Neuroevolution of Hybrid Neural Networks in a Robotic Agent (NHNN-RA)

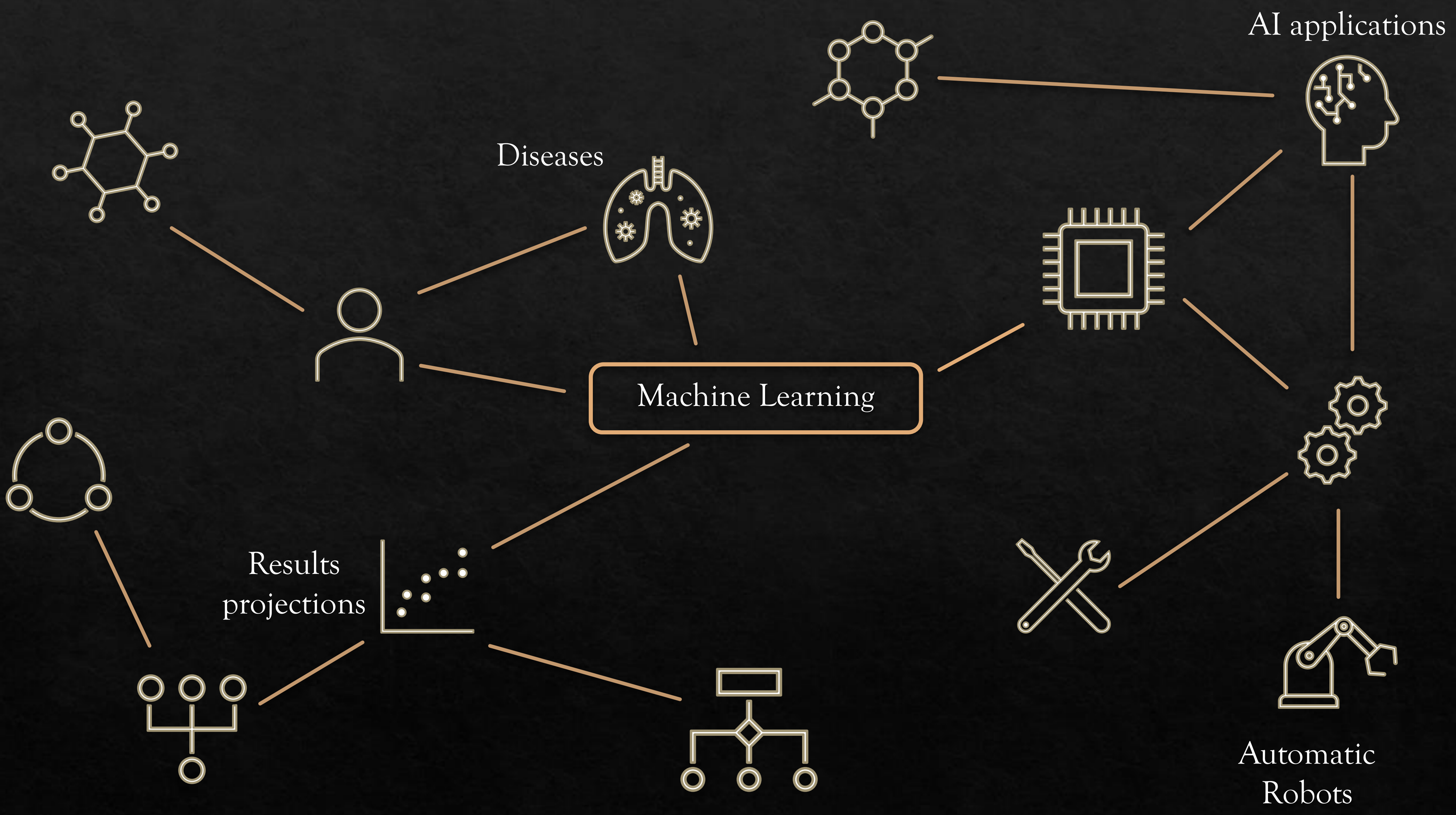
Autor: Carlos Vasquez

Supervisor: Ph.D. Mariko Nakano

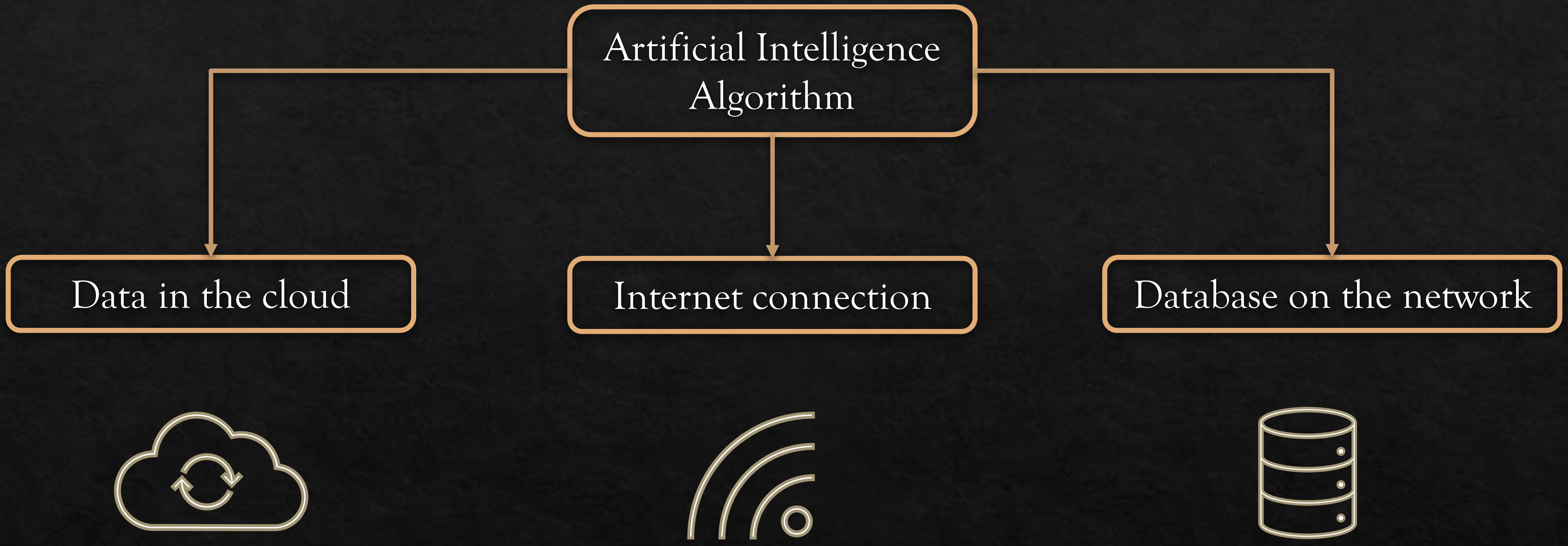
*Engineering in Microelectronics, National Polytechnic Institute,
Mexico City, Mexico

