

Introduction to Deep Learning

Course Introduction

October 10, 2018

Sebastian Stober <stober@ovgu.de>
Jens Johansmeier <???@ovgu.de>



FACULTY OF
COMPUTER SCIENCE



Course Prerequisites

- linear algebra
(e.g. Math I+II @ FIN)
- machine learning
(e.g. “Intelligente Systeme” or “Machine Learning” @ FIN)
- plus: basic Python programming

Introductions

“Living Stats”

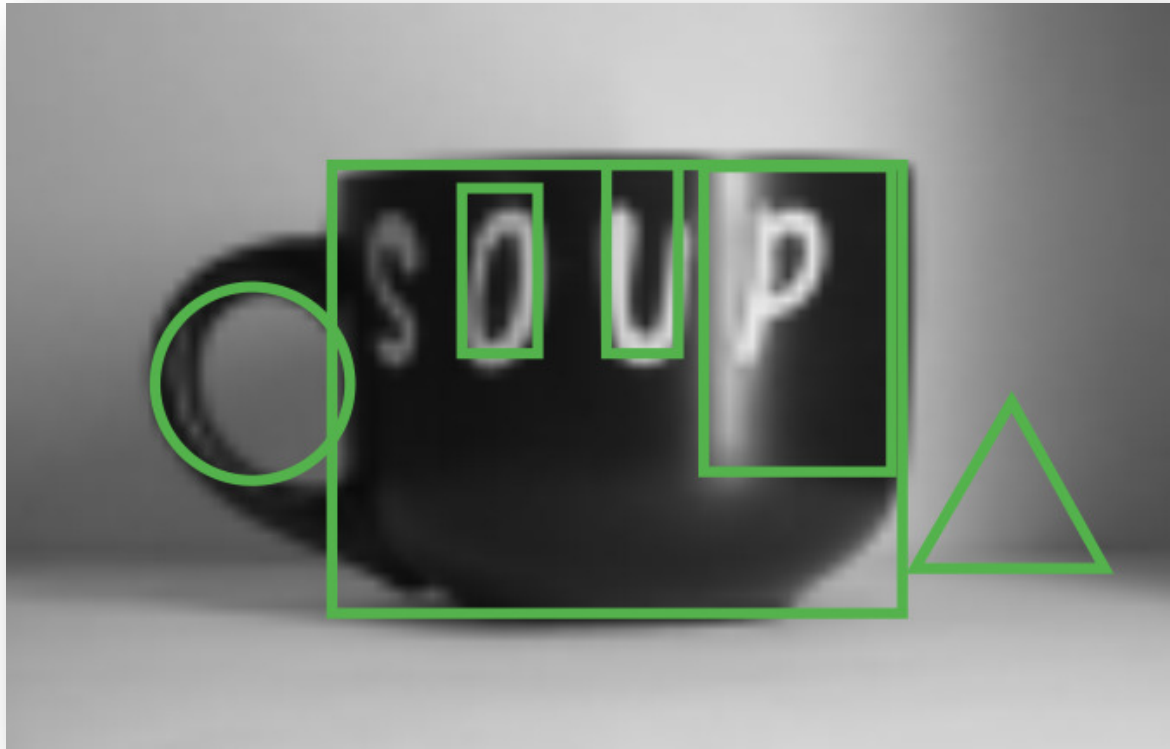
- Where are you from?
- What do you study?
- How experienced are you in ML?
- How experienced are you with Python?

My Experience

1. ML / artificial neural nets > 17 years
2. deep nets > 5 years
3. generative nets > 2 years

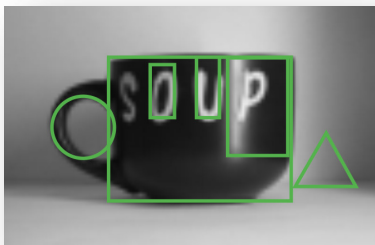
My 1st Serious ML Project

- A coffee mug though the “eyes” of a road-sign detector in 2003:

