Cpt Ugur UYSAL – MILITARY APPLICATIONS FOR MACHINE LEARNING

IDS 6908 – Independent Study – summer 2019 – presentation for MSG-175

ArmA3 in game screenshot

UCF

Purpose

The purpose of this presentation is to summarize to what extent selected **military capabilities** can be **improved** by using **Machine Learning** techniques such as

1. Reinforcement Learning and
2. Image Classification.

UCF

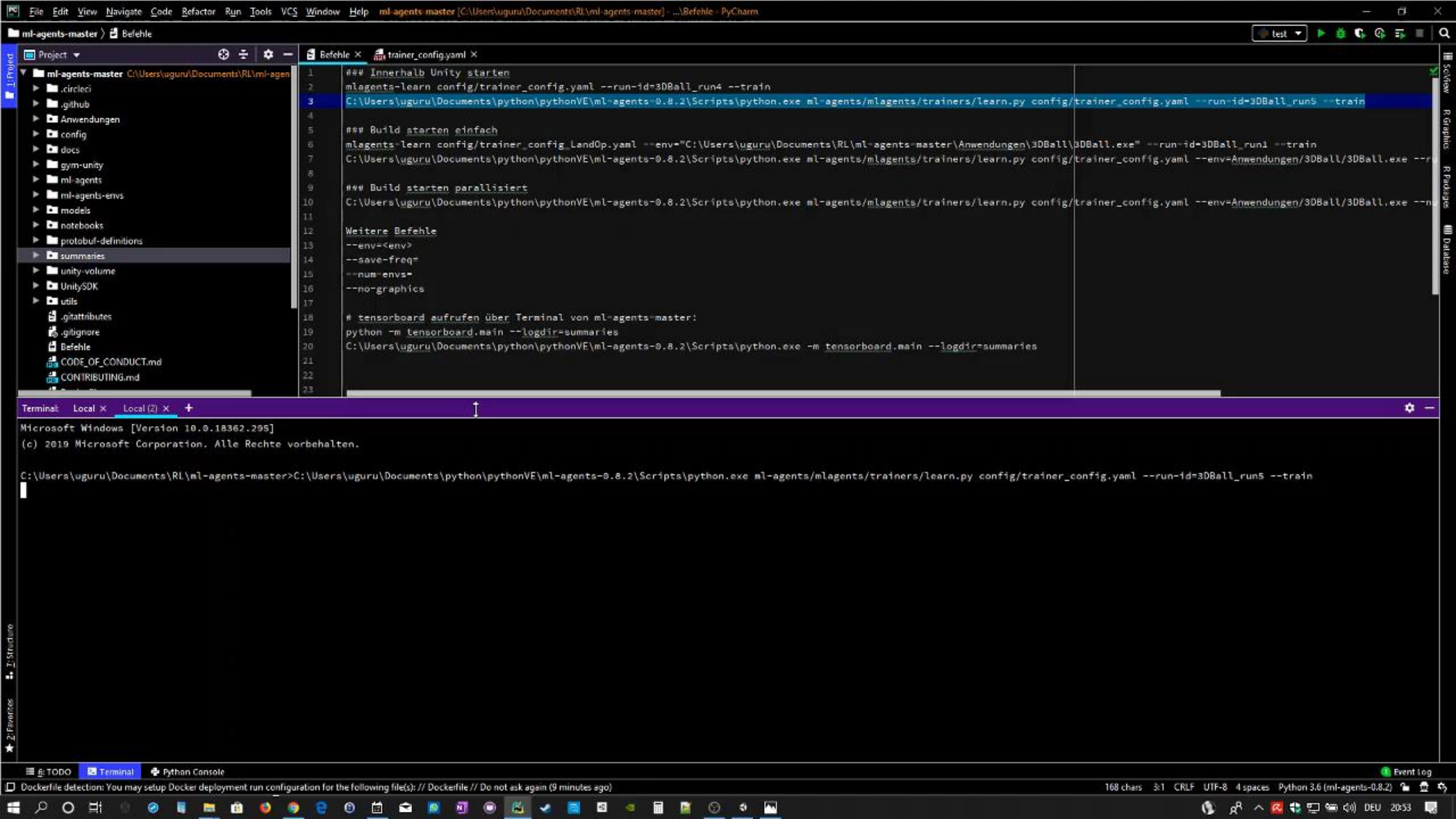
Introduction
Reinforcement Learning (RL)

RL studies algorithms for **training agents** how to take **actions** in an environment to maximize some cumulative **reward**.

