Deep Learning applied to Robotic

Flavien Ronteix–Jacquet
TalTech, Robotic 2018, Estonia

December 19, 2018

Abstract

A revolution in Artificial Intelligent is in Progress. The name of this revolution is Deep Learning. Since a decade, some enablers allowed its fast development. Robotic takes advantage of Deep Learning methods to become smarter to solve some challenges. We found out there are 7 big challenges of AI in Robotic, New dynamic, object recognition, sensor fusion, advanced task planning, learning control policies, advanced manipulation and human-robot interactions. After, a practical approach with 2 examples is presented. Finally, a discussion about limits and future of Deep Learning in robotic.

Contents

1 Introduction

Since 2014, several articles were published on this subject of Deep Learning. Indeed, since 2014, this (not new) method in Artificial intelligence and moreover in Machine Learning, exploded and changed lot of things in many topics. Robotics is not spared by this new revolution, the 4th revolution of AI.

Robots need artificial intelligence. They need it to solve many problems. The first we can imagine is for autonomy. When it is difficult to control remotely the robot, it could be interesting to give it some autonomy to avoid obstacles and take quick decisions. Because of that, since a long time ago, the links between robotic development and AI development are very strong.

According to the father of Artificial Intelligence, John McCarthy, Artificial intelligence is "The science and engineering of making intelligent machines, especially intelligent computer programs". For me what makes something intelligent is its ability to adapt to generic and new problems, and learn from previous experiences.

The new developments of Deep Learning give strong answer to this problems of adaptation and learning.
In a first part, we will present some basis of Deep Learning. How it works, history, different structure,... In a second part, how we can apply this technology for robotics issues. After, two examples of Deep Learning implementations in robots. Finally, a discussion about limits and future improvements.

2 Basics of Deep Learning

Deep Learning is a subtopic of Machine learning that is also a subtopic of Artificial Intelligence. Artificial Intelligence is huge topic and exists since the beginning of informatics in 1950 with Alan Turing (also known as the father of computer) and the paper "Computing machinery and intelligence"[6]. Machine learning appeared in 1957 with the first learning machine, the perceptron (Figure 1). The perceptron algorithm is really simple and try to classify input on 2 classes.

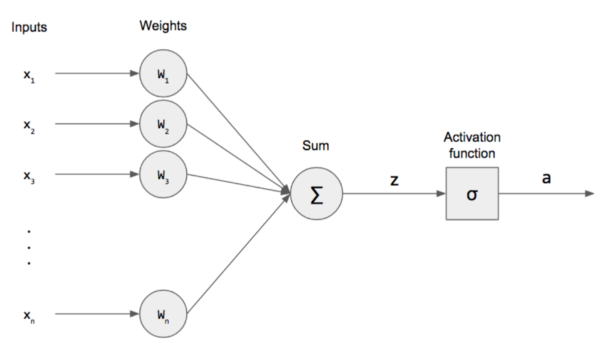

Figure 1: Perceptron

We feed the inputs x1..xn, apply weights, sum the result and apply an activation function (could be a simple step function). The training phase consist in minimizing error in fine-tuning weights.

Quickly, the researchers tried to combine several perceptrons in parallel to create a "layer of perceptron" and after combined layers to create "multi-layer" perceptrons, a deep neural network (Figure 2). We called single-perceptron, neurons because they work a little like neurons in our brain. The problem was to train this structure. Which algorithm ? How to differentiate error from a layer of neurons an error of individual neurons in a layer ? A first answer came in 1980 with the first training method called "back propagation". This method marked the beginning of the deep learning revolution (deep because we have some "hidden" layers).

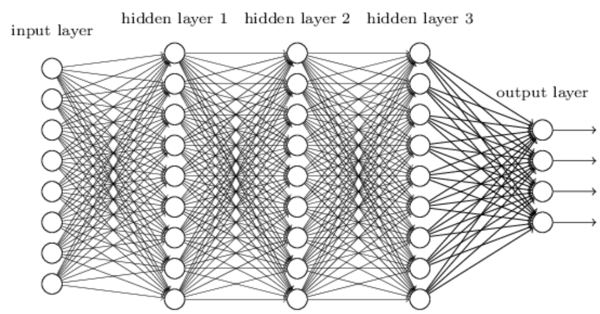

Figure 2: Deep Neural Network structure

However, these neural networks could have only few layers and a limited amount of neurons because of the problem of vanishing gradients. This problem reduces the ability to learn and make training exponentially longer when the neural network grows.

