Robotics Inference

A. Cámara

Abstract: Two image inference projects are studied. In first place, a given project with images of a conveyor belt where bottles and candy boxes are classified using deep neural networks. The second project classifies images of the hand game “stone, paper, scissors”. Popular neural networks has been chosen and carefully trained to get good results in these classification problems.

Introduction

Machine vision is the technology and algorithms used to recognize objects in images. It is an important field of study in robotics because vision can give machines a lot of information about their environment.
In first place, vision requires acquisition of images, usually using cameras and lightning that has been designed to provide the differentiation required by the following processing. Once good images are acquired, multiple stages of processing are applied to get the desired classification result. A typical image processing starts with filters that modify the image, followed by objects extraction, then objects data extraction to end with data analysis to get final results.

A modern approach to image classification is the use of deep neural networks (DNNs). They allow to learn all the necessary processing stages making faster and easier prototyping a machine vision system. DNNs require a lot of examples of classified images to learn and evaluate progress and a big quantity of computing power but they avoid the engineer the hard image processing work.

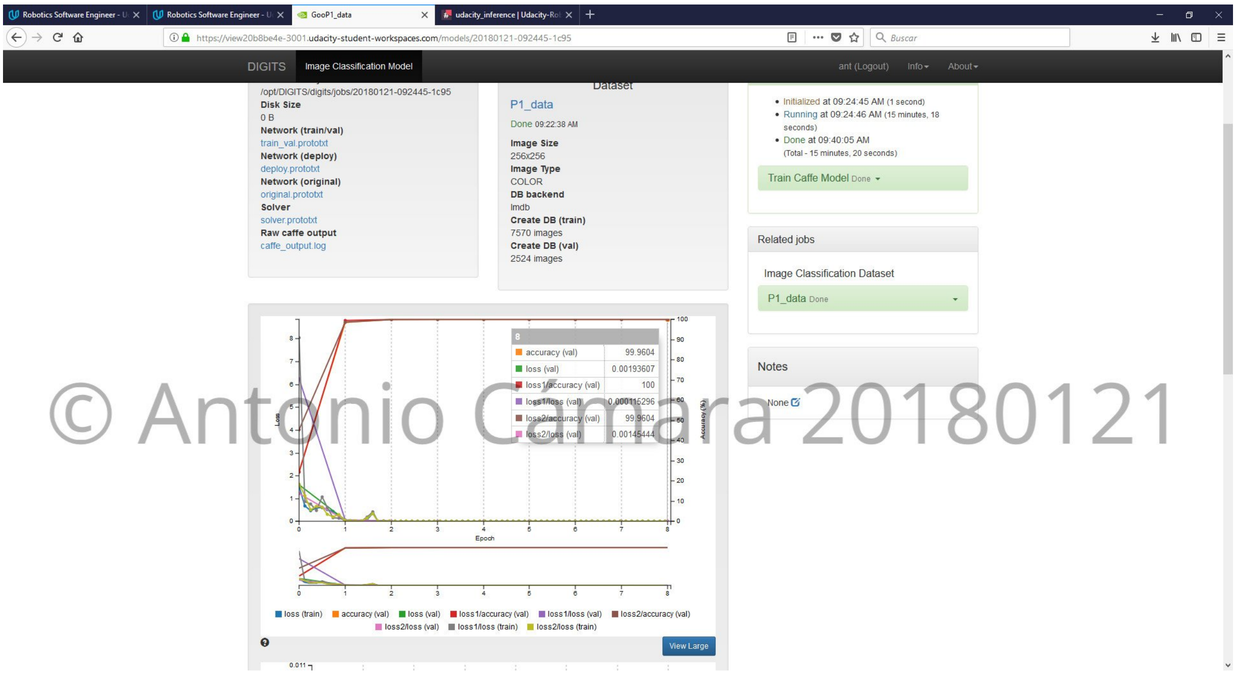
Recent success of DNNs has brought a good number of predesigned networks that can be used for fast prototyping like LeNet, AlexNet and GoogleNet. Moreover, there are several libraries (CNTK, Tensoflow, Caffe) and development environments like NVIDIA DIGITS that allow to make, train and deploy a DNN project in very little time.

This work is divided in two projects. The first one requires to build a DNN using DIGITS to classify a given set of images. There are 3 different types of images: candy boxes, bottles and nothing (empty conveyor belt). Images are 256x256 color pictures. Once the training set is loaded into DIGITS, a network is chosen to be trained in order to achieve an inference time of 10ms or less and an evaluation accuracy of more than 75 percent.

