Deep Learning guinea pig image classification using NVidia DIGITS and GoogLeNet

Lukasz Zmudzinski
University of Warmia and Mazury in Olsztyn
lukasz@zmudzinski.me

Abstract—In this paper guinea pig classification using deep learning imaging methods was performed on the Nvidia DIGITS 6. Models capable of distinguishing skinny, abyssinian and crested fur types were created in the process. To increase the classification accuracy empty images (with only the background) were added to the data set. Upon evaluation, the created model recognized the animals correctly from images taken in various household backgrounds.

Index Terms—classification, deep learning, animal recognition, robotics, DIGITS.

1 Introduction

Robotic systems are nowadays increasingly introduced in various animal care facilities like farms, daries and shelters [1], [2]. With the growing need of automating work more methods of image classification and decision making are needed.

In this work, deep learning techniques were utilized to create a classification model of guinea pigs in different home environments, to explore the possibilities of bringing such systems into the world of household animals.

1.1 Nvidia DIGITS

DIGITS (NVIDIA Deep Learning GPU Training System) was used in this project. It allows to rapidly train deep neural networks (DNNs) for image classification, segmentation and object detection tasks [3], [4].

The solution comes with pre-trained models for example:

- GoogLeNet,
- AlexNet,
- UNET and more.

In this paper GoogLeNet was used as the model of choice.

2 Background / Formulation

Deep Learning is growing in popularity over the past few years. It is connected to the fact that it becomes more useful than before thanks to the amount of available training data and advances in computer hardware/software [5].

Another problem with neural networks in general was vanishing gradients and unpredictable weight initialization [6]. This becomes solved by the use of RELUs (Rectified Linear Units) [7] and more recently the GoogLeNet deep learning model [8].

In neural network algorithms we can tweak several parameters, that will greately impact the final output: number of epochs, learning rate, solver type and many more.

- **Epoch** - one complete pass of the data in the data set. The amount of epochs should be determined through tests. Small number of epochs often leads to bad predictions, while a big number leads to overfitting.
- **Learning rate** - describes the rate at which the network abandons old beliefs, for new ones to take their place. The value must be correct, values that are too big or too small may lead to bad predictions.
- **Solver type** - contains information about how the weights are updated for the network. This project used one of the most popular ones: Stochastic Gradient Descent (SGD) [9].

Optimization algorithms help in minimizing (or maximizing on that matter) the *error function*. Wide variety of pre-made optimizers are available in the DIGITS framework from the simplest gradient descents to momentum based techniques like Adagrad [10].

GoogLeNet connects various techniques known from Deep Learning like convolutions, pooling, adding softmax and more [11], to distinguish all the objects that are present in the image. It implements so called *Inception modules*, that range from 245 filters to 1024 in top inception modules. The consequence of this is the possibility to remove fully connected layers on top completly [12].

3 Data Acquisition

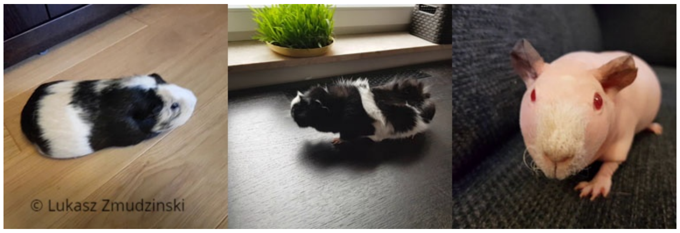


Fig. 1. Example images for crested, abyssinian and skinny guinea pigs from the provided data set.

The data was collected by recording a square video of each guinea pig over a period of 30 seconds in different environments and then extracting each frame as an image. Each frame was then lowered in resolution to 256x256 px in order to fit GoogLeNet requirements. Example images can be seen in Fig. 1.

To ensure good accuracy of the model, each set had to cover front, side and back angles of the guinea pig body. Moreover empty images (without guinea pigs) were added of the provided backgrounds, to increase classification accuracy in distinguish items.

Guinea pigs used in the experiment:

- Fifi (4 years, abyssinian),
- Rey (2 years, crested),
- Asajj (2 years, skinny).

Using the data, four labels were produced and assigned to the following neural network classes:

- **None** - when there is no guinea pig in the image,
- **Abyssinian**, **Crested** and **Skinny** - fur type.

The entire data set consisted of 1098 images, from which 25% were taken for validation and 10% were excluded for testing purposes. Moreover 32 photos in different environments were taken after the training to test the model behaviour. The visual representation can be seen on Fig. 2.

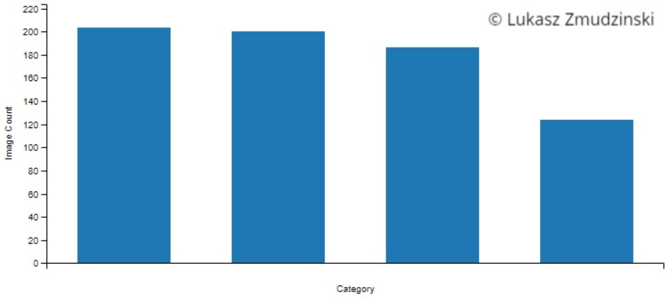


Fig. 2. Data representation in the training data set. Category labels from left: crested, skinny, abyssinian, none.

