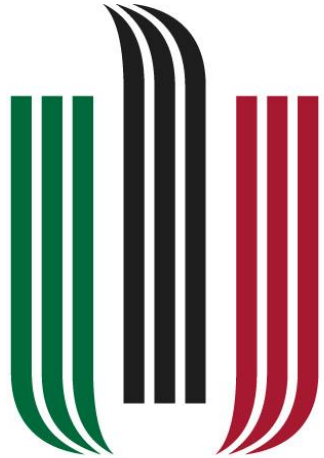


AGH AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Wydział Fizyki i Informatyki Stosowanej

Prezentacja pracy dyplomowej inżynierskiej

Interfejs wykorzystania modeli sieci neuronowych w aplikacji Unity3d

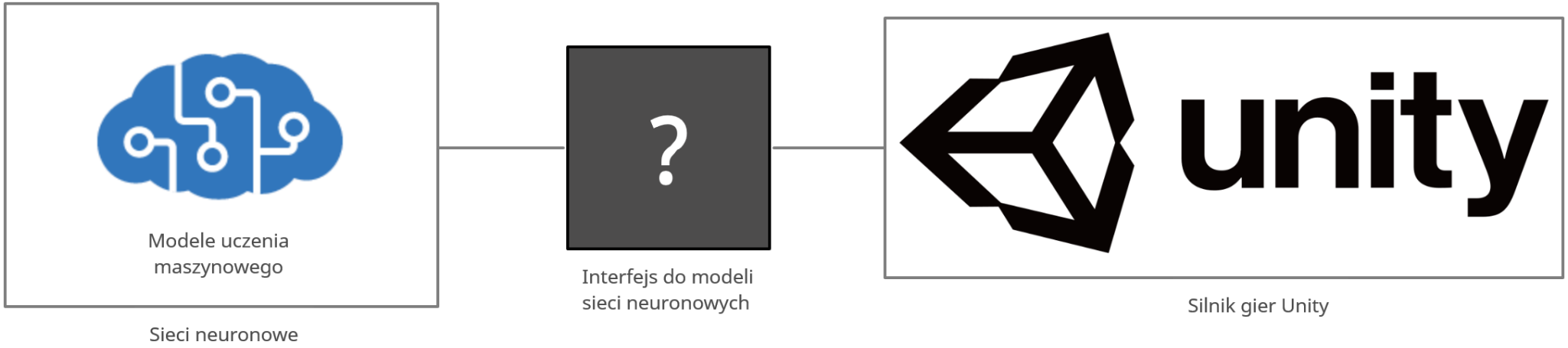
Imie i nazwisko: Michał PABJAN
Kierunek studiów: INFORMATYKA STOSOWANA

Opiekun pracy: dr. inż. Bartłomiej RACHWAŁ

Cel i głowne założenia pracy

- Celem niniejszego projektu inżynierskiego było stworzenie generycznego interfejsu dla modeli sieci neuronowych pod aplikacje rozwijane w silniku Unity 3D.