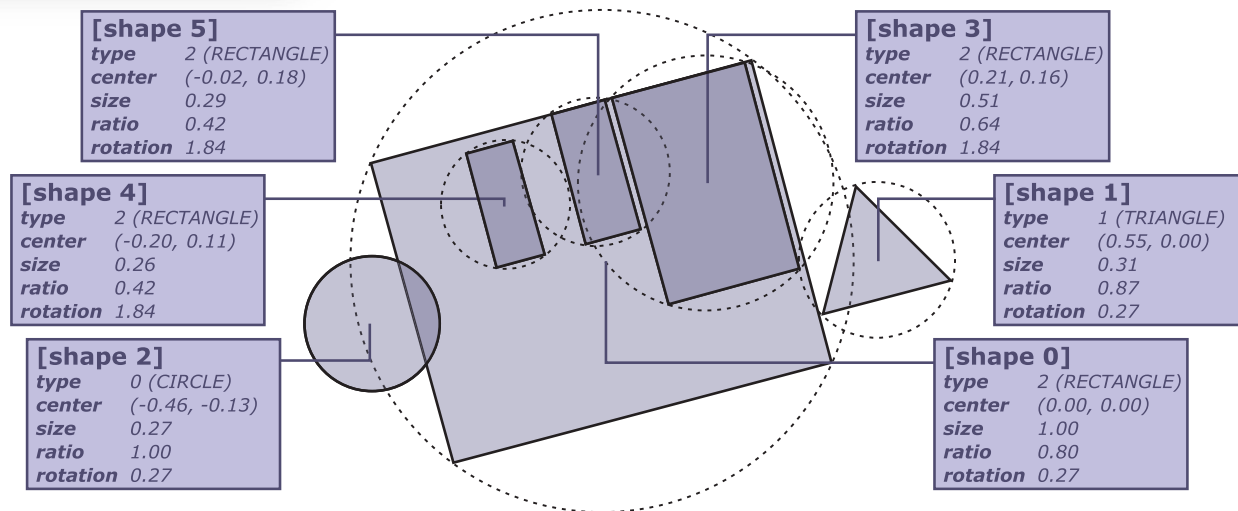


My 1st Serious ML Project

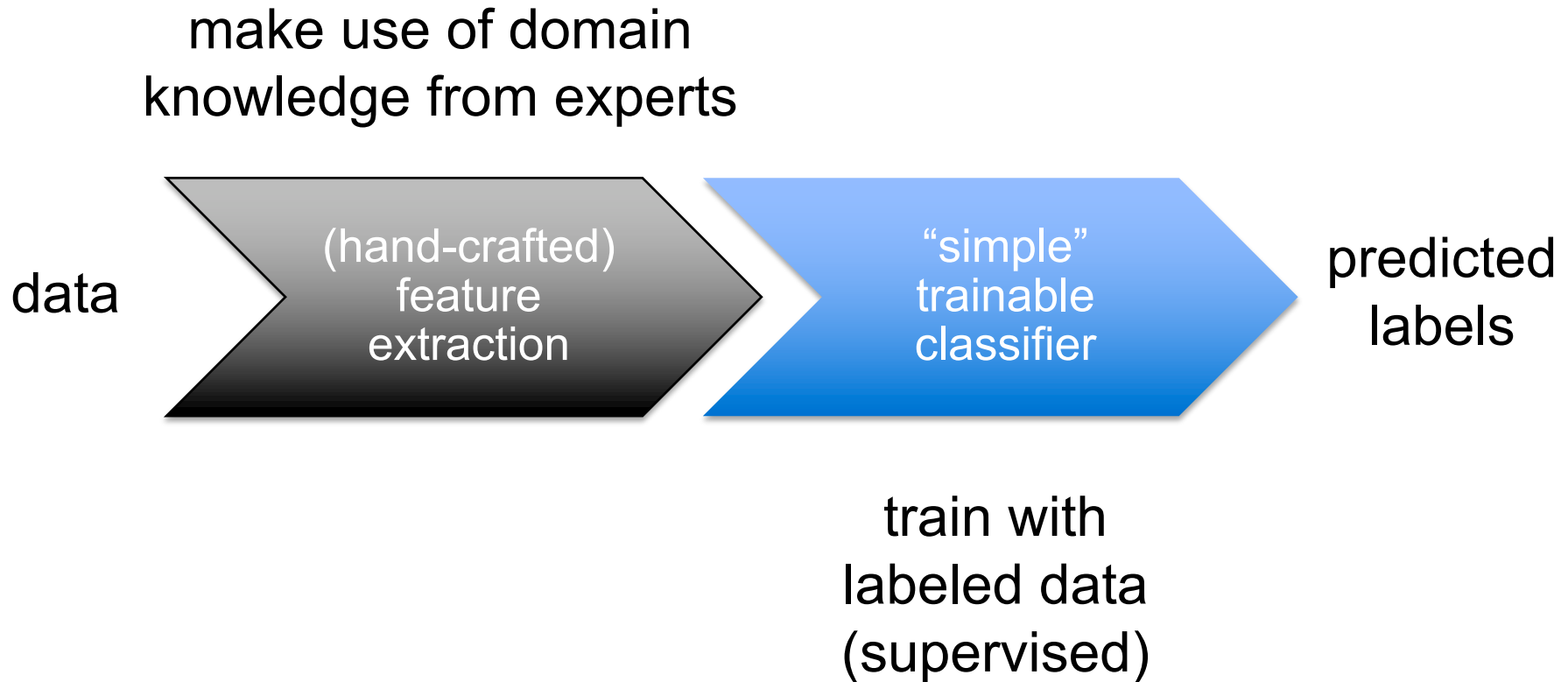
- A coffee mug though the “eyes” of a road-sign detector in 2003:



