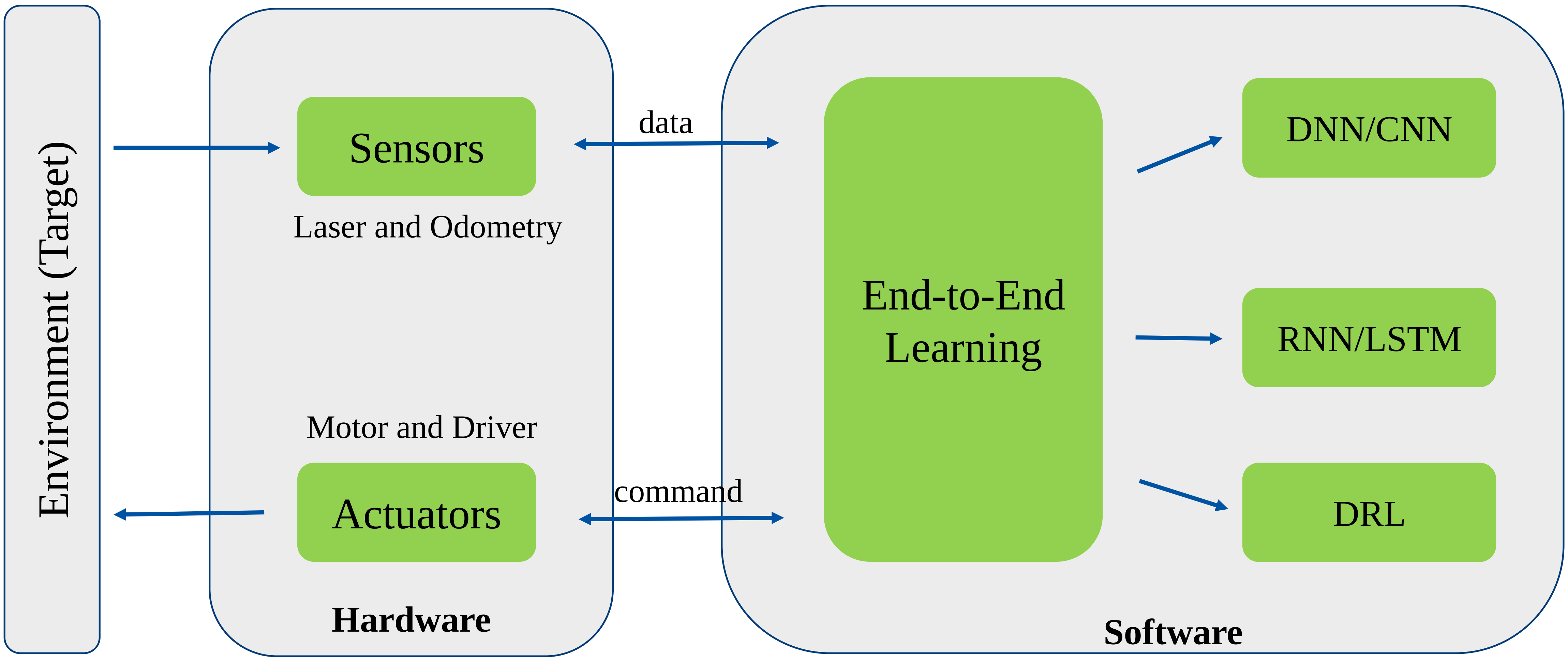
Application of End-to-end Learning

Benchun Zhou

School of Automation Science and Electronic Engineering
Beihang University

- **Framework**


Environment (Target)
Sensors
Laser and Odometry
Motor and Driver
Actuators
Hardware
data
command
End-to-End Learning
DNN/CNN
RNN/LSTM
DRL
Software

• CNN/DNN: End to End Learning for Self-Driving Cars, 2016

- Contribution: CNN+FC architecture
- Application: lane keeping
- Input: raw pixels from a single front-facing camera
- Label: human steering angle
- Training data: video(72h human driving data) and real data.

