Visual Recognition of Cooking Ingredients

Image Classification using Deep Learning

Agenda

Introduction

- Problem Description
- Experimental Setup

Results

- Performance of the Trained Model
- Examples

Conclusion and Outlook

About me

Kiril Schewzow (kiril.schewzow@gmail.com)

Masters and PhD in Physics (Medical Imaging)

Former IT-Consultant (1.5 years working experience)

Numerous Online Courses in Data Science and Machine Learning

Some Motivation

You have food in the fridge but you don't know what to cook

→You order something less healthy

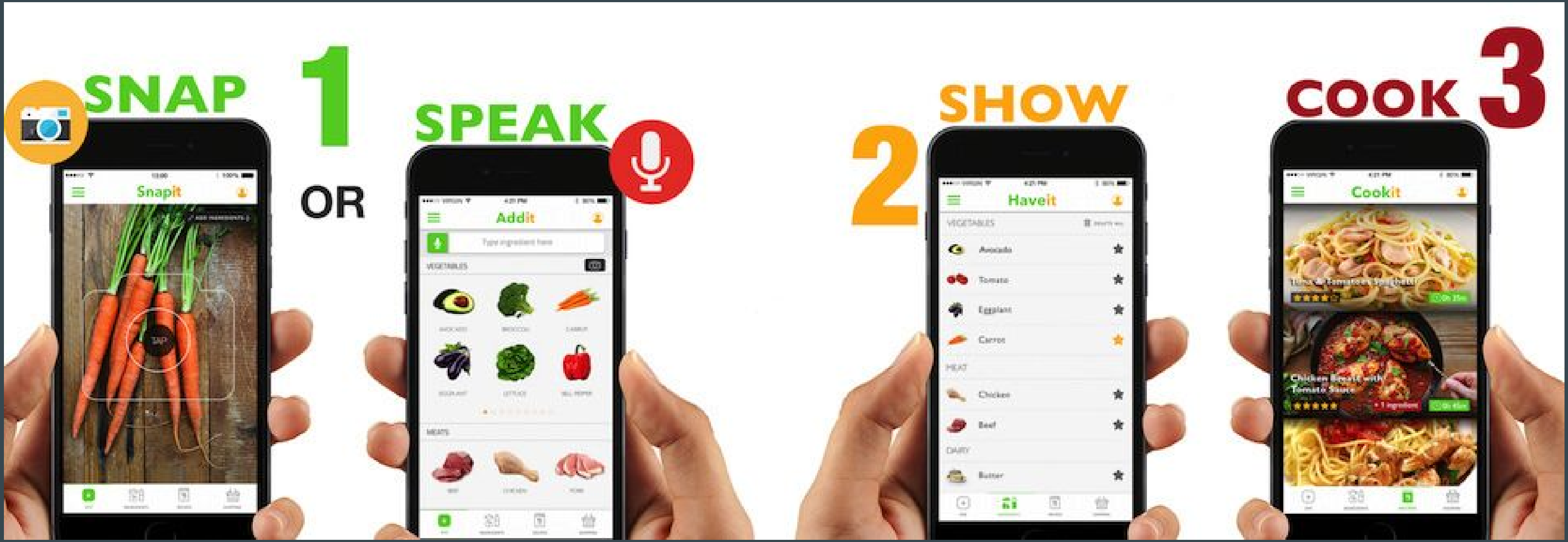
→ You spend more money

→ The food in the fridge spoils

On average a person in Vienna throws away 40 kg of food every year

Scoodit.com

Snap pictures of what you have ⇒ Recipe



Visual recognition of ingredients

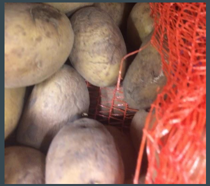
Typing is slow

Speaking to the phone still feels awkward for many people (including me)

Snapping pictures is fun! :)