In the second project, a complete image inference project is implemented. Recognizing hand gestures of “stone, paper, scissors” game has been chosen as the goal of the project due to the importance of robots and human interaction. Gestures is a way of communication people use constantly and it is important for a social robots to understand and react to them. Our system should have into account 4 different classes: 3 gestures and nothing. After obtaining the necessary images, DIGITS is used again to train a network to do the classification. At the end, the network is deployed to a Jetson TX2 system and tested working in real time.

Background / Formulation

In these projects DIGITS has been used to train DNNs that do image classification. In the robotics inference project, GoogleNet has been chosen due to the high accuracy required of 75 percent in the evaluation. Using that network with the default training parameters and 8 epochs (observed as enough in previous uses of DIGITS) was enough to meet the expectations.


Robotics Software Engineer - U
Robotics Software Engineer - U
GooP1_data
udacity_inference | Udacity-Ro
https://view20b8be4e-3001.udacity-student-workspaces.com/models/20180121-092445-1c95
Buscar
DIGITS
Image Classification Model
ant (Logout)
Info
About
Dataset
Loss
Accuracy (%)
Epoch
0
1
2
3
4
5
6
7
8
10
20
30
40
50
60
70
80
90
100
View Large
0.011

/opt/DIGITS/digits/jobs/20180121-092445-1c95

Disk Size
0 B
Network (train/val)
train_val.prototxt
Network (deploy)
deploy.prototxt
Network (original)
original.prototxt
Solver
solver.prototxt
Raw caffe output
caffe_output.log

P1_data
Done 09:22:38 AM
Image Size
256x256
Image Type
COLOR
DB backend
lmdb
Create DB (train)
7570 images
Create DB (val)
2524 images

- Initialized at 09:24:45 AM (1 second)
- Running at 09:24:46 AM (15 minutes, 18 seconds)
- Done at 09:40:05 AM
  (Total - 15 minutes, 20 seconds)

Train Caffe Model Done

Related jobs

Image Classification Dataset
P1_data Done

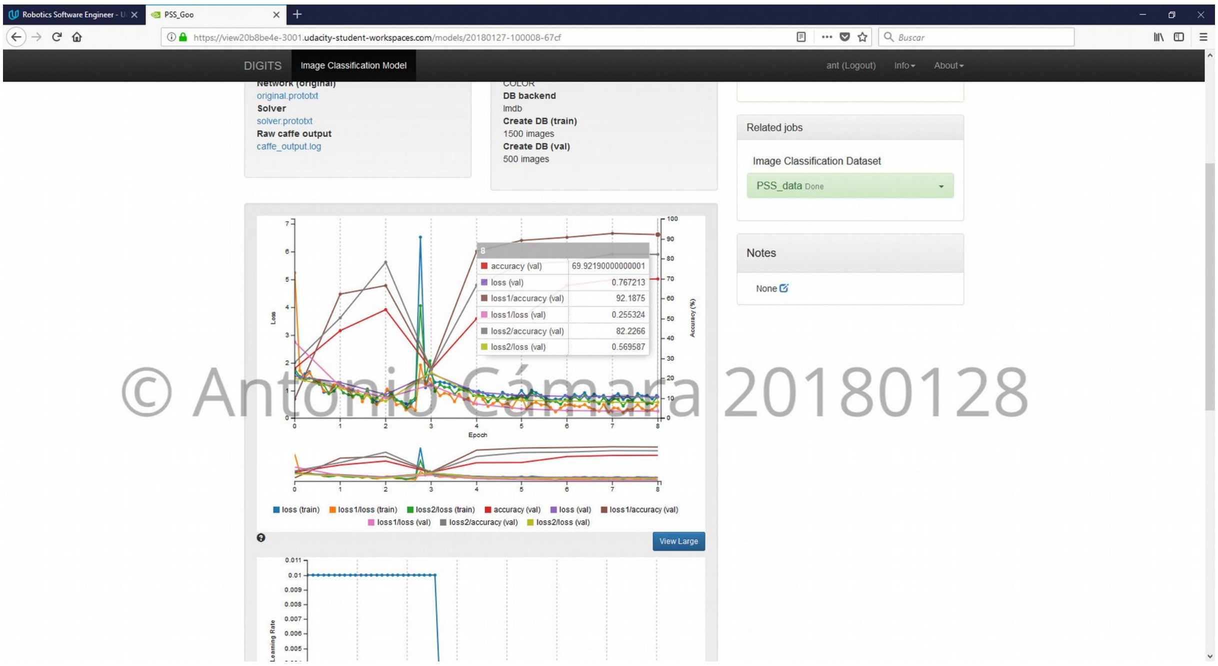
Notes

None

8	
accuracy (val)	99.9604
loss (val)	0.00193607
loss1/accuracy (val)	100
loss1/loss (val)	0.000115296
loss2/accuracy (val)	99.9604
loss2/loss (val)	0.00145444