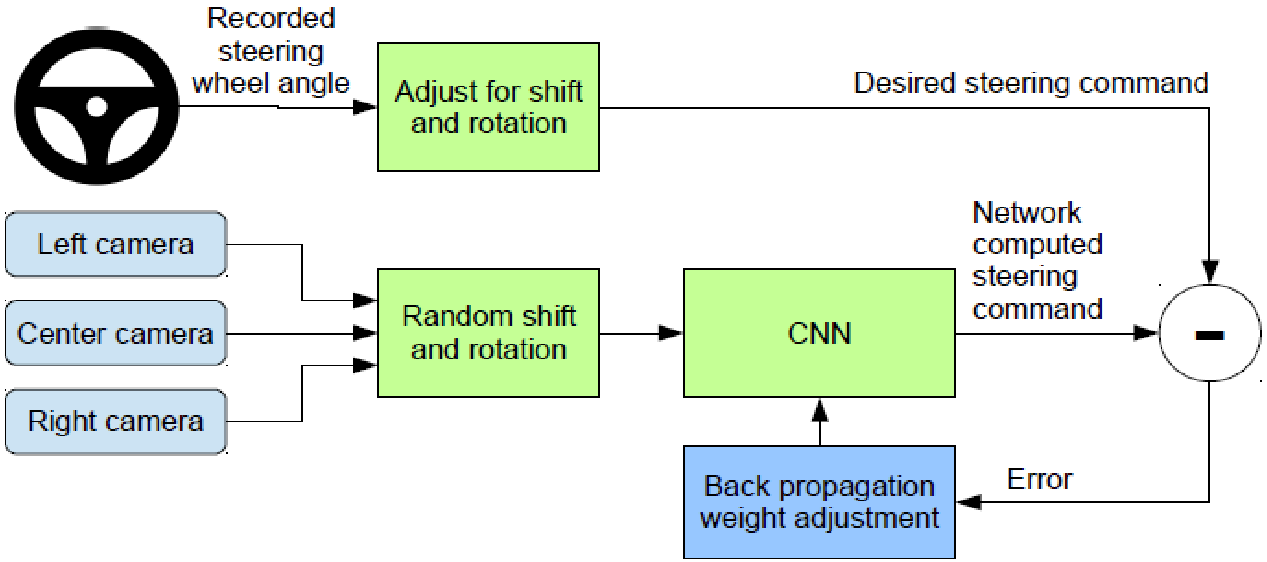


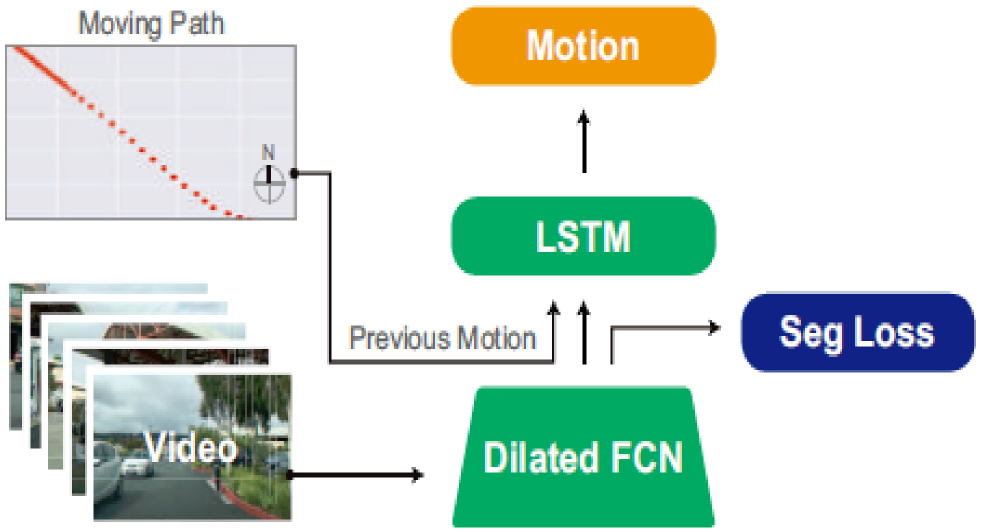
Figure 2: Training the neural network.

- **CNN/DNN: End-to-end Learning of Driving Models from Large-scale Video Datasets, 2016**

- Contribution: FCN-LSTM architecture

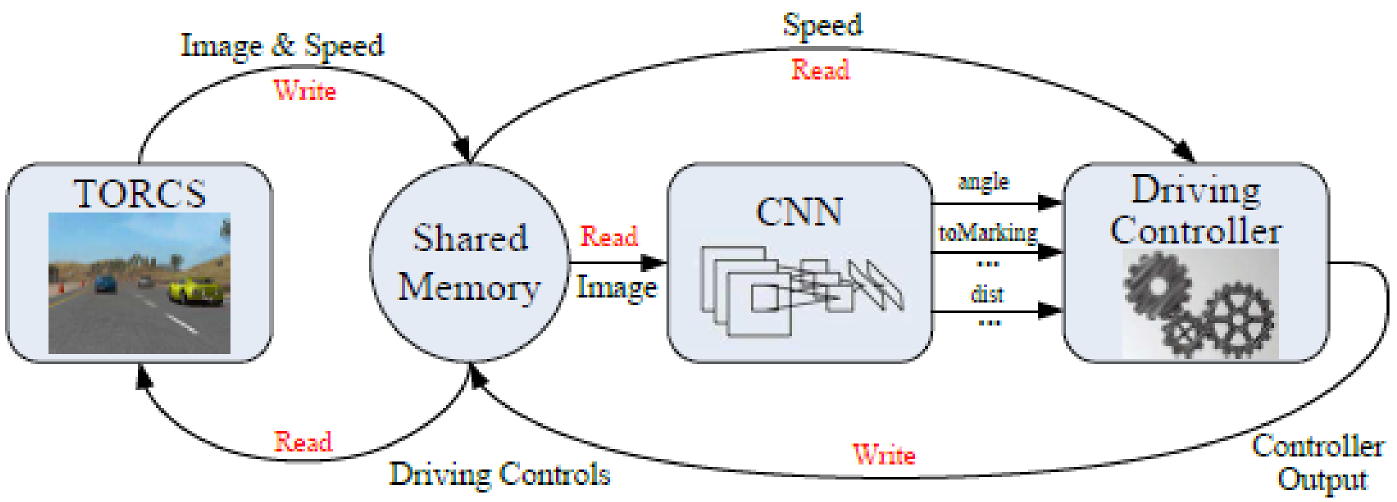
  Leverages available scene segmentation side tasks

- Application: urban environment
- Input: camera information
- Label: human diver behavior
- Training data: large scale crowd-sourced video data



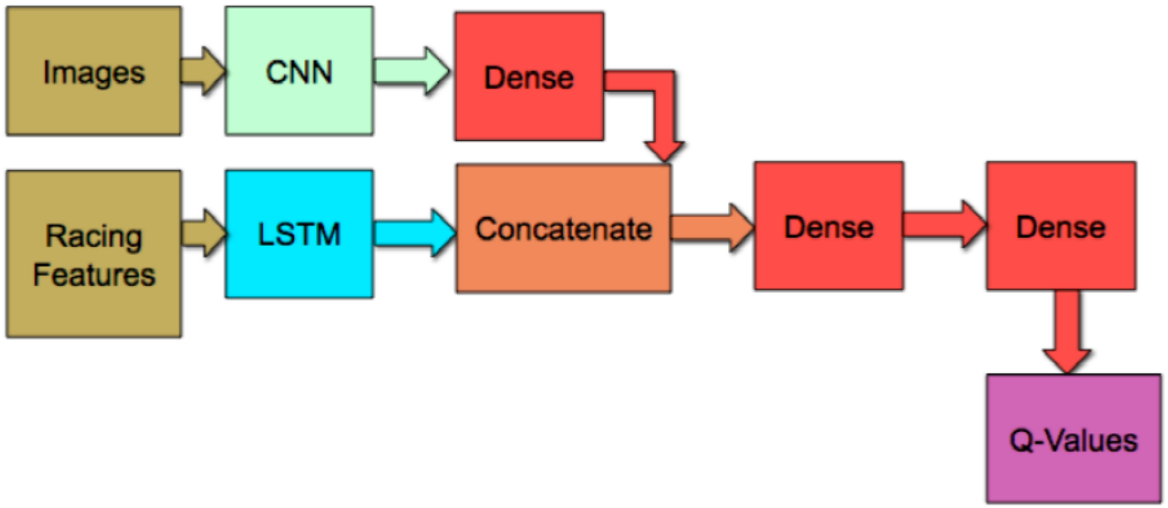
- **CNN/DNN: DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving, 2015**

- Contribution: direct perception (combine mediated perception and behavior reflex)
- Application: highway
- Input: camera information
- Label: human driver behavior
- Training data: TORCS, KITTI