How it works

Scope of my Project



Deep Learning Magic

Garlic

Red Onion

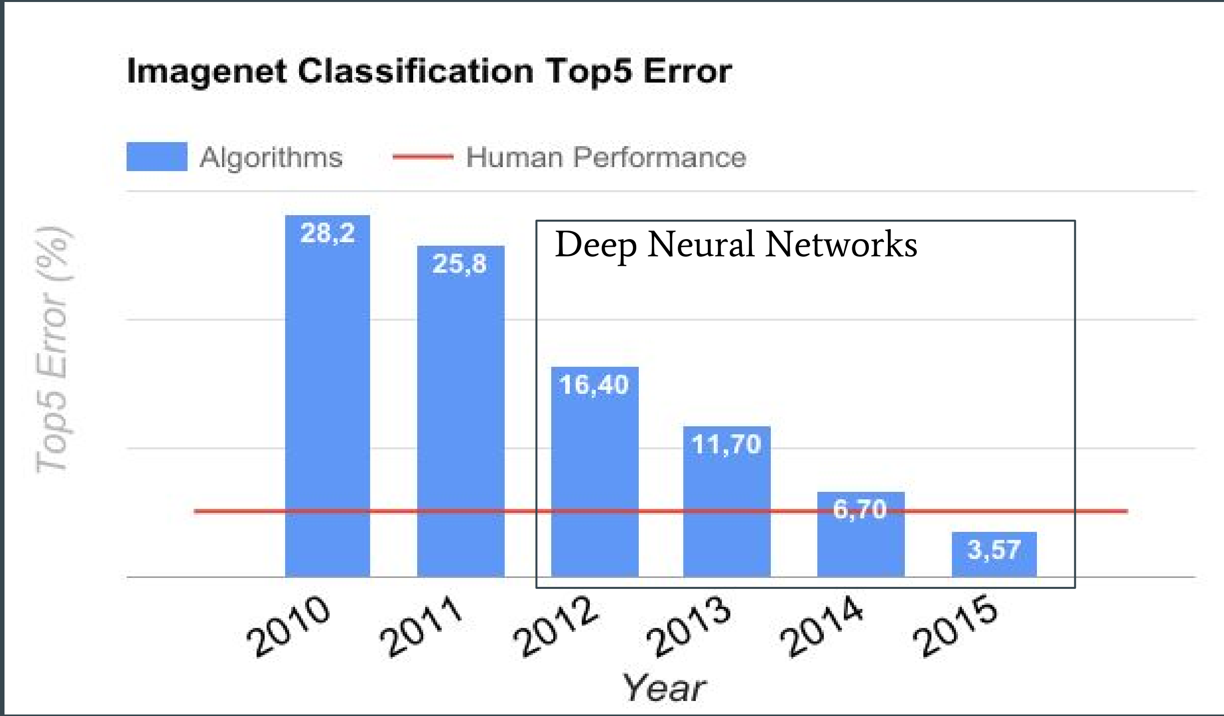
Champignons

Potatos

Ingredients-Recipe Matching

Delicious Meal

Why Deep Learning



Challenges in Deep Learning

- Large amount of data required
  - Use Imagenet images as starting point
- High Computational power required (GPU)
  - Use AWS GPU instances
- Time to train a model from scratch can be in order of days or even weeks
  - Use pretrained Models

Our setup and dataset

AWS p2.16xlarge instance

- 16 NVIDIA K80 GPUs (192 GB GPU Memory)
- 64 vCPUs (732 GB RAM)
- 20 Gbps Network

178 Imagenet Classes images (fruits, vegetables, other ingredients)

- 180K Imagenet images for training
- 10K Imagenet images for validation
- 4K own images for validation (not all 178 classes present)

Deep Learning Frameworks

Aggregate popularity (30•contrib + 10•issues + 5•forks)•1e-3

#1:	97.53	tensorflow/tensorflow
#2:	71.11	BVLC/caffe
#3:	43.70	fchollet/keras
#4:	32.07	Theano/Theano
#5:	31.96	dmlc/mxnet
#6:	19.51	deeplearning4j/deeplearning4j
#7:	15.63	Microsoft/CNTK
#8:	13.90	torch/torch7
#9:	9.03	pfnet/chainer
#10:	8.75	Lasagne/Lasagne

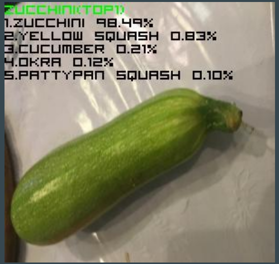
Results

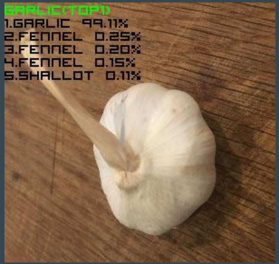
Training time: ~10h (not optimized)

Performance:

- Imagenet validation set (10K images): 92% Top5 accuracy
- Our own preliminary validation set (4K images): 80% Top5 accuracy
    - many images were extracted as frames from short videos