■ loss (train) ■ accuracy (val) ■ loss (val) ■ loss1/accuracy (val) ■ loss1/loss (val) ■ loss2/accuracy (val) ■ loss2/loss (val) ■ loss1/loss (train) ■ loss2/loss (train)

Because GoogleNet is the best in accuracy of the pre-supplied networks in digits and given the good results of it with the first project, it has been used again to train the “stone, paper, scissors” project. Using the same parameters to train the second project gave a not very good training result of 69 percent of validation accuracy.



Network (original)
original.prototxt
Solver
solver.prototxt
Raw caffe output
caffe_output.log

COLOR
DB backend
lmdb
Create DB (train)
1500 images
Create DB (val)
500 images

Related jobs

Image Classification Dataset
PSS_data Done

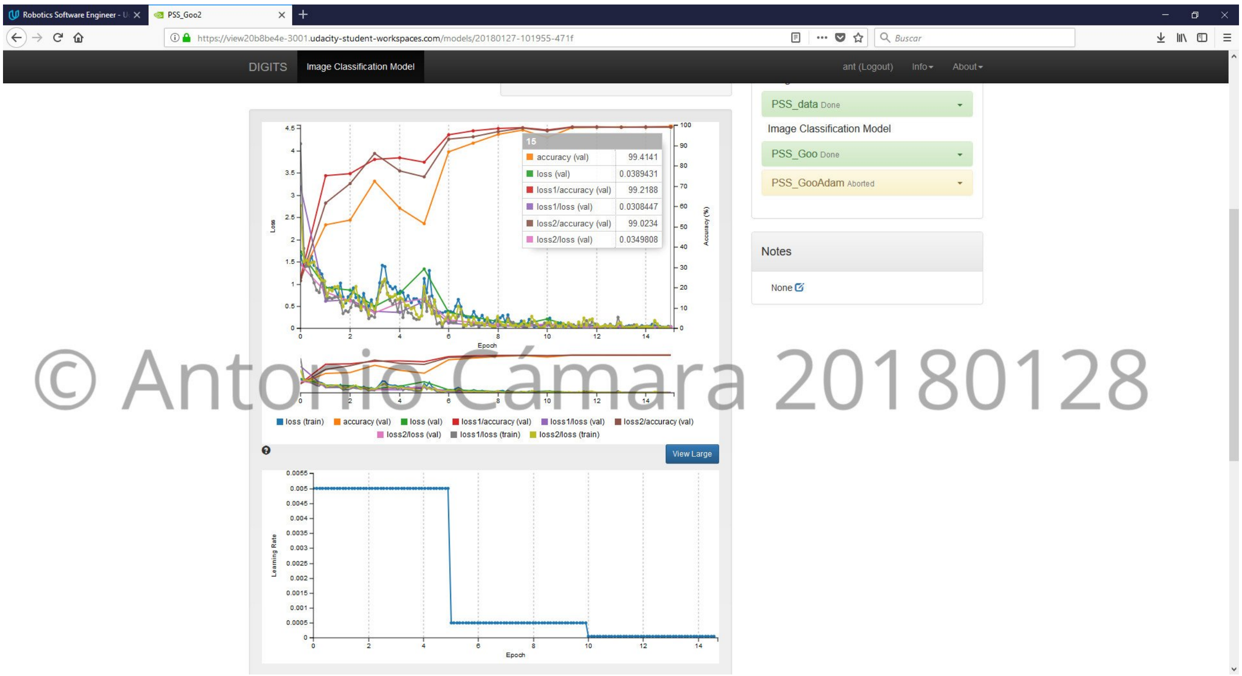
Notes

None

8	
accuracy (val)	69.92190000000001
loss (val)	0.767213
loss1/accuracy (val)	92.1875
loss1/loss (val)	0.255324
loss2/accuracy (val)	82.2266
loss2/loss (val)	0.569587

■ loss (train) ■ loss1/loss (train) ■ loss2/loss (train) ■ accuracy (val) ■ loss (val) ■ loss1/accuracy (val) ■ loss1/loss (val) ■ loss2/accuracy (val) ■ loss2/loss (val)

After lowering the base learning rate to 0.005 and increasing the number of epochs to 15 (to avoid problems of the first training doing it slower and training more to get to the desired validation accuracy), a good training result was obtained.


https://view20b8be4e-3001.udacity-student-workspaces.com/models/20180127-101955-471f