In the 2000s and especially since 2014, Deep Learning exploded and spread among all the subjects of computer sciences, engineering and more. This fast development of Deep learning comes from 3 enablers:

1. *New algorithm* training algorithm and model to overcome the training problems and allow uses of Deep Learning in wide range of topic
2. New and powerful *hardware* like graphic card with mass parallelization. Indeed, compute an inference or a training phase, it is only multiply, inverse and combine huge matrix. GPU with thousand of simple core designed for matrix calculation are very powerful to accomplish this task.
3. To train a model to do something, we need *lot of data.* Big Data with huge mass of data gives to AI everything is needed to learn.

In fact, how it works ?

First, we need to define a structure for the neural network. How many layers, how many neurons, what kind of neurons, which activation function,...

After we need to define how to train this model, the training algorithm, the error function,...

To train a neural network, the most important things is the dataset. This dataset need to contain lot of examples with the desired output in the case of supervised learning (in opposite of unsupervised learning where the AI will try to find clusters, group of responses). This dataset is important because to have efficient learning and a good generalization, the number and the diversity of input will impact a lot.

The training phase consists on feeding neural network, calculate error and modify parameters (weights) to fit more. This phase is critical because if there is an overfitting (the model fits to much the training dataset) or an underfitting (the model didn't have enough data and then had a problem of generalization), then the model is useless to take decision or extract informations.

The implementation is commonly in two phases. First a training on big computer and then inference (prediction) locally on the robot.

What is used in Robotic ? 5 structures :

Function approximator (Figure 3)

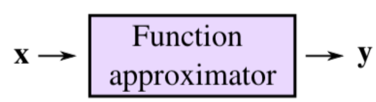

Figure 3: function approximator

Uses for regressing an arbitrary function. Cross-entropy is commonly used to calculate error. We can use SGD (Stochastic Gradient Descent), RMSProp or Adam as optimization methods. For a given input x (a vector commonly), the model predict an output y (also a vector).

Autoencoder (Figure 4)

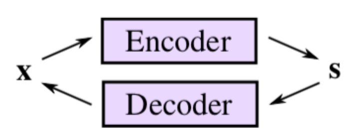

Figure 4: Autoencoder

This structure is used for "unsupervised learning" then a learning where we give only inputs. With the encoder part, the model will try to understand the input and encode it, we can call it "internal representation" of the input. After we have a decoder that will try to reconstruct the input from the internal representation. The idea is to extract informations from the input (that humans can't understand) and reduce the size of them. This informations will be used by a decoder or by an other AI's part.

Convolutionnal Neural Network (Figure 5)

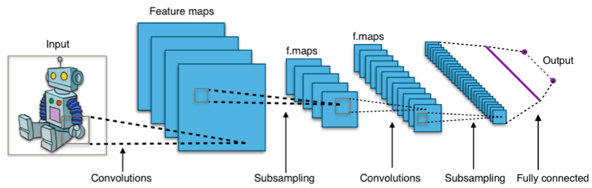

Figure 5: Typical Deep Convolutionnal Neural Network

This neural network looks like the function approximates seen before with multilayer neurons but with a convolution system. Mostly used in computer vision.

A convolutional layer is a neurons layer that uses convolution to link the previous layer with the next layer. Convolution is a dot product between 2 matrix, the previous layer and the filter that produces an activation map. For DCNN, we learn the coefficient of filters.

Recurrent neural network or RNN (Figure 6)

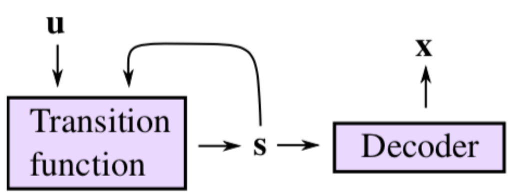

Figure 6: Recurrent Neural Network

The structures previously presented had one problem. It's only state-less machine. It means the current input doesn't depend on the previous output (no correlations between input xn and xn+1). Recurrent neural network take as input the internal state (previous input) and the current input. It allows modeling dynamic systems, and robots are dynamic systems ! The transition function approximates how the input will affect state over the time. The internal state is almost the same as for auto encoder then we can use it for another task.

Control policy (Figure 7)

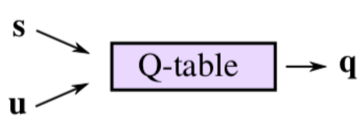

Figure 7: Deep Control Policy

Also named Q-learning, not specific to Deep Learning but for Q-table we can use a DNN. DNN evaluates the utility or quality q of potential control vector. This method could be used when the objective of task is known but the human don't how to solve it. Then, by rewarding robot and train a function of rewarding, the robot will try to solve the task. It is the reinforcement learning. Interesting when we have a score but we don't have the rules.

