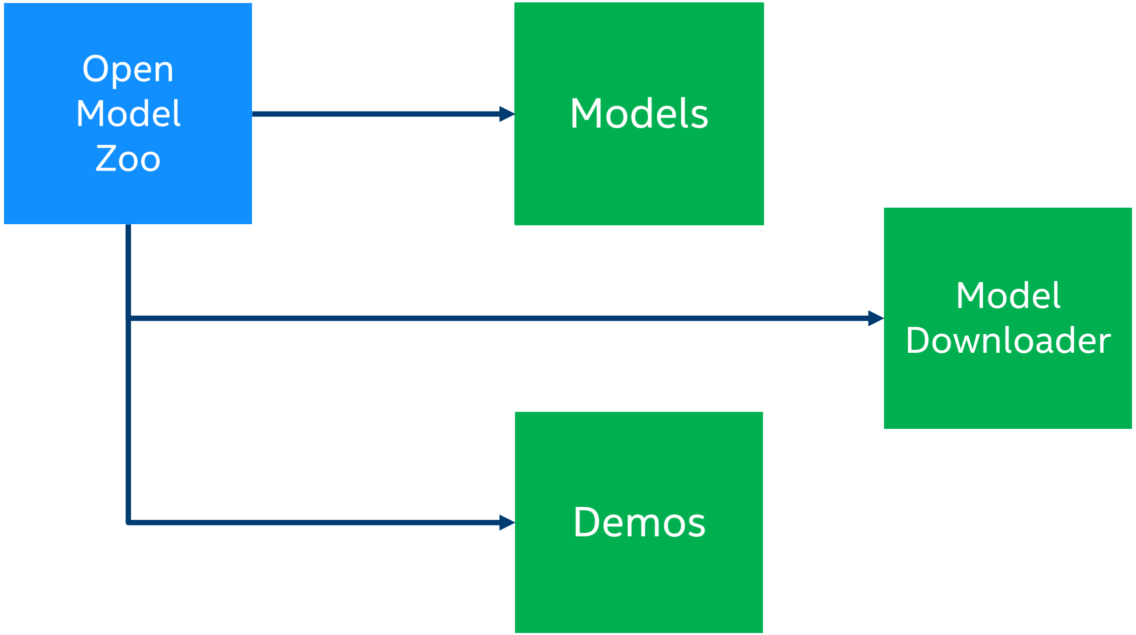
```cpp
#include <opencv2/opencv.hpp>
#include <iostream>

using namespace cv;
using namespace std;

int main(int argc, char** argv)
{
    Mat image = imread("D:/OCV/lol.jpg");
    imwrite("C:/OCV/lol.jpg", image)
    return 0;
}
```

