Application of Deep Learning Technique in UAV's Search and Rescue Operations: Proceedings of the 2018 Intelligent Systems Conference (IntelliSys) Volume 1

Chapter · January 2019

DOI: 10.1007/978-3-030-01054-6_62

4 authors:

Kyaw Min Naing
University of Wolverhampton

Ahmad Zakeri
University of Wolverhampton

Oliver Iliev
FON University

Navya Venkateshaiah
University of Wolverhampton

Intelligent Systems Conference 2018
6-7 September 2018 | London, UK

Application of Deep Learning Technique in UAV's Search and Rescue Operations

Kyaw Min Naing, Ahmad Zakeri
School of Engineering
University of Wolverhampton, UK
A.Zakeri@wlv.ac.uk

Oliver Iliev
Faculty of Information and Automation
FON University
Macedonia
oliver.iliev@fon.edu.m

Navya Venkateshaiah
School of Engineering
University of Wolverhampton, UK
nav.venkateshaiah@gmail.com

***Abstract*—This paper is concerned with the application of Deep Learning techniques for analyzing image data for search and rescue operations of Unmanned Aerial Vehicle (UAV). It uses Keras and its Tensorflow backend to model a deep Convolutional Neural Network (CNN) Learning technique and train the model with MNIST digits dataset to predict the hand-written word from the image data received from the ground level. The paper explains the stages involved in the implementation of LeNet method of Deep Learning techniques for developing a classifier for long distance recognition of handwritten words.**

***Keywords*—Deep neural network; deep learning; convolutional neural network; MNIST; LeNet; image processing; Unmanned Aerial Vehicle; Tensorflow; Keras**

I. Introduction

Unmanned Aerial Vehicles (UAVs) are currently being extensively applied in civilian tasks [5] such as security, surveillance, and disaster rescue. The use of Unmanned Aerial Vehicle in search and rescue activities in particular has increased dramatically over the past three years in different areas/fields such as marines, mountains, wild environments [6]-[10]. The main purpose of a search and rescue (SAR) operation is to identify and rescue the target in the shortest possible time, and this is critical as any delay will possibly reduce the chances of survival of the victims [14]. For effective and quick identification, and in particular, for long distance recognition of images, UAVs need to be equipped with intelligent devices and systems that can help them quickly recognise the images from long distances [2].

Image recognition or object recognition as the process of identifying and detecting an object or a feature in a digital image, has been the subject of research in many field of engineering and science, with many types of algorithms being developed to facilitate these processes, for example, image recognition algorithms include Optical character recognition, or Object recognition algorithms rely on matching, learning, or pattern recognition algorithms using appearance-based or feature-based techniques.

Current development in image recognition or processing involves object detection with trained data set. It is called machine learning (ML), it enables Artificial Intelligence (AI) systems to learn from data. Machine Learning can now be part of image processing [13]. The currently available Machine Learning algorithms can facilitate supervised, unsupervised and reinforcement learning. In supervised learning, algorithms are presented with a dataset containing a collection of features. Additionally, labels or target values are provided for each sample. This mapping of features to labels of target values is where the knowledge is encoded. Once it has learned, the algorithm is expected to find the mapping from the features of unseen samples to their correct labels or target values. Deep learning algorithms are a subset of Machine Learning algorithms that typically involve learning representations at different hierarchy levels to enable building complex concepts out of simpler ones.

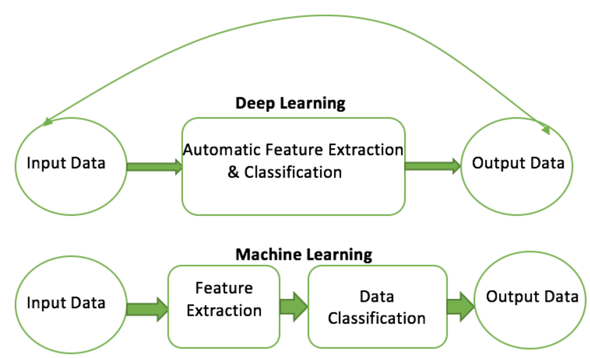

Fig. 1. Deep learning vs Machine learning.