representation
to be used by classifier

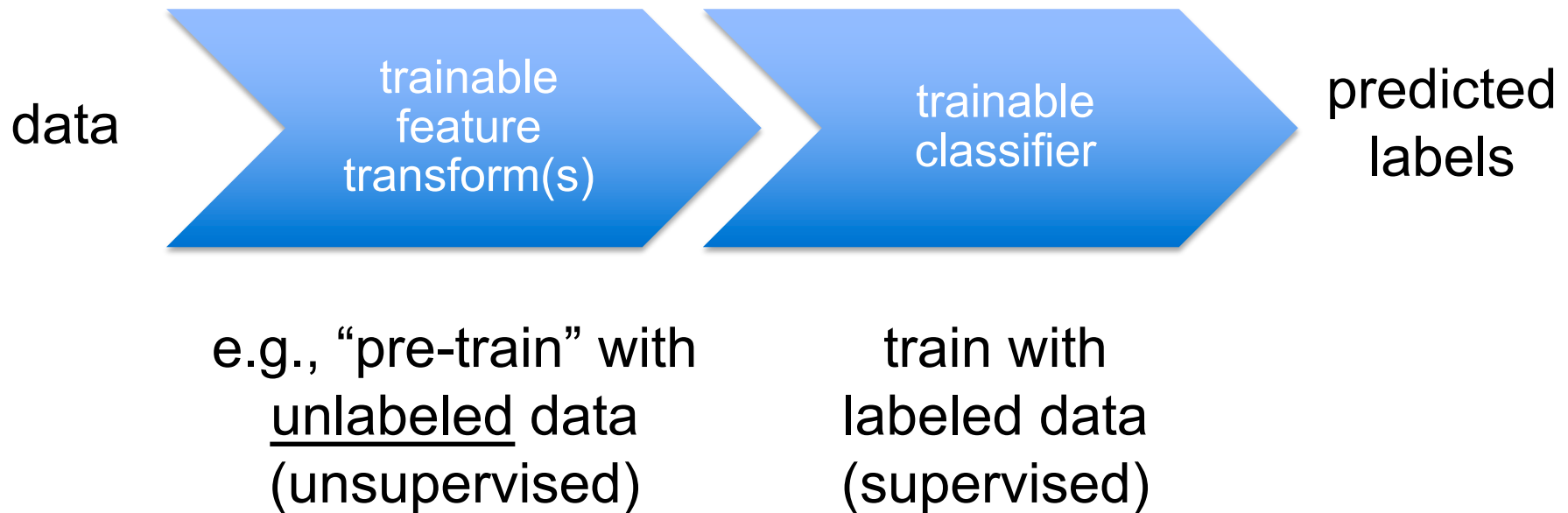


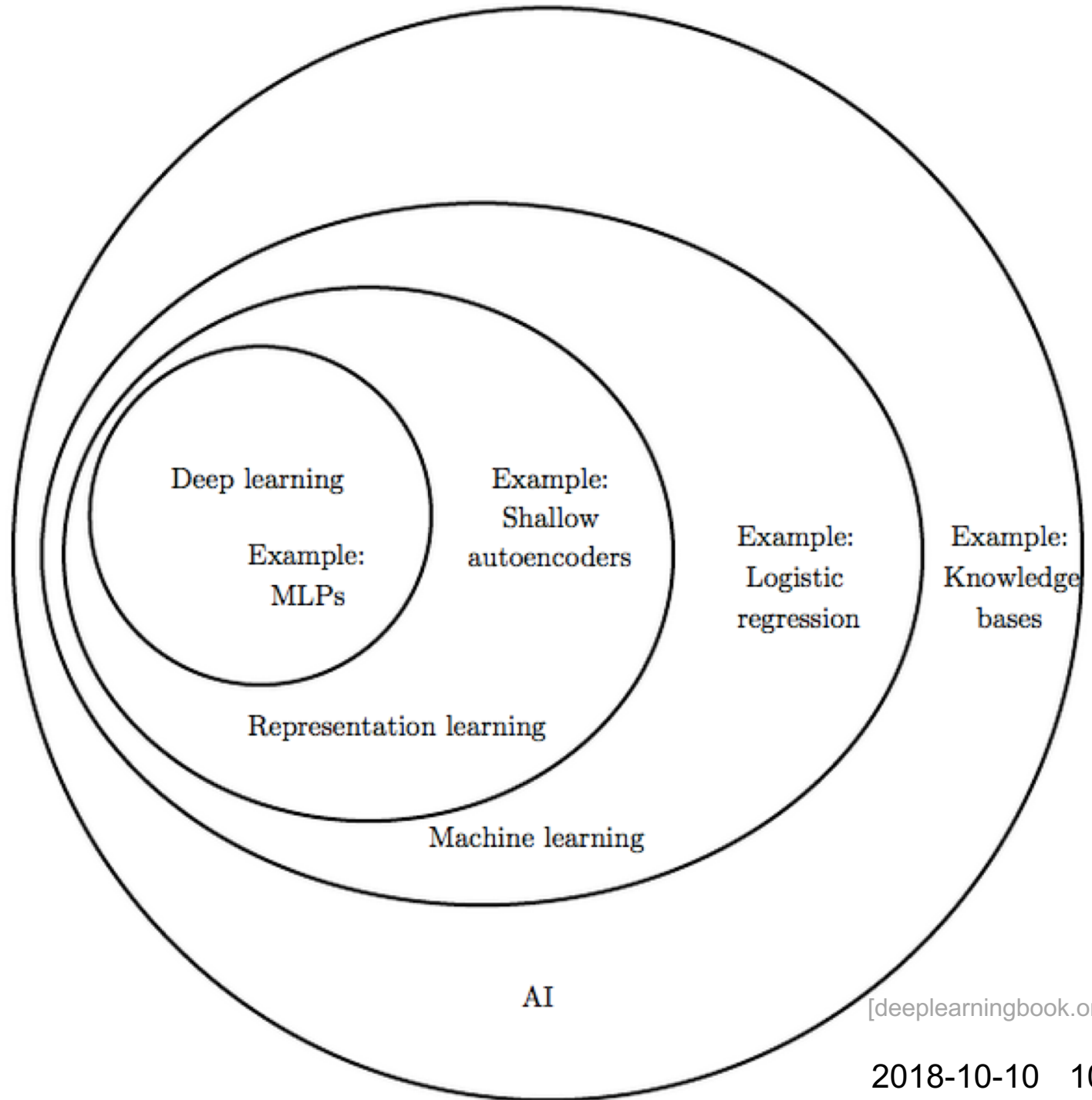
Typical Machine Learning Workflow (for Classification)