4 Results

The final classification model was gathered from running the described image set over 15 epochs of GoogLeNet model on Nvidia DIGITS. As you can see on Fig. 3 the accuracy after 15 epochs rose to over 90%, while the finall value loss was around 0.3 with a learning rate of 0.01. Using more epochs and a different learning was tested, but it led to overfitting of the data.

After the model was created, a series of tests were performed to check, if the inference is correct. The first set of tests consisted of images from the previously gathered data set. The prediction was always right for provided images, with the value between 70%-95%. You can see an example result on Fig. 4.

After acquiring enough information on the predictions, tests on new images were performed (different environments, same guinea pigs) giving satisfactory results - guinea pigs were classified correctly on each provided picture containing the animal. Example result on Fig. 5.

One failed prediction was encountered, when a picture of a background with a cat was used. The cat was badly classified as an abyssinian guinea pig.

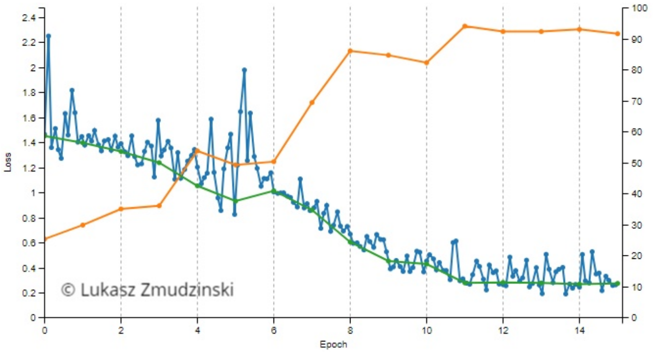
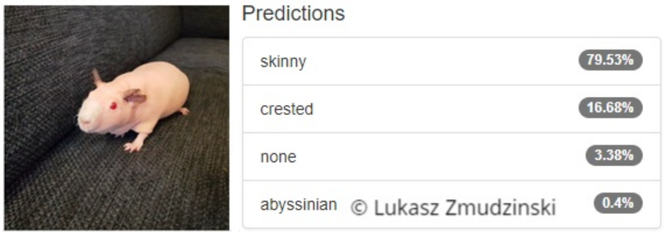


Fig. 3. Graph showing the value accuracy (orange), test data loss (blue) and value loss (green) after running the GoogLeNet model.


Predictions

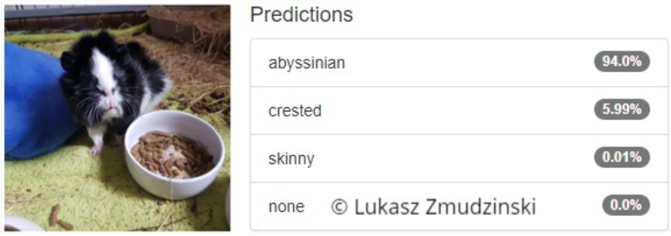
skinny 79.53%

crested 16.68%

none 3.38%

abyssinian 0.4%

Fig. 4. Example correct prediction results (79.53%) using previously captured images in selected environments using GoogLeNet.


Predictions

abyssinian 94.0%

crested 5.99%

skinny 0.01%

none 0.0%

Fig. 5. Example correct prediction results (94.0%) using post-model captured images in animal cage environment using GoogLeNet.

5 Discussion

Modern robotics highly depend on sensor readings of the surrounding environment. They often use camera input as one of the parameters to perceive the world. Due to that, imaging methods for decision-making were introduced.

Related works about animal classification were published in regard of wild animal monitoring [13], [14] and classifying different animals from their own data sets [15]. The authors follow a similar pattern when using deep neural networks, but none of them concern household animals. Both projects face different problems from those, that could be encountered in a safe, indoor environment. Hence, different data acquisition techniques had to be used.

In this paper Deep Learning was implemented for guinea pig classification in order to explore the possibilities of introducing household animal care using robotics and automation, while keeping them safe.

The provided GoogLeNet model from NVidia DIGITS has proven successful in identifying the guinea pigs in different environments, even when taking images that weren't originally added to the data set. Some errors were observed (false-positives) when no guinea pigs were present in the tested image. The model will behave poorly when other animals are present in the pictures, since the classification was based purely on guinea pigs.

Increasing the accuracy of the model, can greately improve the robot-animal interactions, allowing to tailor behaviours to specific beings. This could be achieved by using modifying the learning rate, using more images, creating more labels or finally using a different optimizer or prebuilt model. There are many possible variables to take into consideration, when building a model for a defined task.

The guinea pig classification model after building with GoogLeNet, was able to provide results almost instantly. This is important, if used for robotics, because while waiting for an action, the environment can change quite drasticly. Moreover, such model are built to be deployed on an autonomous platform (Raspberry, Jetson TX2 or any other), so the memory usage will be limited, increasing the inference calculation time.