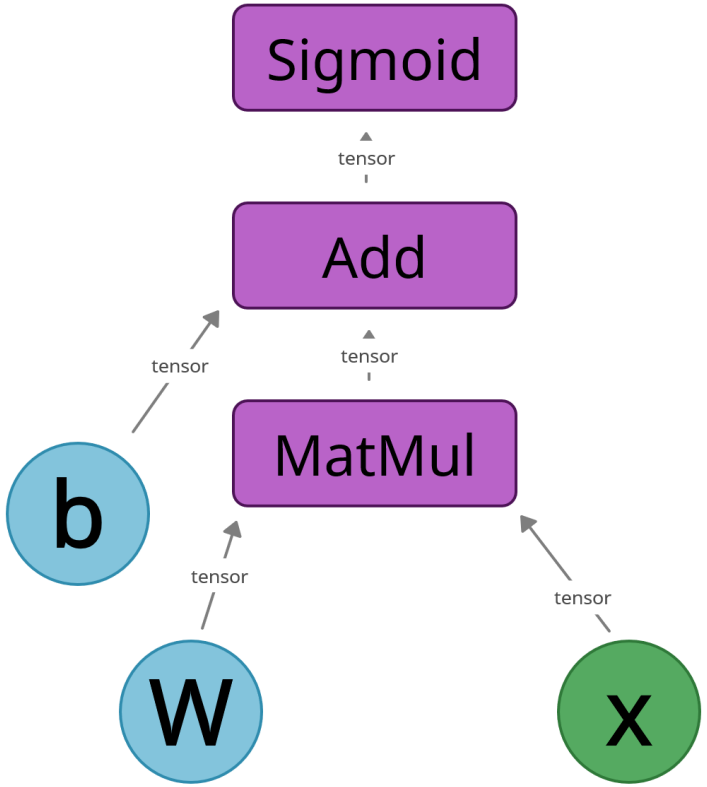
- Jako środowisko testowania ewentualnych rozwiązań, miała posłóżyć aplikacja AR Unity 3D rozwijana przez Pana D. Kobyra. Przeznaczeniem owej aplikacji, miało być wykrywanie objektów na obrazie, pochodzącym z kamer urządeń mobilnych.



Rysunek 1 : Szkic prototypu interfejsu do modeli sieci neuronowych jako API pomiedzy modelami a silnikiem gier

Graf obliczeniowy w TensorFlow

TensorFlow (TF) :
- otwarto źródłowa biblioteka programistyczna napisana przez Go-ogle Brain.
- wykorzystywana jest jest w uczeniu maszynowym i głębokich sieci neuronowych.



Rysunek 2 : Przykładowa warstwa sieci neuronowej w rozumieniu grafu TF

```python
import numpy as np
import tensorflow as tf

# Definicja grafu :

b = tf.Variable(tf.zeros((100,)))
W = tf.Variable(tf.random_uniform((784, 100), -1, 1))

x = tf.placeholder(tf.float23, (100, 784))
h = tf.nn.sigmoid(tf.matmul(x, W) + b)

# Uruchomienie grafu w sesji :

sess = tf.Session()
sess.run(tf.initialize_all_variables())
sess.run(h, {x: np.random.random(100, 784)})
```

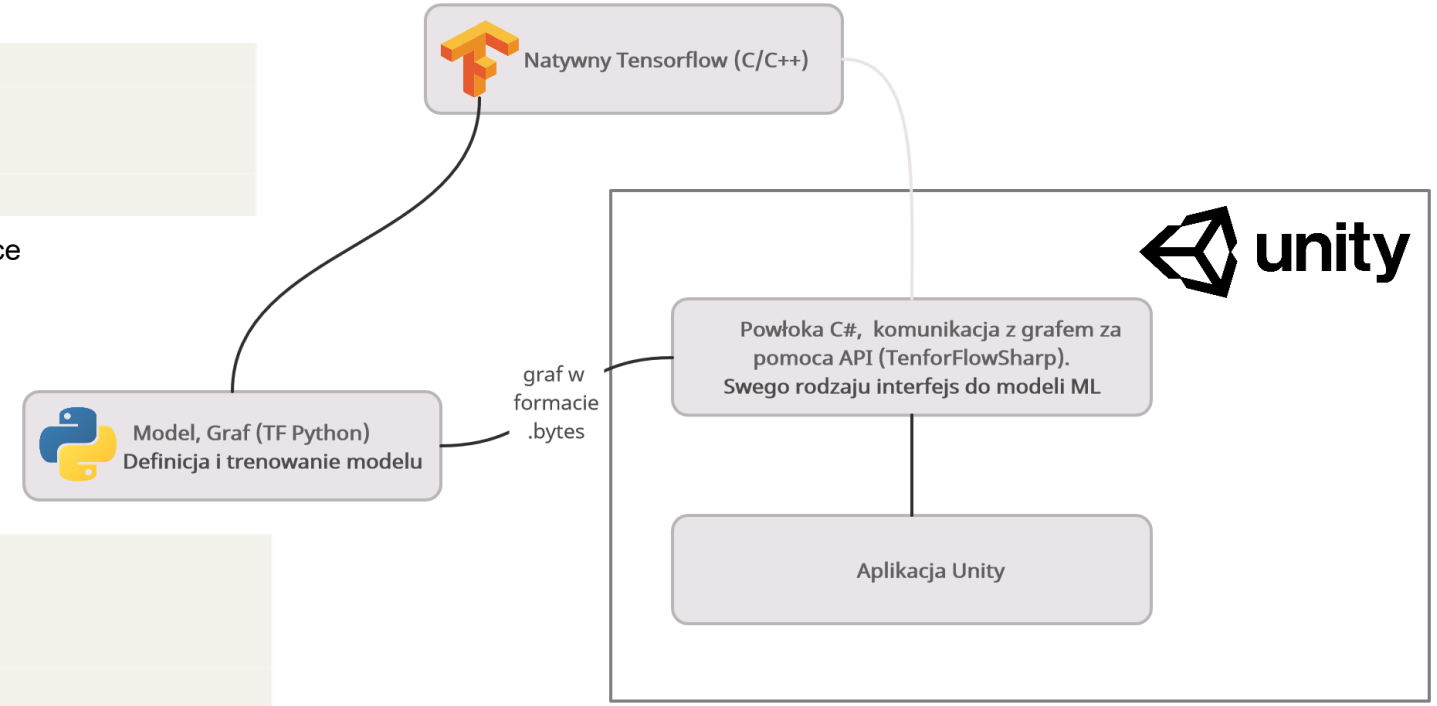
Listing 1 : Graf napisany w języku Python na podstawie Rysunku 2