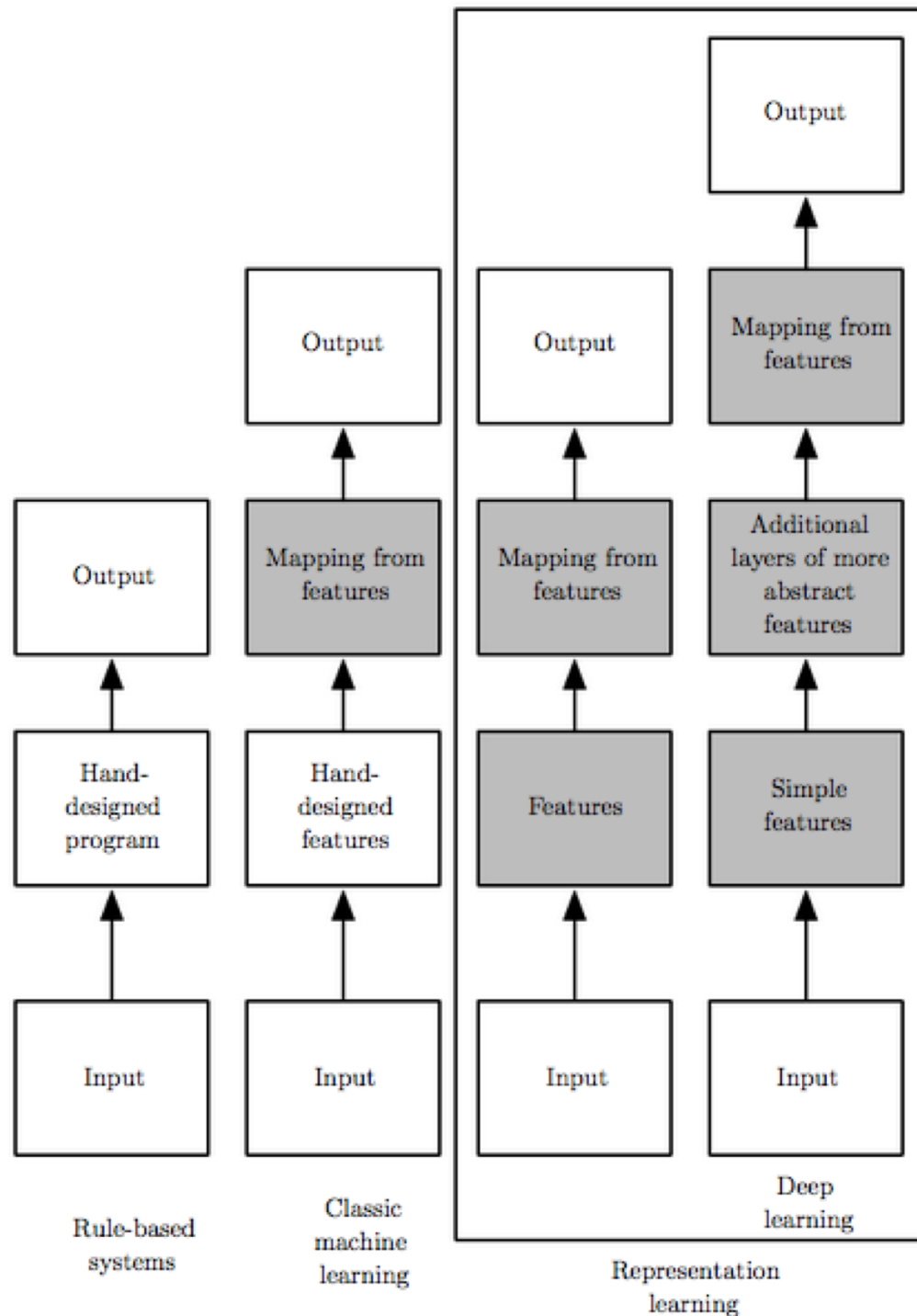
Typical Deep Learning Workflow (for Classification)