Deep learning is recently showing outstanding results for solving a wide variety of robotic tasks in the areas of perception, planning, localization, and control [12]. Its excellent capabilities for learning representations from the complex data acquired in real environments make it extremely suitable for image processing applications. Fig. 1 compares the working of Deep Learning vs Machine Learning.

As shown in Fig. 1, the traditional Machine Learning approaches worked in a way that required first design of a feature extraction algorithm, which generally involved a lot of heavy mathematics (complex design), and didn't perform too well at all (accuracy level just wasn't suitable for real-world applications). After doing all of that you would also have to design a whole classification model to classify your

978-1-5386-4185-9/18/$31.00 ©2018 IEEE

input given the extracted features. With deep networks, we can perform feature extraction and classification in one shot, which means we only have to design one model. Fig. 2 compares the performance of Deep Learning vs Machine Learning. When the amount of training data increases in machine learning, the performance of ML decreases but the Deep learning get the maximum performance with huge amount of data with the same situation.

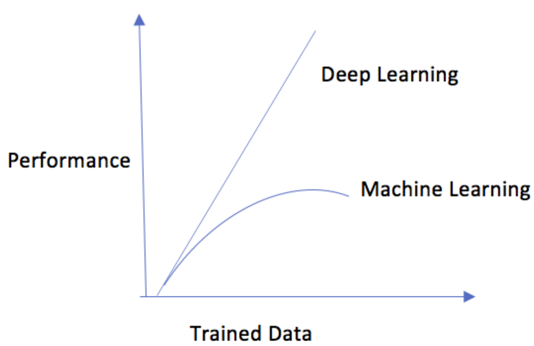

Fig. 2. Performance comparing of two models.

II. Deep Convolutional Neural Networks (CNNs) and the Training (Learning) Process

There are several deep learning technologies available in supervised learning; the most relevant algorithms nowadays in supervised learning are:

- Feedforward Neural Networks, a popular variation of these called Convolutional Neural Networks (CNNs),
- Recurrent Neural Networks (RNNs), and
- a variation of RNNs called Long Short-Term Memory (LSTM) models.

Feedforward Neural Networks, also known as Multilayer Perceptrons (MLPs) are the most common supervised learning models. Their purpose is to work as function approximators: given a sample vector with features, a trained algorithm is expected to produce an output value or classification category that is consistent with the mapping of inputs and outputs provided in the training set. The approximated function is usually built by stacking together several hidden layers that are activated in chain to obtain the desired output. The number of hidden layers is usually referred to as the depth of the model, which explains the origin of the term deep learning.

This paper uses Keras and Tensorflow backend platform in python to design and train a designated deep Convolutional Neural Network architecture with the given sample image data set, and then applies the trained network to new image data taken by UAV camera for identification and recognition of the image.

III. Modelling Deep Learning with Tensorflow

Tensorflow software library help tackle implementation of machine learning and deep learning methods. These libraries are very broad enough to implement many types of methods and algorithms which are included seeding, loading data, designing neural network architecture, compiling and training of the machine learning or deep learning model. The technique has been used in areas such as robotics, voice recognition devices, map extracting, object detection [15], etc. [1]

In addition, the Tensorflow backend called Keras [3] is also used for detecting hand written digits or MNIST data. The Convolutional neural network (CNN) [11] is used to determine MNIST image into words by transforming multi-Dimensional images into characters [4]. Open source Tensorflow made by Google team to model CNN with MNIST digits. The reason of using MNIST digits for modelling deep learning is because a person called Yann LeCunn already released hand written datasets in 1998 which then main source for data scientists and researchers for classifying objects in Deep Learning.

IV. Modelling and Simulating Deep Convolutional Neural Network using Tensorflow and Keras

A deep convolutional neural network is built in this paper using Keras with Tensorflow backend library tool for modelling and simulating of hand written digits by training with MNIST dataset sample image data of hand written images. The flow diagram for the proposed model system is shown in Fig. 3.

