Robotics Inference

Jose A Sanchez De Lucio

Abstract—Image processing has become one of the most developed research areas in recent years. This can be seen in our daily lives, from the use of social networks to the devices that we normally use, which probably were assembled using techniques based on the processing of images. In this work are presented two examples of classification of images to be used as a framework for the solution of similar problems. For both cases the main idea was the design and training of a convolutional neural networks for the classification of specific images. For the first case images of common goods were used as classes, while on the second case images of different toys were used. The results obtained shown that it is possible the classification and recognition of images following the proposed framework, based on the first project of the second term of the Udacity Robotics Nanodegree course.

Index Terms—CNN, Image processing, Robotics, Inference.

1 Introduction

The processing of images in robotics is one of the most important topics for the robots to perceive and interact with their environments. Furthermore, is an important topic because of all the benefits related, such as the use of robots to increase precision in surgeries to reduce risks by compensating surgeons movements accordingly with their environments [1], [2], or robots used to help people with different physical limitations to increase their quality of life [3], [4]. Moreover, the processing of images are also used for robots to recognize persons in social networks [5] or by recognizing and grasping objects in rpoduction lines [6], among others examples.

Different attempts to understand human vision can be tracked back to ancient times [7]. However, it was until the first half of the *20th* century that a significant result, obtained by *Max Wertheimer*, allowed to have a better understanding of perceptual organization, and therefore, helping to the development of computer vision [8]. Nowadays there exist many available techniques for the processing of images, and their applications have multiplied in relation with the increment of computational processing power.

Convolutional Neural Networks (CNN) is one of the available techniques for the processing of images that have shown success on it [9]. In the project was used a CNN with the AlexNet [10], [11] architecture for the classification of images, with the objective of classify toys in a production line. In the following section is described with more details the objective of this work, while in sections 3 and 4 are presented the results obtained and their discussion. Additionally, in the last section are presented some conclusions about the obtained results and proposed future work to increase the robustness of the proposed solution.

2 Background / Formulation

Reducing failures while producing any article is crucial in order to reduce costs and to improve the quality of the produced articles [12]. In this case the assemble line or production line was simulated by using vehicles assembled with *legos* to represent articles produced. The main idea was to generate a CNN based on the alexnet architecture to recognize the assembled vehicles, which can be used to identify any article produced under a specific tag that won't correspond to that specific class. Nonetheless, in order to show the performance of a CNN with the mentioned characteristics, a specific problem was solved as example, based on the first project of the term 2 of the Udacity Robotics Nanodegree program.

2.1 P1_Data Network

The Udacity example is based on a CNN trained with pictures of candy boxes, bottles, and nothing (empty conveyor belt) for the purpose of real time sorting, example of these pictures can be observed better in the next figure. The network was trained using a stochastic gradient descent algorithm (SGD) to relate the provided figures with specific labels in order to build a relationship based on the figures characteristics, where the training was accomplish using the DIGITS platform provided by Udacity. Furthermore, the training process is presented in figure 2.

Fig. 1. Set of images used with the Udacity example problem.

Furthermore, inference of the trained model was tested using the *DIGITS* platform too, obtaining the results presented in figure 3.

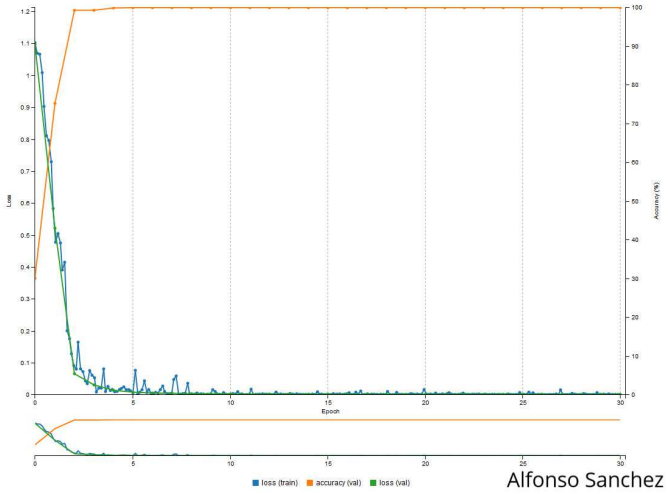

Fig. 2. Training a CNN for the Udacity example problem.

```
root@3d4c3a6d0c83:/home/workspace#
root@3d4c3a6d0c83:/home/workspace#
root@3d4c3a6d0c83:/home/workspace#
root@3d4c3a6d0c83:/home/workspace#
root@3d4c3a6d0c83:/home/workspace#
root@3d4c3a6d0c83:/home/workspace#
root@3d4c3a6d0c83:/home/workspace#
root@3d4c3a6d0c83:/home/workspace#
root@3d4c3a6d0c83:/home/workspace#
root@3d4c3a6d0c83:/home/workspace#
root@3d4c3a6d0c83:/home/workspace# evaluate

Do not run while you are processing data or training a model.

Please enter the Job ID: 20180521-210508-4e77

Calculating average inference time over 10 samples...
deploy: /opt/DIGITS/digits/jobs/20180521-210508-4e77/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20180521-210508-4e77/snapshot_iter_1800.caffemodel
output: softmax
iterations: 5
avgRuns: 10
Input "data": 3x227x227
Output "softmax": 3x1x1
name=data, bindingIndex=0, buffers.size()=2
name=softmax, bindingIndex=1, buffers.size()=2
Average over 10 runs is 4.70252 ms.
Average over 10 runs is 4.70957 ms.
Average over 10 runs is 4.68917 ms.
Average over 10 runs is 4.20317 ms.
Average over 10 runs is 4.19984 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14601  100 12285  100  2316   1051    198  0:00:11  0:00:11 --:--:--  2135

Your model accuacy is 75.4098360656 %
root@3d4c3a6d0c83:/home/workspace#
```