make use of abundant data
and (GPU) compute power



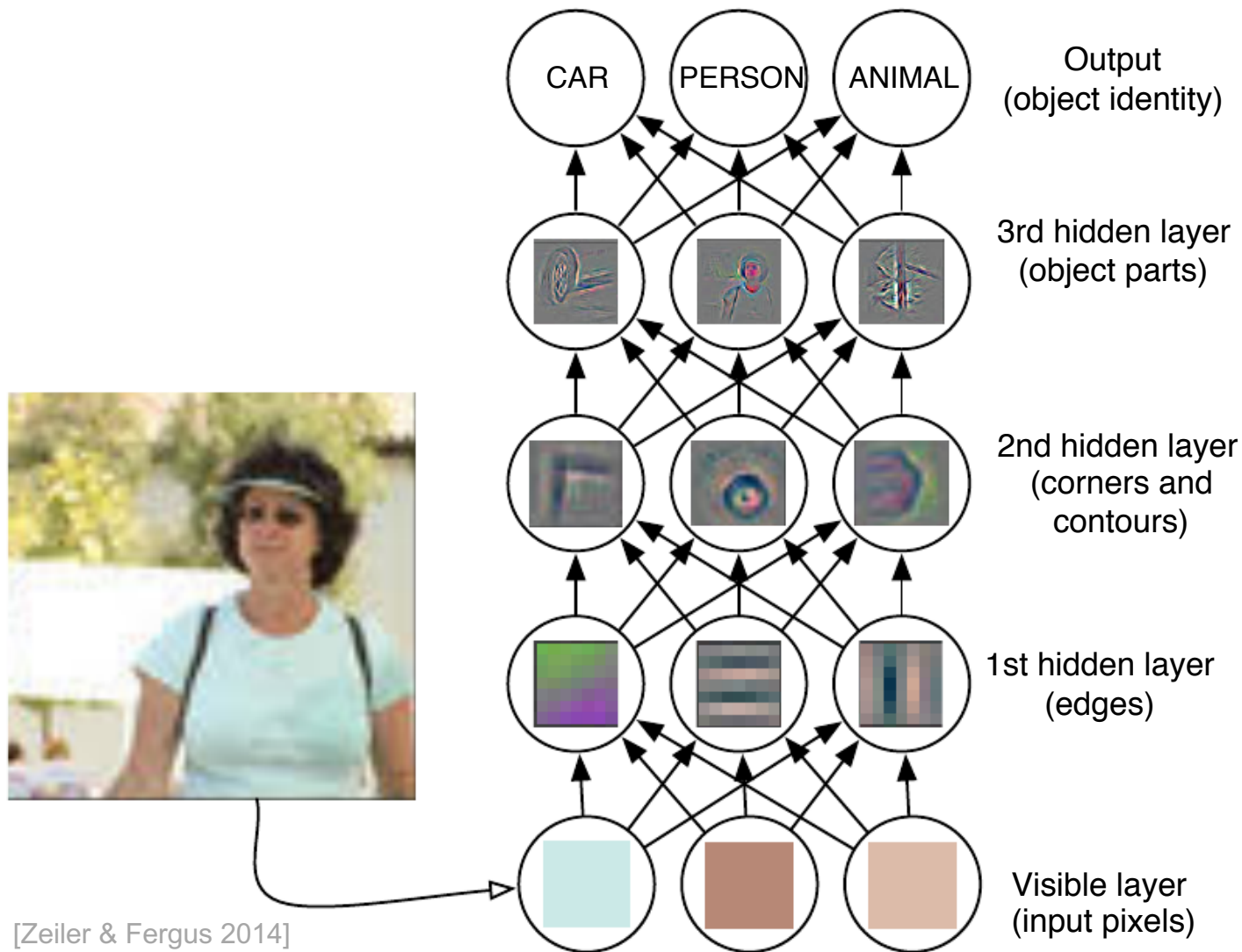


[deeplearningbook.org]



[deeplearningbook.org]

An Example Deep Net



[Zeiler & Fergus 2014]

The Promise of Deep Learning

- learn suitable feature representations along with the actual learning task
- using a general-purpose learning procedure

Course Rationale & Design

Learning Goals

- Think – Pair – Share
 1. Think about your personal learning goals for this course!
 2. Discuss with your neighbors & create a ranking!
 3. Name your most important one!

Overall Learning Goals

- At the end of the course, you are able to ...
 - **confidently** apply DL techniques to develop a solution for a given problem
 - follow recent DL publications and **critically** assess their contributions
 - formulate **hypotheses** and design & conduct DL experiments to **validate** them
 - **document** progress & design decisions for **reproducibility** and transparency

Please add your goals!

No Free Lunch!

This is **not** how
you will learn...

you will need to
participate



The Nuremberg Funnel (1647)

Disclaimer



this course may not be suitable for ...

- mere credit collectors
- passive attendees
- remote students
- the lighthearted ;-)

cc-by-nc-nd Lord-Psymon
<http://www.deviantart.com/art/Here-Be-Dragons-172141393>

Supportive Learning Environment

- encourage active participation
- be inclusive & open
 - leverage diversity
 - stick to course language
- build on mutual respect
- promote positive failure culture
- minimize distractions
 - arrive in good time
 - switch devices to silent mode

Course Design

Preparation

(session summary)

reading

- book chapters
- papers
- blogs

issues / discussions

weekly blog posts*

- new insights
- hints, tricks & hacks
- open questions

3-6h per week

In Class

“last episode on IDL”
(3-min summary)

literature
discussion / Q&A
small-group activity

project / assignments
discussion / Q&A

150h total

Practical Work

weekly programming
assignments
(~ until week 8)

issues / discussions

course project
working in teams

- 2-4 team members
- scrum-style
weekly sprints
- team progress blog

3-6h per week

grading: written exam (2h)