Media SDK

OpenVINO™

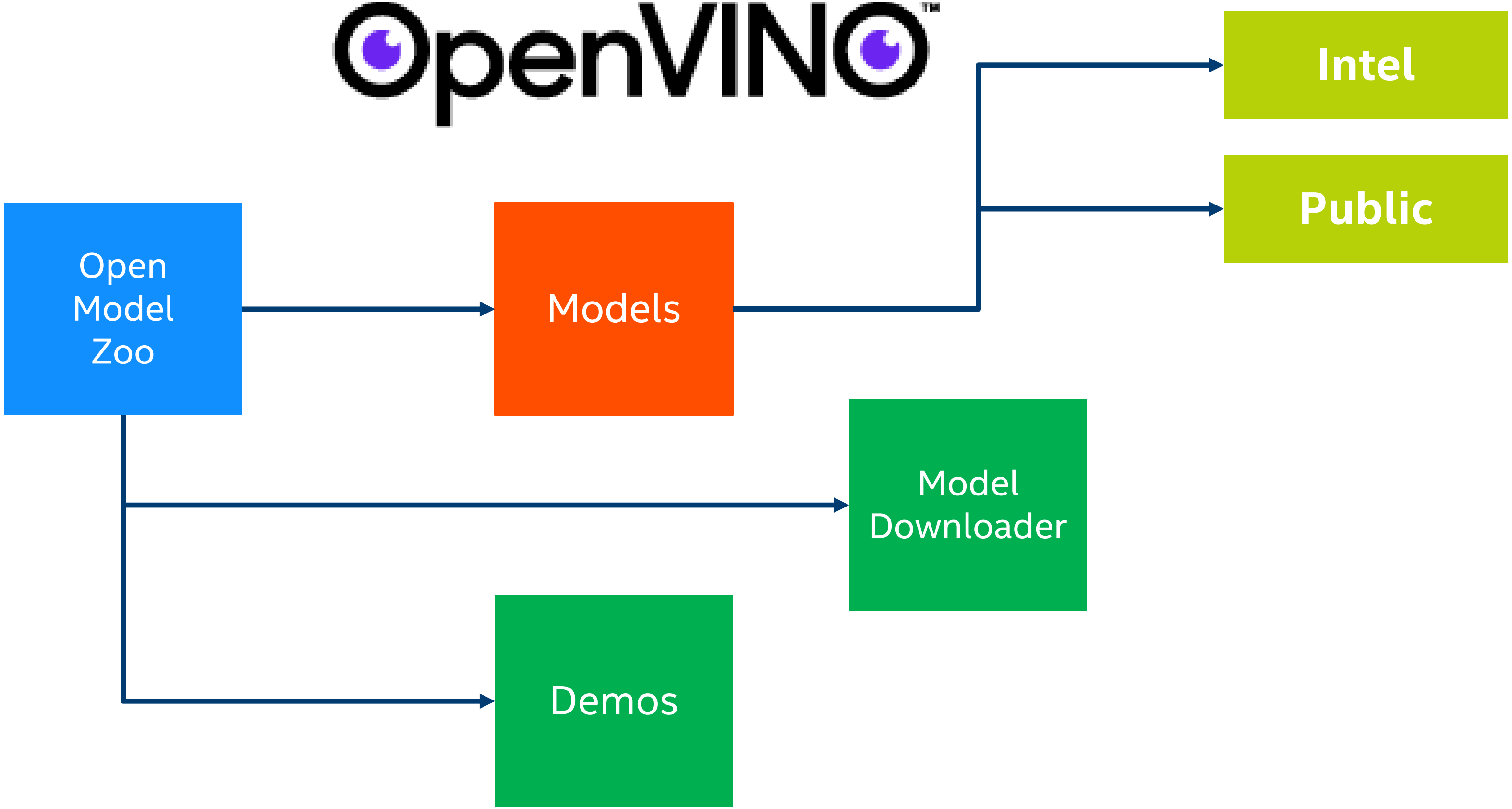


OpenVINO™



https://github.com/openvinotoolkit/open_model_zoo

OpenVINO™



Open Model Zoo

Models

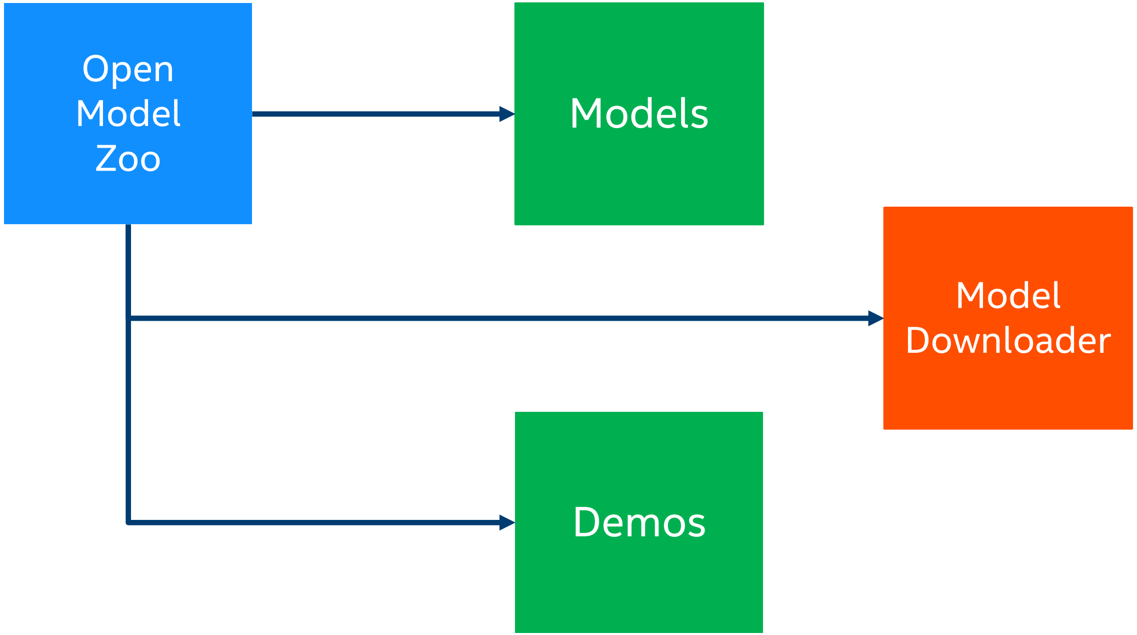
Intel

Public

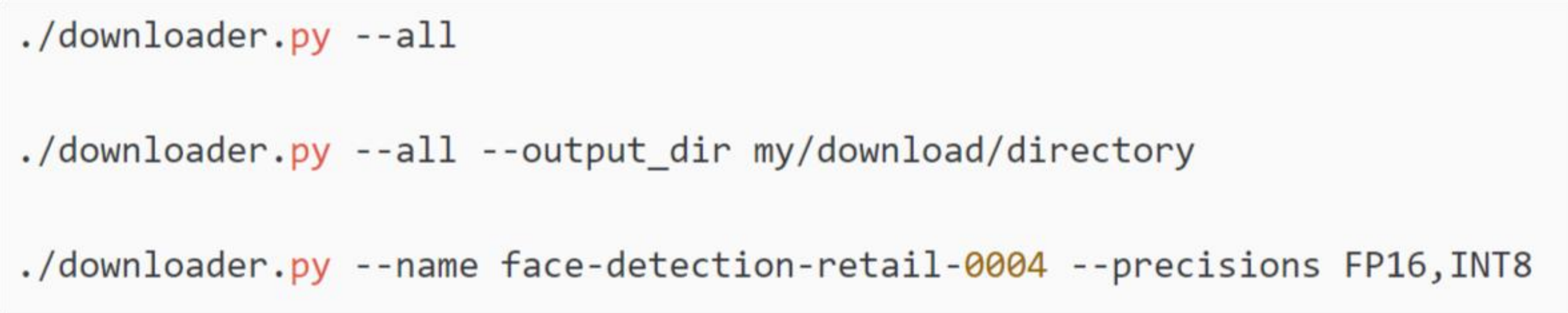