- **DRL: CARMA: A Deep Reinforcement Learning Approach to Autonomous Driving, 2016**

- Contribution: CNN-RNN DQN architecture
- Application: track keep
- Input: image and racing features (speed, angle, distance…)
- Training data: racing simulator (Vdrift)



- **DRL: Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving, 2018**
- Contribution: 1)Policy Gradient does not really require the Markov assumption
  2)decompose policy function into desired and constraints $\pi_\theta = \pi^{(T)} \circ \pi_\theta^{(D)}$
  3)employ a hierarchical temporal abstraction 'DAG'
- Application: Double Merge Scenario
- Input: sensor information(geometry of lanes, location, velocity, heading…)
- Training data: simulator

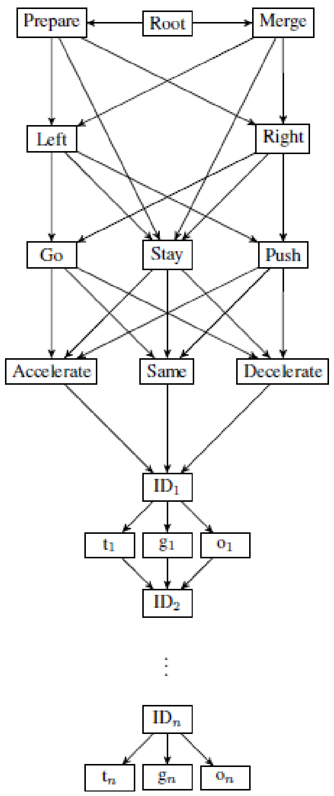
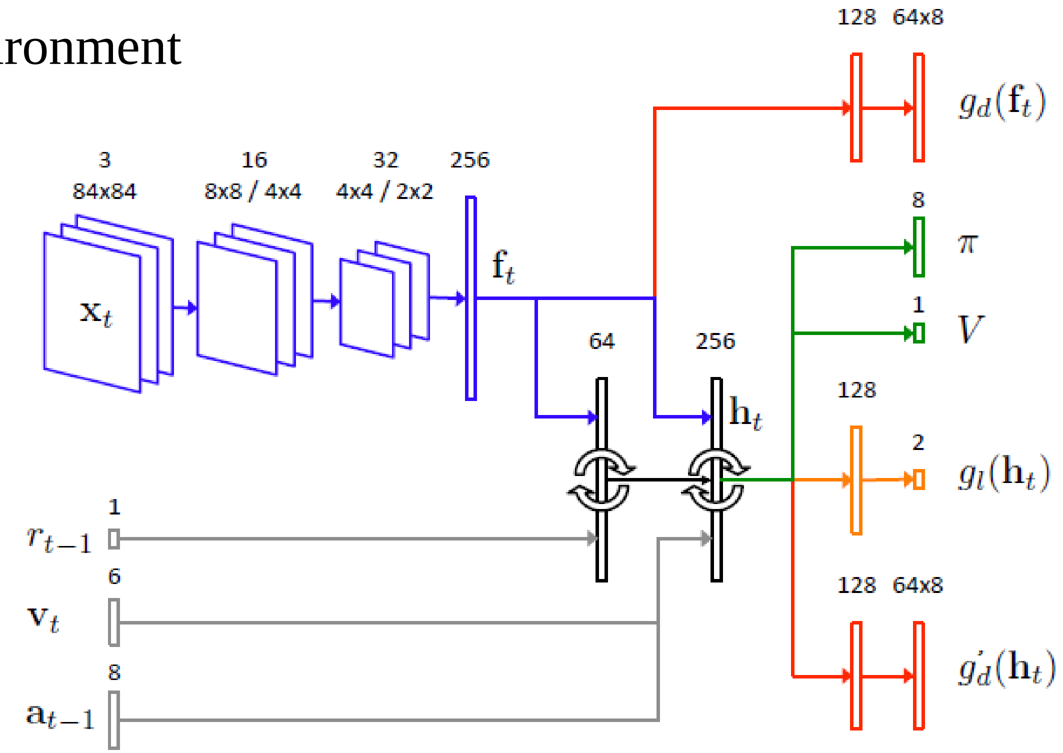


Figure 2: An options graph for the double merge scenario.

• DRL: Learning to navigate in complex environments, 2017

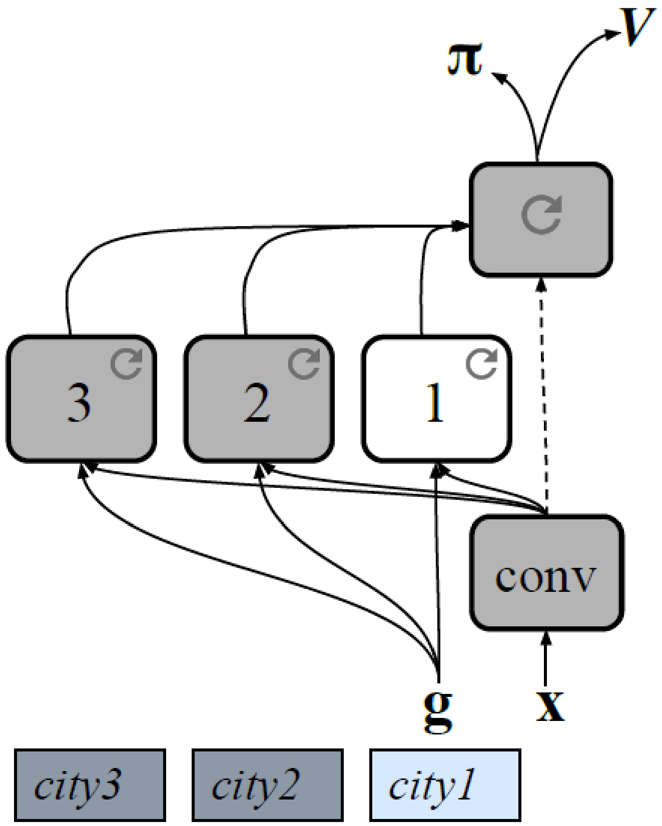
- Contribution: 1)Nav A3C+CNN+LTSM+Predicting Loop Closure architecture
  2)auxiliary objectives develop more general navigation strategies.
- Application: Navigation Task in Maze
- Input: RGB image
- Training data: maze environment



• DRL: Learning to Navigate in Cities Without a Map, 2018

- Contribution: MultiCityNav architecture (IMPALA+ LSTM)