With this five structures, we have an overview of how to insert Deep Learning into robots. In the next section, robotic challenge that Deep learning can solve.

3 Applying Deep Learning in Robotics

There is a lot of robot type. From industrial welding robots to autonomous cars passing by the receptionist robots. All of them became more and more intelligent since few years. Autonomous cars (also robot) are only possible with intelligent hardware.

The "Deep learning in robotics" find **seven robotic challenges** that can be solved by Deep Learning. Dynamic, object recognition, sensor fusion, planning, control, manipulation, human actions. For each, we give examples and state-of-the-art of the research.

3.1 Novel dynamics

A simple definition of robot is a combination of mechanic, electronic and informatics to accomplish a task in an environment. The problem is the environment can be difficult and will be dynamic. This is especially the case of autonomous moving robot with a dynamic and complex "real" environment.

Commonly, we give to the robot some hand-made model to adapt but it requires a human experience, it is time consuming. Also, making such models robust to uncertainty is difficult and full state information is often unknown.

This facts give a motivation to use Deep Learning for this first challenge because different model based on Deep Learning as the property to be dynamic and quite easy to train and implement. We can easily find examples of Deep Learning model applied to this challenge in the research.

1. Deep learning helicopter dynamic model [7]
2. Perceiving physical object properties by integrating a physics engine with deep learning [8]
3. Deepmpc: learning deep latent features for model predictive control [9]
4. Continuous control with deep reinforcement learning [10] or Deep-Q learning

3.2 Object recognition

This is my main research subject for 2 years and my main motivation to apply Deep Learning to Robotic.

Humans observe the world through their eyes. Then it is normal to want to give robots the opportunity to "see" the world.

The first used Deep Learning model in industry was the "LeNet" from Y. Lecun [11] to read hand-written numbers on cheque. Since Deep Learning shown its amazing capacity to understand images particularly a structure of Neural Network, the Deep Convolutionnal Neural Network. We already present this structure of neural network.

There is lot of models that produce different output. The challenges in computer vision are:

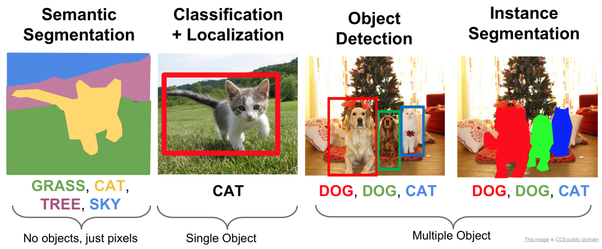

Figure 8: Challenges of Deep Learning in Computer vision

For autonomous car for example, we want to have segmented scene to decide where is the road, where are the obstacles, the humans?... Because there is a plenty of models, I decided, arbitrary to present a recent one and one of my favorite for the accuracy of its results, RefineNet with Mobilenet feature extractor (presented in Figure [**?**].

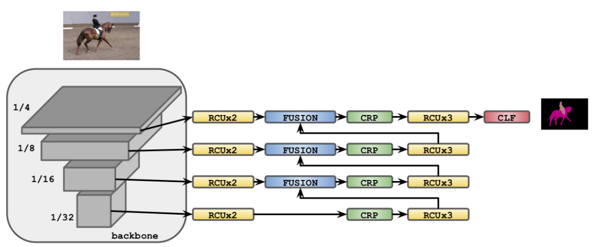

Figure 9: Model structure of RefineNet for Real-time semantic segmentation

There is lot of model because not only for robotic. How to recognize object and segment a scene. We choose the state-of-the-art in realtime image segmentation RefineNet with MobileNet [12].

The problem here is the problem of dynamic. The detection doesn't depend on previous frames, they are totally independent. Because of that, we can have some pixels that are detected as road that becomes tree on the next frame and conversely. It is why research try to use video instead of frames and combine RNN with DCNN like in this publication.

Object detection in video sequence [13]

But for now, the most effective methods remain DCNN methods.

An other problem of DCNN is the requirements in term of computation and memory. For example, the best version of RefineNet we studied before, needs around 50 Billions FLOPs (Floating Operation), a Raspberry Pi 3 has a computation capabilities of 2.7GFlops (4 cores). It means that a little robot based on Raspberry pi board can only process one image every 5 seconds in the best case, impossible for real-time ! It is for this reason that autonomous cars have big graphic cards to perform computation in real-time (Nvidia Pascal 20TFLOPS for Tesla autopilot).