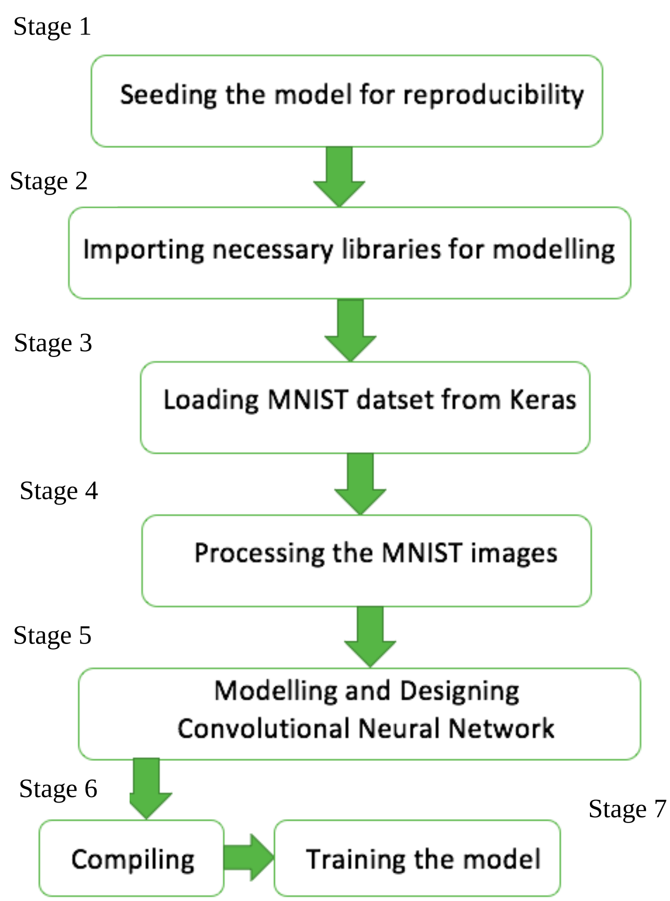

Fig. 3. Model system approach flow diagram.

The modelling process for testing and training include seven stages described as follows:

A. *Stage-1*

In the first stage, Neural network may not be predicted precisely because of instability of the algorithms being trained with same dataset with unusual results. To overcome this complexity, random seeding to start the system with random weights applied on the neurons respectively. This cause the system to learn automatically each time if the identical neural network train with identical result.

B. *Stage-2*

Importing Keras library to use all the necessary libraries from that framework to model and simulate the MNIST datasets. Given dataset can be loaded into the system to classify hand written images already uploaded in the Keras. After that, a sequential model needed to perform and design neurons by stacking layers of that neurons using Dense layer which implement the operation of the neural network. The system can cause overfitting by using complicated neurons in the model. To avoid that overfitting neural layer was used using with the Dropout layer which set a fraction rate of input to 0 at each update during training and flatten the input using Flatten layers. To create successful deep convolution neural network for 2-Dimensional images, Maxpooling2D layer was used to operate spatial data of each pixel of the image which presents same pixel detection to get accurate output.

C. *Stage-3 & 4*

In this stage, loading the hand-written images from the dataset stored. For testing and training the data, it included in the dataset are images with its respective labels. When training the input data, the total of digits is 60000 and only 10000 are being tested for training and testing the data. The value of each pixel is 255 pixels in processing the MNIST images.

D. *Stage-5*

- Using Sequential model to initialise and stack all the neural layers.
- 2-Demensional convolutional layer uses lenet-5 have 32 neurons which is the first convolutional layers. The size of the pool size is 3 by 3 kernel size pixels. As with the Dense layer, a recommended neuron called Relu should be activated. MNIST digits input is (28 by 28) pixels with 1 depth black and white image.
- Second 2-D Convolutional layer, on the other hand uses 64 layers and remaining values are same as first layer.
- Third layer is 2-D MaxPooling, for every 4 pixels reduce them to one with 2 by 2 pool size.
- Dropout layer produce results from training apply well to the validation data, going to dropout 0.25 of the neuron.
- We have to flatten everything, so we have more than one dimension at this point , dense fully connected layer is going to able recombine all these possible representation stored by convolutional neuron, represent 3 by 3 corner , feed that into dense layer with any kind of configuration. Multi dimension to one dimension by flattening.
- Configuring Dense layer with 128 neurons and relu activation.
- We are going to dropout half in this layer.
- Configuring Dense layer with 10 neurons and softmax activation.