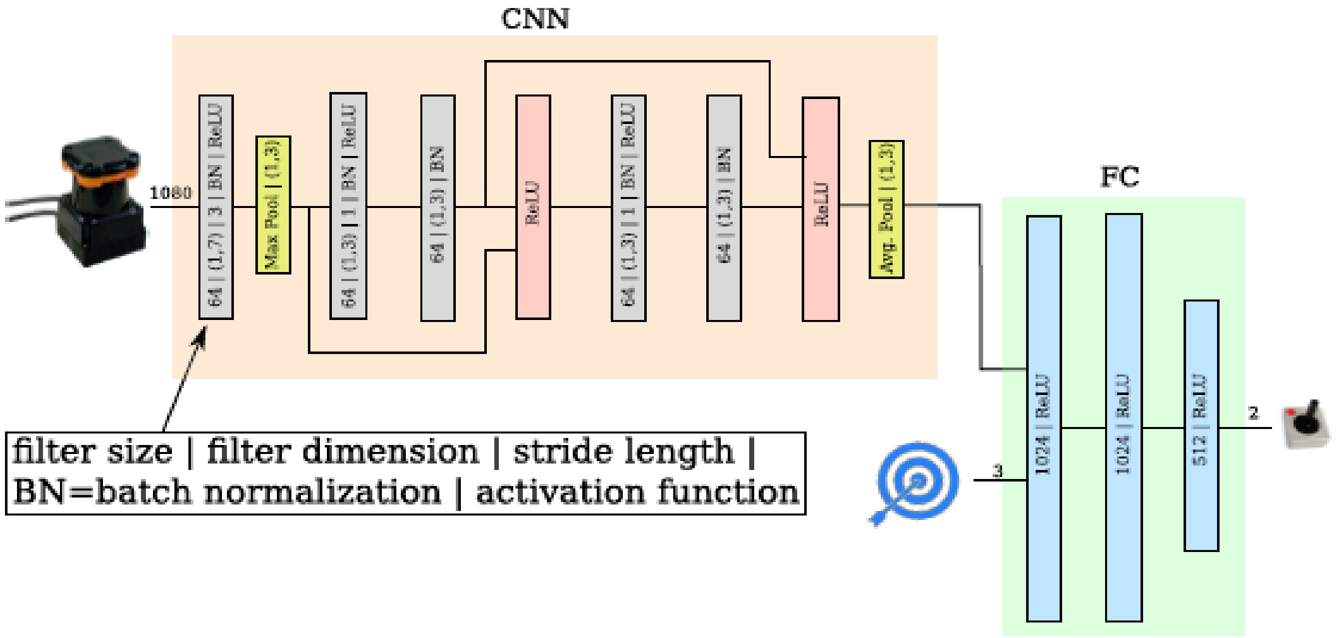
  Transfer to multiple cities
- Application: Navigation Task in Cities
- Input: Camera
- Training data: google street view



2. Mobile Robotics

- **CNN/DNN: From Perception to Decision A Data-driven Approach to End-to-end Motion Planning for Autonomous Ground Robots, 2016**

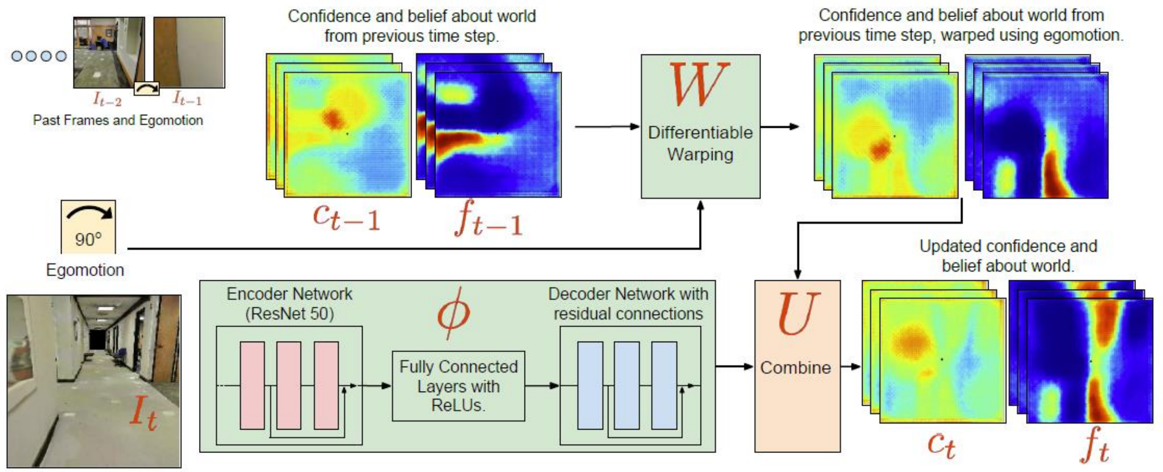
- Contribution: ResNet+FC architecture
- Application: indoor navigation
- Label: ROS navigation
- Input: lidar data
- Training data: simulation