March 2022

Introduction



Introduction



Problems in robotics

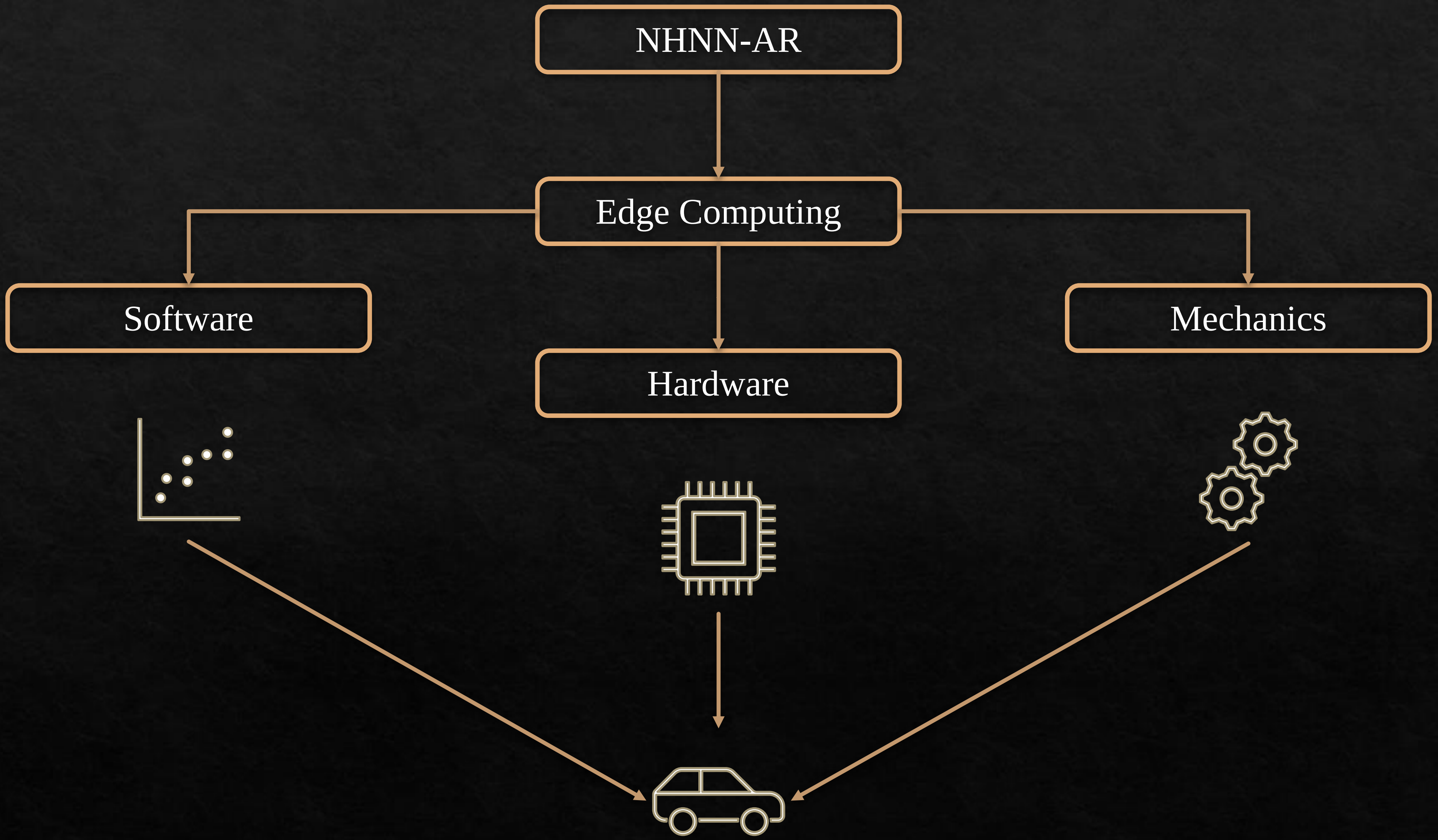
Environments

- Avoid latency
- No internet access
- No database access

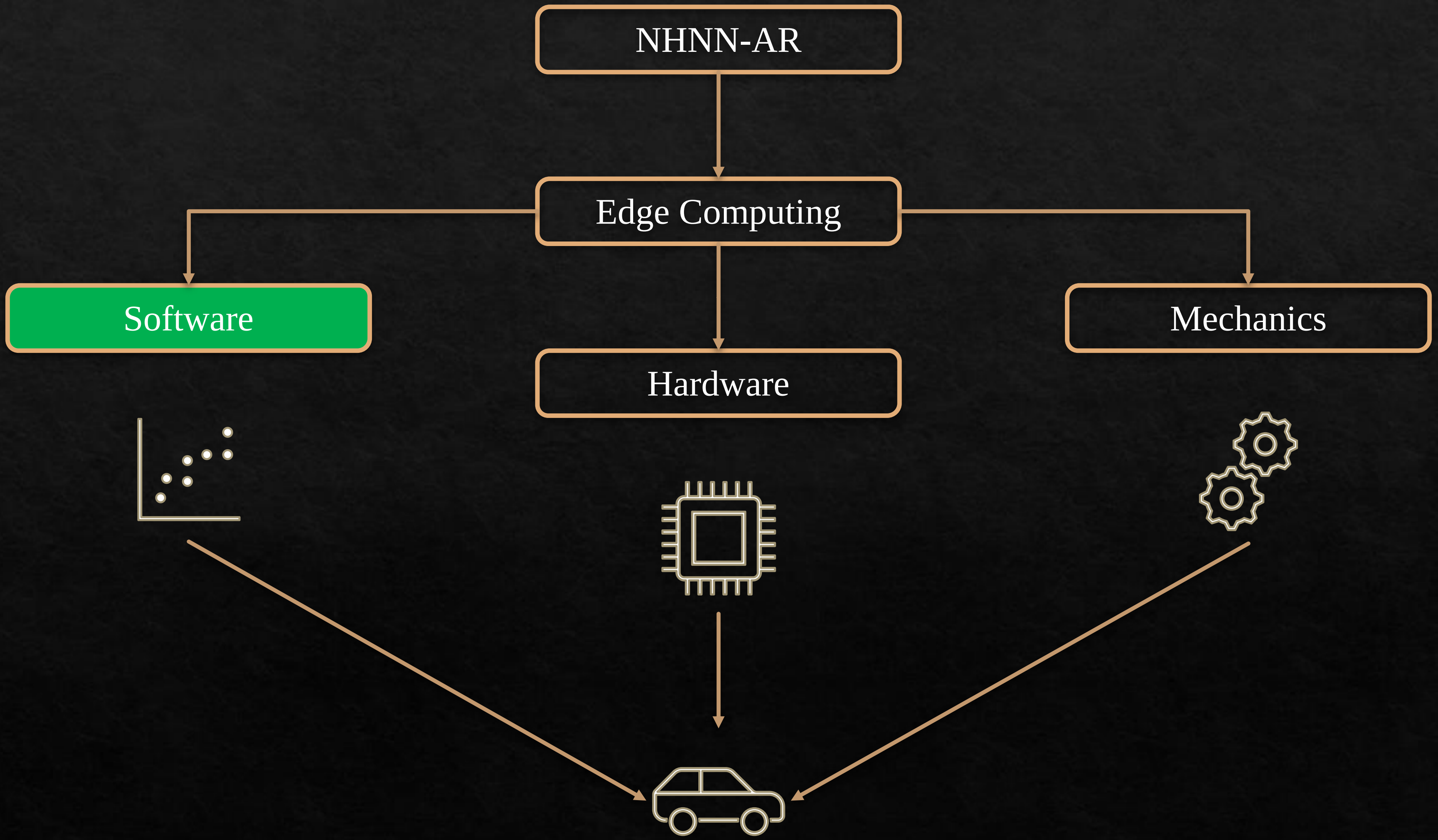
Instant actions

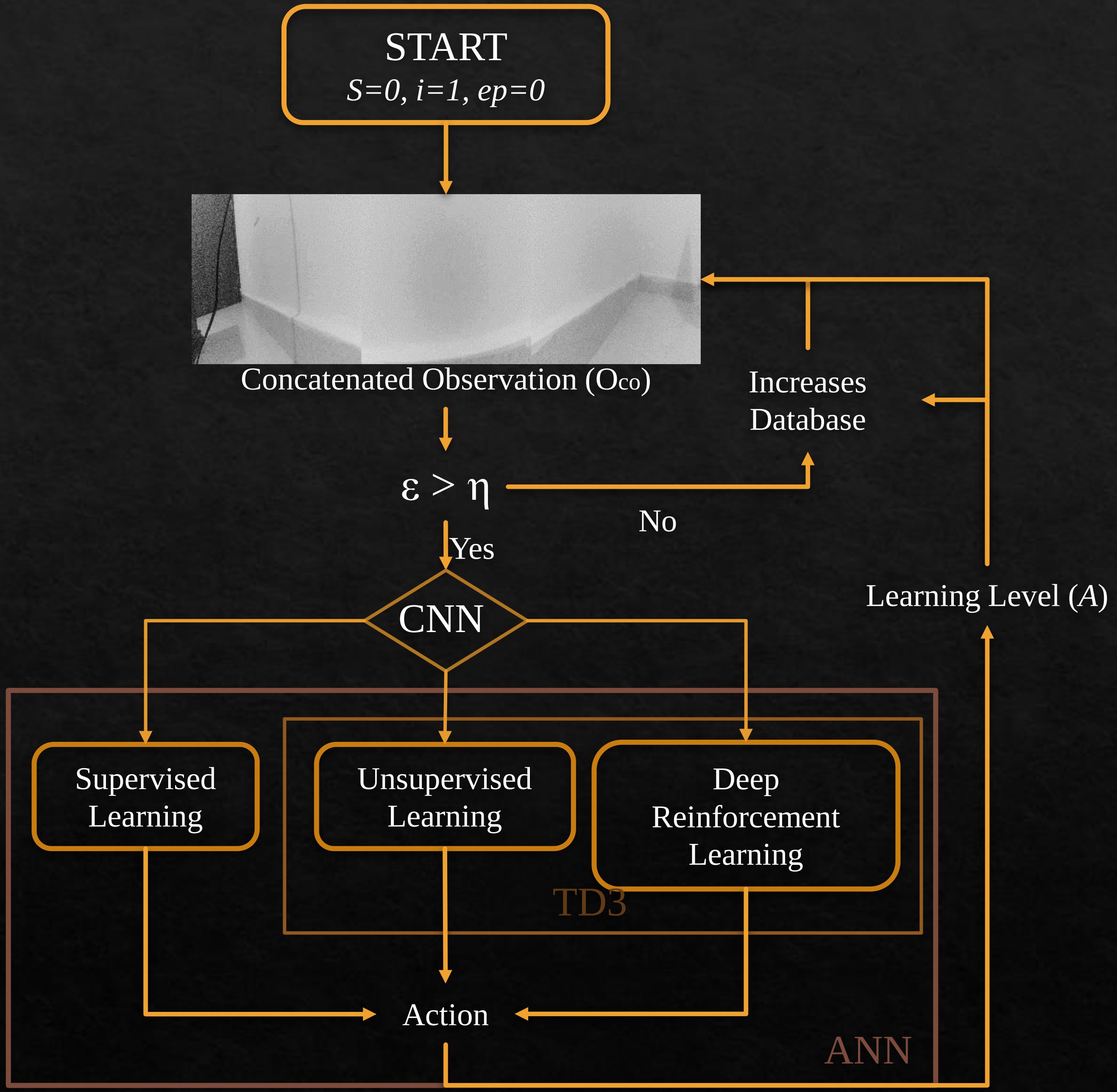
Inhospitable places

Algorithm performance

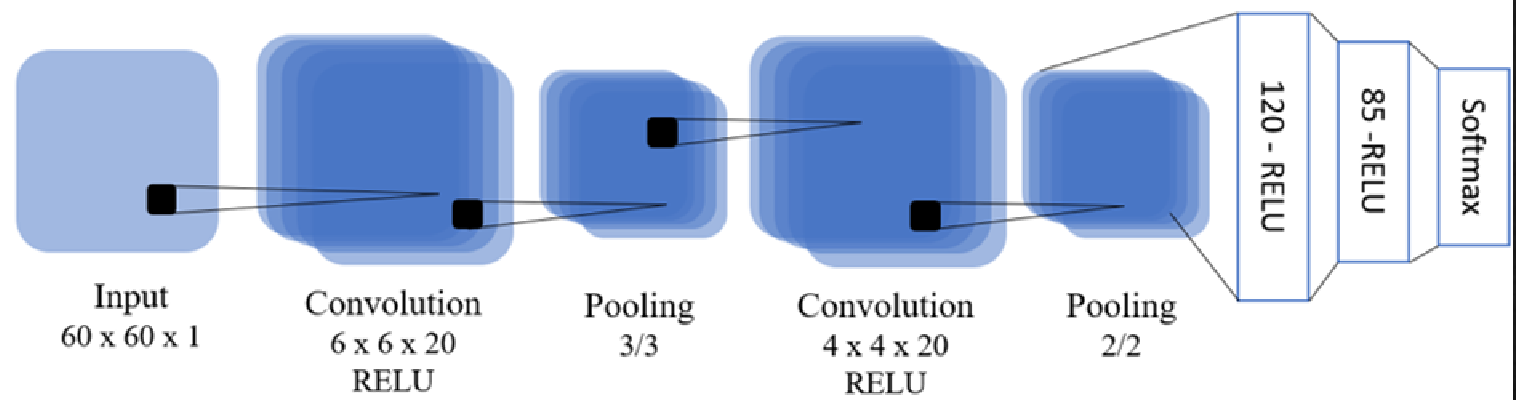


Algorithm performance




START
S=0, i=1, ep=0
Concatenated Observation (Oco)
ε > η
Yes
No
Increases
Database
Learning Level (A)
CNN
Supervised
Learning
Unsupervised