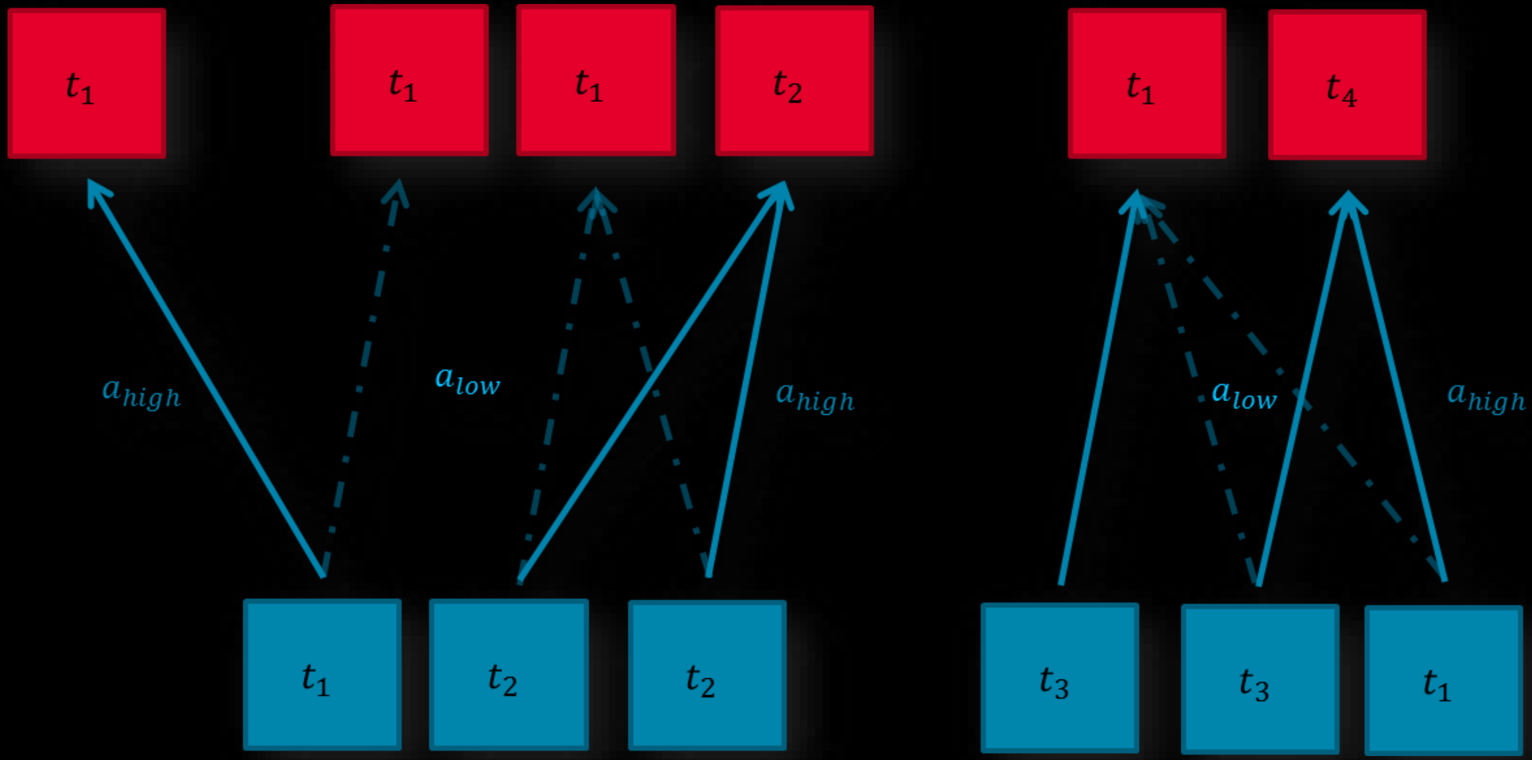
Introduction
Unity ML-Agents Toolkit

"By taking advantage of **Unity** as a simulation platform, the toolkit enables the **development of learning environments** which are rich in sensory and physical complexity, provide compelling cognitive challenges, and support dynamic multi-agent interaction." **[1]**

UCF


File Edit View Navigate Code Refactor Run Tools VCS Window Help ml-agents-master [C:\Users\uguru\Documents\RL\ml-agents-master] - ...\Befehle - PyCharm
ml-agents-master 〉 Befehle
Project
ml-agents-master C:\Users\uguru\Documents\RL\ml-agen
.circleci
.github
Anwendungen
config
docs
gym-unity
ml-agents
ml-agents-envs
models
notebooks
protobuf-definitions
summaries
unity-volume
UnitySDK
utils
.gitattributes
.gitignore
Befehle
CODE_OF_CONDUCT.md
CONTRIBUTING.md
Befehle
trainer_config.yaml
Innerhalb Unity starten
mlagents-learn config/trainer_config.yaml --run-id=3DBall_run4 --train
C:\Users\uguru\Documents\python\pythonVE\ml-agents-0.8.2\Scripts\python.exe ml-agents/mlagents/trainers/learn.py config/trainer_config.yaml --run-id=3DBall_run5 --train
Build starten einfach
mlagents-learn config/trainer_config_LandOp.yaml --env="C:\Users\uguru\Documents\RL\ml-agents-master\Anwendungen\3DBall\3DBall.exe" --run-id=3DBall_run1 --train
C:\Users\uguru\Documents\python\pythonVE\ml-agents-0.8.2\Scripts\python.exe ml-agents/mlagents/trainers/learn.py config/trainer_config.yaml --env=Anwendungen/3DBall/3DBall.exe --r
Build starten parallisiert
C:\Users\uguru\Documents\python\pythonVE\ml-agents-0.8.2\Scripts\python.exe ml-agents/mlagents/trainers/learn.py config/trainer_config.yaml --env=Anwendungen/3DBall/3DBall.exe --n
Weitere Befehle
--env=<env>
--save-freq=
--num-envs=
--no-graphics
tensorboard aufrufen über Terminal von ml-agents-master:
python -m tensorboard.main --logdir=summaries
C:\Users\uguru\Documents\python\pythonVE\ml-agents-0.8.2\Scripts\python.exe -m tensorboard.main --logdir=summaries
Terminal: Local Local (2)
Microsoft Windows [Version 10.0.18362.295]
(c) 2019 Microsoft Corporation. Alle Rechte vorbehalten.
C:\Users\uguru\Documents\RL\ml-agents-master>C:\Users\uguru\Documents\python\pythonVE\ml-agents-0.8.2\Scripts\python.exe ml-agents/mlagents/trainers/learn.py config/trainer_config.yaml --run-id=3DBall_run5 --train
6: TODO Terminal Python Console Event Log
Dockerfile detection: You may setup Docker deployment run configuration for the following file(s): // Dockerfile // Do not ask again (9 minutes ago)
168 chars 3:1 CRLF UTF-8 4 spaces Python 3.6 (ml-agents-0.8.2)
DEU 20:53

