Department 07
Master Computer Science

Deep learning - Dog Breed Classification

Realization of an native Android app using deep learning algorithms

Alice Bollenmiller, Andreas Wilhelm
WS 17/18 IG
January 22, 2018

Contents

1 Introduction

1.1 Deep learning

- what is deep learning -¿ purpose, usage, current research projects, state of the arts

1.2 Terms of Referencee

- dog breed analyzer -¿ goals, purpose,
-¿ high perfomance computing but native android app

2 Methodological fundamentals

2.1 Common Frameworks for Deep Learning Applications

- some examples, tensorflow (tensorflow slim -¿ High level api for easier use, tensorflow lite), Caffe, Keras, Torch, PyTorch, ...
https://datahub.packtpub.com/deep-learning/top-10-deep-learning-frameworks/

2.2 Common Models in Deep Learning Applications

- short differences between different architecuteres (?, CNN, RNN)
- AlexNet, Mobilenet, Inception, VGG, -¿ short decsription, useCases, important things, differences

2.3 Qualified Models for mobile App Integration

- Mobilenet, Inception etc -¿ short decsription, useCases, important things, differences

2.4 Key requirements for an appropriate dataset

- generall why you need a huge dataset -¿ different backgrounds
- self trained needs a huge dataset, a lot of computing performance and time
-¿ so use pre trained, if small dataset.
-¿ pretrained used millions of pictures (e.g. ImageNet)

3 Concept

3.1 Frameworks

- tensorflow -¿ why

3.2 Model based Architectures

- general architectures of models -¿ Mobilenet, Inception

3.3 Application based Architecture

4 Realization

4.1 dataset

4.2 hardware environment

used CPU, GPU -¿ NVIDIA, handys

4.3 software environment

- Bazel, Java, Android Studio, Python, Operating System
- Android system

4.4 installation of software

- software environment

4.4.1 Tensorflow based on Python

4.4.2 Tensorflow based on Bazel

- e.g. Workspace changes for Android SDK, msse4.2

4.4.3 Installing Android Studio and its Delevopment Kit

- also possible with bazel but easier Android studio (needs correct versions of sdk, ndk)
- SDK, NDK
- IMPORTANT: tf versions updaten (same as trained)

4.5 building the models

-¿ evtl extra subsubsection:
- execution methods -¿ Bazel and Python (incompatible versions)
- Mobilnet -¿ steps, optimierung
- Inception -¿ steps, optimierung
- time related differences of execution
-¿ time CPUs/GPU

4.6 Output Tests and Validation

- test pictures and if it works -¿ label image
- validation script?!

4.7 Implementation of an native Android App

- list all necessary things to do (e.g. tensorflow version, Interpreter -¿ load Model)

4.8 Deployment and Validation

5 Evaluation

- prio von nierdig zu hoch
- regarding implementation time

- regarding performance
- regarding quality in accuracy
- handy perfomance?

6 Conclusion

- tutorials not complete, different
- which model is better
- prospects, improvements, Recommendations

Beispiele fürs referenzieren:
In Figure 1 ist das HS München Logo zu sehen.

Figure 1: FH-Logo

Oder auch eines Codes wie in listing 1.

```
1
2  bottleneck_path_2_bottleneck_values = {}
3
4
5  def create_bottleneck_file(bottleneck_path, image_lists, label_name, index,
6                             image_dir, category, sess, jpeg_data_tensor,
7                             decoded_image_tensor, resized_input_tensor,
8                             bottleneck_tensor):
9    """Create a single bottleneck file."""
10   tf.logging.info('Creating bottleneck at ' + bottleneck_path)
11   image_path = get_image_path(image_lists, label_name, index,
12                               image_dir, category)
13   if not gfile.Exists(image_path):
14     tf.logging.fatal('File does not exist %s', image_path)
15   image_data = gfile.FastGFile(image_path, 'rb').read()
16   try:
17     bottleneck_values = run_bottleneck_on_image(
18         sess, image_data, jpeg_data_tensor, decoded_image_tensor,
19         resized_input_tensor, bottleneck_tensor)
20   except Exception as e:
21     raise RuntimeError('Error during processing file %s (%s)' % (image_path,
22                                                                  str(e)))
23   bottleneck_string = ','.join(str(x) for x in bottleneck_values)
24   with open(bottleneck_path, 'w') as bottleneck_file:
25     bottleneck_file.write(bottleneck_string)
```

Listing 1: Some python code

Sectionrefs: In section 2 ist vieles noch nicht fertig.
SubSectionrefs: In section 2.1 wird dann näher auf den Inhalt eingegangen.
SubSubSectionrefs: In section 4.4.1 gehts ans eingemachte.

Beispiele fürs zitieren:

Für einen noch besseren Überblick, kann das Buch von Butler et al. (2017) hinzugezogen werden. Wenn in klammern und Seitenzahl (Butler et al., 2017, p. 3)

als compared, aber ohne Seitenzahl (cmp. Butler et al., 2017)
als compared mit Seitenzahl, das nd heißt ”no date”, da keine Jahrezahl vorhanden (cmp. Wang et al., nd, p. 5)

References

Butler, R., Butler, R., and Pettey, C. (2017). Implementation of a tensorflow convolutional neural network to discriminate similar gene functions.

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. (n.d.). Locality-constrained linear coding for image classification.

List of Figures