Learning
Deep
Reinforcement
Learning
TD3
Action
ANN

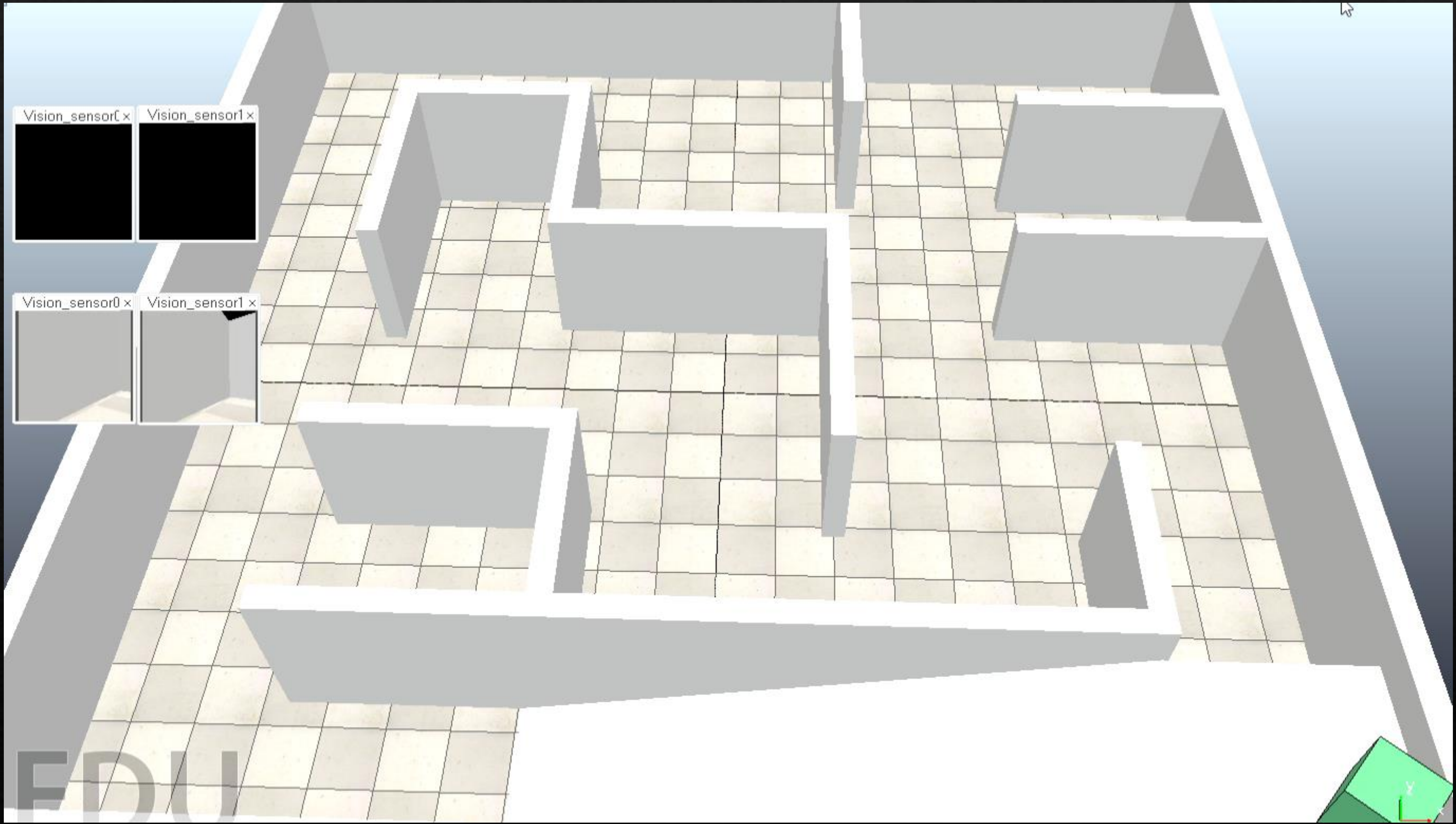
Convolutional Neuronal Network (CNN)



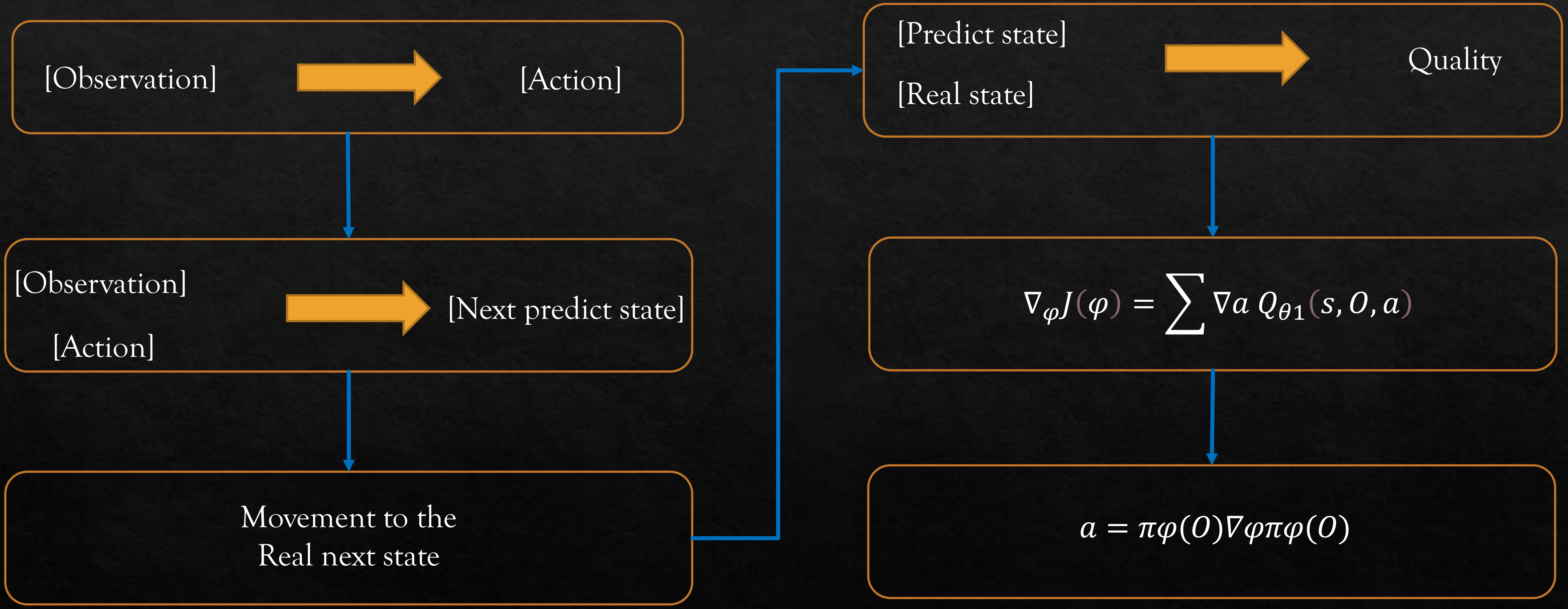
The CNN is in charge of classifying the input images when they collide or do not collide. [4,5]

[4] Cleverpy [En línea] Available: https://cleverpy.com/red-convolucional-pytorch/. Último acceso: 10/2021.
[5] Álvaro Artola. Clasificación de imágenes usando redes neuronales convolucionales en Python. Dpto. de Teoría de la Señal y Comunicaciones, Sevilla, 2019.