TensorFlow w Unity (TensorFlowSharp)

TensorFlowSharp (TF#) :
- złącze językowe platformy .NET, do natywnej biblioteki TensorFlow.
- słowo ’Sharp’ pochodzi od C#, standardowego języka framework’u silnika Unity.

```
output_graph="/graph-model.pb"
with tf.gfile.GFile(output_graph, "wb") as f:
    f.write(output_graph_def.SerializeToString())
sess.close()
```

Listing 2 : Instrukcja zamrażania grafu zdefiniowanego w bibliotece TensorFlow (Python)



Rysunek 3 : Szkic prototypu struktury interfejsu do grafów TF w Unity

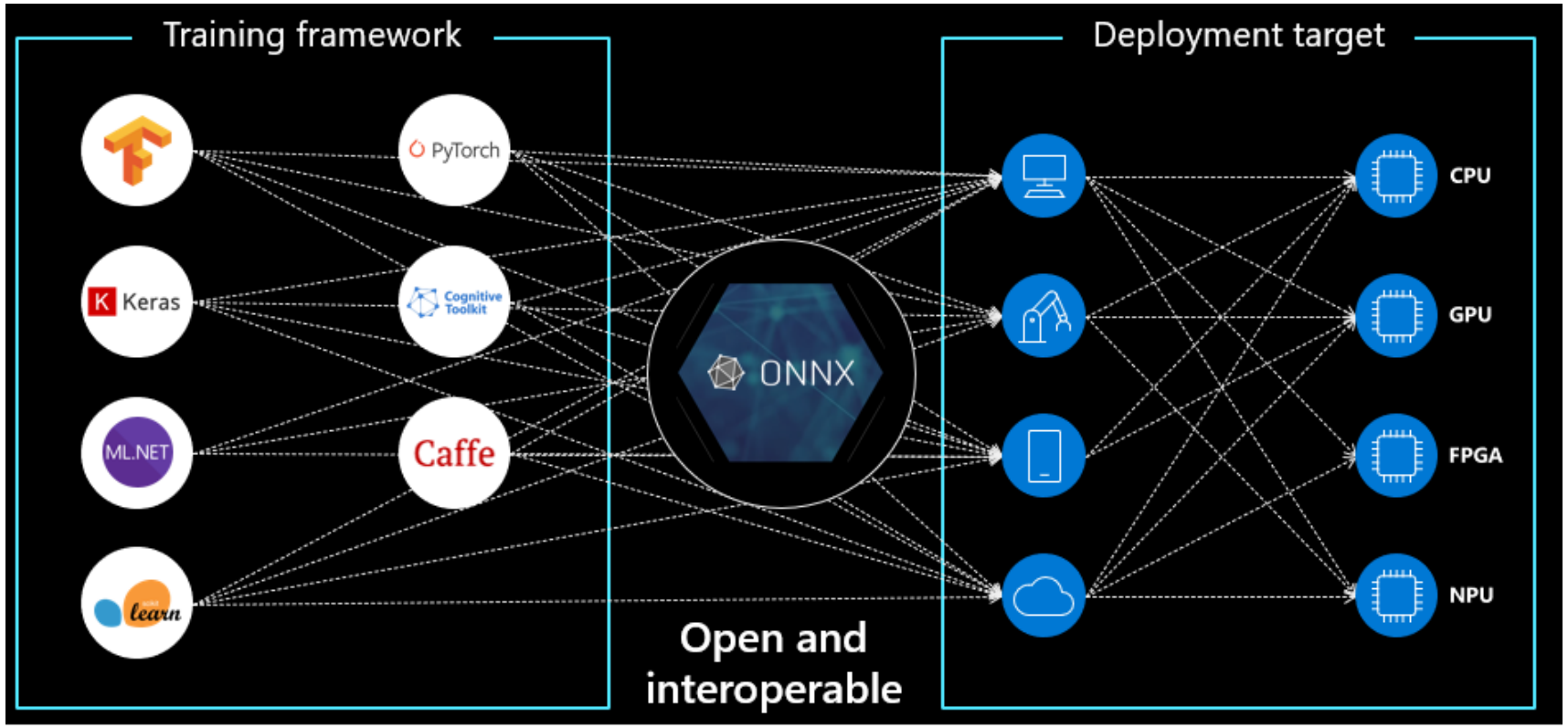
```
public TextAsset graphModel;