Intelligent fire distribution



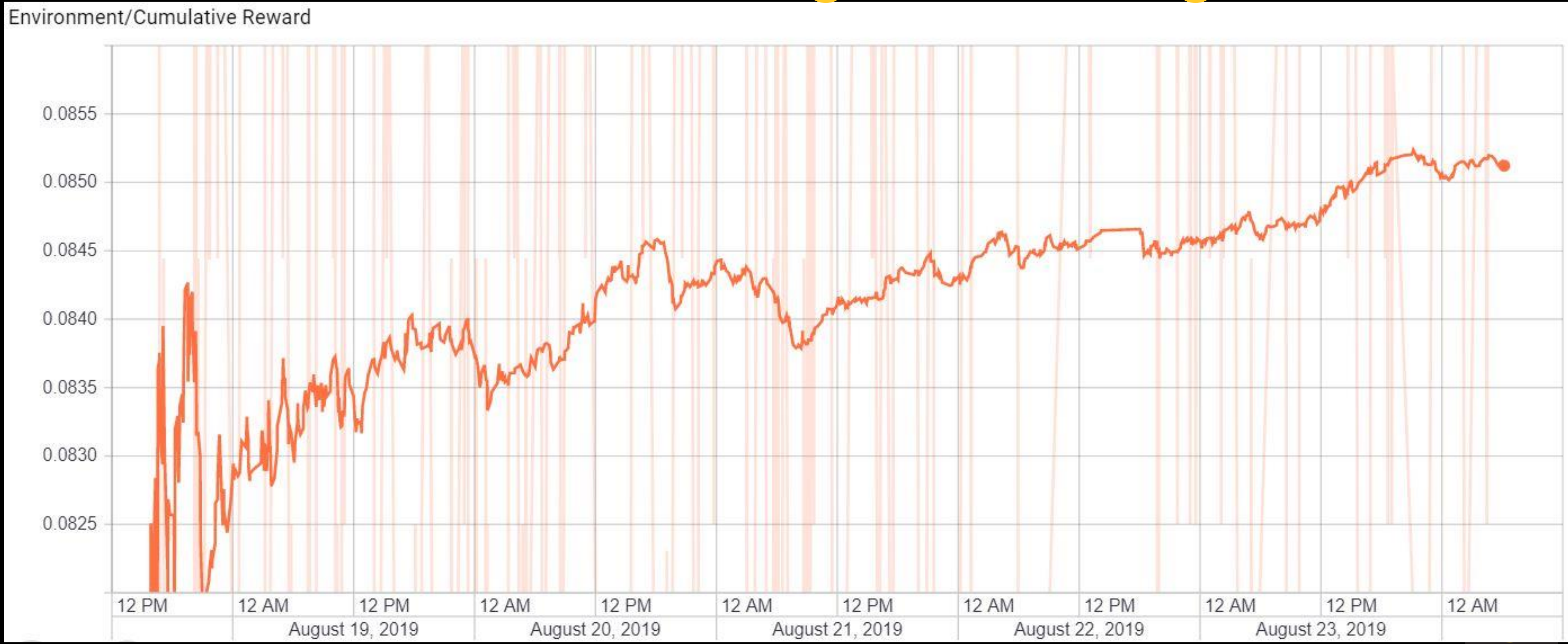
t_i: combat vehicle type

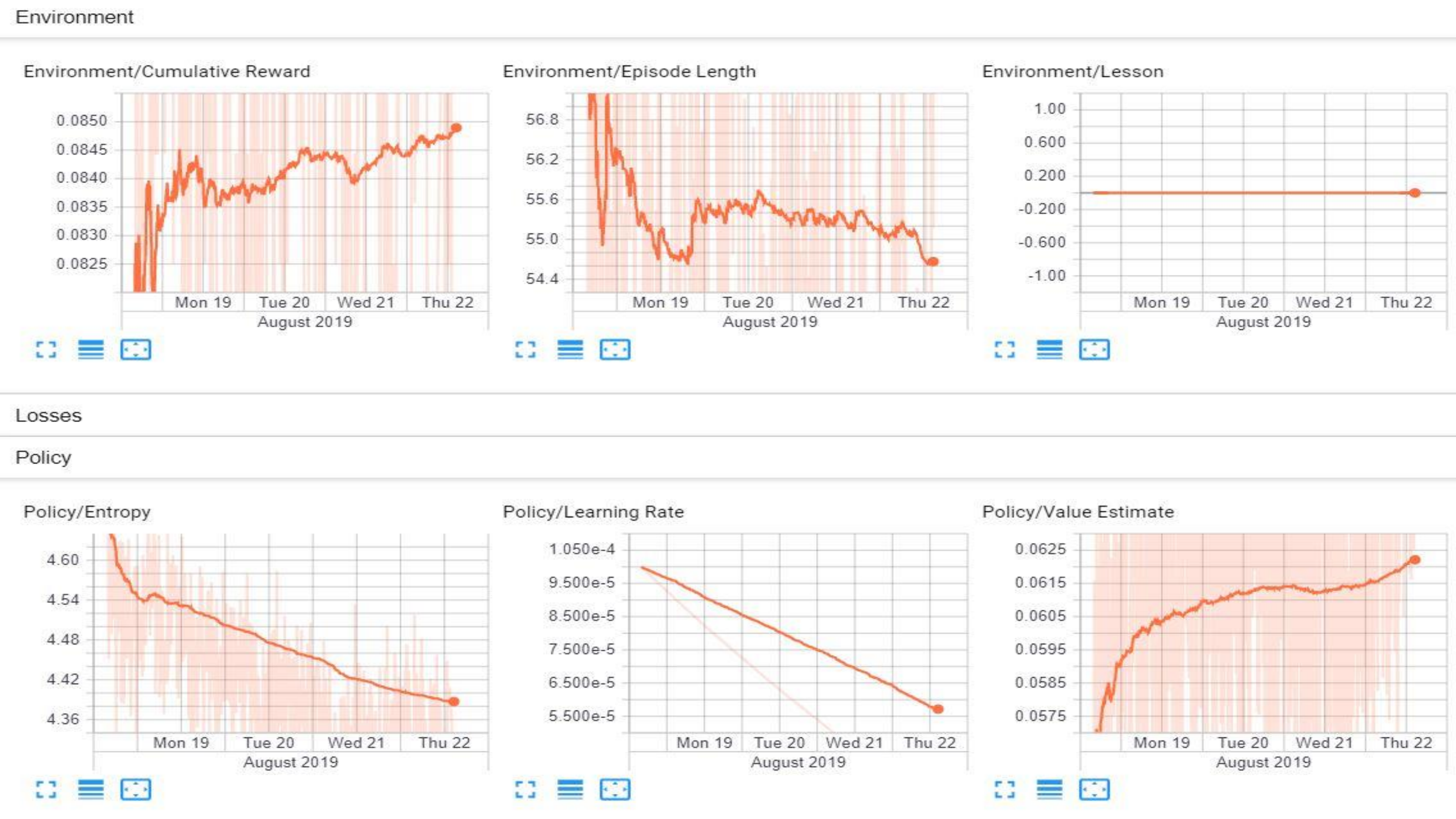
a_p: action with priority

Intelligent fire distribution
Unity simulation model


Unity 2017.4.30f1 Personal (64bit) - NoMovement.unity - Landop-0.8.2 - PC, Mac & Linux Standalone <DX11>
File Edit Assets GameObject Component ExtendedReality Mobile Input Window Help
Center Global
Collab Account Layers Layout
Game
Display 1 Free Aspect Scale 1x
Maximize On Play Mute Audio Stats Gizmos
MBT_RED4
MBT_RED6
IFV_Medium_RED7
IFV_Heavy_RED2
MBT_RED9
IFV_Heavy_RED5
MBT_BLUE1
MBT_BLUE3
IFV_BLUE6
IFV_BLUE9
IFV_BLUE7
IFV_BLUE10
409,96 Episode done: All Blue Destroyed | Total Reward: -1
DEU 20:32