Environment 2

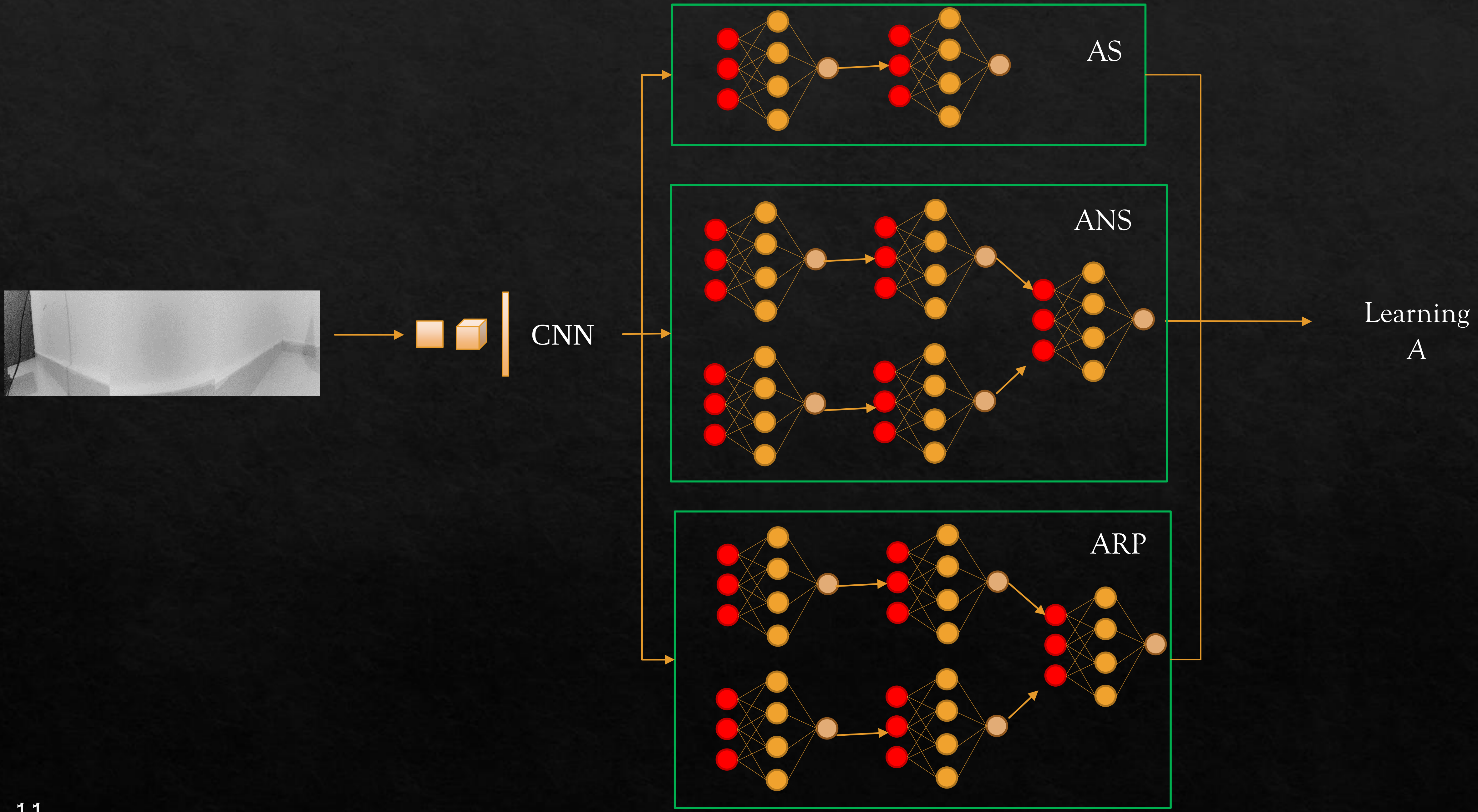


ALGORITHM OPERATION [1]

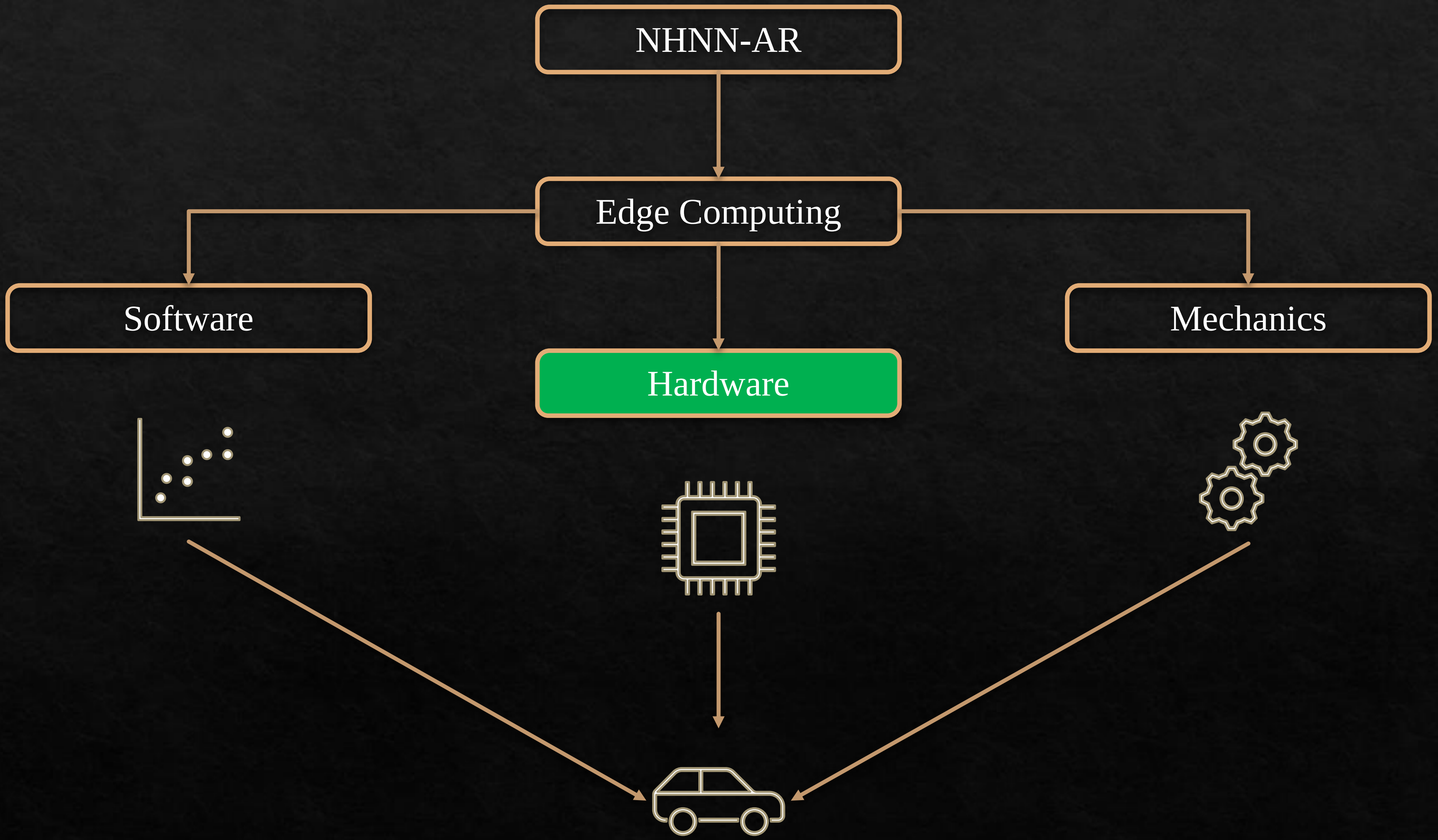


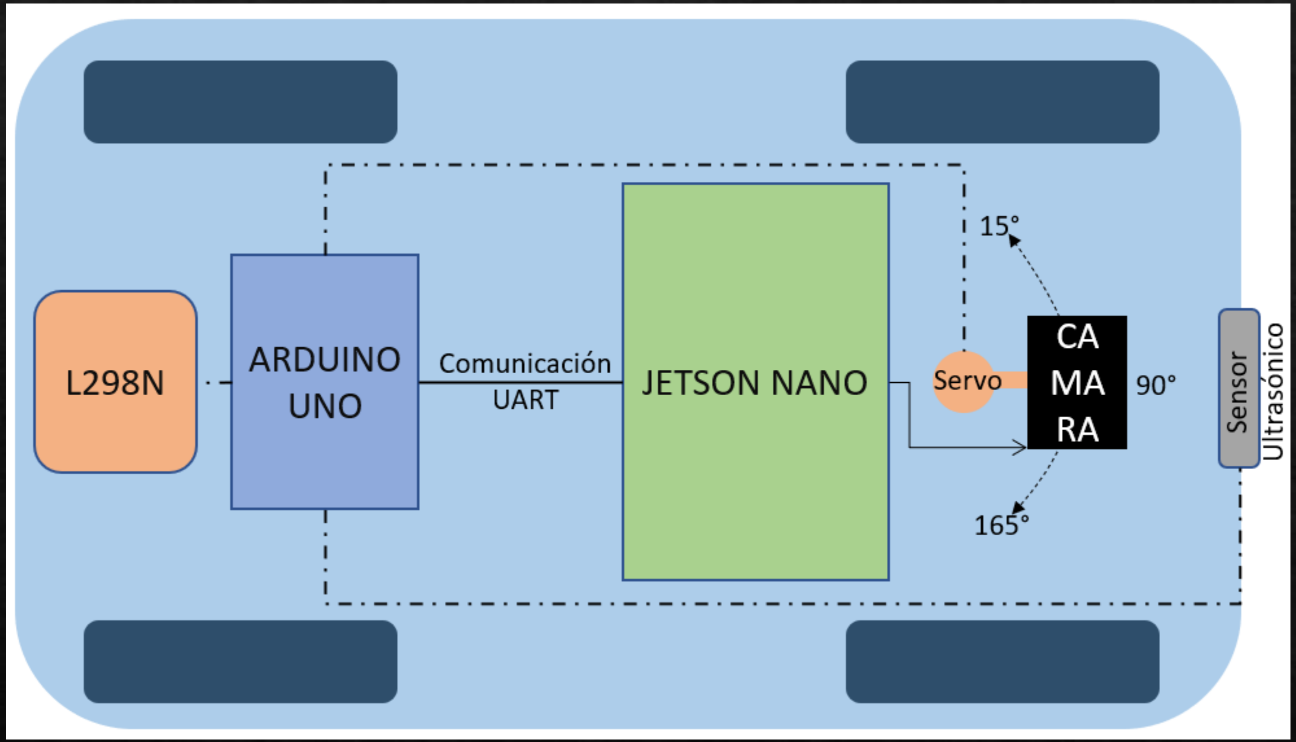
[1] Carlos Vasquez-Jalpa, Mariko Nakano-Miyatake. "A deep reinforcement learning algorithm based on modified Twin delay DDPG method for robotic Applications". ICCAS 2021.