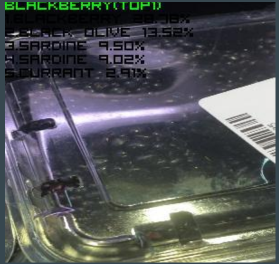
Examples: Top1

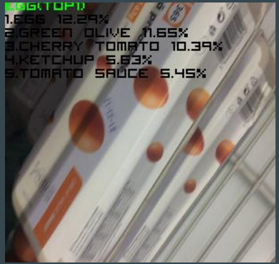


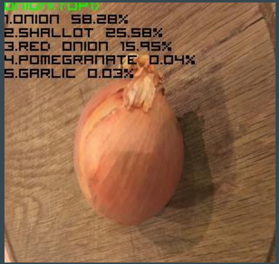


Examples: Top1

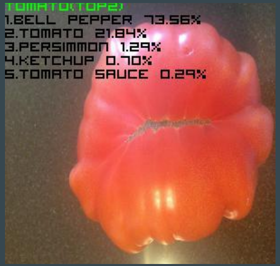
Examples: Top1

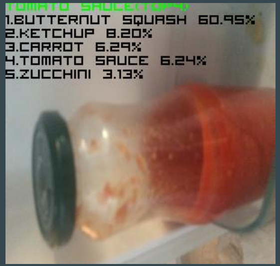




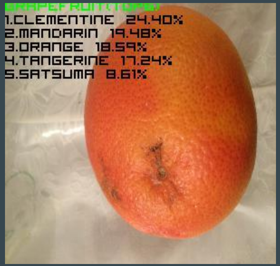


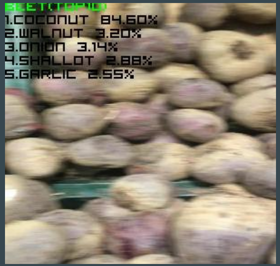
Examples: Top2-Top5

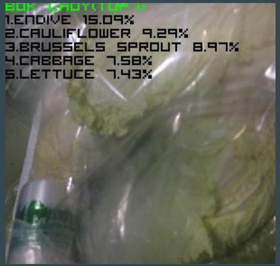


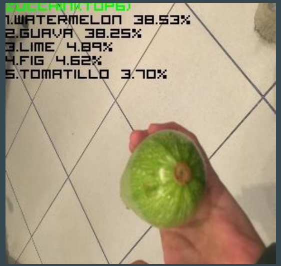


Examples: Top6-Top10

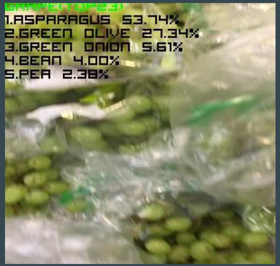


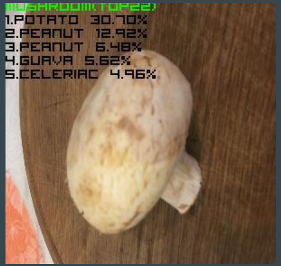






Examples: not even close





Next Steps

Model Deployment via Tensorflow-Serving or AWS Lambda

Experiments with more than 178 Classes

Continuous improvement using the images produced by the app

Multi Object Detection

Conclusion

Deep Learning Models show unmatched performance in image recognition

Fine tuning the pretrained models instead of training from scratch reduces the training time dramatically.

The frameworks are in active development, there can be incompatibilities due to different version (Framework, GPU-Driver, Model)