Intelligent fire distribution
Reinforcement learning with ml-agents





Environment
Environment/Cumulative Reward
0.0850
0.0845
0.0840
0.0835
0.0830
0.0825
Mon 19
Tue 20
Wed 21
Thu 22
August 2019
Environment/Episode Length
56.8
56.2
55.6
55.0
54.4
Mon 19
Tue 20
Wed 21
Thu 22
August 2019
Environment/Lesson
1.00
0.600
0.200
-0.200
-0.600
-1.00
Mon 19
Tue 20
Wed 21
Thu 22
August 2019
Losses
Policy
Policy/Entropy
4.60
4.54
4.48
4.42
4.36
Mon 19
Tue 20
Wed 21
Thu 22
August 2019
Policy/Learning Rate
1.050e-4
9.500e-5
8.500e-5
7.500e-5
6.500e-5
5.500e-5
Mon 19
Tue 20
Wed 21
Thu 22
August 2019
Policy/Value Estimate
0.0625
0.0615
0.0605
0.0595
0.0585
0.0575
Mon 19
Tue 20
Wed 21
Thu 22
August 2019

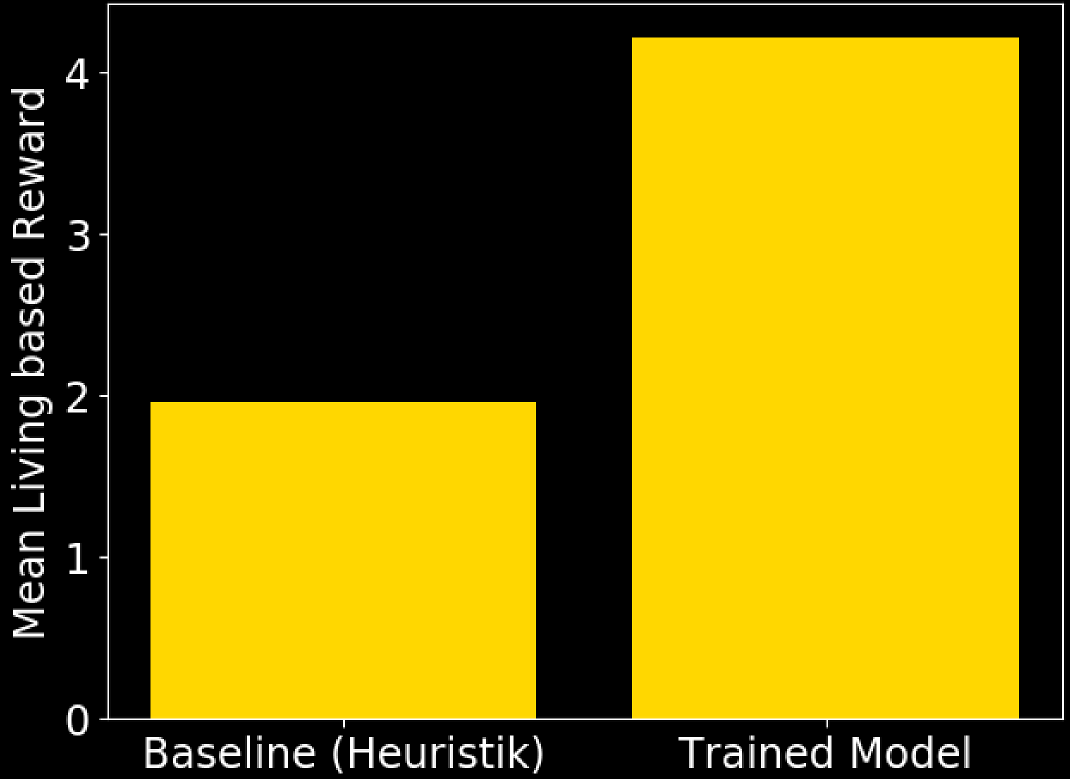
Intelligent fire distribution
Mean Reward after 1000 test episodes

Episode Reward =

Active Blue tanks

-

Active Red tanks



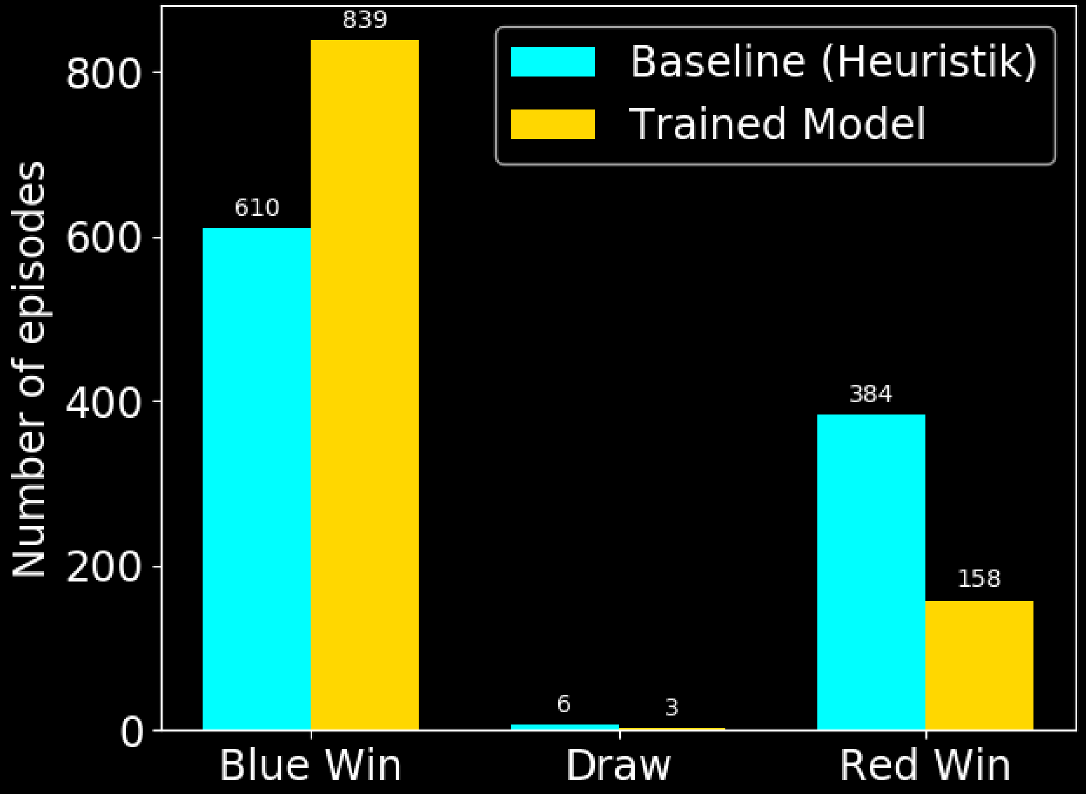
Intelligent fire distribution

Episodes won after 1000 test episodes

Blue Win:
if 0 active Red tanks

Red Win:
if 0 active Blue tanks

Otherwise Draw



Introduction
Image Classification (IC)

IC was one of the earliest successes of deep learning, which enabled computers to classify objects in images.