Alfonso Sanchez

Fig. 3. Inference obtained using the DIGITS Udacity platform.

2.2 Classifying Toys

The classification of toys was possible using legos to represent three classes of interest, each of them representing a different toy. The classes of interests, or toys used, can be observed better in figures 4, 5 and 6, for each of the toys used, respectively. As in the Udacity example, a CNN with an AlexNet architecture was used to relate the classes of interest images with specific labels, using a SGD algorithm with a learning rate (α) of .001. Unfortunately, it was not possible the use of the DIGITS Udacity platform for calculating the inference of the trained network, because the data used was not part of the Udacity provided Data for the previously mentioned program. Nonetheless, the accuracy of the proposed network was tested using additional images for each of the classes of interest.

Fig. 4. a

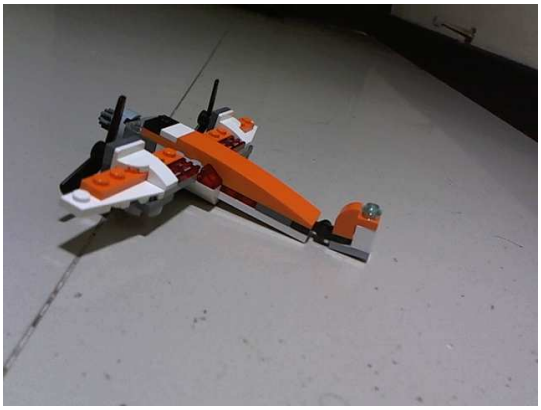

Fig. 5. b

Fig. 6. c

Fig. 7. Three classes of interest used to training the network to identify toys in a production line. Top: Class used to represent a F1 toy car, Middle: class used to represent a toy airplane and on Bottom: Class used to represent a toy Truck, all of the previous classes were generated using legos.

3 Data Acquisition

The acquisition of images for training the network to classify toys was possible using a *Logitech B910* webcam, in combination with the script provided by Udacity for the acquisition of images. For each class of interest were acquired 400 images, considering different distances from the camera, different positions for the toys and using the same background for all the acquired images. Additionally, taking advantage of the use of legos figures, some part were removed from the toys in order to increase the robustness of the trained network when identifying the same toys but with small differences, such as a missing wheel, a missing tire or a missing propeller.

4 Results

The results presented in this section are related with the network used with the simulation of a production line, where the objective of the network is to classify images of the produced articles. The first result presented is related with the training of the network, achieved using the DIGITS Udacity platform, results presented in figure 8. Moreover, the accuracy of the trained network is presented briefly using figures 9, 10 and 11, which are used to represent the accuracy of the network identifying images of the classes of interest that were not used for training the network.

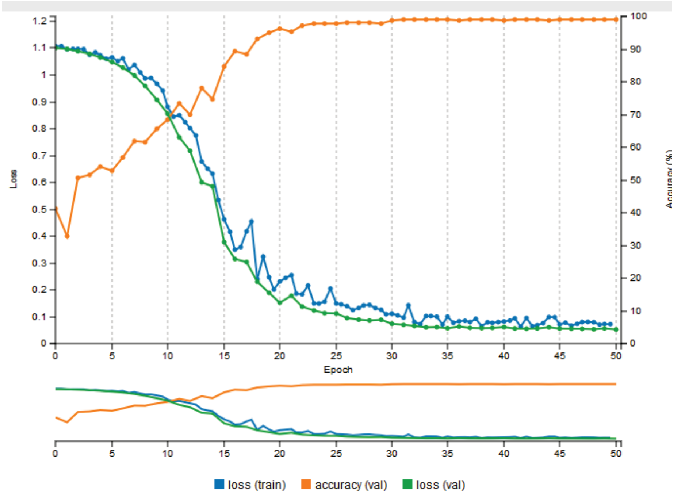

Fig. 8. Training a CNN to classify toys using the DIGITS Udacity platform.

5 Discussion

CNNs have been used for the classification of images in different fields and for different applications. The solution proposed for the presented problem proved that it is also possible the use of CNN for the classification of images to recognize specific objects in a simulated environment used to represent a production line of toys assembled using legos. Unfortunately, not all variables in a production line were considered in this work, it also needs to be considered the speed of the production line, especially because we were not able to calculate the inference times for the trained network.