graph = new TFGraph (); // Tworzenie grafu
graph.Import (graphModel.bytes); // Importowanie grafu
session = new TFSession (graph); // Uruchamianie grafu w sesji
```

Listing 3 : Instrukcja odtwarzanie zaimportowanego grafu TF w formacie .bytes przy wykorzystaniu TF# (C# - Unity)

Open Neural Network Exchange

ONNX (Open Neural Network Exchange) :

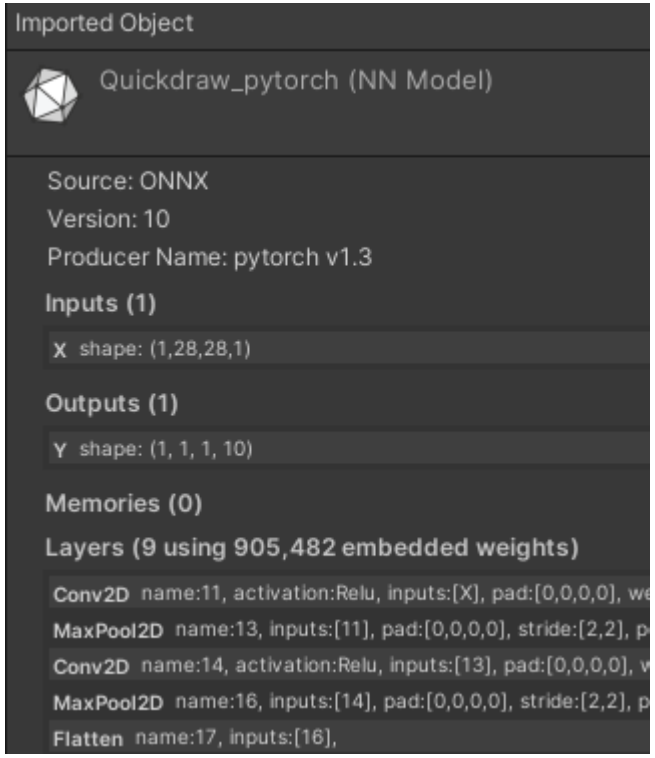
- otwarty format stworzony do reprezentowania modeli uczenia maszynowego.
- jedna z pierwszych technologii umożliwiających wymianę modeli uczenia maszynowego, pomiędzy najpowszechniej używanymi frameworkami tej dziedziny.