Neural representation



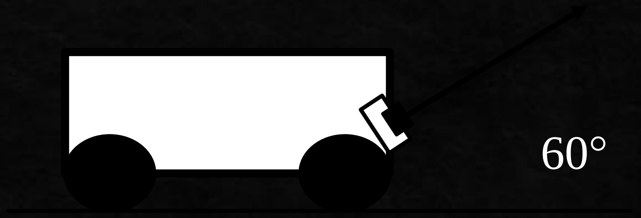
Algorithm performance



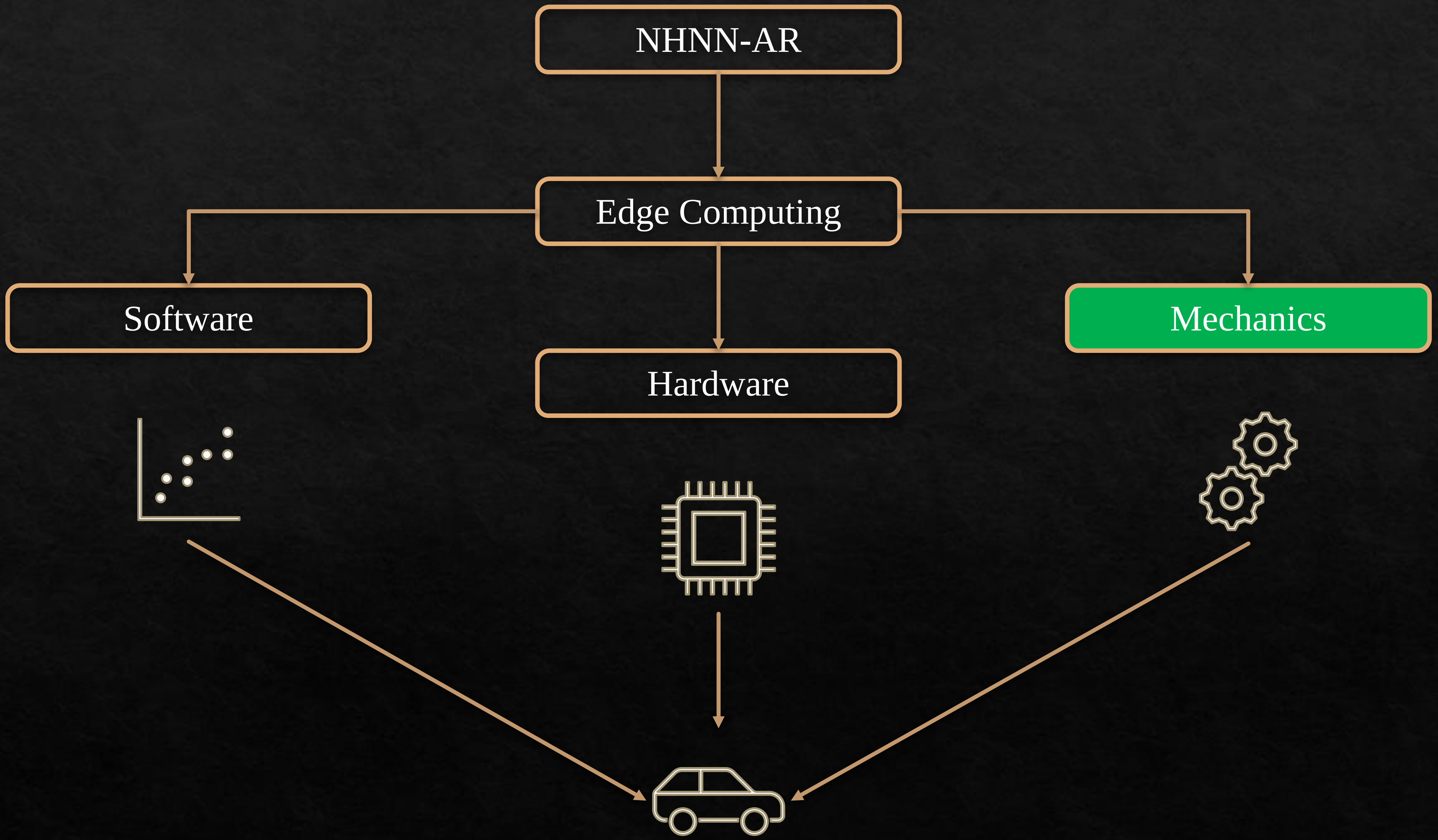


Configuration and connection of electronic components and hardware of the robotic agent.

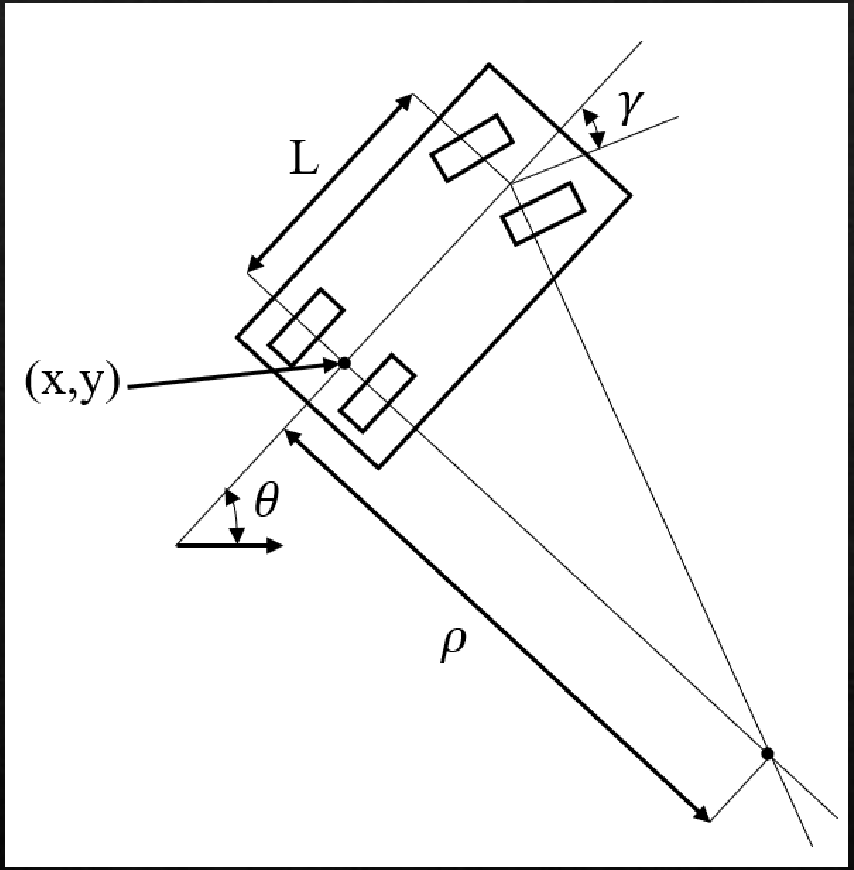
- Jetson Nano and an Arduino connected (UART protocol)
- Rate of 115200 baud/second.
- L298N Motor Control Module
- Servomotor, place the camera at 15° , 90° , 165° to capture the characteristics of the environment respectively.
- Ultrasonic sensor positioned at 60°
- CSI camera, collects data from the environment you navigate.



