Abstract

**Deep learning** is recently showing outstanding results for solving a wide variety of robotic tasks in the areas of perception, planning, localization, and control. Its excellent capabilities for learning representations from the complex data acquired in real environments make it extremely suitable for many kinds of autonomous robotic applications. **In parallel, Unmanned Aerial Vehicles (UAVs) are currently being extensively applied for several types of civilian tasks in applications going from security, surveillance, and disaster rescue to parcel delivery or warehouse management.** In this paper, a thorough review has been performed on recent reported uses and applications of deep learning for UAVs, including the most relevant developments as well as their performances and limitations. In addition, a detailed explanation of the main deep learning techniques is provided. We conclude with a description of the main challenges for the application of deep learning for UAV-based solutions.

1. Introduction

Recent successes of deep learning techniques in solving many complex tasks by learning from raw sensor data have created a lot of excitement in the research community. However, deep learning is not a recent technology. It started being used back in 1971, when Ivakhnenko [1] trained an 8-layer neural network using the Group Method of Data Handling (GMDH) algorithm. The term deep learning began to be used during the 2000s, when Convolutional Neural Networks (CNNs), a computational original model from the 80s [2] but trained efficiently in the 90s [3], were able to provide decent results in visual object recognition tasks. At the time, datasets were small and computers were not powerful enough, so the performance was often similar to or worse than that of classical Computer Vision algorithms.

The development of CUDA for Nvidia GPUs which enabled over 1000 GFLOPS per second and the publication of the ImageNet dataset, with 1.2 million images classified in 1000 categories [4], were important facts for the popularization of CNNs with several layers ( to  connections and  to  parameters). These deep models show great performance not only in Computer Vision tasks but also in other tasks such as speech recognition, signal processing, and natural language processing [5]. More details about recent advances in deep learning can be found in [6, 7].

An evidence of the suitability of deep learning for many kinds of autonomous

communication system are responsible for establishing an adequate communication with human operators and/or other robots

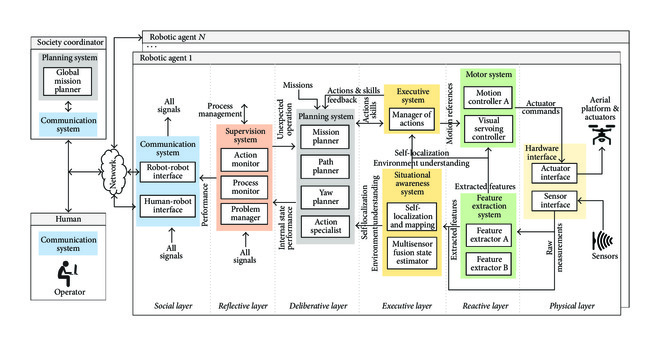


Figure 1

Aerostack architecture, consisting of a layered structure, corresponding to the different abstraction levels in an unmanned aerial robotic system. The architecture has been applied here to systematically classify deep learning-based algorithms available in the state of the art which have been deployed for applications with Unmanned Aerial Vehicles.

The remainder of this paper is as follows: firstly, Section [2](https://www.hindawi.com/journals/js/2017/3296874/#sec2) covers a description of the currently relevant and prominent deep learning algorithms. For the sake of completeness, deep learning algorithms have been included regardless of their direct use in UAV applications. Section [3](https://www.hindawi.com/journals/js/2017/3296874/#sec3) presents the state of the art in deep learning for feature extraction in UAV applications. Section [4](https://www.hindawi.com/journals/js/2017/3296874/#sec4) surveys UAV applications of deep learning for the development of components of planning and situation awareness systems. Reported applications of deep learning for motion control in UAVs are presented in Section [5](https://www.hindawi.com/journals/js/2017/3296874/#sec5). Finally, a discussion of the main challenges for the application of deep learning for UAVs is covered in Section [6](https://www.hindawi.com/journals/js/2017/3296874/#sec6).

2. Deep Learning in the Context of Machine Learning

Machine Learning is a capability enabling Artificial Intelligence (AI) systems to learn from data. A good definition for what learning involves is the following: “a computer program is said to learn from experience E with respect to some class

layer  given its  inputs (outputs of the previous layer ) is given by the following equation:

During the process of learning, the weights in each unit are updated using backpropagation in order to optimize a cost function, which generally indicates the similarity between the desired outputs and the actual ones.