15	
accuracy (val)	99.4141
loss (val)	0.0389431
loss1/accuracy (val)	99.2188
loss1/loss (val)	0.0308447
loss2/accuracy (val)	99.0234
loss2/loss (val)	0.0349808

PSS_data Done

Image Classification Model

PSS_Goo Done

PSS_GooAdam Aborted

Notes

None

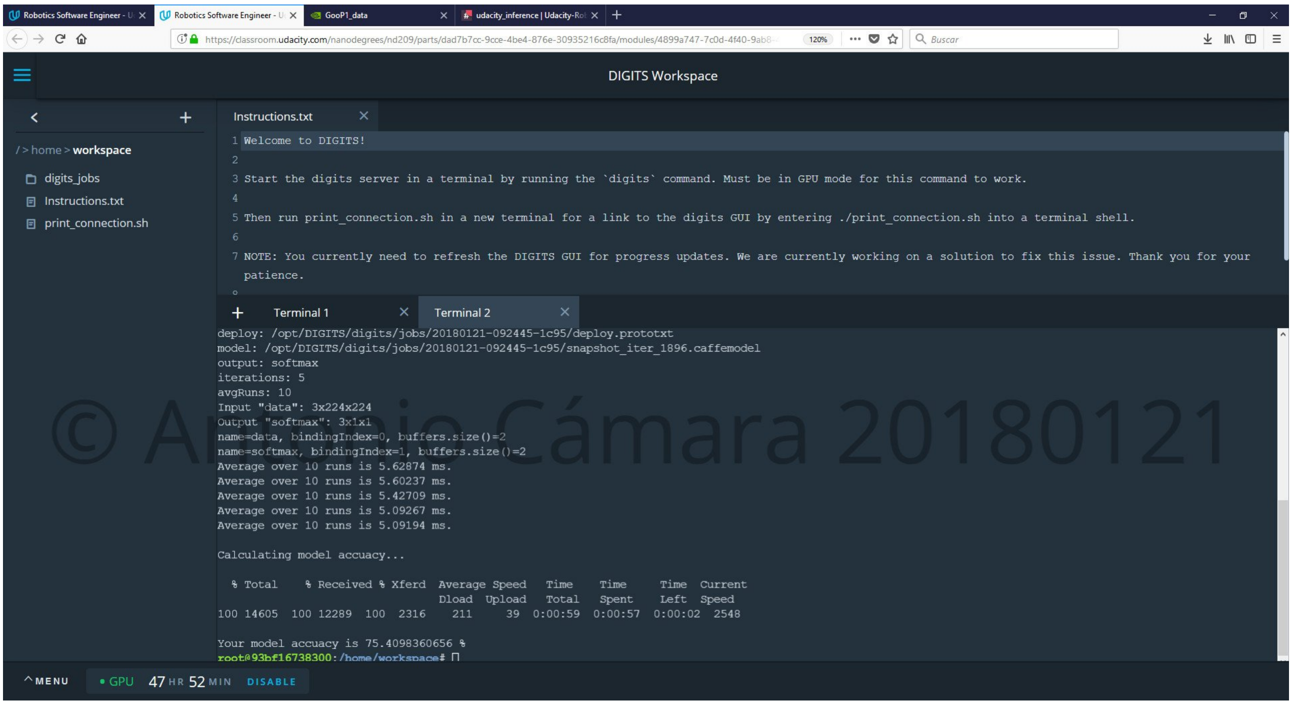
Data acquisition

In the second project, four classes are learned from images: nothing, stone, paper, scissors. 500 RGB images were obtained per class. They are enough to make a fast DNN prototype.
Images were acquired using Jetson TX2 and its included camera. They were obtained with a 256x256 resolution and a fixed background. A controlled lightning environment was used with a lateral natural light source.

Color images with 256x256 resolution were chosen to train them easily with the selected GoogleNet DNN.