Algorithm performance



Mechanical Feature



Ackerman configuration

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

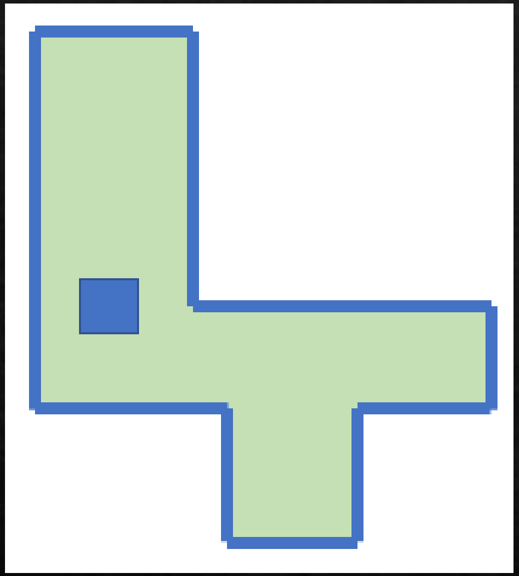
$$\dot{x} = u_s \cos\theta$$

$$\dot{y} = u_s \operatorname{sen}\theta$$

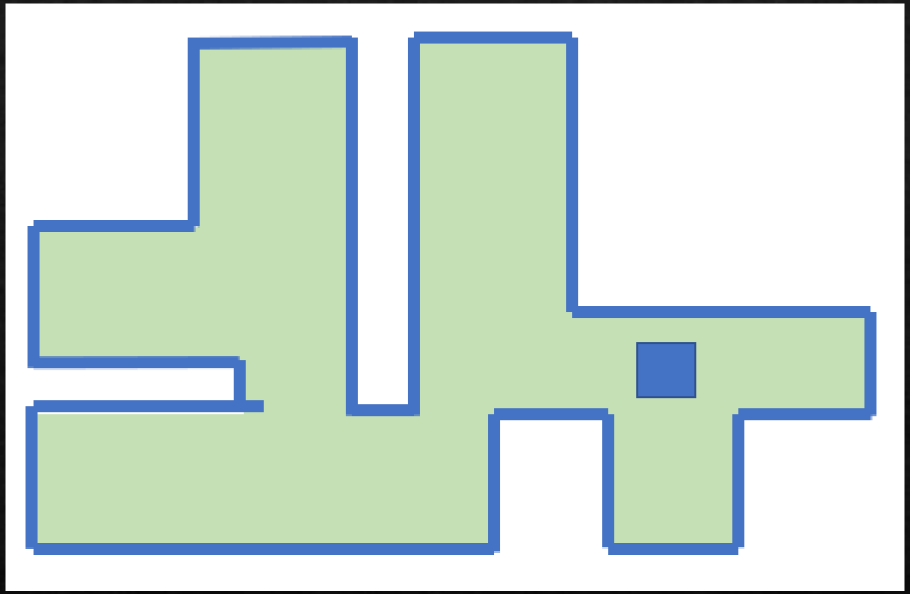
$$\dot{\theta} = \frac{u_s}{L} \tan u_\gamma$$