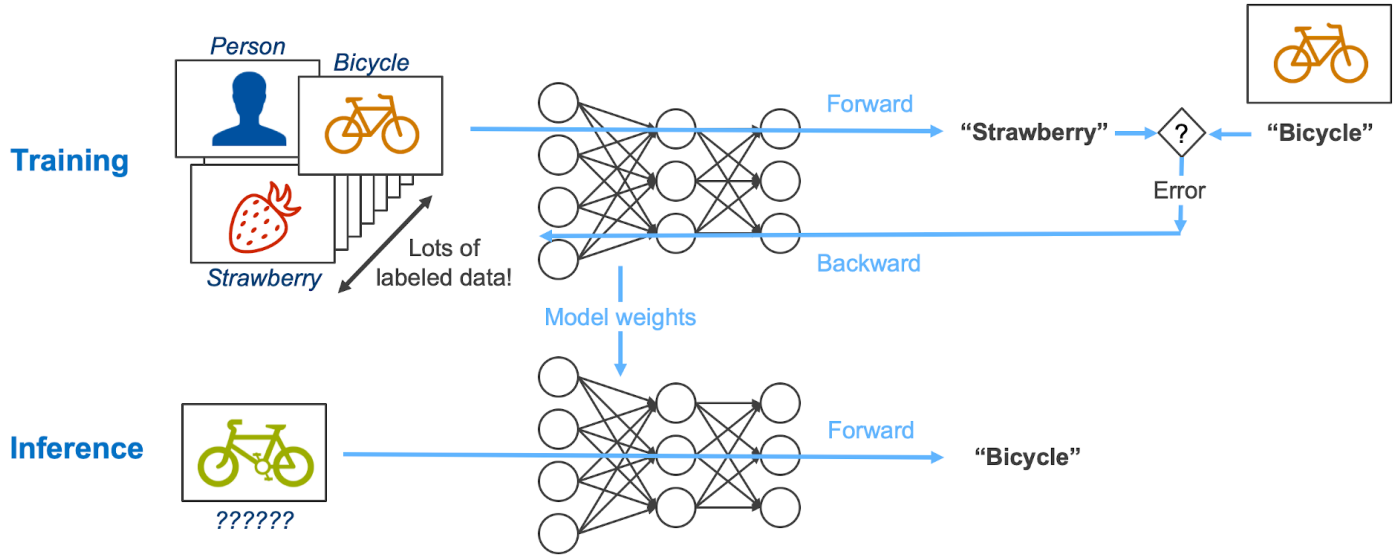
Rysunek 4 : ONNX jako punkt zbieżności framework'ów uczenia maszynowego

Biblioteka inferencyjna – Unity Barracuda

Unity Barracuda :
- biblioteka inferencyjna sieci neuronowych
- składa się z silnika inferencyjnego dla modeli uczenia maszynowego, dostępnych w formacie ONNX.
- odpowiedzialna za wdrażanie modelu uczenia maszynowego oraz ewentualne monitorowanie jego wydajności.



Rysunek 5 : Przykład ’asset’u’ typu NNModel w Unity – modelu o formacie .onnx zdefiniowanego w Pytorch