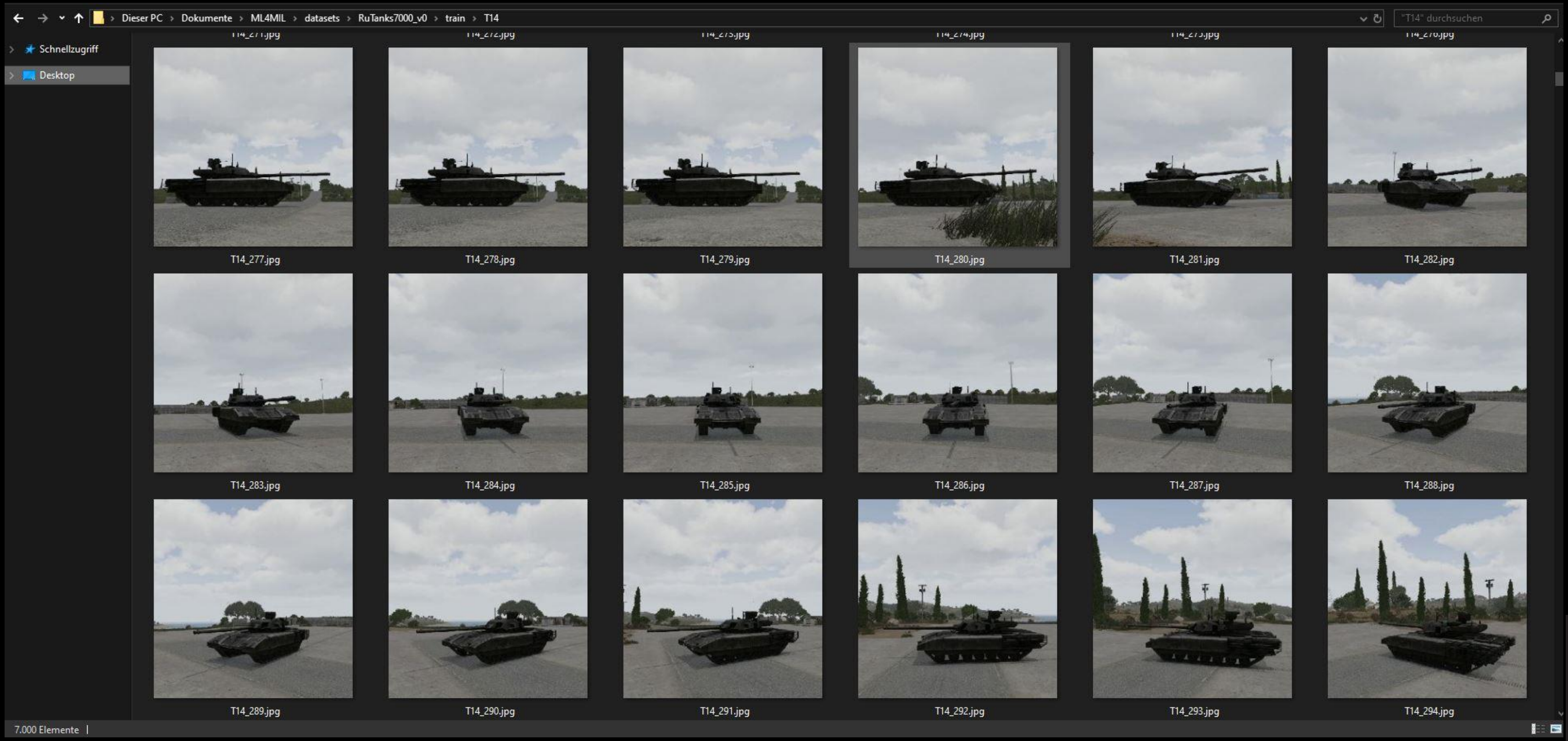
Image classification of military vehicles

Prediction:
79.28% T14
20.19% T90
0.41% Buk-M1-2
0.05% Background
0.05% BMP2
0.02% ZSU23
0.00% Civilian Car

[3]