Static environments

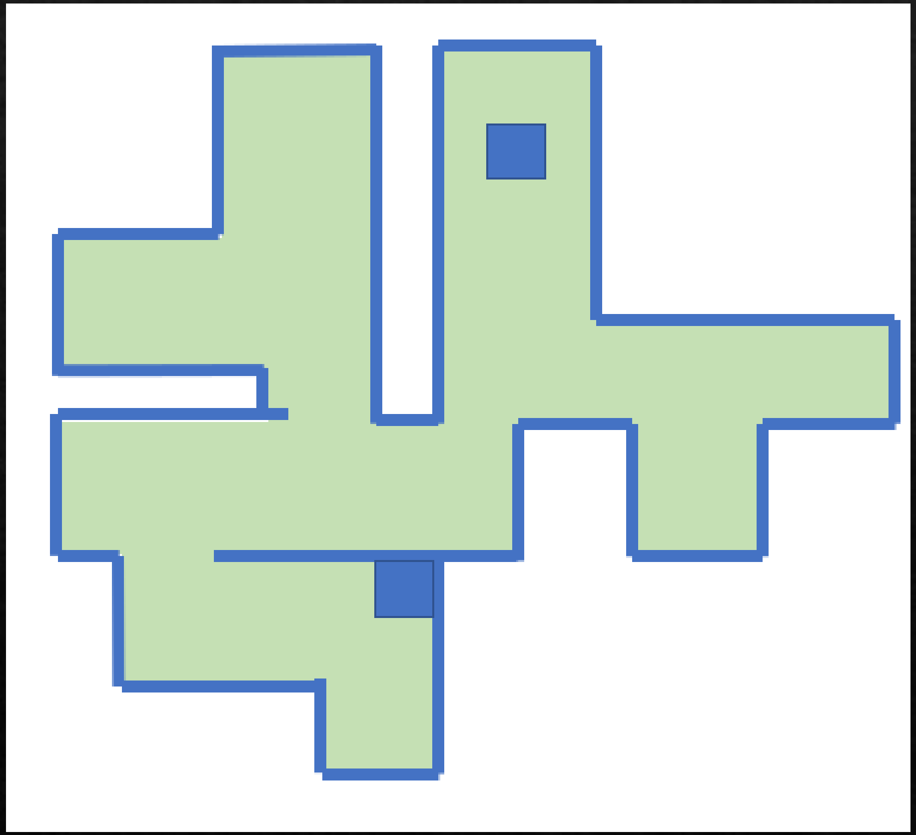
1



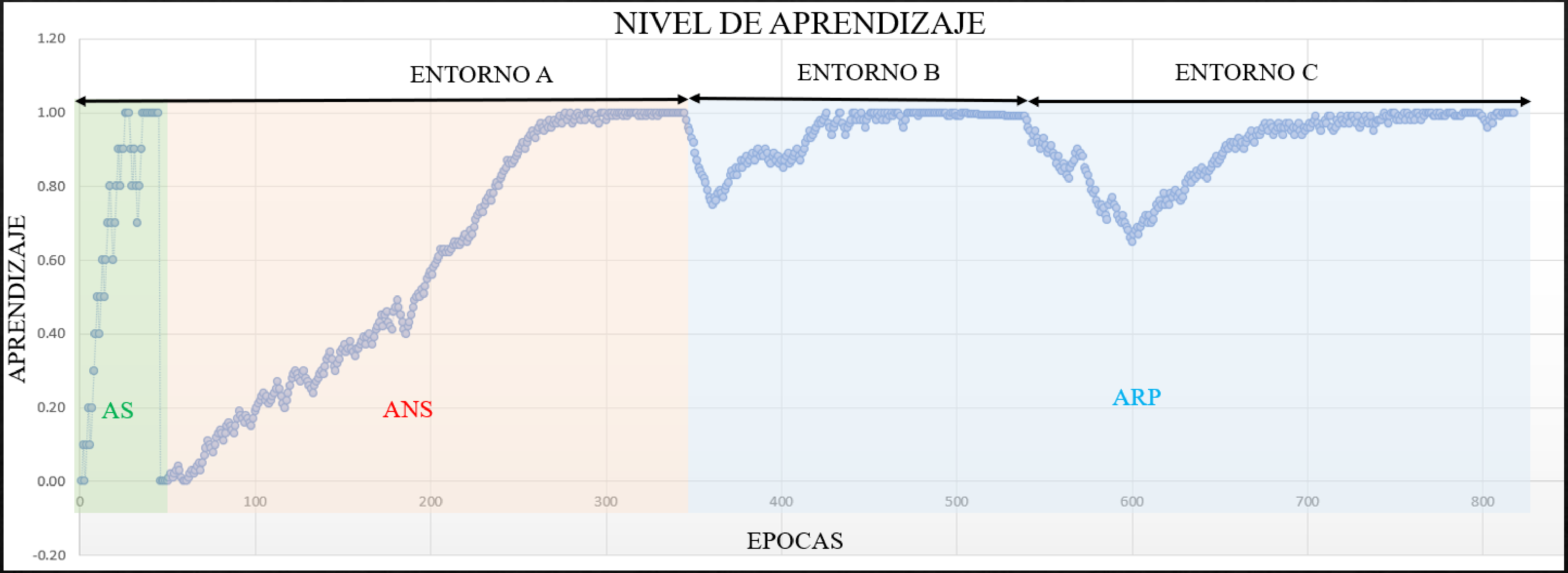
2

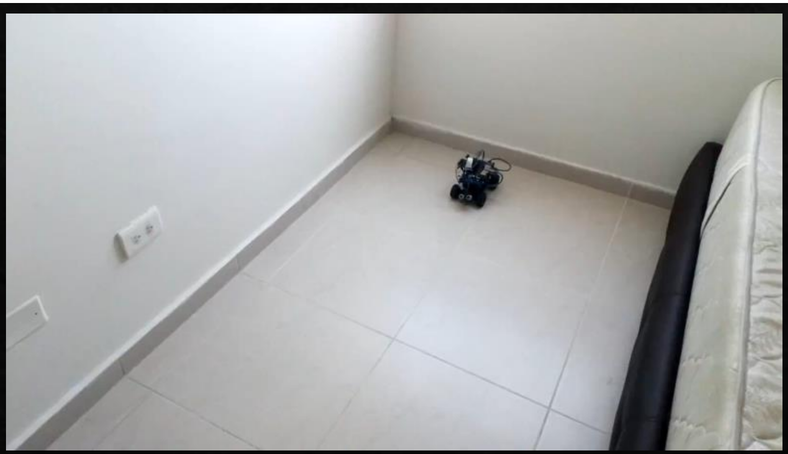


3



Results

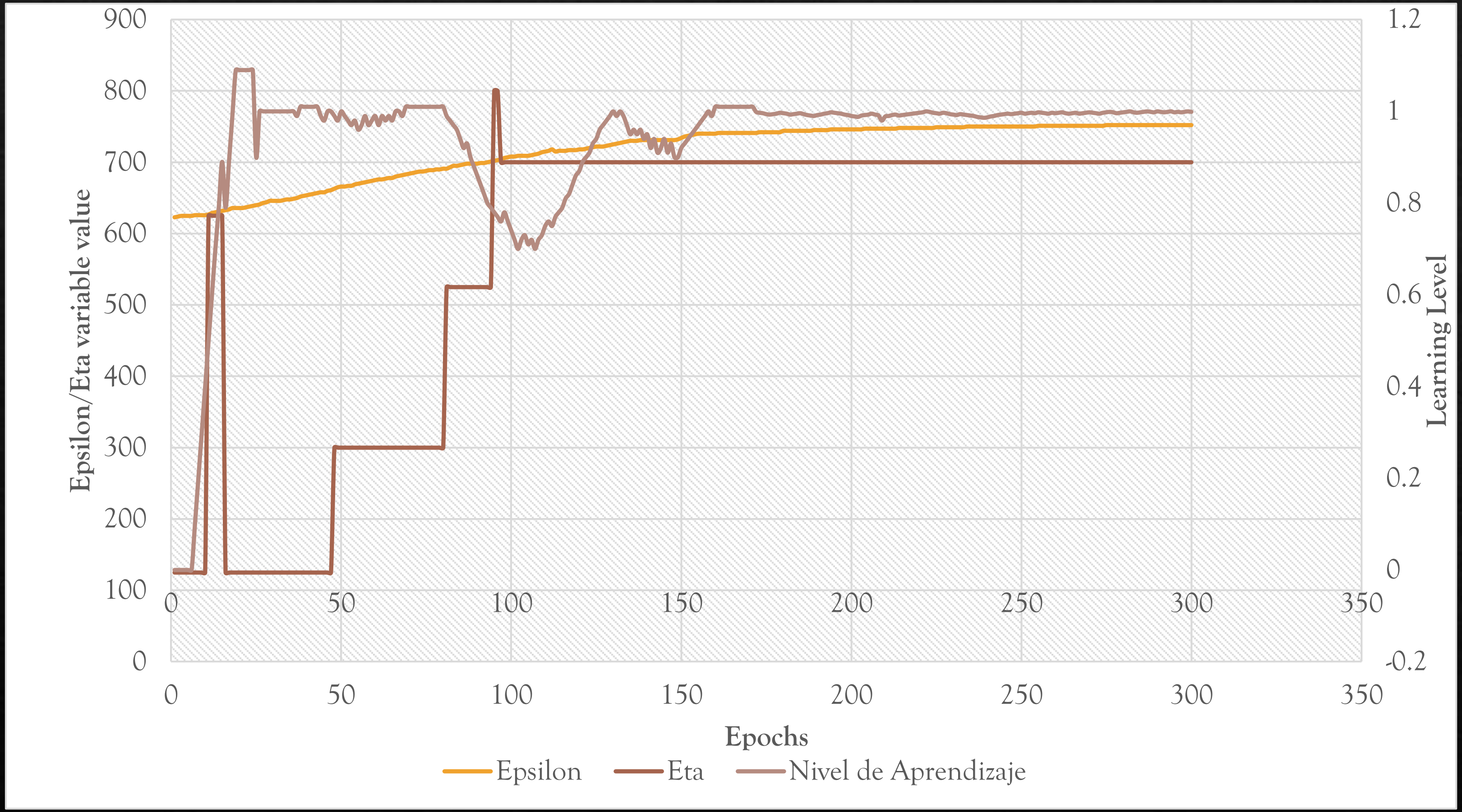

NIVEL DE APRENDIZAJE
ENTORNO A
ENTORNO B
ENTORNO C
APRENDIZAJE
1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
AS
ANS
ARP
0
100
200
300
400
500
600
700
800
EPOCAS