Results

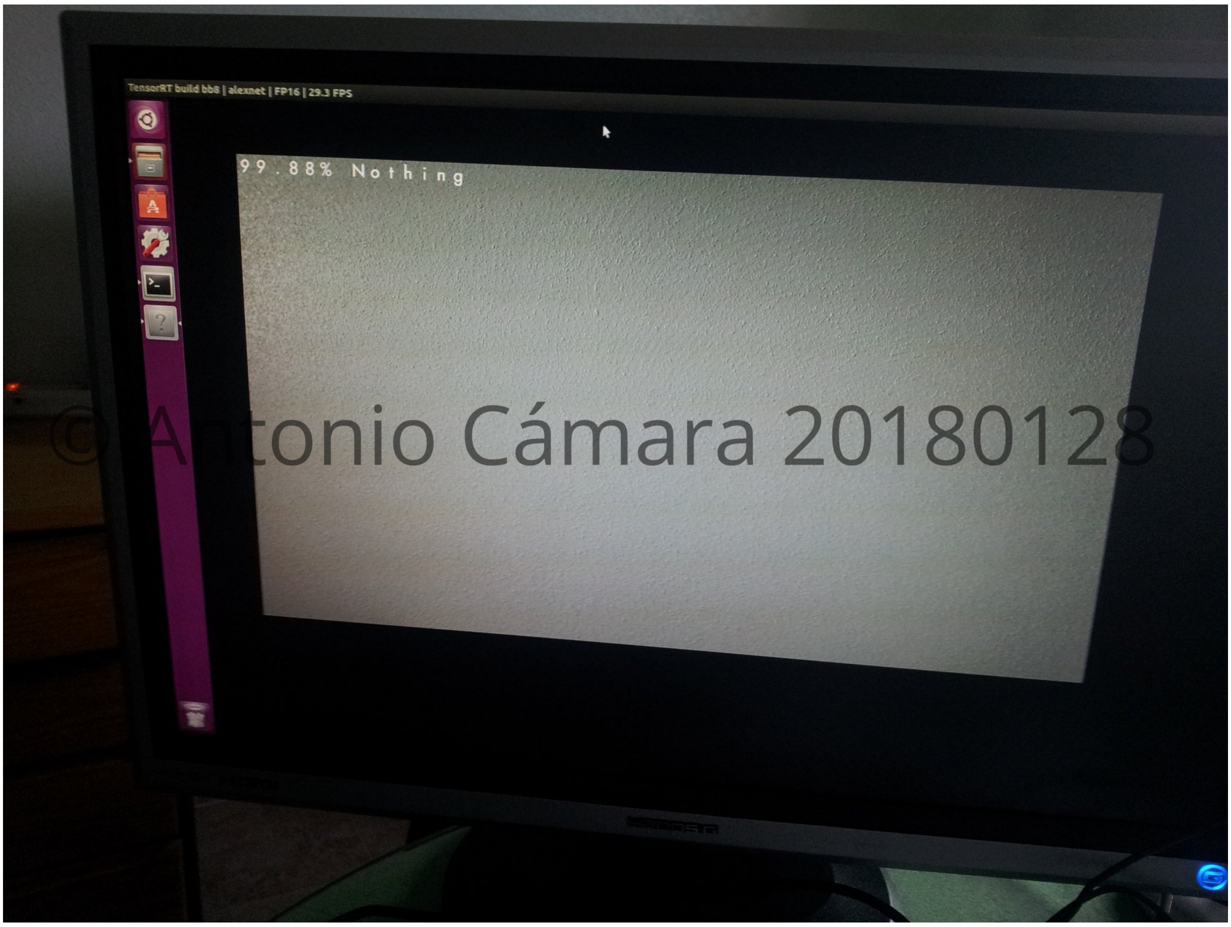
The robotics project model (P1) achieve the required 75 percent of accuracy and less than 10ms inference time as shown in the attached picture.