Image classification of military vehicles

Creating a synthetic image dataset with ArmA3



UCF

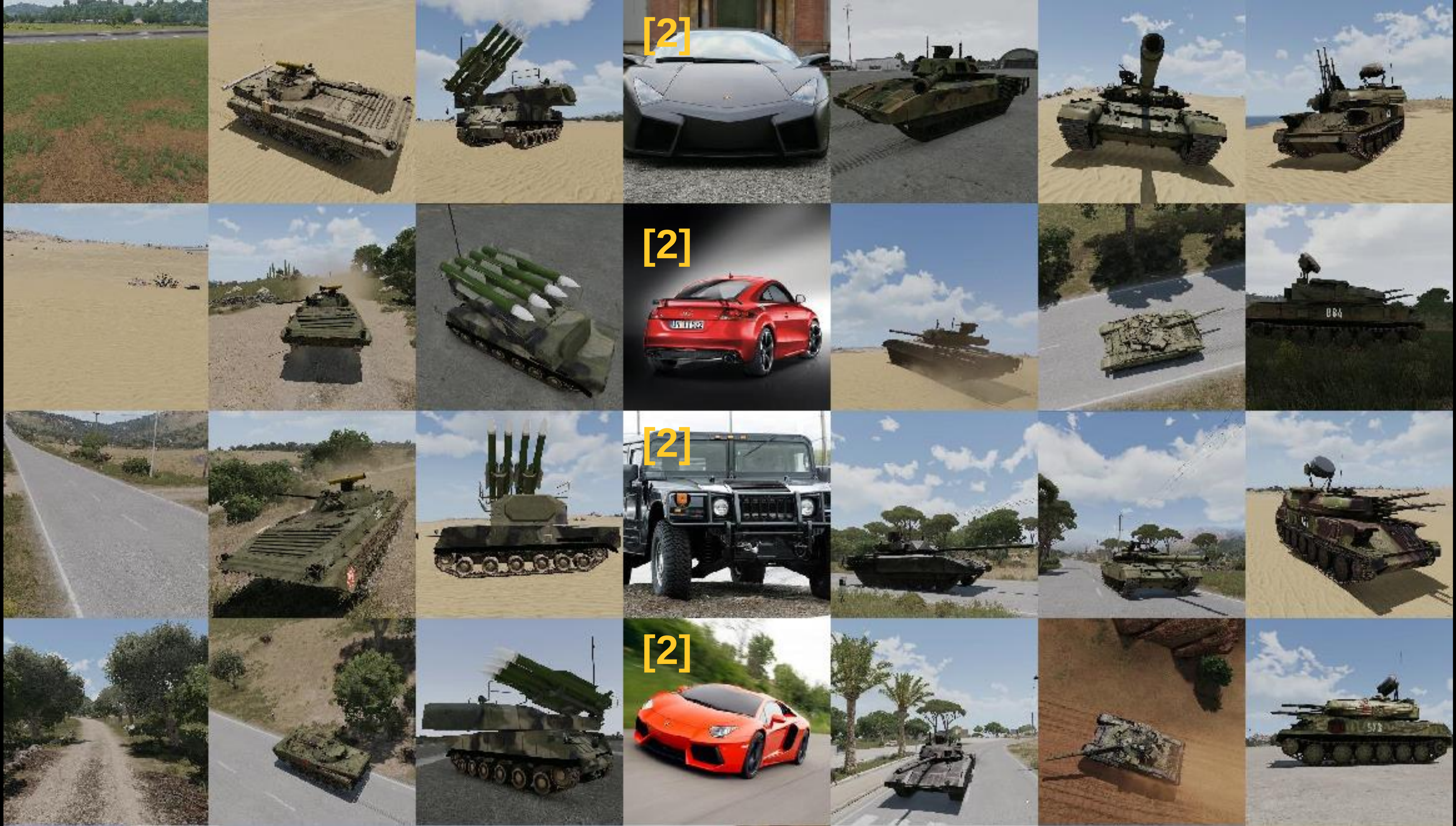

[2]
[2]
[2]
[2]

Image classification of military vehicles
Deep learning using transfer learning

The Python packages Keras 2.2.4 and Tensorflow-gpu 1.14.0 were used to fine-tune [4] the pretrained network NASNet [5]. **Test accuracy of 91%** (115 test images) was achieved after **519 minutes training** on a Nvidia Geforce RTX 2060.

UCF

Test Results of the trained Model:

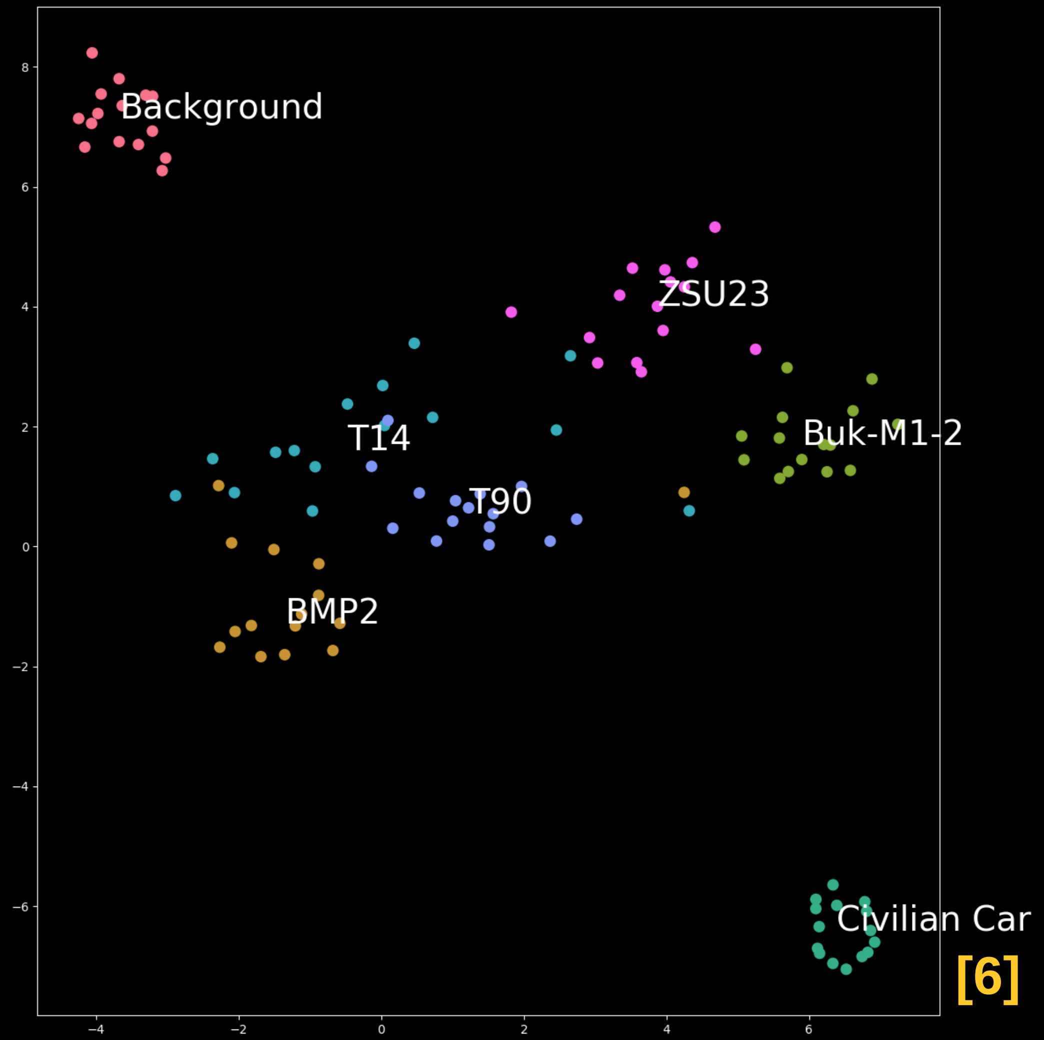
Test loss: 0.3384579775078911 , test accuracy: 0.9142857142

Confusion Matrix

```
[[15  0  0  0  0  0  0]
 [ 0 14  0  0  1  0  0]
 [ 0  0 15  0  0  0  0]
 [ 0  0  0 15  0  0  0]
 [ 0  2  0  0 10  2  1]
 [ 0  0  0  0  2 13  0]
 [ 0  0  1  0  0  0 14]]
```