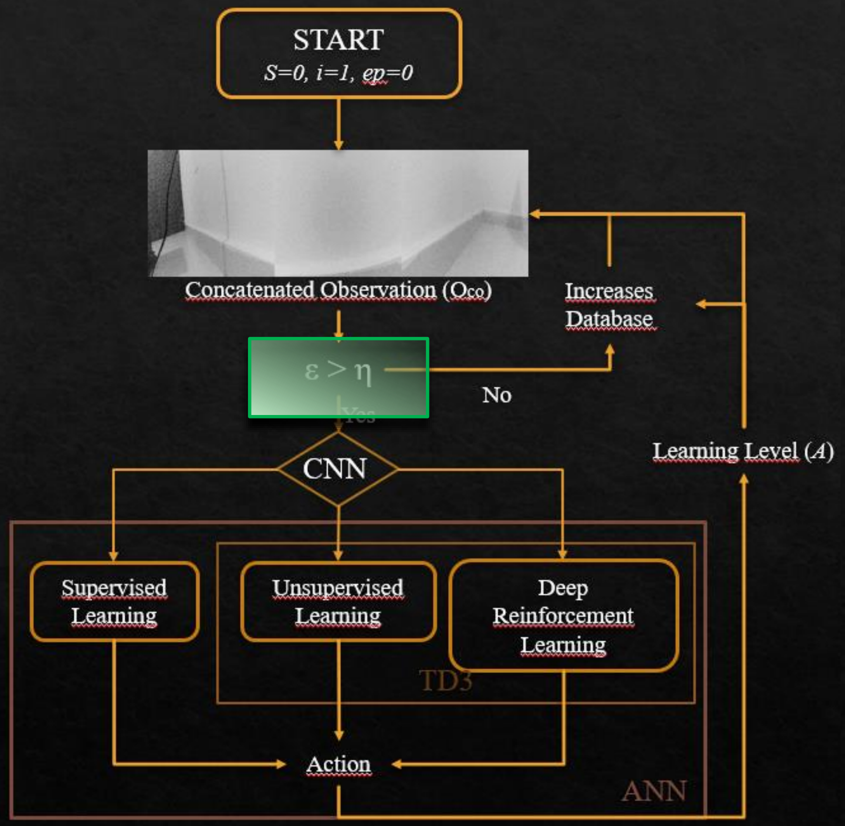


Dynamic environments

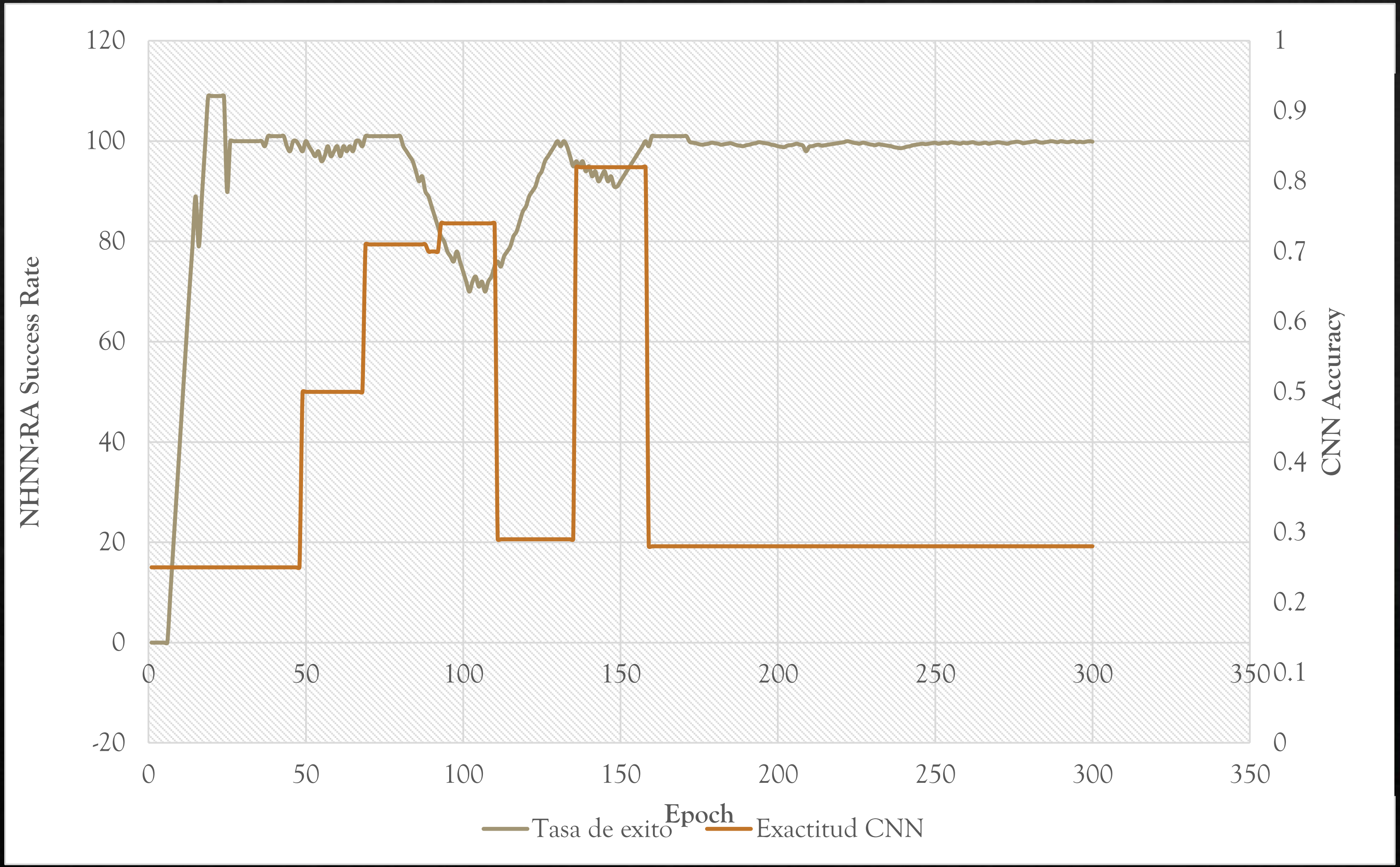
Less controlled
environment

Results

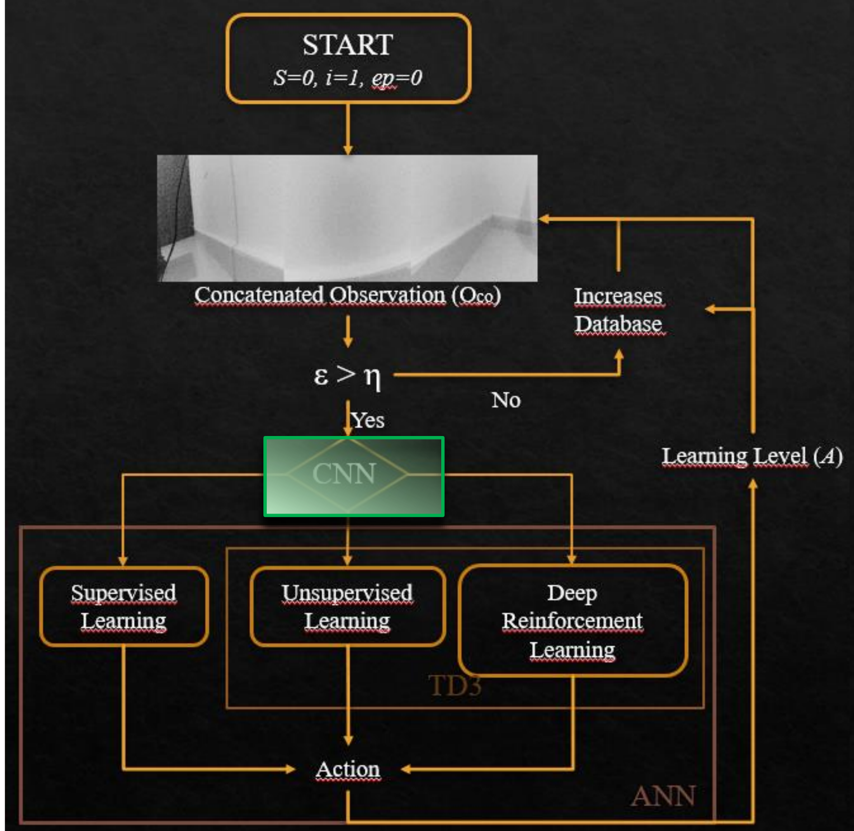

Epsilon/Eta variable value
Learning Level
Epochs
Epsilon
Eta
Nivel de Aprendizaje


START
S=0, i=1, ep=0
Concatenated Observation (Oco)
Increases Database
ε > η
No
Learning Level (A)
CNN
Supervised Learning
Unsupervised Learning
Deep Reinforcement Learning
TD3
Action
ANN

Results

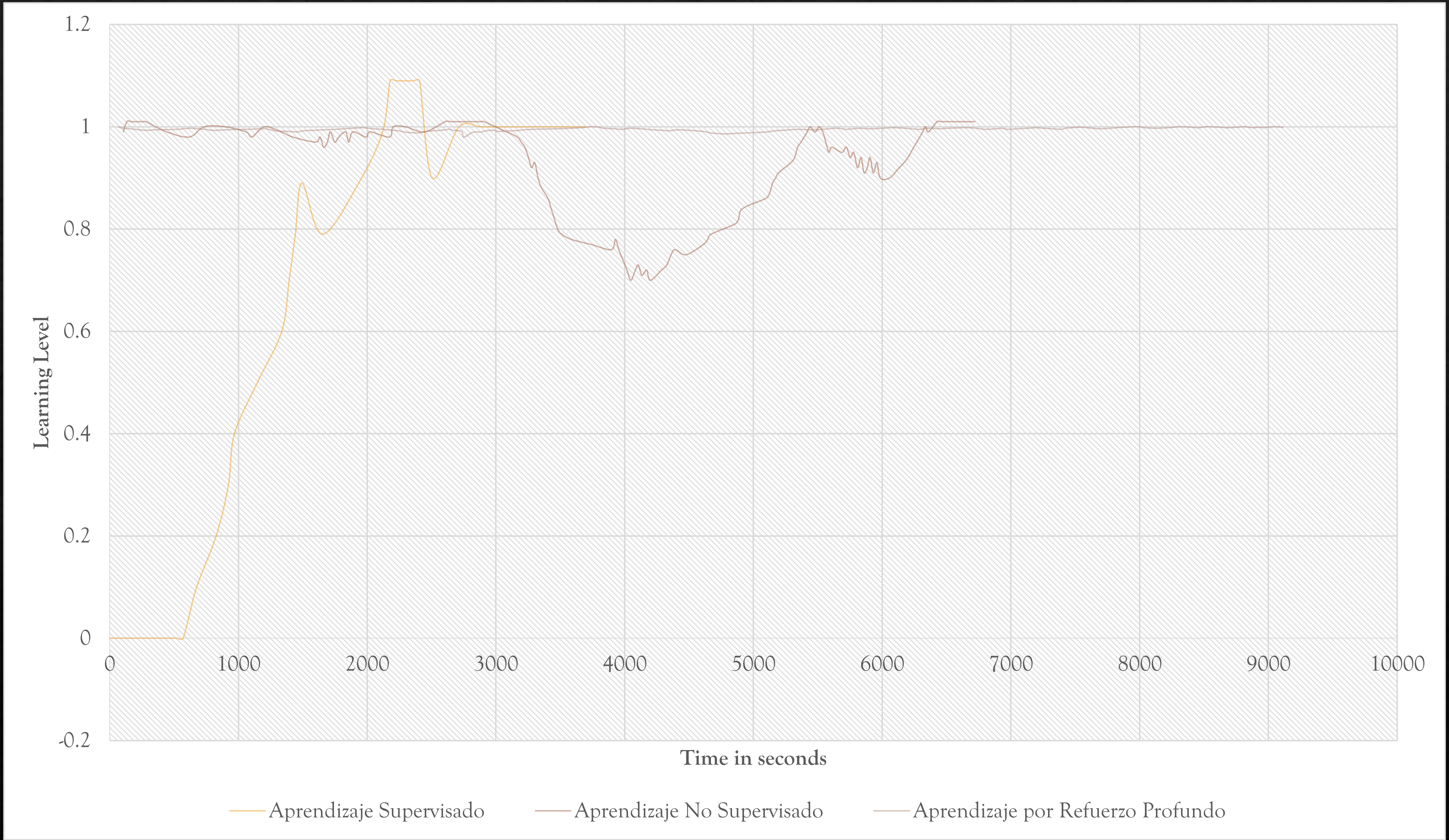


NHNN-RA Success Rate
CNN Accuracy
Epoch
Tasa de exito
Exactitud CNN



START
S=0, i=1, ep=0
Concatenated Observation (Oco)
Increases Database
ε > η
No
Yes
CNN
Learning Level (A)
Supervised Learning
Unsupervised Learning
Deep Reinforcement Learning
TD3
Action
ANN

Results



1.2
1
0.8
0.6
0.4
0.2
0
-0.2
Learning Level
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Time in seconds
Aprendizaje Supervisado
Aprendizaje No Supervisado
Aprendizaje por Refuerzo Profundo

https://www.youtube.com/watch?v=SUtYk2Ee1DU

Conclusion

- Using Jetson Nano is possible to use it for reinforced, unreinforced and deep reinforcement learning trainings, however, it does not have the optimal computational capacity to perform a fluid NHNN-RA network training.
- The performance of the NHNN-RA algorithm depends on the condition $\varepsilon > \eta$ and the accuracy of the CNN implemented, the higher the ACCURACY of CNN the Learning Level of the NHNN-RA will decrease.
- The varied the input images in the agent, the longer it will take for the network to stabilize in a range of NA=0.997 to NA=1. In our case, 129 images taken were necessary to stabilize in the dynamic environment and 623 in the static environment.
- The TD3 configuration is physically demonstrated to provide more stable results than the DDPG.
- The average number of steps in each epoch is 11 when evaluating the algorithm on a physical robot.
- The NHNN-RA algorithm works on the robotic agent, however, for better results the mechanical block and hardware will be improved considering the functionality in size, weight and design. In addition, to have a continuous and faster training you must have a durable energy system.
- Also as future work, NHNN-RA will be implemented in different types of robots to check how viable the algorithm is.

References

[1] Carlos Vasquez-Jalpa, Mariko Nakano-Miyatake. “A deep reinforcement learning algorithm based on modified Twin delay DDPG method for robotic Applications”. ICCAS 2021.

[2] Cheng, KS (Cheng, Kai-Sheng) Lin, HY (Lin, Huei-Yung). “Stereo Matching with Bit-Plane Slicing and Disparity Fusion”. IEEE International Conference on Systems Man and Cybernetics Conference Proceedings, 2015. Doi: 10.1109/SMC.2015.71

[3] Gehrig, S Schneider, N Franke, U. “Stereo vision during adverse weather - Using priors to increase robustness in real-time stereo vision”. DEC, 2017. Doi: 10.1016/j.imavis.2017.07.008

[4] Cleverpy [En línea] Available: https://cleverpy.com/red-convolucional-pytorch/. Último acceso: 10/2021.

[5] Álvaro Artola. Clasificación de imágenes usando redes neuronales convolucionales en Python. Dpto. de Teoría de la Señal y Comunicaciones, Sevilla, 2019.

THANK YOU FOR LISTENING

cvasquezj1200@alumno.ipn.mx