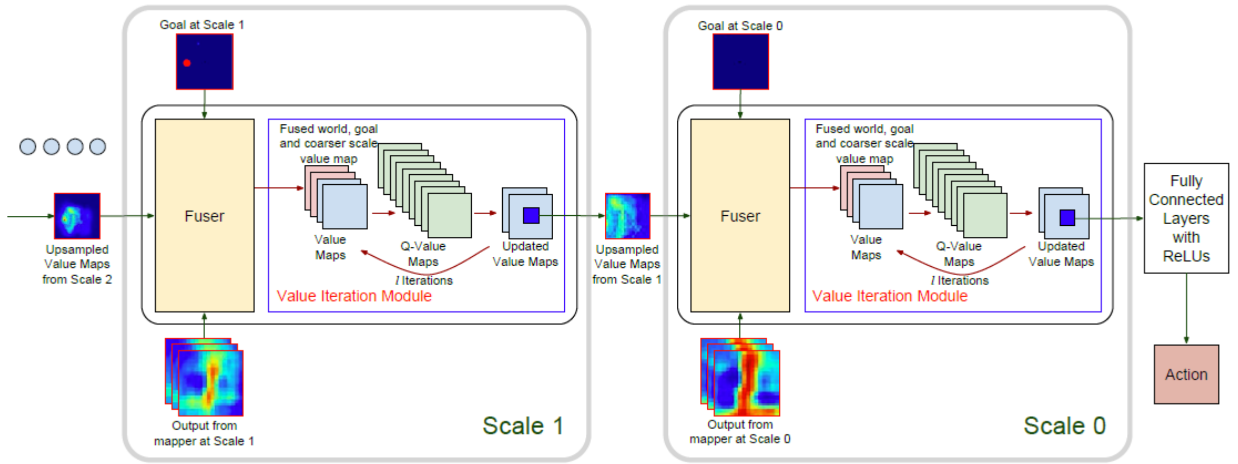


2. Mobile Robotics

- **CNN/DNN: Cognitive Mapping and Planning for Visual Navigation, 2017**

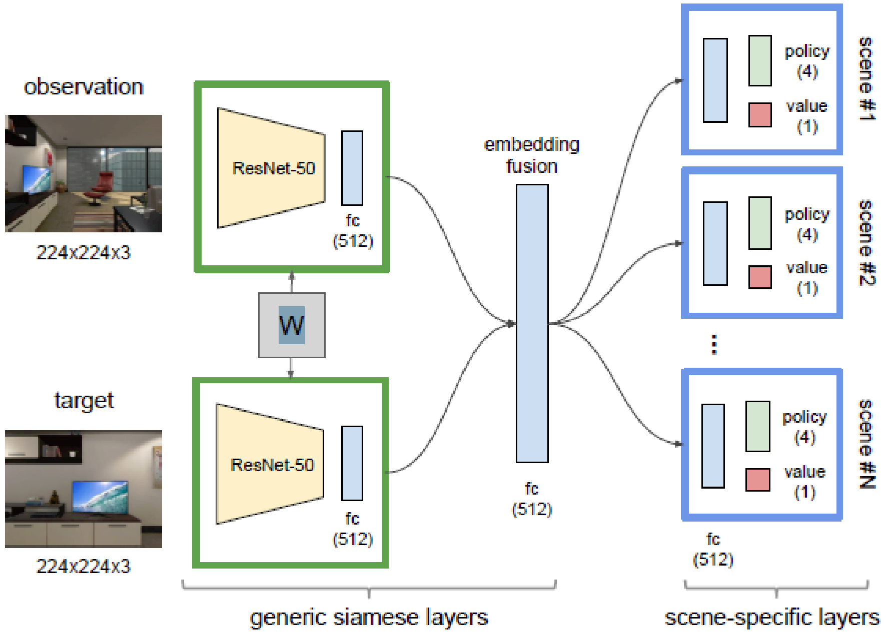
- Contribution: Cognitive Mapper and Planner architecture
- Application: indoor navigation
- Input: RGB image
- Label: DAGGER navigation
- Training data: DAGGER



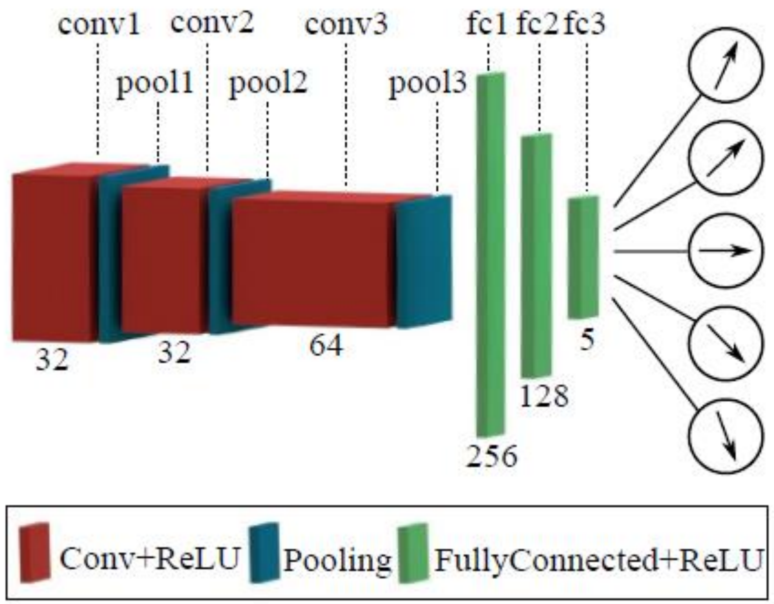


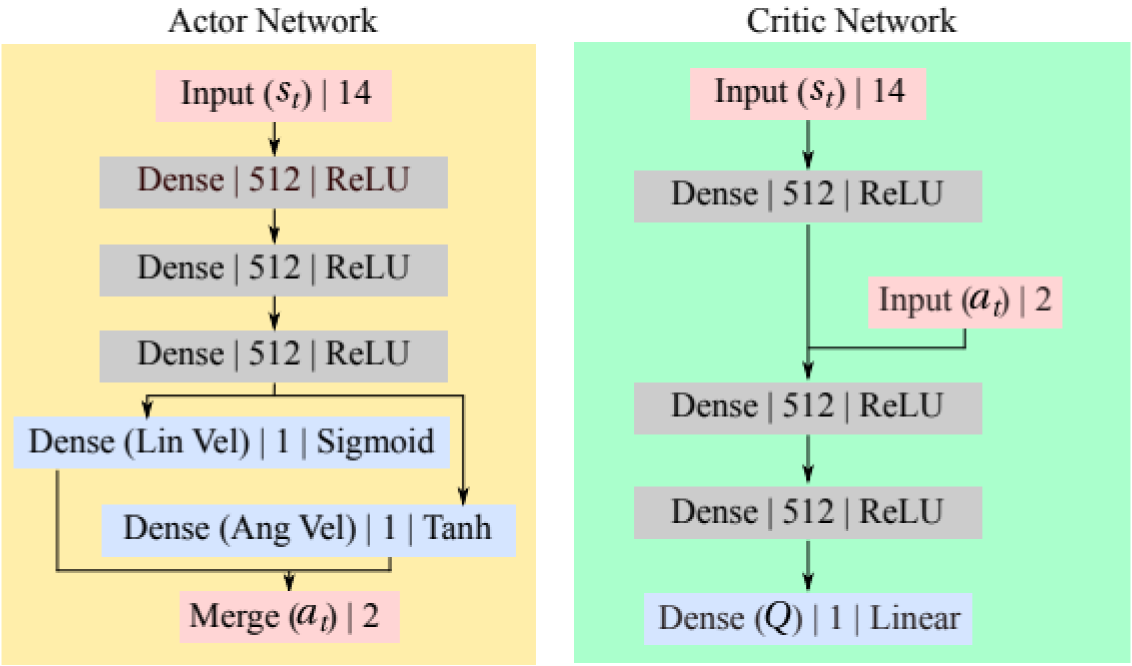
• DRL: Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning, 2017