3.3 Sensor fusion

We talk before about autonomous cars, this challenge of sensor fusion is particularly important in this case because of **the numbers of sensors** and the variety of them :

1. GPS (Global Positioning System) : Get the location on the earth
2. Radar (RAdio Detection And Ranging) : Radar is useful for detecting obstacles, vehicles and pedestrians around a vehicle tracking multiple targets at once is a primary use for an automotive radar
3. Camera : to "see" the environment, we already study this before.
4. LiDAR (Light Detection and Ranging) : is a method that uses light in the form of a pulsed laser to measure distance to an object based on signal time of flight. LiDAR is useful for perceiving surroundings when 3-dimensional, high-resolution, light-independent.
5. UltraSonar : detect near obstacles with ultrasounds.

These sensors are also used for "normal" robot. In 2002, Elmenreich [19] found following **benefits** of sensor fusion over single source:

1. Reliability
2. Extended spatial coverage
3. Extended temporal coverage
4. Increased Confidence
5. Reduced Uncertainty
6. Robustness against noise
7. Increased Resolution

The problem we see when we have some sensors is how we can combine them to have a result better than the sum of each sensor. Deep Learning make this sensor fusion more easy because the model will learn the best way to combine sensors' data and extract informations. In this paper [18] Bohez and al. proposed a method to use Deep reinforcement Learning for controlling robot by sensor fusion.

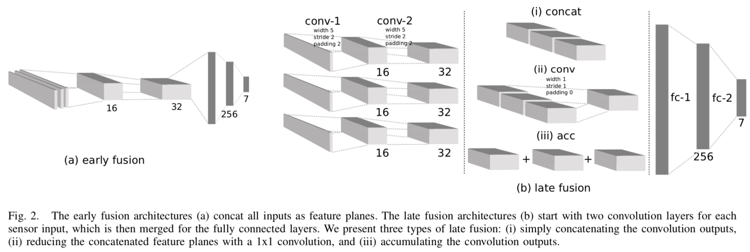

Figure 10: Sensor fusion with Deep Learning

By using an encoder structure, we can also reduce the space needed for informations. From dozens of sensor with huge bandwidth of data, we can have after the Deep learning model less data but with useful informations to take decisions as presented in "DeepSense: a Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing" [20].

3.4 Advanced task planning

The previous challenges give to the robot the ability to understand its environment, now it can take decision to accomplish its goals. First, how to plan tasks ?

With MPNet (Motion Planning Network, Figure 11), Qureshi and al. [14] proposed a solution of Fast and efficient motion planning algorithm for many robotics applications such as self-driving cars.

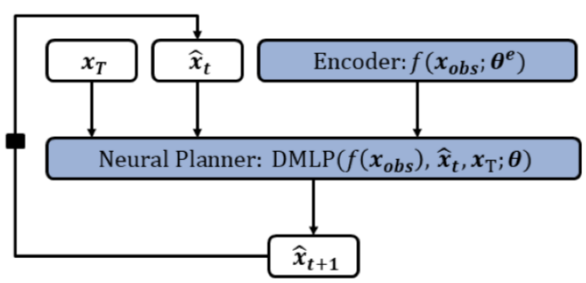

Figure 11: Motion Planning network

The idea is simple. From the result of an encoder that plays the role of sensor fusion system, they take value of goal and current computed output. They put all these informations on a Deep Neural Network, the Neural Planner and after we have a new planned tasks.

This paper [15] proposed an end-to-end deep visuomotor from the object recognition to motor control using a Deep convolutional neural network and a policy search method for a reinforcement learning part.

3.5 Learning control policies

As for high-level task planning, we can learn to control motors, and different part of robots, it is called control policies challenge.

For this problem, it is possible to use Q-learning or DNN function approximator. In the paper published in 2015 by Polydoros and al. [21] during the International conference on intelligent robots ans systems.

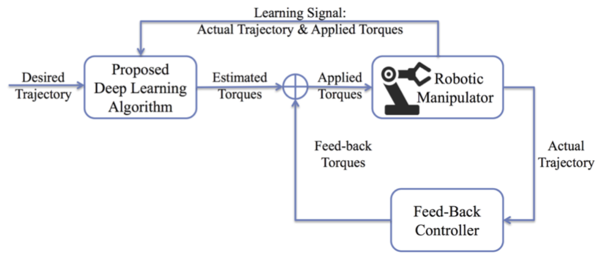

Figure 12: Deep learning for manipulator

In Figure 12 the Deep Learning model help to control manipulator based on the goal, the current state and what the model learned from previous experience (during the training). Because of this method, no need anymore an expert to develop complex system, the model will learn by itself during the training how to respond according to goals and current state.