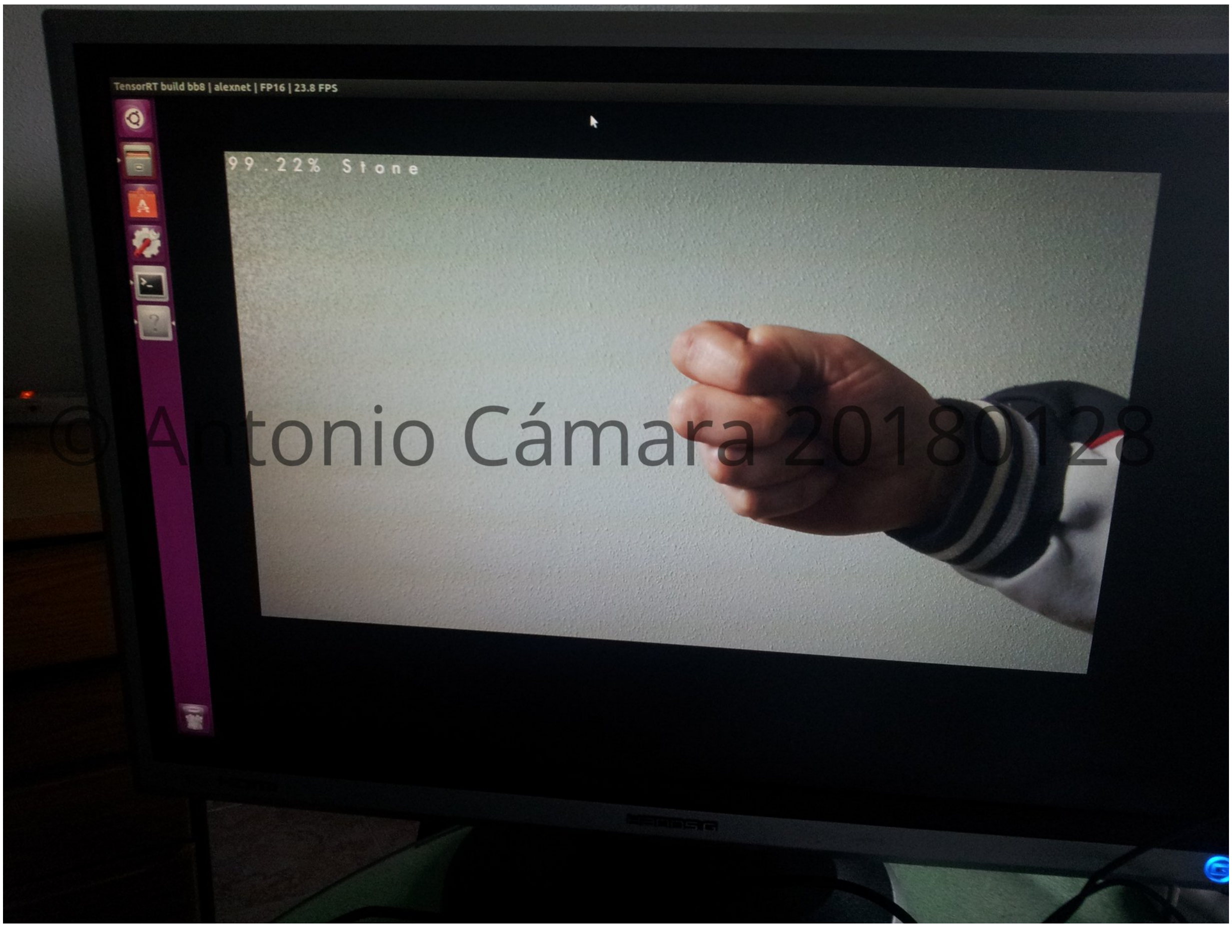

Robotics Software Engineer - U
Robotics Software Engineer - U
GooP1_data
udacity_inference | Udacity-Rol
https://classroom.udacity.com/nanodegrees/nd209/parts/dad7b7cc-9cce-4be4-876e-30935216c8fa/modules/4899a747-7c0d-4f40-9ab8
120%
Buscar
DIGITS Workspace
/ > home > workspace
digits_jobs
Instructions.txt
print_connection.sh
Instructions.txt
Terminal 1
Terminal 2
MENU
GPU 47 HR 52 MIN
DISABLE

```
Welcome to DIGITS!

Start the digits server in a terminal by running the `digits` command. Must be in GPU mode for this command to work.

Then run print_connection.sh in a new terminal for a link to the digits GUI by entering ./print_connection.sh into a terminal shell.

NOTE: You currently need to refresh the DIGITS GUI for progress updates. We are currently working on a solution to fix this issue. Thank you for your
patience.
```