- Contribution: 1) AC + ResNet architecture for better generation
  2) AI2-THOR framework to build high-quality 3D scenes.
- Application: indoor navigation
- Input: image
- Training data: AI2-THOR



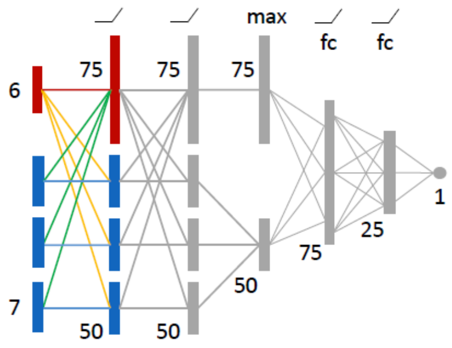
- **DRL: Towards cognitive exploration through deep reinforcement learning for mobile robots, 2016**
- **DRL: Virtual-to-real Deep Reinforcement Learning: Continuous Control of Mobile Robots for Mapless Navigation**



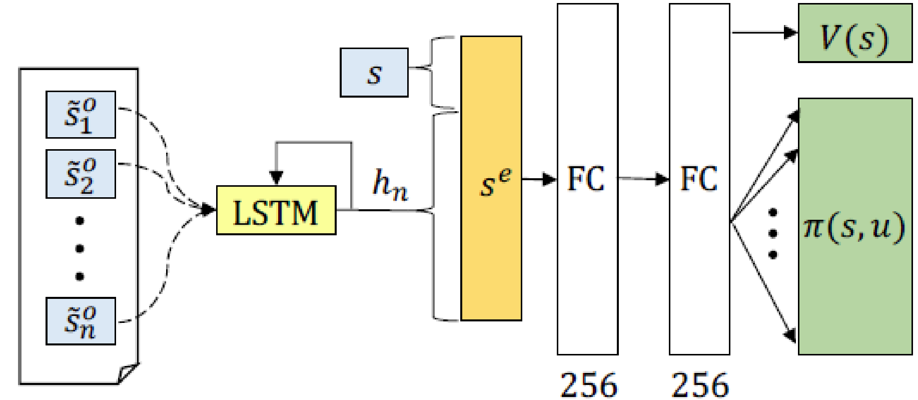


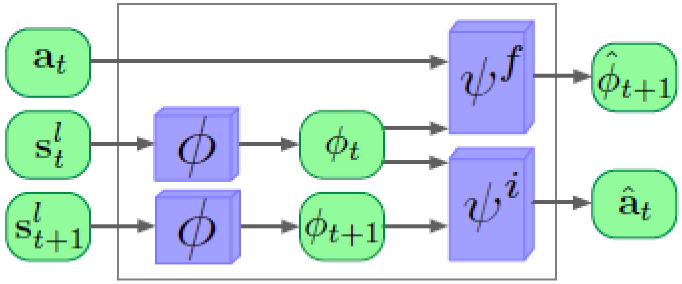
2. Mobile Robotics

- **DRL: Socially Aware Motion Planning with Deep Reinforcement Learning, 2017**
- **DRL: Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning, 2018**
- **DRL: Curiosity-driven Exploration for Mapless Navigation with Deep Reinforcement Learning , 2018**



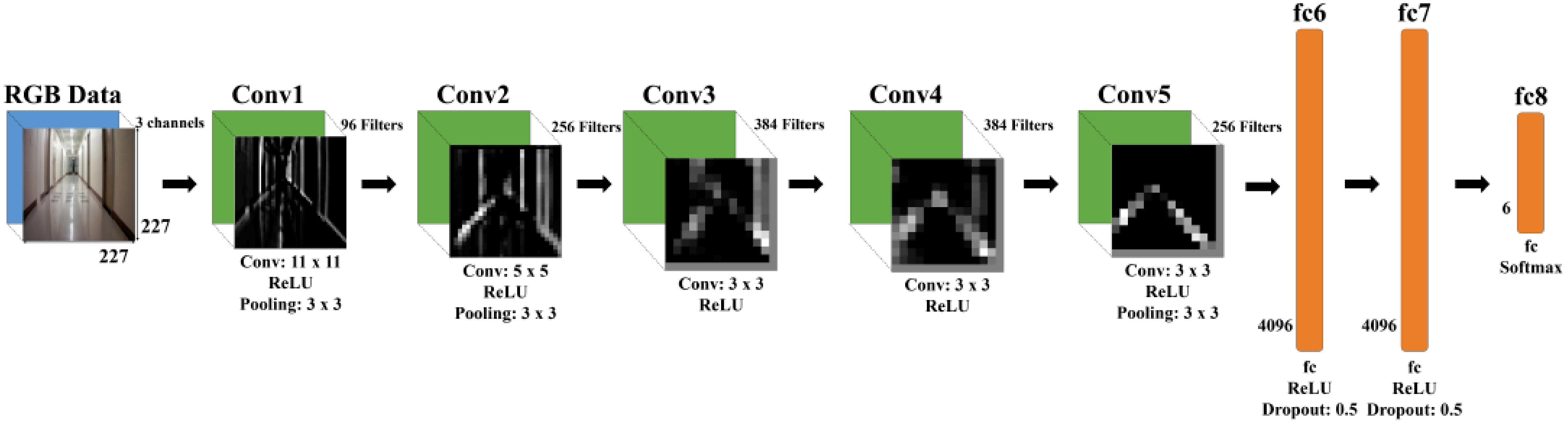
(b) symmetric multiagent net





- **CNN/DNN: Learning Monocular Reactive UAV Control in Cluttered Natural Environments**

- **CNN/DNN: Deep Neural Network for Real-Time Autonomous Indoor Navigation**



- **DRL: Autonomous Navigation of UAV in Large-scale Unknown Complex Environment with Deep Reinforcement Learning**