[22] proposed a solution to control a robotic arm using Deep Q-learning policy.

3.6 Advanced manipulation

The last paper present a solution to control robotic arm more smartly. The manipulation is also a challenge for Deep Learning applied to robotic. The idea in "smart manipulation" is to achieving a "synergy" in cognitive behavior of humananoids, that is the title of this paper "Achieving ?synergy? in cognitive behavior of humanoids via deep learning of dynamic visuo-motor-attentional coordination" [16]

It exists hundreds of video that show "intelligent" arm on internet to sort objects.

3.7 Human-Robot interactions

The last but not least challenge in intelligent robotic is the Human-robot interaction. In the next year, there will have more and more robot in direct contact with human. Thus, it is necessary to implement in robots some mechanisms to understand and interact with humans.

At the beginning of an interaction between two humans, there is an evaluation of the sentiment. The smile, the eyes,... and after there is a conversation. Therefore, A robot need to understand the human sentiments and be able to talk with them according to this sentiment and the context. A paper try to solve this big challenge, "Deep Learning and Sentiment Analysis for Human-Robot Interaction" from Cagliari university published by Springer during the ESWC 2018.

Deep Learning and Sentiment Analysis for Human-Robot Interaction [17]

This challenge is so important, not only for robot, that lot of AI API providers propose sentiment analysis system like IBM with IBM Watson or Amazon cloud service. Then we can use this systems to robot today.

During this part, we saw the seven challenges of robotic that Deep Learning can solve. There is already lot of solution especially from few years. This revolution is not finished and we will see more and more publication and project about this subject, making robotic more intelligent. Now we will see how to implement Deep Learning in Real life with two examples, a module for ROS and a project.

4 Real implementation in ROS and a super-intelligent robot

4.1 Deep Learning and ROS

ROS (Robot Operating System) is a robotics middleware that aims to simplify the task of creating complex and robust robot behavior. It is normal then to have some nodes and packages developed by the community to implement Deep Learning models. The node we test is dnn_detect (http://wiki.ros.org/dnn_detect) and is used for object recognition based on Deep Learning model.

The configuration for the node we used is

```
<node pkg="dnn_detect" name="dnn_detect"
   type="dnn_detect" output="screen" respawn="false">
   <param name="image_transport" value="compressed"/>
   <param name="publish_images" value="true" />
   <param name="data_dir" value="model"/>
   <remap from="/camera/compressed"
        to="camera/image/compressed"/>
   <remap from="/camera_info" to="camera/camera_info"/>
 </node>
```

With the model SSD-mobilenet for object recognition because this is the best model currently. In the simulation environment the robot only gives in the console what it saw.

4.2 Example of robotic project to implement Deep Learning

Since 6 months now, I work with 3 others student of my home university in a project of super-intelligent robot (Figure 14). The idea is to apply as much as possible Deep Learning to create a smart robot. We work on this to learn more about Deep Learning and Robotic but also to show that this new technology is a revolution for robotic.

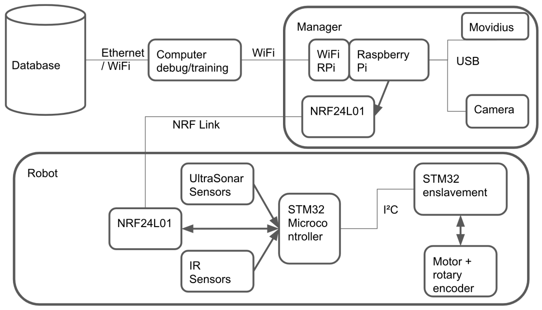

Figure 13: Functionnal architecture of the robot

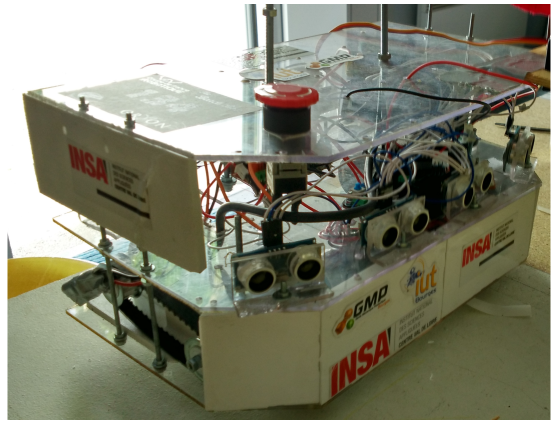

Figure 14: Picture of the robot

In fact what we made so far and what is the plan (Figure 13 for the current functional architecture).