Convolutional Neural Networks (CNNs), depicted in Figure [2](https://www.hindawi.com/journals/js/2017/3296874/fig2/), are a specific type of models conceived to accept 2-dimensional input data, such as images or time series data. These models take their name from the mathematical linear operation of convolution which is always present in at least one of the layers of the network. The most typical convolution operation used in deep learning is 2D convolution of a 2-dimensional image  with a 2-dimensional kernel , given by the following equation:

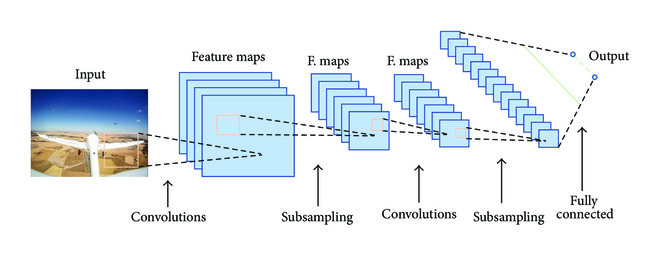
[](https://www.hindawi.com/journals/js/2017/3296874/fig2/)

Figure 2

A generic example of a Convolutional Neural Network model. The usual architecture alternates convolution and subsampling layers. Fully connected neurons are used in the last layers.

The output of the convolution operation is usually run through a nonlinear activation function and then further modified by means of a pooling function, which replaces the output in a certain location with a value obtained from nearby outputs. This pooling function helps make the representation learned invariant to small translations of the input and performs subsampling of the input data. The most common pooling function is max pooling, which replaces the output with the maximum activation within a rectangular neighborhood. Convolution and pooling layers are stacked together to achieve feature learning in a hierarchical way. For example, when learning from images, layers closer to the input learn low-level feature representations (i.e., edges and corners) and those closer to the output

immeasurably large or extremely small. These problems are referred to as exploding gradients and vanishing gradients, respectively. Exploding gradients are easier to solve, as they can be truncated or squashed, whereas vanishing gradients can become too small for networks to learn from and for the resolution of a computer to enable its representation.

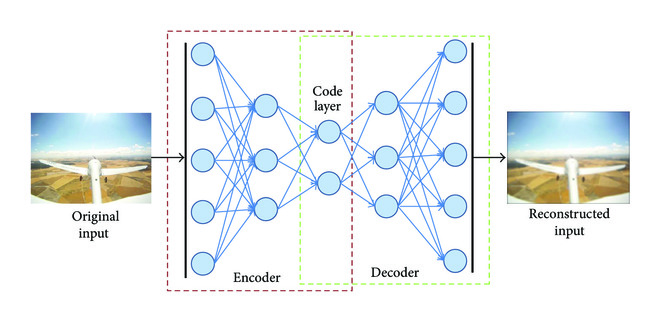
Long Short-Term Memory (LSTM) models are a type of RNN architecture proposed in 1997 by Hochreiter and Schmidhuber [17] which successfully overcomes the problem of vanishing gradients by maintaining a more constant error through the use of gated cells, which effectively allow for continuous learning over a larger number of time steps. A typical LSTM cell is depicted in Figure [3](https://www.hindawi.com/journals/js/2017/3296874/fig3/). The input, output, and forget gate vector activations in a standard LSTM are given as follows:

Figure 3

A long-short term memory model, adapted from the original figure in [14]. Learned weights control how data enter and leave and are deleted through the use of gates.

The cell state vector activation is given by the following equation:where  represents the Hadamard product. Finally, the output gate vector activation is given by the following equation:

As it has been already stated, LSTM gated cells in RNNs have internal recurrence, besides the outer recurrence of RNNs. Cells store an internal state, which can be written to and read from them. There are gates controlling how data enter and leave and are deleted from this cell state. Those gates act on the signals they receive, and, similar to a standard neural network, they block or pass on information based on its strength and importance using their own sets of weights. Those weights, as the weights that modulate input and hidden states, are adjusted via the recurrent network’s learning process. The cells learn when to allow data to enter and leave or be deleted through the iterative process of making guesses, backpropagating error, and adjusting weights via gradient descent. This type of model architecture allows successful learning from long sequences, helping to capture diverse time scales and remote dependencies. Practical aspects on the use of LSTMs and other deep learning architectures can

[](https://www.hindawi.com/journals/js/2017/3296874/fig4/" \t "https://www.hindawi.com/journals/js/2017/3296874/_blank)