Programming Assignments

- can be done in groups of up to 3 students (everybody must be able to explain solution)
- do not have to be completely correct (substantial effort!)
- copied/adapted code requires reference(s)
- solutions discussed in exercise groups
- failing to explain solution => +1 fail
2 fails => exclusion from exam

Qualifying for the Exam

- $\geq 80\%$ of the weekly assignments in time (including being able to present the solution on request during the respective exercise session)
- significant contribution to the course project (at least one blog post and progress report) and the resulting group paper

for further details, see website
<https://ovgu-ailab.github.io/idl2018/>

Exercise Groups

- group 1
Thursdays 13-15:00, G29-K058
start: Oct 18
- group 2
Fridays 11-13:00, G29-335
start: Oct 19
- group 3
Mondays 11-13:00, G29-E037
start: Oct 22

*Sign up in LSF until Oct 14!
Specify priorities!*

Topics (Tentative)

- MLPs, Gradient Descent & Backpropagation
- Convolutional Neural Networks
- Recurrent/Recursive Neural Networks
- Auto-Encoders
- Regularization Techniques
- Advanced Regularization Techniques
- Introspection & Inception
- Optimization Techniques
- Advanced Training Strategies

Online Tools

- Mattermost (~Slack) TBA
 - messaging, discussions
- GitLab <https://code.ovgu.de>
 - issues: Q&A
 - wiki: blogs, documentation
 - snippets
- Google Colab <https://colab.research.google.com>
- GPU compute environment TBA
 - shell access & jupyterhub for notebooks

Mattermost

The screenshot displays the Mattermost web interface. On the left is a dark sidebar with a list of channels. The top of the sidebar shows the user 'IDL 2018 @sstober'. Below this are sections for 'FAVORITE CHANNELS', 'PUBLIC CHANNELS', and 'PRIVATE CHANNELS'. The 'Town Square' channel is selected and highlighted. The main area on the right shows the chat history for 'Town Square'. At the top of the chat area, it says 'Town Square' with a dropdown arrow, and links to 'provide feedback' and 'course website'. There are icons for 28 users, a star, a search bar, and icons for mentions and flags. The chat history shows several messages:

- jens** 09:25: sorry, I forgot about that. I'm adding some comments right now and then I'll upload it
- jens** 09:37: see [#Assignments](#) channel
- Tue, Jul 10, 2018**
- marius** 21:02: Servus! I'm struggling a bit with masking data gaps. In most of our 2D data features containing seismogram amplitudes we have a couple of traces that are broken. Means there is no data to fill into our data array for that trace. Imagine a 2D image where entire rows of pixels are empty. First attempt was to fill them with zeros, hoping the network would learn to ignore those 'infinite' gradients at the edges. But that is actually what bothers me. So, next we will try filling them with random numbers but I was wandering if there are more elegant or efficient ways to handle data gaps. (edited)
- sstober** 21:13: would it make sense to interpolate from the neighboring rows?
- marius** 21:19: Nice idea, we will try that! That will add some bias which might have an influence on other parameters which we would like to invert in a later stage, but for location I guess this will be the best way. Thanks 🙌
- Wed, Jul 25, 2018**
- sstober** 15:10: @all There will be a power shutdown by the university on Friday morning at 4am. This will impact the complete cluster and the Mattermost system. We will use this opportunity also for some maintenance. The plan is to be fully operational again in the afternoon. Please make sure to complete any running jobs by Thursday evening! (edited)
- Thu, Jul 26, 2018**
- maxi** 10:00: I just wanna let you know that I'm uploading my lecture summaries for learning purposes in the channel "x maxi". If you find errors or have questions, let me know.
- Fri, Jul 27, 2018**
- sstober** 13:45: @all The cluster is up again. You should be able to log in. Please post if you encounter any problems!