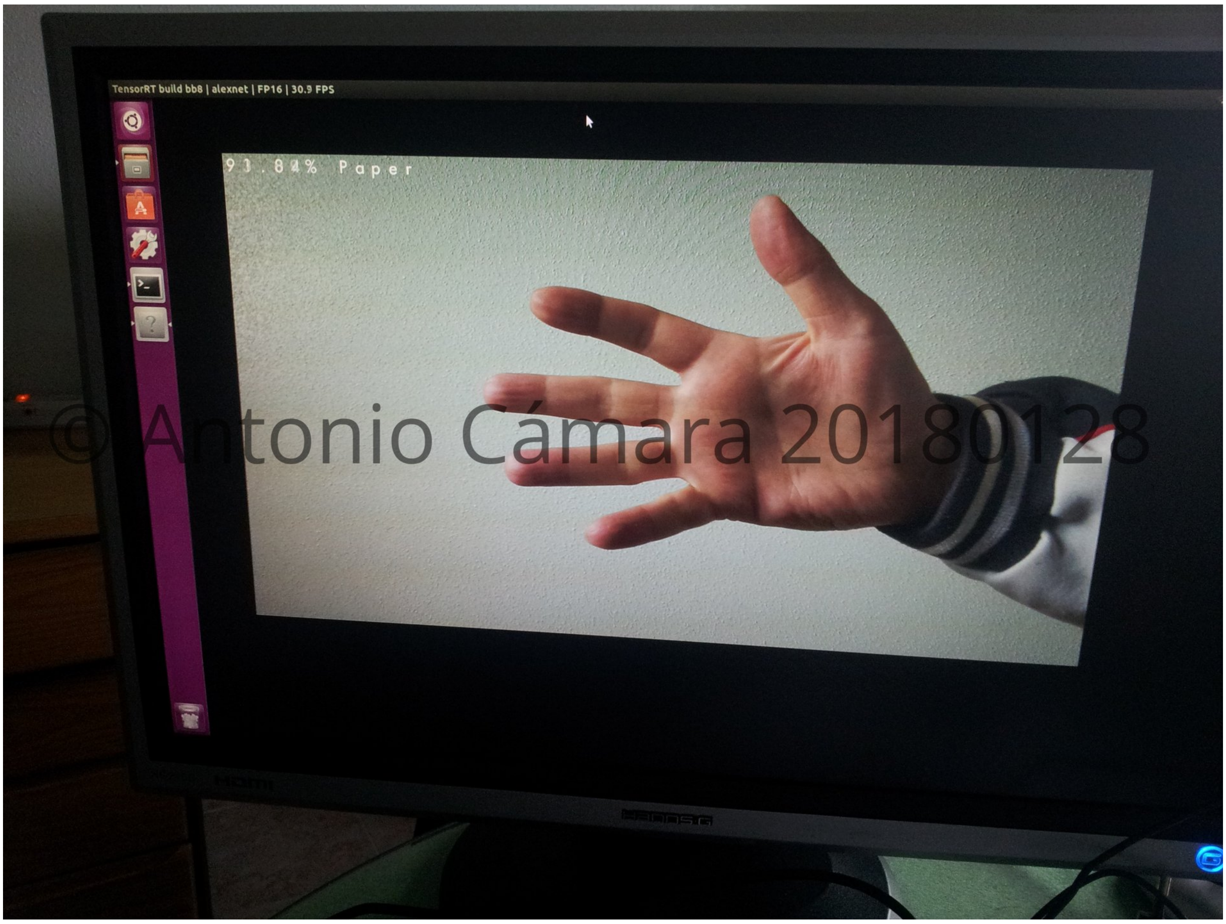
```
deploy: /opt/DIGITS/digits/jobs/20180121-092445-1c95/deploy.prototxt
model: /opt/DIGITS/digits/jobs/20180121-092445-1c95/snapshot_iter_1896.caffemodel
output: softmax
iterations: 5
avgRuns: 10
Input "data": 3x224x224
Output "softmax": 3x1x1
name=data, bindingIndex=0, buffers.size()=2
name=softmax, bindingIndex=1, buffers.size()=2
Average over 10 runs is 5.62874 ms.
Average over 10 runs is 5.60237 ms.
Average over 10 runs is 5.42709 ms.
Average over 10 runs is 5.09267 ms.
Average over 10 runs is 5.09194 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14605  100 12289  100  2316    211     39  0:00:59  0:00:57  0:00:02  2548

Your model accuacy is 75.4098360656 %
root@93bf16738300:/home/workspace# 
```