When training machine learning models, users often want to able to examine the state of various aspects of the model, and how this state changes over time. To this end, Tensorflow and Keras supports a collection of different summary operations that can be inserted into the graph later in the results. Fig. 4 shows convolutional neural network model summary.

```
In [8]: model.summary()

Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 12, 12, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 9216)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               1179776
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                1290
=================================================================
Total params: 1,199,882
Trainable params: 1,199,882
Non-trainable params: 0
_________________________________________________________________
```

Fig. 4. Convolutional neural network model summary.

E. *Stage-6 & 7*

In this stage included compiling and training on 10000 from 60000 sample MNIST images with validation.

F. *Results*

This model is testing with only 1 epoch and the validation accuracy is 98.36 % (see Fig. 5).

```
In [10]: model.fit(X_train, y_train, batch_size=128, epochs=1, verbose=1, validation_data=(X_test, y_test))

Train on 60000 samples, validate on 10000 samples
Epoch 1/1
60000/60000 [==============================] - 319s - loss: 0.2410 - acc: 0.9267 - val_loss: 0.0518 - val_acc: 0.9836

Out[10]: <keras.callbacks.History at 0x7f5202ce6898>
```

Fig. 5. Trained data results.

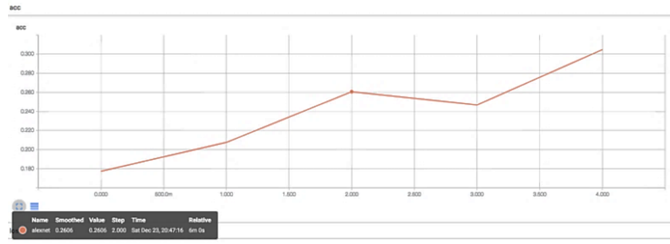

Fig. 6. Accuracy results.

978-1-5386-4185-9/18/$31.00 ©2018 IEEE

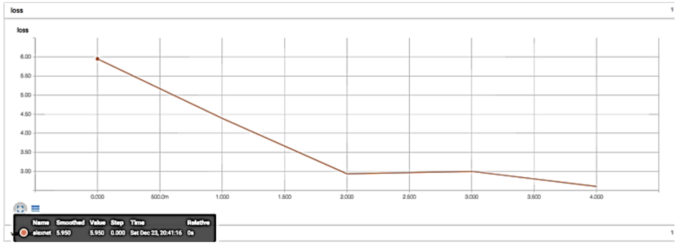

Fig. 7. Loss values to the training data decreases.

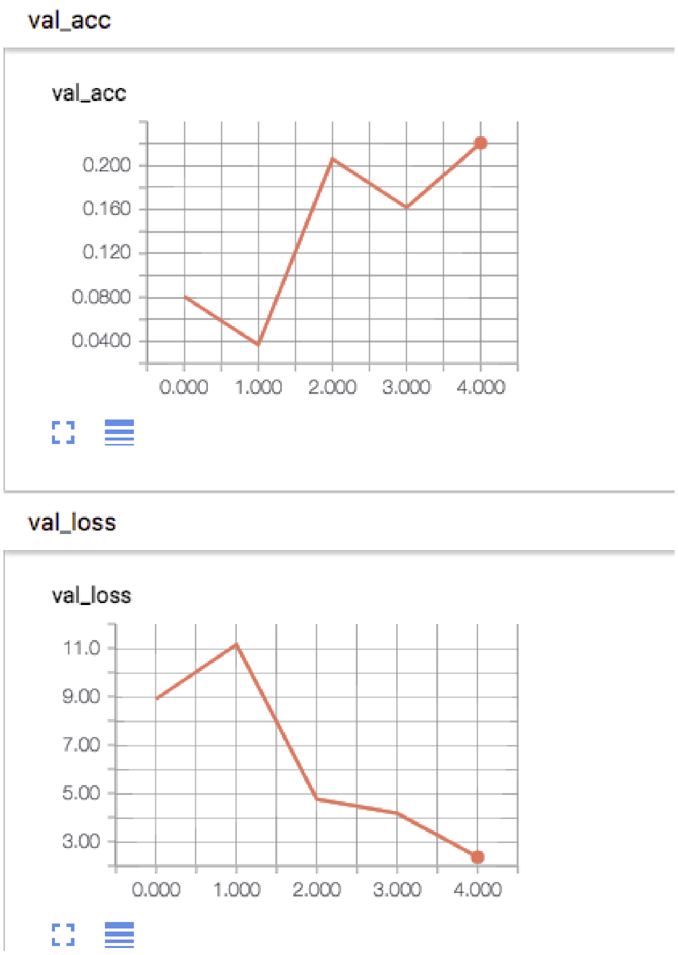

Fig. 8. Validated accuracy and loss after feedback from output.