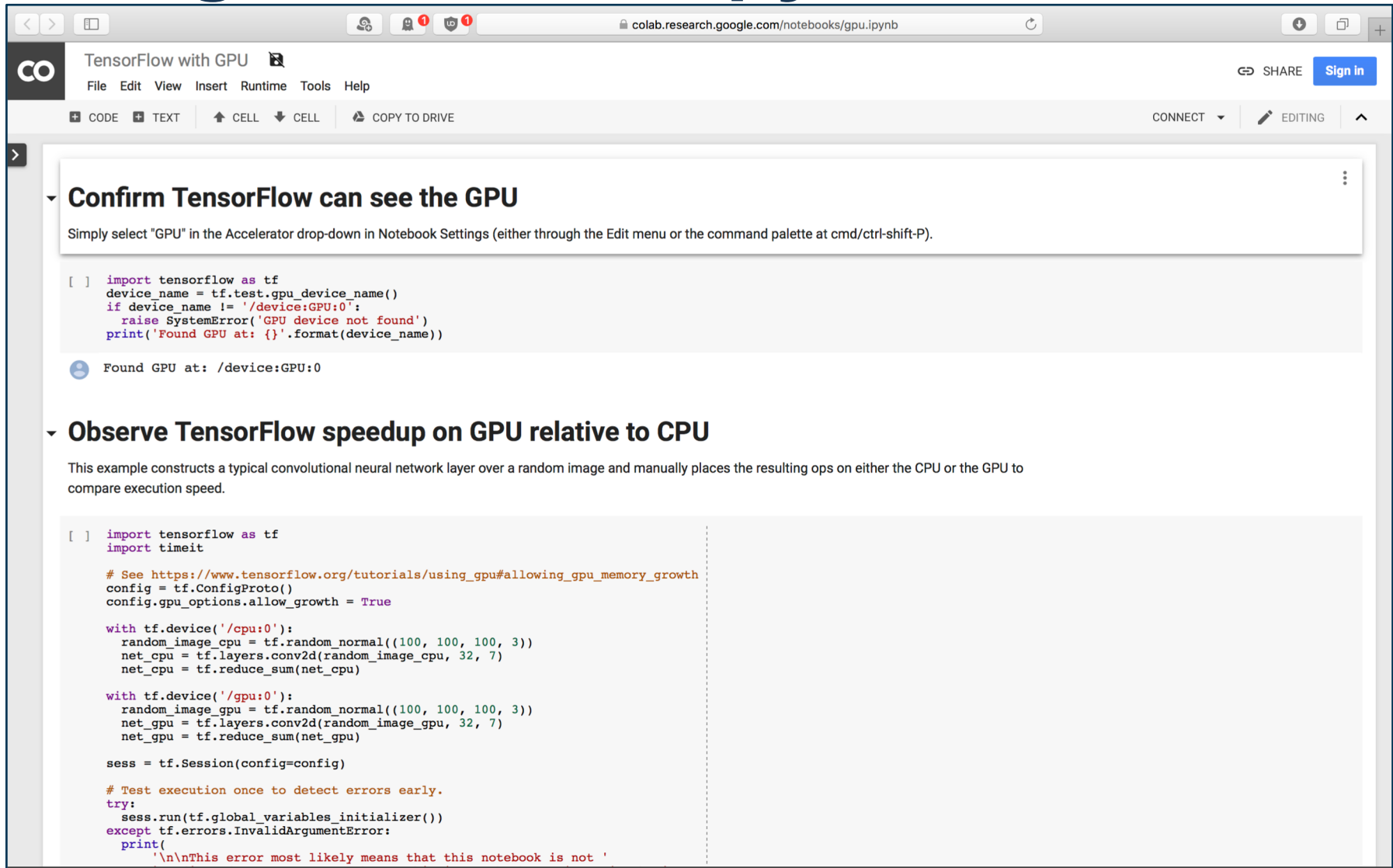
At the bottom of the chat area is a text input field with the placeholder 'Write a message...' and icons for attachments and emojis. In the bottom right corner of the interface are links for 'Preview' and 'Help'.

The screenshot shows a web browser window displaying a GitLab Snippets page. The browser's address bar shows the URL `code.ovgu.de/stober/idl2018/snippets/3`. The GitLab navigation bar at the top includes links for Projects, Groups, Activity, Milestones, Snippets, and a search bar. The left sidebar shows the user 'idl2018' and navigation options: Project, Issues (0), Wiki, Snippets (selected), and Settings. The main content area shows the snippet details for 'Sebastian Stober > idl2018 > Snippets > \$3'. It indicates the snippet was authored 22 hours ago by Sebastian Stober and provides 'Edit', 'Delete', and 'New snippet' buttons. The snippet title is 'Check for available GPUs'. Below the title, the file `tf_test_gpu.py` (51 Bytes) is shown with a copy icon. The code content is as follows:

```
1 import tensorflow as tf
2 tf.test.is_gpu_available()
```

Below the code, there are buttons for thumbs up (0), thumbs down (0), and a smiley face. At the bottom of the snippet view is a red 'Report Abuse' icon. Below this is a comment section with a 'Write' tab (selected) and a 'Preview' tab. The 'Write' tab contains a text area with the placeholder 'Write a comment or drag your files here...' and a rich text editor toolbar with icons for bold, italic, quote, code, list, link, and image. A note states 'Markdown is supported' with a link to 'Attach a file'. At the bottom of the comment section is a 'Comment' button and a dropdown arrow. The bottom of the sidebar has a 'Collapse sidebar' button.

Google Colab / Jupyterhub



The screenshot shows a Google Colab notebook interface. The browser address bar displays `colab.research.google.com/notebooks/gpu.ipynb`. The notebook title is "TensorFlow with GPU". The menu bar includes "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". Below the menu bar, there are tabs for "CODE", "TEXT", "CELL", and "COPY TO DRIVE". On the right side, there are buttons for "CONNECT", "EDITING", and "Sign in".

The notebook content is divided into two sections:

- Confirm TensorFlow can see the GPU**
Simply select "GPU" in the Accelerator drop-down in Notebook Settings (either through the Edit menu or the command palette at cmd/ctrl-shift-P).
- Observe TensorFlow speedup on GPU relative to CPU**
This example constructs a typical convolutional neural network layer over a random image and manually places the resulting ops on either the CPU or the GPU to compare execution speed.