6 Conclusion / Future work

The created model proves that guinea pig fur recognition for robotic systems is possible. The project gave good results - the created model recognized the animals correctly from images taken in various household backgrounds. The prediction was acquired fast making the inference time low. This is especially important for robotic systems that deal with live animals, because the reaction times need to be rapid.

Future work might include robotic systems that monitor the state of specific animals, adjust food distribution depending on image readings or alert when the guinea pig suffers from any kind of illness.

Moreover, different types of models can be employed to see, which one fits the needs the most. The project shouldn't be limited to GoogLeNet.

References

[1] J. J. Roldán, J. del Cerro, D. Garzón-Ramos, P. Garcia-Aunon, M. Garzón, J. de León, and A. Barrientos, "Robots in agriculture: State of art and practical experiences," *Service Robots Antonio Neves, IntechOpen*, 2018. DOI: 10.5772/intechopen.69874. Available from: https://www.intechopen.com/books/service-robots/robots-in-agriculture-state-of-art-and-practical-experiences.

[2] M. BD, S. Adil, and S. Ranvir, "Robotics: An emerging technology in dairy and food industry: Review," *International Journal of Chemical Studies*, 2018.

[3] "Nvidia DIGITS." https://developer.nvidia.com/digits.

[4] B. Erickson, P. Korfiatis, Z. Akkus, T. Kline, and K. Philbrick, "Toolkits and libraries for deep learning," *Journal of Digital Imaging vol. 30 pp. 400-405*, 2017.

[5] I. Goodfellow and Y. Bengio, *Deep Learning*. Cambridge, Massachusetts: The MIT Press, 2016.

[6] "GoogLeNet presentation," Available from: https://goo.gl/pPgJGn.

[7] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier networks," *In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics.* JMLR W&CP Volume (Vol. 15, pp. 315-323).

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Deeper with convolutions," Available from: http://arxiv.org/abs/1409.4842.

[9] L. Bottou, "Large-scale machine learning with stochastic gradient descent," *Proceedings of COMPSTAT 2010*.

[10] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research 12 pp. 2121-2145*, 2011.

[11] J. Long, E. Shelhammer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *arXiv: 1411.4038*, (v2) 2015.

[12] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2017.

[13] A. Gomez, A. Salazar, and F. Vargas, "Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks," 2016.

[14] G. Chen, T. X. Han, Z. He, R. Kays, and T. Forrester, "Deep convolutional neural network based species recognition for wild animal monitoring," *International Conference on Image Processing (ICIP) IEEE pp. 858-862*, 2014.

[15] X. Yu, J. Wang, R. Kays, P. A. Jansen, T. Wang, and T. Huang, "Automated identification of animal species in camera trap images," *EURASIP Journal of Image and Video Processing*, 2013.

7 Udacity provided data

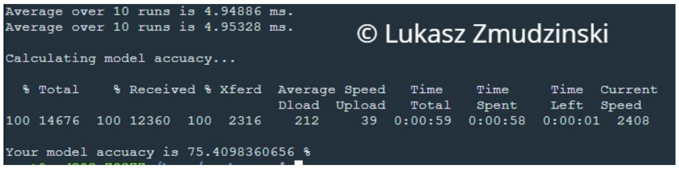
All the necessary project requirements for the supplied data were placed in this section. My plan is to publish the paper as an actual scientific article, so need to separate the Udacity challenge from the actual guinea pig project.

The following steps were taken to create the Conveyor Belt model:

1) The DIGITS platform was started,
2) Provided image data set was downloaded to ”Conveyor Belt” image classification set. The data settings were:
   - Image dimensions: 256x256
   - Image type: Color
   - Resize Transformation: Squash
   - DB Backend: lmdb
   - Image encoding: png
   - DB Compression: none

   The whole dataset took: 675 mb.
3) A new Image Classification Model was created with the following settings:
   - Training epochs: 15,
   - Solver type: Adagrad,
   - Base Learning Rate: 0.01,
   - Network used: GoogLeNet standard network.

   After some time, I broke the model creation, because after 6 epochs, no changes were seen on the graph.
4) The **evaluate** was run on the model, providing the results as seen on Fig. 6:
   - Average time: 4.95ms
   - Model accuracy: 75.41%



```
Average over 10 runs is 4.94886 ms.
Average over 10 runs is 4.95328 ms.

Calculating model accuacy...

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14676  100 12360  100  2316    212     39  0:00:59  0:00:58  0:00:01  2408

Your model accuacy is 75.4098360656 %
```

Fig. 6. The results of the evaluate methods for Udacity data set.

7.1 Conclusion

The Conveyor Belt project ended up quite similar to the Guinea Pig that was created earlier. The inference time achieved was really good - around 5ms. That means that the robot, will get the data about the object almost instantly. The accuracy was just over 75%, which makes the passing rate, although could be improved, by playing with the model parameters.