1. Implement a voice recognition to set robot's goal remotely
2. Use object recognition for obstacle avoidance and learn more about the environment
3. Sensor fusion between cloud point from kinect and ultrasonic
4. Compute path with Deep Q-learning

5 Limits and future

The limits are numerous because as we saw before, Deep Learning is not a new topic but there is an explosion of methods since a couple of years and then it is difficult to have a perfect method to replace "traditional" methods. Some roboticians are afraid by uncertainty introduce by unproveable model of Deep Learning. But it is a big potential for the new years.

An other limits is the amount of data needed to have a good model and the training methods. Big Data is an enabler but this technology needs more data especially in the case of robotic. The training methods needs also to be improved especially for complicated models like RNN.

Finally, there is the problem of hardware limitation. Autonomous cars need a chest full of computer to have the ability to see, take decision,... in realtime. For the moment, it is impossible for most of "little" robot. Specific hardware need to be develop but this is an interesting opportunity. Intel and many others develop hardware like "Neural Compute Stick" for this purpose.

6 Conclusion

Deep Learning is a technological revolution in progress. We can apply many model of Deep Learning to some important Robotic challenges. We can use it today for object recognition or sensor fusion but that is really interesting with this technology is that in the next few years, new hardware, new method, new data will make robots smarter, more adaptive and more collaborative with humans. This is why I think it is important to be aware about this field of research in artificial intelligence for roboticians.

References

[1] Harry A. Pierson & Michael S. Gashler, *Deep learning in robotics: a review of recent research*, Advanced Robotics, 2017.

[2] Niko Snderhauf et al., *The Limits and Potentials of Deep Learning for Robotics*, 2018.

[3] Sergey Levine, *Deep learning in robotics*, podcast, 2018.

[4] Ian Lenz, *Deep Learning for robotics*, Cornell University, 2016.

[5] Javier A. Perez et al., *Artificial Intelligence and Robotics*, UKRAS, 2018.

[6] Alan Turing, *Computing machinery and intelligence*, Mind, 1950.

[7] Ali Punjani & Pieter Abbeel, *Deep Learning helicopter dynamics models*, ICRA, 2015.

[8] Jiahun Wu et al., *Galileo: Perceiving physical Object Properties by integrating a physics engine with Deep Learning*, MIT, 2016.

[9] Ian Lenz et al., *DeepMPC: Learning Deep latent feature for model predictive control*, Cornell University, 2015.

[10] Timothy P. Lillicrap et al., *Continuous control with deep reinforcement learning*, Google Deepmind, 2016.

[11] Yann LeCun et al., *Gradient-based learning applied to document recognition*, IEEE, 1998.

[12] Vladimir Nekrasov et al., *Light-Weight RefineNet for Real-Time Semantic Segmentation*, University of Adelaide, 2018.

[13] Joonas Haapala, *Recurrent neural networks for object detection in video sequences*, Aalto University, 2017.

[14] Ahmed H. Qureshi et al., *Motion planning network*, University of California, 2018.

[15] Serguey Levine, Chelsea Finn et al., *End-to-end training of deep visuomotor policies*, University of California, 2016.

[16] Jungsik Hwang et al,. *Achieving "Synergy" in Cognitive Behavior of Humanoids via Deep Learning of Dynamic Visuo-Motor- Attentional Coordination*, Korea advanced institute of Science and Technology, 2015.

[17] Mattia Atzeni and Diego R. Recupero, *Deep Learning and sentiment analysis for Human-Robot interaction*, University of Cagliari, The semantic Web : ESWC 2018 Satellite events, 2018.

[18] Shaun M. Howard, *Deep Learning for Sensor fusion*, case wester reserve university, 2017.

[19] Wilfried Elmenreich, *An Introduction to Sensor Fusion*, Vienna university of technology, 2002.

[20] Shuochao Yao et al., *DeepSense: a Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing*, University of Illinois, 2017.

[21] Athanasios S. Polydoros et al., *Real-time Deep learning of robotic manipulator inverse dynamic*, IROS, 2015.

[22] Stephen James & Edward Johns, *3D Simulation for Robot Arm Control with Deep Q-Learning*, Imperial College London, 2016.