Model Downloader

Demos

OpenVINO™

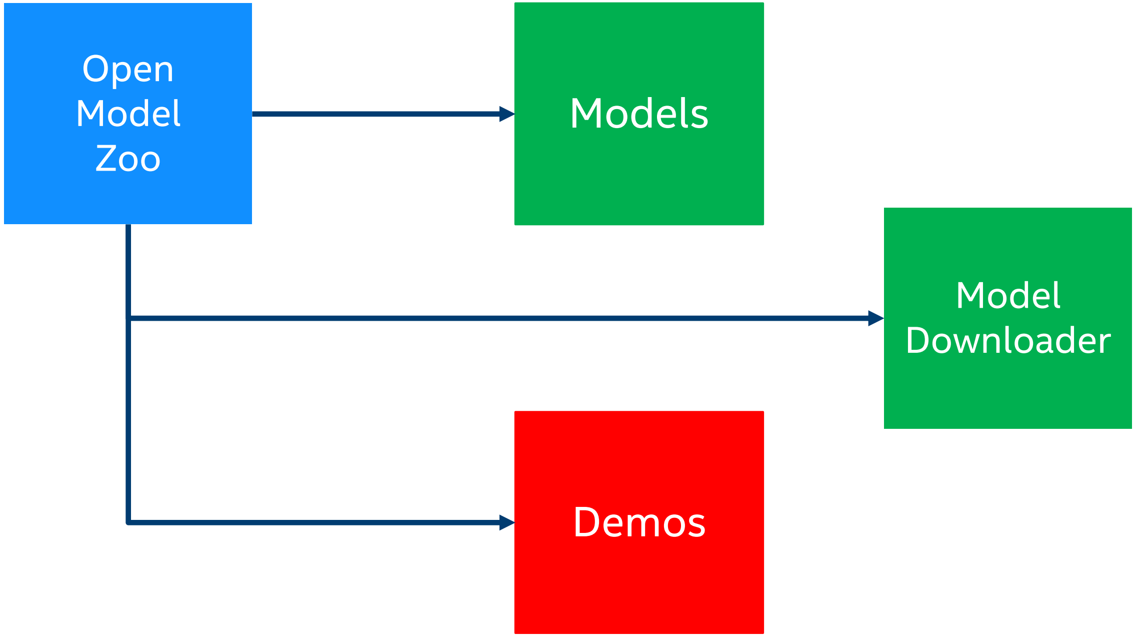


OpenVINO™



```
./downloader.py --all

./downloader.py --all --output_dir my/download/directory

./downloader.py --name face-detection-retail-0004 --precisions FP16,INT8
```