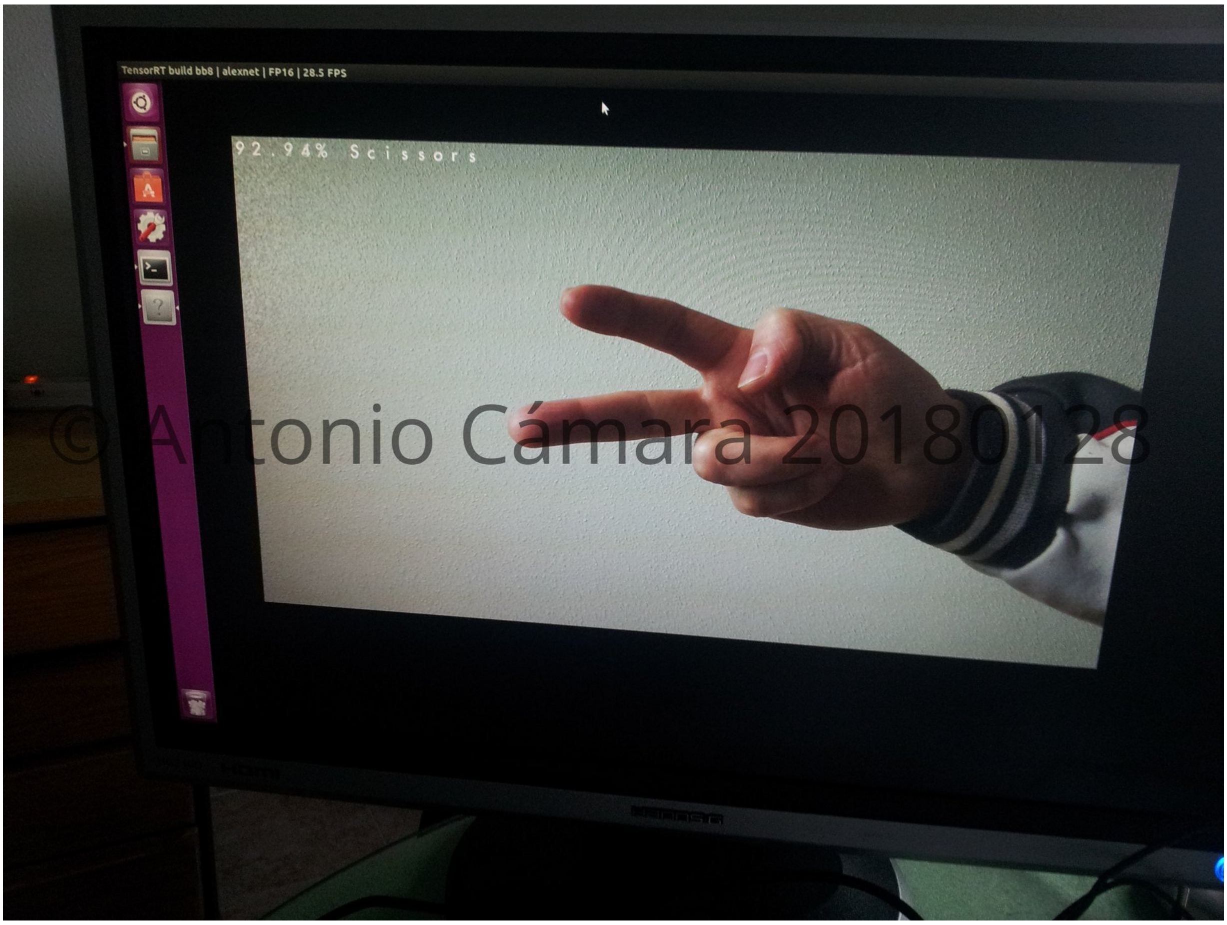
The second model obtained a good training accuracy and has been tested deploying it on the Jetson TX2. Using the inference-camera software, a good classification has been done. Nothing class is always well labeled. Paper and stone are almost always well classified. Scissors class is not always well identified been labeled as stone o paper in about 40% of the times.





```
TensorRT build bb8 | alexnet | FP16 | 23.8 FPS
99.22% Stone
```


TensorRT build bb8 | alexnet | FP16 | 30.9 FPS
93.84% Paper


TensorRT build bb8 | alexnet | FP16 | 28.5 FPS
92.94% Scissors

Discussion

The robotics inference project has achieved its desired results with the first attempt because of the good election of the network (GoogleNet). Better accuracy results could have been achieved using deeper models like ResNet or Inception.

Second project has good accuracy results except for scissors class that is sometimes mistakenly labeled. The training set of images is not very large, so with more images it can show better results. Another way to improve accuracy could be using better networks like ResNet or Inception.
As shown in the analysis (Canziani et al, 2016) accuracy and inference time are in a hyperbolic relationship. A little better accuracy costs a lot of computational time. Our robotics inference project requires real time response to the eyes of the person that interacts with it, therefore we need 0.2s of inference time or less. Having into account that the trained GoogleNet shows a maximum inference time of 0,05s there is margin to improve accuracy using Inception which needs 3 times more inference time.
After testing the network in the Jetson, it is recommended a larger testing set of images with different backgrounds and lights to achieve a high quality classifier that can perform well in a wide range of situations. Increasing the number of training images does not affect inference time but can produce better final results.

Conclusion / Future work

Two image inference projects have been done. The first project has soon achieved desired results because of the good election of the network. The second project has been trained and deployed successfully. It has good results but has to improve scissors class results witch can be done with more training images.

The second project can be used in a social robot that will be able to play the popular stone, paper, scissors game like another human using only vision. That can be useful in leisure places like parks or in interactive museums. Another useful application. Another useful application of this project is to help working with children with social difficulties like autistic children

As a future work, a more complex gestures recognition can be done. Understanding the language of signs or gestures people do while they interact with others, could be a very important skill for robots that should work with humans. Therefore, the suggested future work is doing a more capable gestures classifier to allow robots understand humans better.