The code in the first cell is as follows:

```
[ ] import tensorflow as tf
device_name = tf.test.gpu_device_name()
if device_name != '/device:GPU:0':
    raise SystemError('GPU device not found')
print('Found GPU at: {}'.format(device_name))
```

The output of the first cell is:

```
Found GPU at: /device:GPU:0
```

The code in the second cell is as follows:

```
[ ] import tensorflow as tf
import timeit

# See https://www.tensorflow.org/tutorials/using_gpu#allowing_gpu_memory_growth
config = tf.ConfigProto()
config.gpu_options.allow_growth = True

with tf.device('/cpu:0'):
    random_image_cpu = tf.random_normal((100, 100, 100, 3))
    net_cpu = tf.layers.conv2d(random_image_cpu, 32, 7)
    net_cpu = tf.reduce_sum(net_cpu)

with tf.device('/gpu:0'):
    random_image_gpu = tf.random_normal((100, 100, 100, 3))
    net_gpu = tf.layers.conv2d(random_image_gpu, 32, 7)
    net_gpu = tf.reduce_sum(net_gpu)

sess = tf.Session(config=config)

# Test execution once to detect errors early.
try:
    sess.run(tf.global_variables_initializer())
except tf.errors.InvalidArgumentError:
    print(
        '\n\nThis error most likely means that this notebook is not '
```


Your Personal Learning Blog

@GitLab wiki

- document your learning / project progress
 - ~one post per week
 - share your experiences!
 - visible only to course participants
- examples:
 - <https://deeperandommumbling.wordpress.com/>
 - <http://bartvanmerrienboer.nl/#blog>
- guidelines:
 - <https://www2.uwstout.edu/content/profdev/rubrics/blogrubric.html>

Session Summaries

last episode on
“Introduction to Deep Learning”

...

- short summary blog post (in GitLab wiki)
 - + 3-min intro recap at next session
 - key topics
 - results of the discussion
 - optional photos

Q&A – GitLab Issues

- guide for what is covered in class
deadline: Tuesday noon (12:00)
- do not hesitate to post questions!
(If you got one, you are probably not the only one!)
- upvote questions you find important!
- post a comment if you know the answer!

Contribute!

- ask – file new issues / upvote existing ones
- comment / like / discuss
- answer
- document
 - hints, tricks & hacks => snippets
- recommend
 - additional readings (papers, blogs, etc.)
- give (constructive) feedback

Next Steps for You!

- decide whether you want to take this class
- sign up for exercise groups
 - students accepted to an exercise group will receive invitations to Mattermost & GitLab
- start with the first reading and programming assignments listed at <https://ovgu-ailab.github.io/idl2018/schedule.html>

Course Project

Course Project

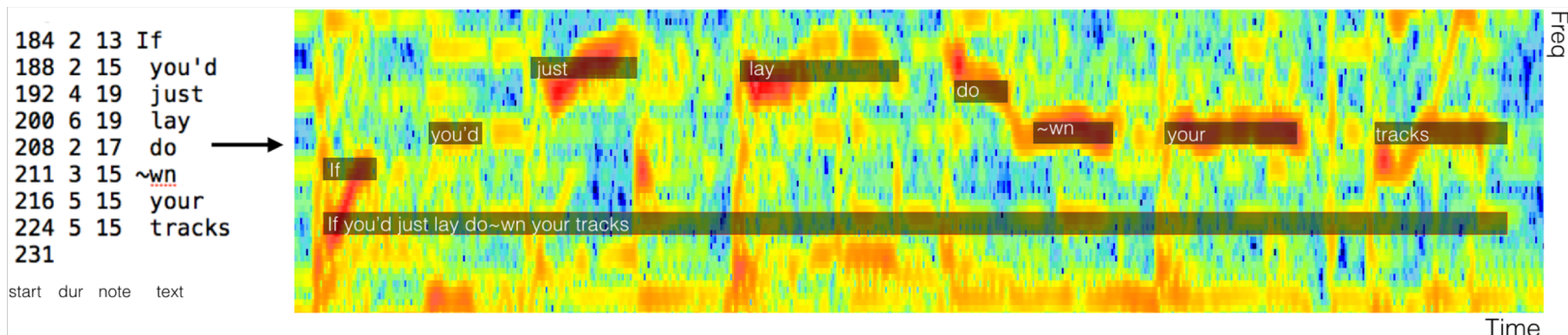
- collaborative effort (within exercise groups)
- “coopetition” (cooperative + competition)
- multiple teams of 2-4 students:
 - discuss ideas and form team
 - self-organized (heterogeneous if possible)
 - scrum-like approach
 - focus on different goals / aspects / strategies

Course Project Teams

- weekly progress report (blog + presentation)
 - similar to scrum
 - compare original goals with outcomes
 - What has worked well?
 - What did not work / had to be changed?
 - outline plan for next week
 - What would you like to try / investigate next?
- rotating job (each team member has to present at least once)

Course Project Topic

- speech recognition for sung lyrics
 - a) without background music (solo voice)
 - b) with background music



[http://ismir2018.ircam.fr/doc/pdfs/35_Paper.pdf]

Course Project Topic

- speech recognition for sung lyrics
 - a) without background music (solo voice)
 - b) with background music
- potential datasets from recent work:
 - <https://ccrma.stanford.edu/damp/>
 - http://ismir2018.ircam.fr/doc/pdfs/30_Paper.pdf
 - http://ismir2018.ircam.fr/doc/pdfs/35_Paper.pdf