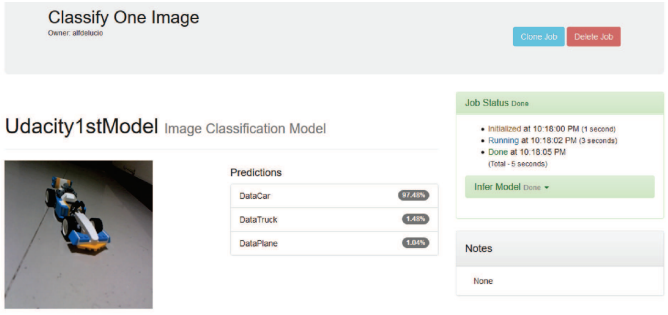

Fig. 9. a

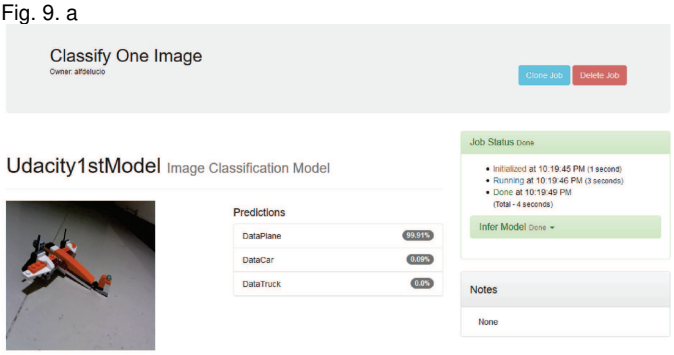

Fig. 10. b

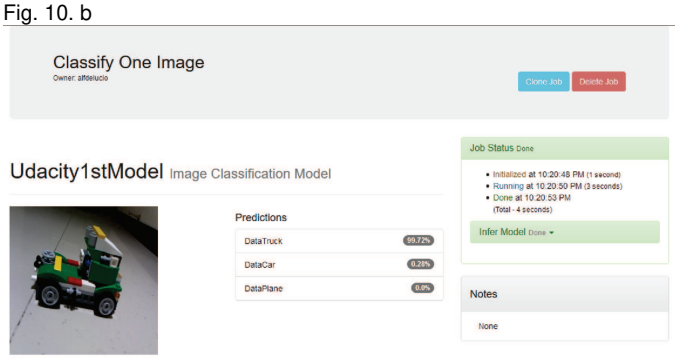

Fig. 11. c

Fig. 12. Performance of the trained network classifying three images related with the classes of interest that were not used during the training process.

6 Conclusion / Future work

The results obtained let us know that it is possible the use of a CNN based on the AlexNet architecture with a SGD learning algorithm and with $\alpha = .001$ for the recognition of toys with specific characteristics. Therefore, the probability that this network can be use in a production line to decrease costs due to misclassified toys is big. However, because of the limitations for the estimation of the inference times of the trained network, its performance is unknown for on-line applications. Nonetheless, based on the inference times obtained for the Udacity example problem and because of the networks similarities, it can be concluded that the trained network should achieve a good performance for the mentioned task.

Furthermore, a more robust analyst needs to be implemented in order to measure the performance of the proposed network for recognizing objects of interest in

production lines, which can be considered as the future work of this project, summarized in the following points:

- Increase the number of images for each of the classes to get a validation set.
- Use of the validation set to obtain a more robust accuracy measurements.
- Implemention of the trained network on the *Nvidia Jetson TX2*.
- Once the network is implemented on the TX2, calculate the inference times.
- Measurements of the accuracy when the line production moves at different speeds.

REFERENCES

[1] Y. Nakamura, K. Kishi, and H. Kawakami, "Heartbeat synchronization for robotic cardiac surgery," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 2, pp. 2014–2019, IEEE, 2001.

[2] R. H. Taylor, B. D. Mittelstadt, H. A. Paul, W. Hanson, P. Kazanzides, J. F. Zuhars, B. Williamson, B. L. Musits, E. Glassman, and W. L. Bargar, "An image-directed robotic system for precise orthopaedic surgery," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pp. 261–275, 1994.

[3] H. A. Yanco, "Wheelesley: A robotic wheelchair system: Indoor navigation and user interface," in *Assistive technology and artificial intelligence*, pp. 256–268, Springer, 1998.

[4] P. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Hand-eye coordination for robotic tracking and grasping," in *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, pp. 33–69, World Scientific, 1993.

[5] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[6] J. Krüger, T. K. Lien, and A. Verl, "Cooperation of human and machines in assembly lines," *CIRP Annals-Manufacturing Technology*, vol. 58, no. 2, pp. 628–646, 2009.

[7] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[8] M. Wertheimer, "Laws of organization in perceptual forms," *A source book of Gestalt Psychology*, 1923.

[9] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4277–4280, IEEE, 2012.

[10] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 427–436, 2015.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[12] D. Sannen, M. Nuttin, J. Smith, M. A. Tahir, P. Caleb-Solly, E. Lughofer, and C. Eitzinger, "An on-line interactive self-adaptive image classification framework," in *International Conference on Computer Vision Systems*, pp. 171–180, Springer, 2008.