# Deep Neural Network Evaluation Helper

## Project Presentation

Huang Ru, Li Mingyi, Xaver Wangerpohl

Sommersemester 2020

# Responsibilities

- Li: Environment setup, Team coordination, Software architecture, OOP structure implementation, Code/Algorithm optimization and Quality assurance
- Huang Ru: Deep Neural Network implementation
- Xaver: Questionnaires processing and distortion computation

# Development Tools & Environment

- Slack: Team communication
- Zoom: Online meeting
- PyCharm Community Edition: IDE
- Git & GitHub: Version control
- Anaconda with Python 3.8: Package management and Environment

# Software Architecture

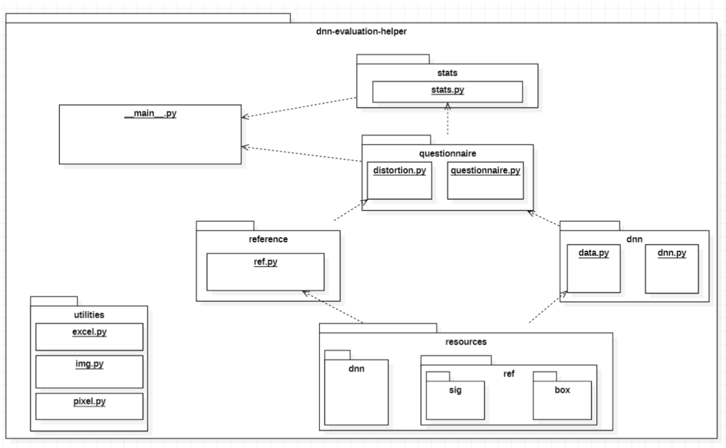

Figure: Strict Layered Architecture

# Highlight 1: Well defined ”utilities” package

Total encapsulation of image and excel libraries access

- protects other teammates from making mistakes
- creates a single point of responsibility
- keeps the other modules clean from multiply library imports

# Highlight 2: Auto-generation of ”masks”

Users and developers are able to add or delete a ”reference point of significance” by simply modifying the corresponding excel file in ”resources” package.

# Highlight 3: Improved algorithm in finding the matching "reference points of significance"

We pre-filtered the interested region in a questionnaire image where the potential matching points are.
By doing this, a 100% accurate match is achieved

## Highlight 4: Extra generating of labelled questionnaire images for quality assurance

Apart from the statistics in excel, the output artifacts also includes labelled questionnaire images for quality assurance.

# Highlight 5: Multiprocessing

Multiprocessing is used to process questionnaire images and generate labelled images.
The run time is about 3.5 times faster on a 4-core machine than the single process version. (no guaranteed statistical significance)

# Problem 1: Unable to achieve Cross-platform

Our application only runs on Windows10.
Due to the unsolved bugs in the "pickle" module that is used to serialize and deserialize the deep neural network object, the trained deep neural network can not be loaded on Mac if the saving happens on Window10.

# Problem 2: Imperfection in our Deep Neural Network

The correctness of our trained deep neural networks can reached up to 99.45% in the validation process.
But still, wrong recognition of crosses and empties are discovered when we applied the trained deep neural network to real questionnaires.

- Not enough or not versatile enough data sets?
- Imperfect parameters?
- The limitation of the algorithm itself?