OpenVINO™



Demos

OpenVINO™

Object recognition

Action recognition

Segmentation

Tracking

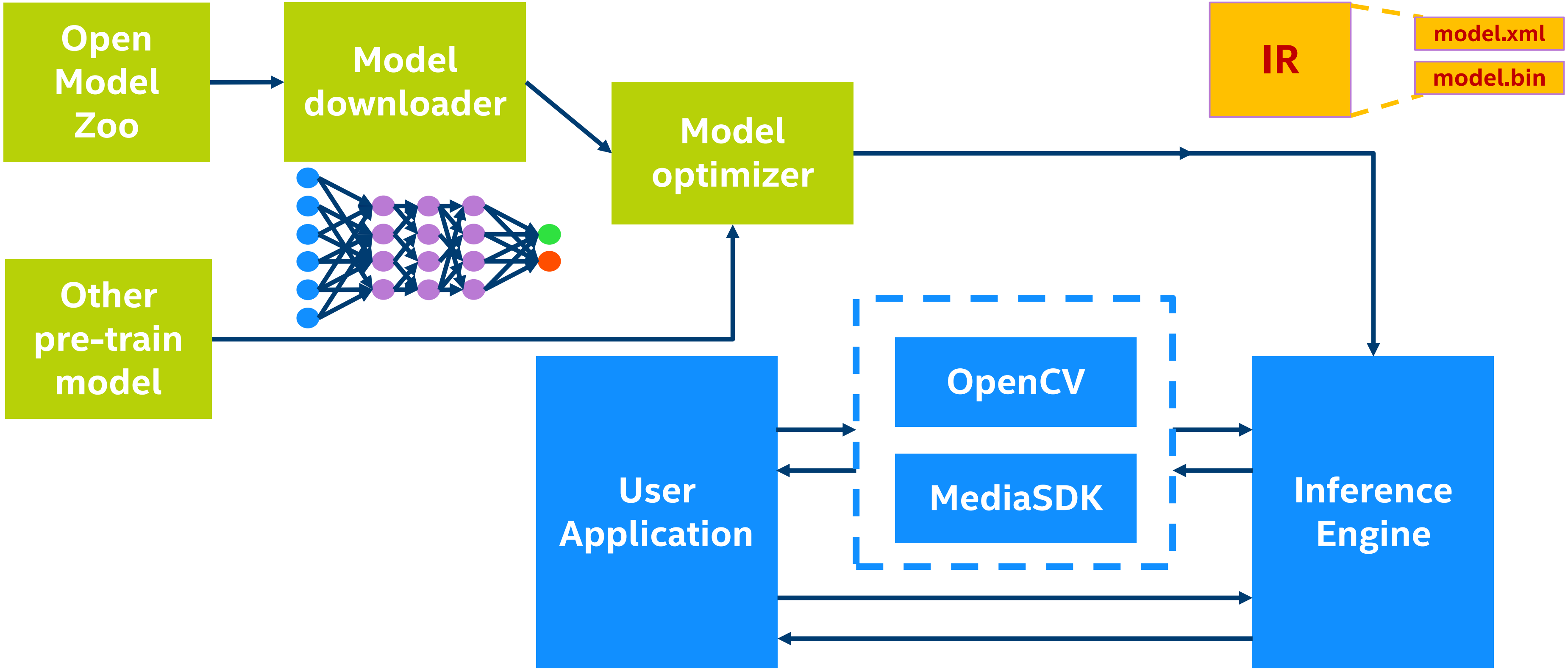
Object detection

Image processing

Text processing

Audio processing

OpenVINO™ pipeline



Open Model Zoo

Model downloader

Model optimizer

Other pre-train model

IR

model.xml

model.bin

OpenCV

MediaSDK

User Application

Inference Engine

Q & A