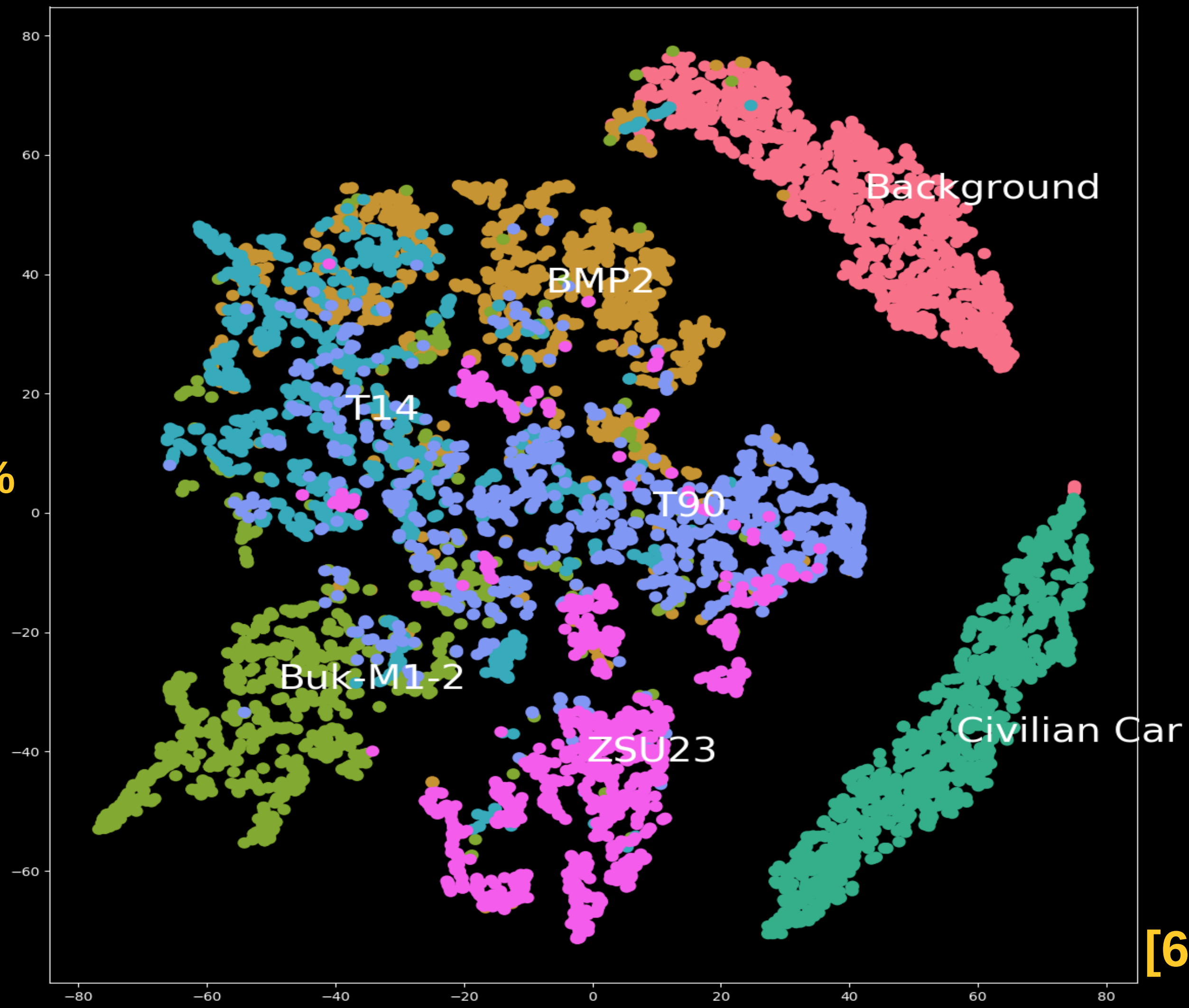
Classification Report

	precision	recall	f1-score	support
Background	1.00	1.00	1.00	15
BMP2	0.88	0.93	0.90	15
Buk-M1-2	0.94	1.00	0.97	15
Civilian Car	1.00	1.00	1.00	15
T14	0.77	0.67	0.71	15
T90	0.87	0.87	0.87	15
ZSU23	0.93	0.93	0.93	15
accuracy			0.91	105
macro avg	0.91	0.91	0.91	105
weighted avg	0.91	0.91	0.91	105



[6]

Test accuracy of 71%
(7000 test images)



[6]

Conclusions

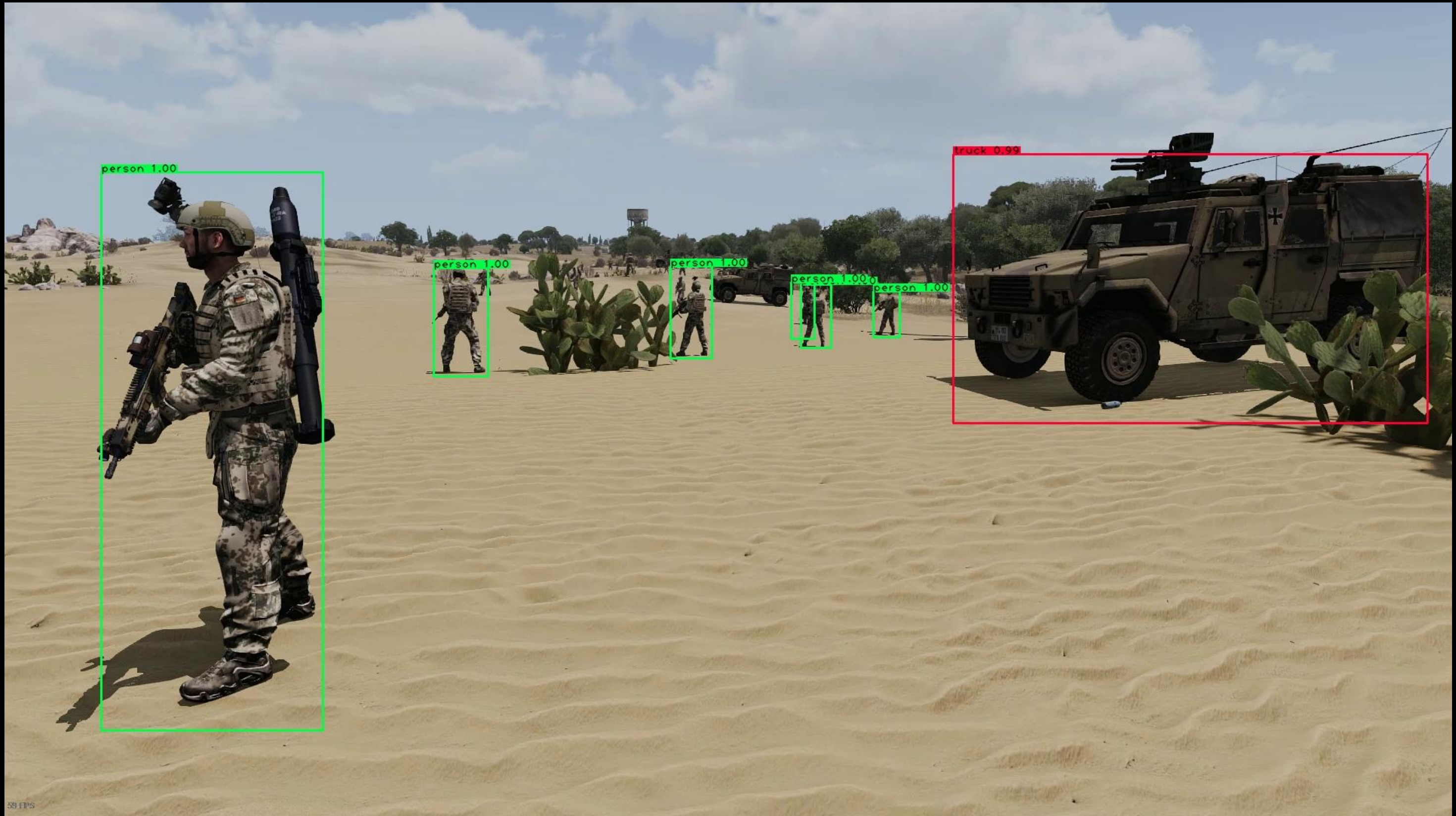
- The use of RL-methods to train agents for decision support tasks to control combat vehicles is advantageous within a simulation environment and promising for real-world military applications.