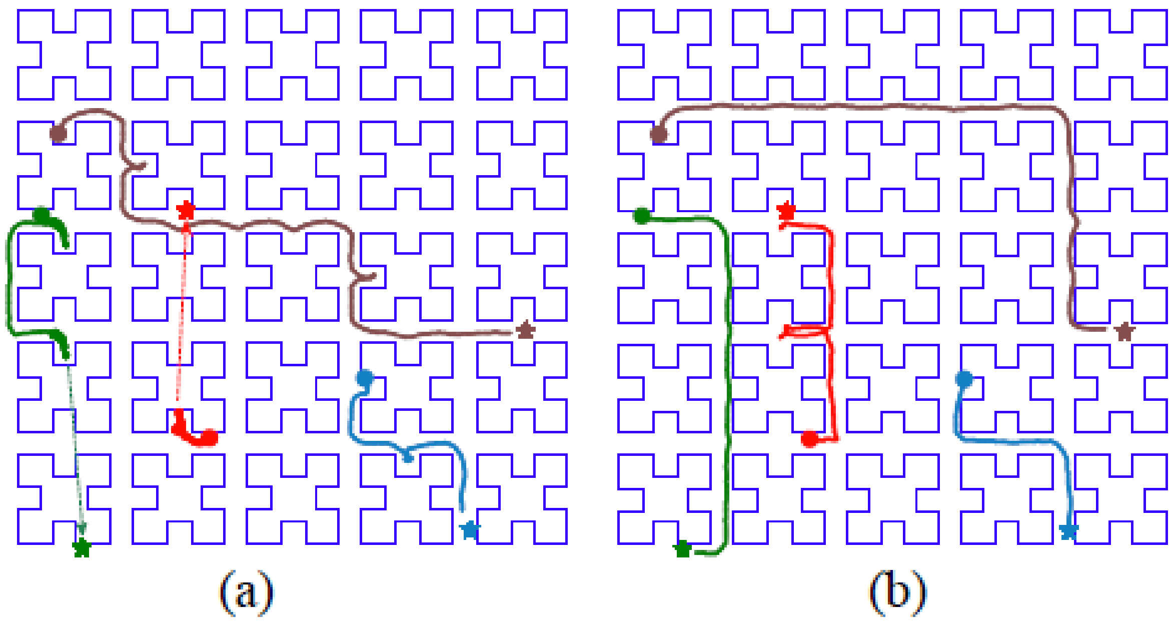


Fig. 3. Illustration of trajectories, where circles represents starting positions and stars represent target positions. (a): trajectories generated by a DDPG agent (b): trajectories generated by a Fast-RDPG agent

3. UAV Control

- **DRL: Autonomous UAV Navigation Using Reinforcement Learning**
- **DRL: Autonomous Navigation of UAV by Using Real-Time Model-Based Reinforcement Learning**

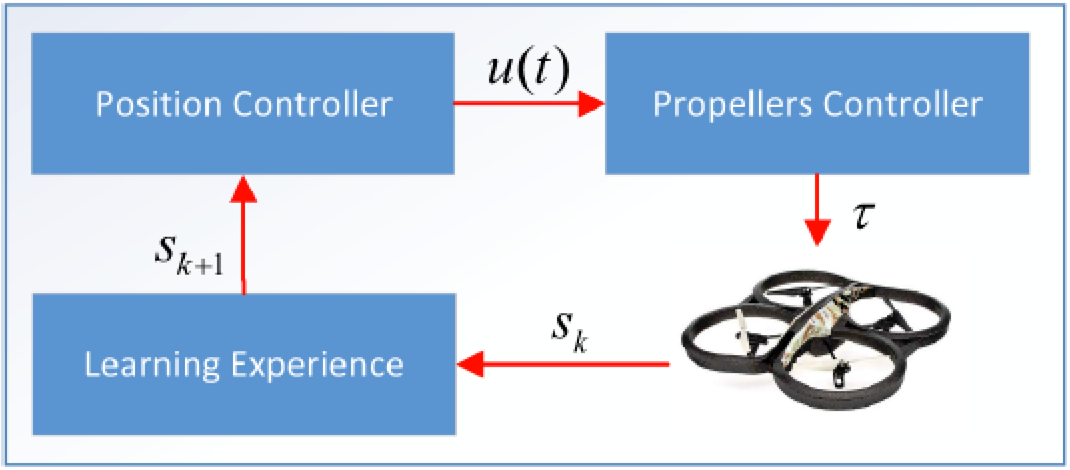


Fig. 2. Reinforcement Learning model.

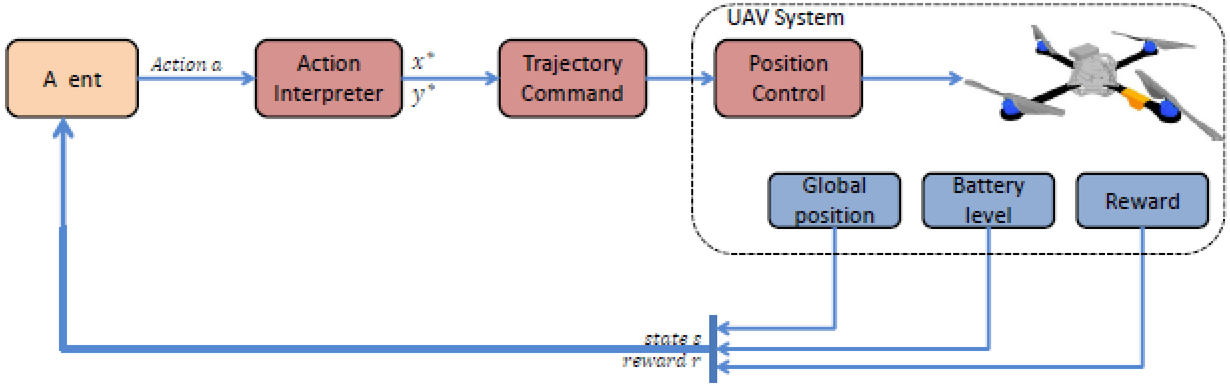


Fig. 2: The control architecture of the UAV.