The results from Fig. 6, 7 and 8 verify that when predicting accuracy increases, the amount of loss data on the trained data decreases over the final result. The validation accuracy and loss get from feedback received from fully connected deep neural network enhance the system performance.

Relu Activation functions were selected in the system to test the accuracy of the classification of the system was used for this purpose and received 98.36% classification accuracy is obtained on the test data.

V. Conclusion

Fast evolving technologies has led to the innovation of powerful unmanned aerial vehicles (UAV), also known as drones, for the purpose of extracting information or high-quality images in dangerous environments which human won't able to travel. In this paper we demonstrated a novel approach in applying deep learning technique (as a part of Machine Learning –ML) to assist image recognition task of Unmanned Aerial Vehicles (UAVs) in their search and rescue (SAR) operations. Tensorflow (Open Source software Library) and Keras (backend platform in python) were used in designing and training of a deep Convolutional Neural Network (NN) architecture. The learning process was conducted over given sample image data set, and the trained NN was applied over new image data taken by UAV camera, with the result achieved indicated the trained system capable of recognising images with high level of accuracy

References

[1] Ciresan D., Meier U., Schmidhuber J., (2012), "Multi-column deep neural networks for image classification", in: Proceedings of the CVPR,

[2] Colomina, I., Molina, P., (2014), "Unmanned Aerial Systems for Photogrammetry and Remote Sensing: A Review" , ISPRS Journal of Photogrammetry and Remote Sensing 92, pp. 79 – 97

[3] Deng L. (2014)," A tutorial survey of architectures, algorithms and applications for deep learning", APSIPA Trans. Signal Inf. Process. 3 e2.

[4] Gundlach, J., (2012), "Design Unmanned Aircraft Systems: A Comprehensive Approach", American Institute of Aeronautics and State University, Blacksburg, Virginia Gupta, S.G.,

[5] Ghonge, M.M., Jawandhiya, P.M., (2013), "Review of Unmanned Aircraft System (UAS)", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol. 2, Issue 4

[6] Howard A.G., (2013), "Some improvements on deep convolutional neural network based image classification", arXiv preprint, arXiv:1312.5402,

[7] Klemas. V.V., (2015), "Coastal and Environmental Remote Sensing from Unmanned Aerial Vehicle: An Overview", Journal of Coastal Research, Vol. 31, No. 5, 1260-1267

[8] Li H., Zhao R., Wang X., (2014), "Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification", arXiv preprint, arXiv:1412.4526,

[9] Nonami, K., Kendoul, F., Suzuki , S., Wang, W., Nakazawa, D., (2010), " Autonomous Flying Robots, Unmanned Aerial Vehicles and Micro Aerial Vehicles"‖, ISBN 978-4-431-53855-4, Springer

[10] Pereira, E., Bencatel, R., Correira, J., Felix, L., Goncalves, G., Morgano, J., Sousa, J., (2009), "Unmanned air vehicles for costal and environmental research", In: da silva, C.P (ed), Proceedings of the ICS, Journal of Coastal Research, Special Issue No. 56, pp. 1557 – 1561

[11] Ren J.S.J., Xu L., (2015), "On vectorization of deep convolutional neural networks for vision tasks," in: Proceedings of the AAAI, 2015.

[12] Schmidhuberc J., (2015), "Deep learning in neural networks: an overview", Neural Netw.61, 85–117.

[13] Simonyan K., Zisserman A., (2015), "Very deep convolutional networks for large-scale image recognition", in: Proceedings of the ICLR,

[14] Szegedy C., Toshev A., Erhan D., (2013), "Deep neural networks for object detection", in: Proceedings of the NIPS,

[15] Tompson J., Goroshin R., Jain A. , et al., (2015), "Efficient object localization using convolutional networks", in: Proceedings of the CVPR.

978-1-5386-4185-9/18/$31.00 ©2018 IEEE