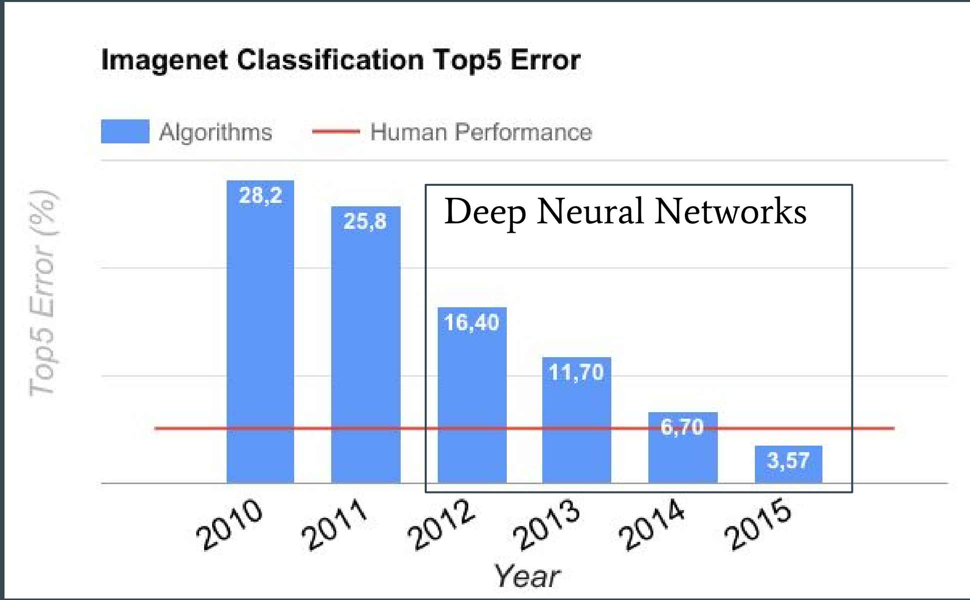
Thank you for your attention

Why Deep Learning

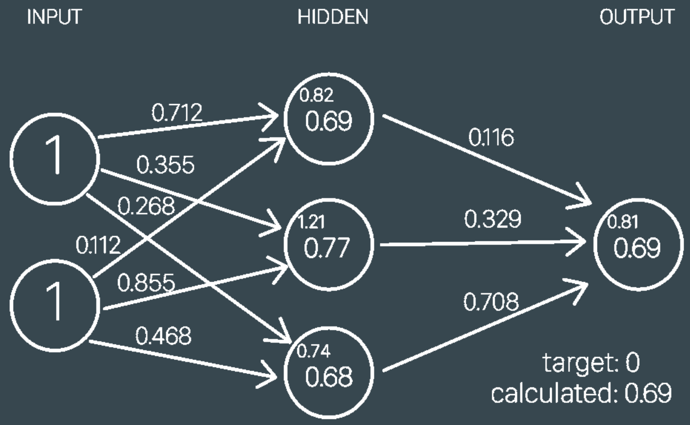
- Superior performance compared to all other techniques
- No domain knowledge required

ImageNet Large Scale Visual Recognition Competition

- 1.000.000 images
- 1.000 image classes
- 1.000 images per class
- if the algorithm return the right class among the top 5 guesses it counts as correct



Neural nets and deep neural nets



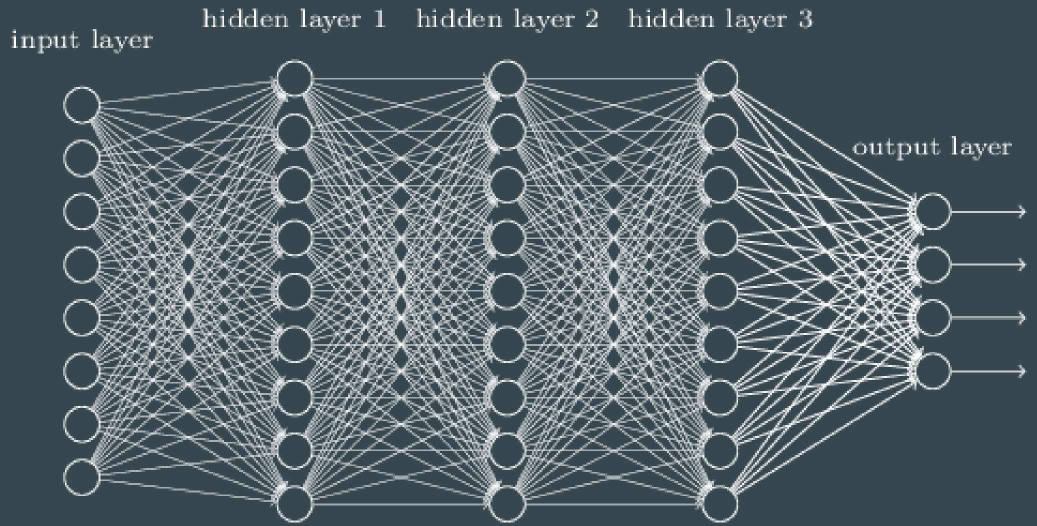


Image classification:

- number of input parameters = image width(pixels) × image height(pixels) × 3 (~150K-300K)
- number of output values = number of classes

Challenges in training of deep neural nets

- Large amount of data required
- High Computational power required (GPU)
- Time to train a model from scratch can be in order of days or even weeks

Deep Learning Model

Inception V3 (Google 2015)

- 93.9% Top-5 Accuracy on the 1000 Classes of Imagenet (Training: 2 weeks on 8 NVIDIA Tesla K40 GPUs)