- **DRL: Control of a Quadrotor With Reinforcement Learning**

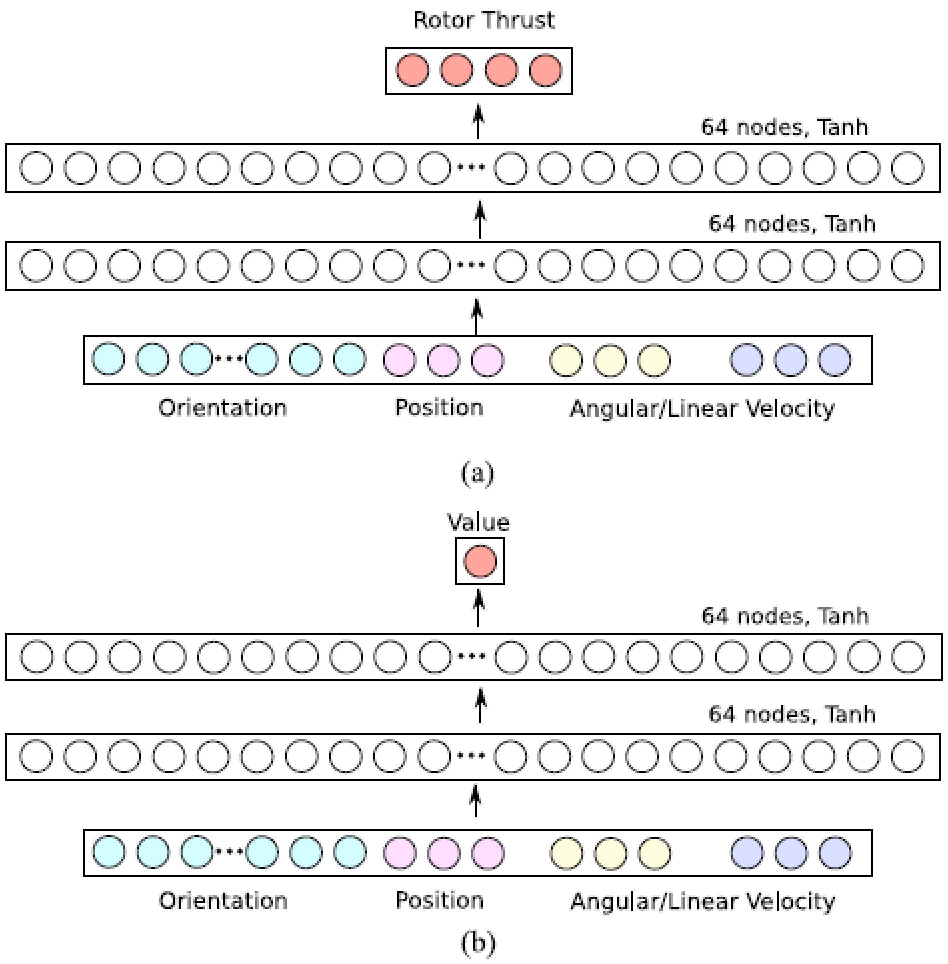


Fig. 2. The two neural networks used in this work are shown. (a) Policy network. (b) Value network.

3. UAV Landing

- **DRL:A Deep Reinforcement Learning Strategy for UAV Autonomous Landing on a Moving Platform**
- **DRL: Autonomous Quadrotor Landing using Deep Reinforcement Learning**

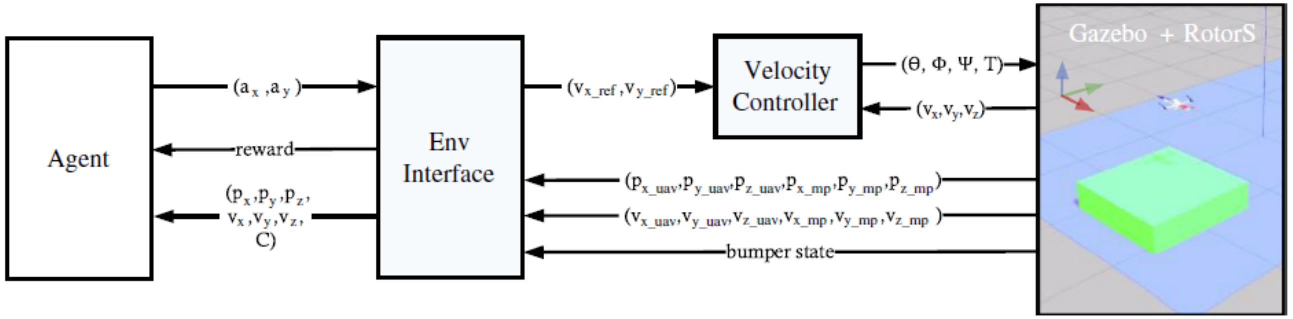
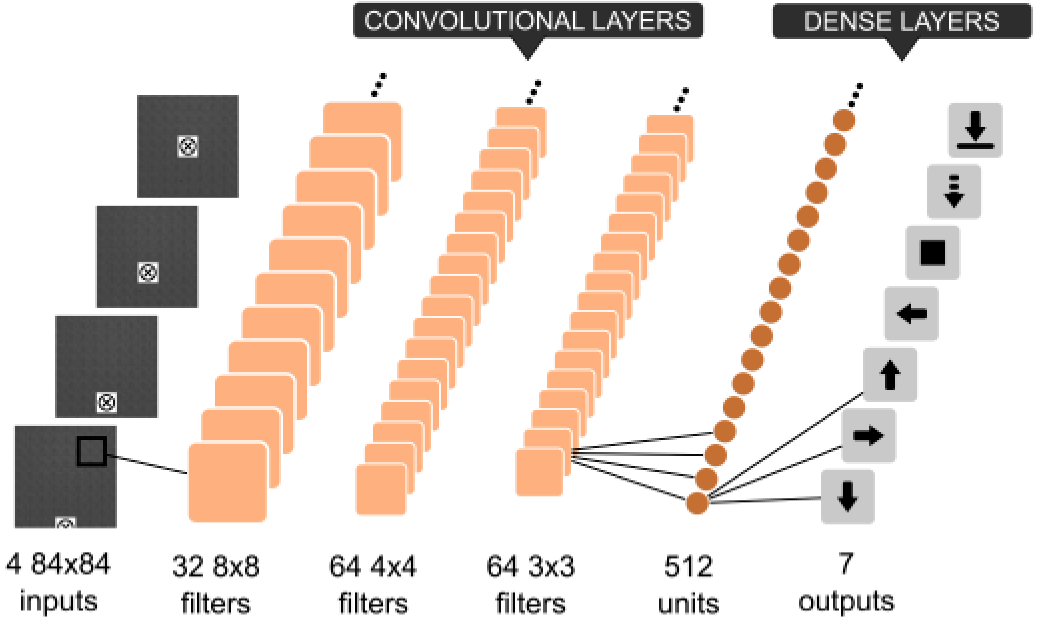


Fig. 2 Architecture of our proposed reinforcement learning framework for the case of the experiment of study



THANKS YOU!