- Synthetic data can be used to train DNN to identify military vehicles in real-world images.

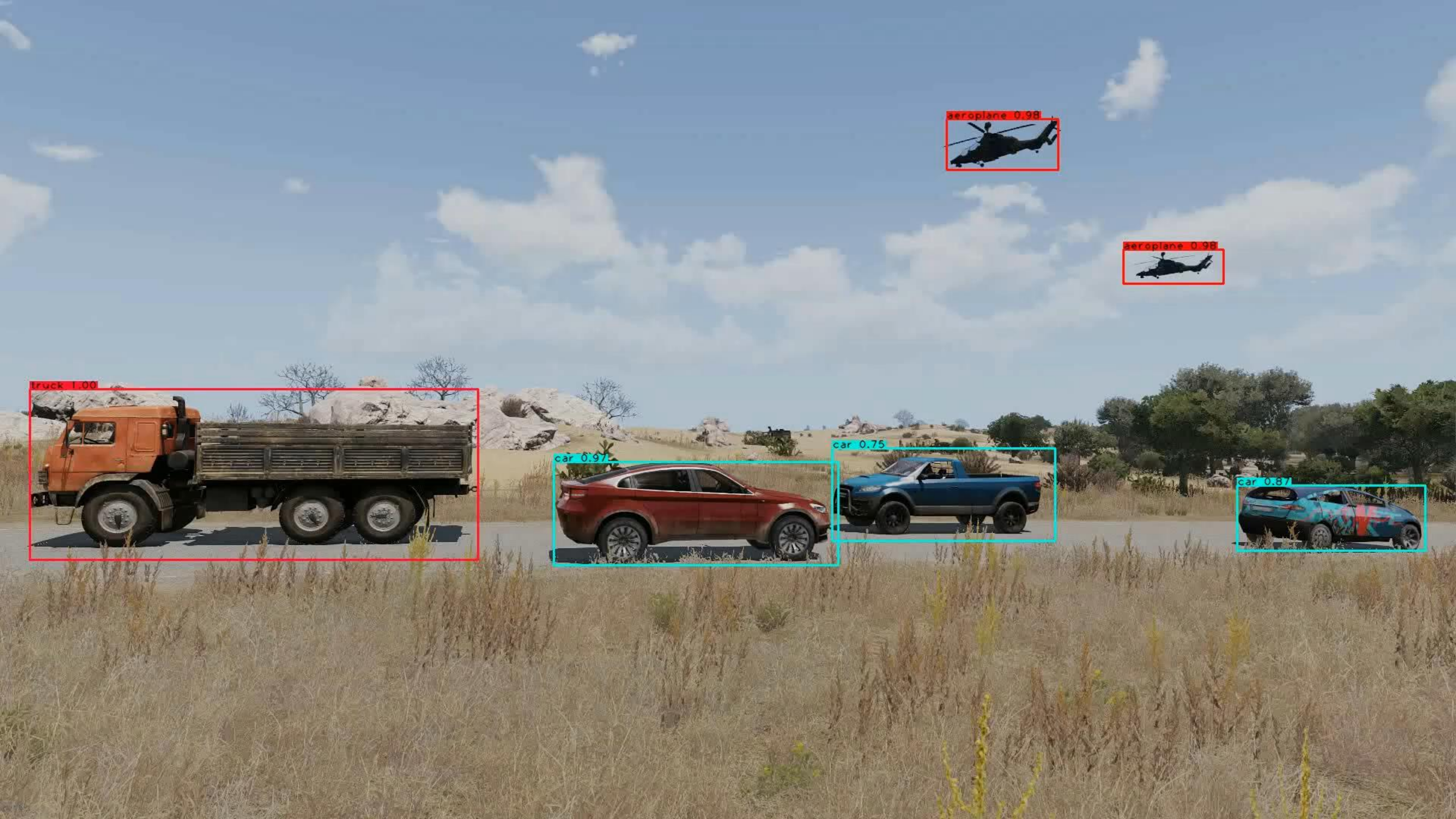
- We are confident that ML will decisively shape the future battlefield.

UCF

Future Work
Object Localization/Detection



[7]


aeroplane 0.98
aeroplane 0.98
truck 1.00
car 0.97
car 0.75
car 0.87

Acknowledgements

We would like to thank Lieutenant-Colonel Doll, who is pushing the topic of Artificial Intelligence in the German Army and has enabled us to participate in this research.

UCF

Visit the GitHub repository https://github.com/UgurUysal86/MLS4MIL for further information

Questions?

UCF

References

[1] A. Juliani, V. Berges, E. Vckay, Y. Gao, H. Henry, M. Mattar, D. Lange. arXiv:1809.02627. *Unity: A General Platform for Intelligent Agents* (2018)

[2] J. Krause, M. Stark, J. Deng, L. Fei-Fei. *4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013* (2013)

[3] B. Mashina. Wikimedia Commons. Cropped to 331x331 pixels. CC BY-SA 4.0 (2018)

[4] F. Chollet. Manning. *Deep Learning with Python*. Section 5.3 (2018)

[5] B. Zoph, et al. *IEEE conference on computer vision and pattern recognition*, pp. 8697-8710 (2018)

[6] M. Pathak. Datacamp. *Introduction to t-SNE* (2018)

[7] J. Redmon, A. Farhadi. arXiv:1804.02767. YOLO-v3: *An Incremental Improvement* (2018)

UCF