Rysunek 6 : Proces trenowania a inferencja modelu sieci neuronowych

Interfejs dla modeli YOLOv2-tiny w aplikacji AR, przy użyciu Barracuda

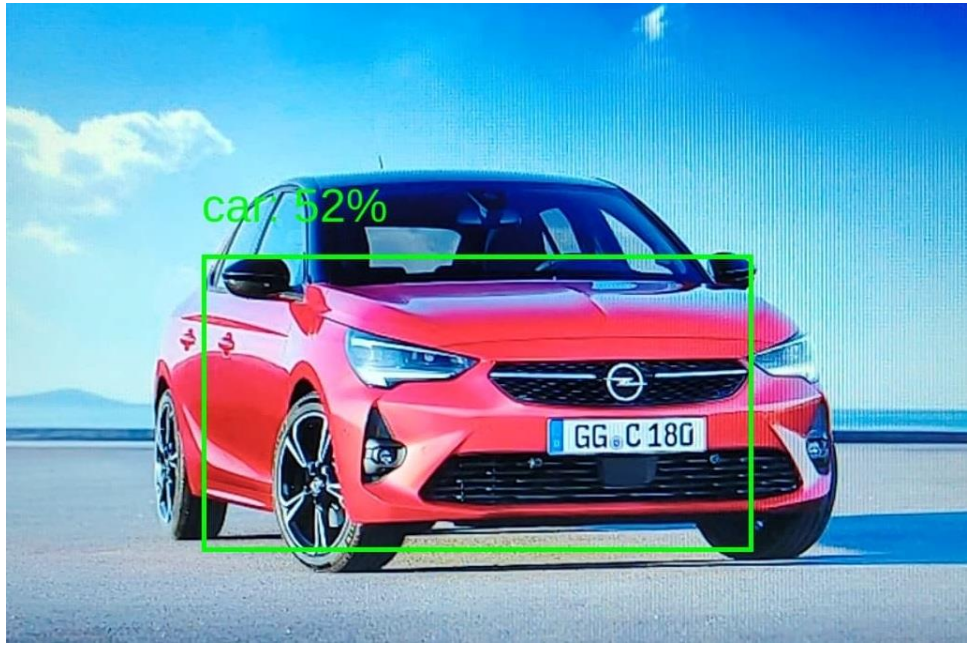
YOLO (You Only Look Once) :
- algorytm do wykrywania obiektów w czasie rzeczywistym, jest obecnie jednym z najefektywniejszych algorytmów tego typu.

Model	**Liczba klas**	**Wejście modelu**	**Wyjscie modelu**
yolov2-tiny-food-freeze.onnx	100	yolo2-tiny/net: shape(0, 416, 416, 3)	yolo2-tiny/ convolutional9/BisadAdd: shape(-1, -1, -1, -1)
yolov2-tiny-voc.onnx	20	input_1: shape(-1, 416, 416, 3)	cov2d_9/BiasAdd: shape(-1, -1, -1, -1)

Tabela 1: Modele YOLO uzyte w aplikacji AR, omowionej w pracy inzynierskiej

Wynik działania modelów YOLOv2-tiny o formacie ONNX w aplikacji AR Unity

Rysunek 6 : Wynik działania pierwszego modelu w analizowanej aplikacji AR



Rysunek 7 : Wynik działania drugiego modelu w analizowanej aplikacji AR

Podsumowanie oraz osiągnięte cele

- Przegląd oraz analiza możliwości technologii pozwalających na wykorzystanie sieci neuronowych w silniku Unity.

- Dostarczenie interfejsu sieci neuronowych dla aplikacji AR Unity 3D opisanej w pracy Pana D. Kobyra, pt. „Aplikacja mobilna AR z wykorzystaniem natywnych modeli sieci neuronowych w środowisku Unity".

- Przetestowanie wydajności inferencji z modelami przeznaczonymi do wykrywania obiektów, przy wykorzystaniu technologii TensorFlowSharp oraz Barracuda.

Bibliografia

[1] Wikipedia. TensorFlow — Wikipedia,.https://pl.wikipedia.org/wiki/TensorFlow, 2021

[2] migueldeicaza. Tensorflowsharp.https://github.com/migueldeicaza/TensorFlowSharp, 2019.

[3] Unity-Technologies.barracuda-release.https://github.com/Unity-Technologies/barracuda-release, 2019

[4] Unity-Technologies. Barracuda.https://docs.unity3d.com/Packages/com.unity.barracuda@1.0/manual/index.html, 2020.

[5] Facebook/Microsoft. Onnx.https://onnx.ai/, 2020.