Teacher Artifact

> Supplementary materials of the article *Inspecting Technology-related Quality of Teaching Artifacts to Understand Teachers' Technology Adoption (under revision)*

Teaching artifacts encapsulate the attributes that can reflect teacher expertise in the authentic context. The paper inspects the implicit quality of teaching artifacts to examine teachers' technology adoption. This repository contains the core code of training machine annotators, samples of training data, and runtime environments addressed in the article.

Code

The code of training machine learning annotators is presented within the Jupyter notebooks. Each notebook demonstrates how we trained an annotator with the artifact samples we collected in the study using deep learning libraries. In each notebook, the data pre-processing, performance metrics, and optimization processes are fully detailed.

Trained Annotators (Machine Learning Techniques):

- Classification of self-created artifact category (a joint deep learning model using integrated data)
- Segmentation of design elements in artifacts (UNet)
- Detection of design elements in artifacts (YOLOv3)
- Classification of material structuring principle (CNN)
- Classification of instruction events (a joint deep learning model using integrated data)

Libraries and Environments

> Running the code in the notebooks requires the dependent libraries or a corresponding environment. The config files of the environment are in the envs folder.
>
> The joint deep leaning model that we trained relies on the fantatic Fastai **v1** and image_tabular library. Before runing the notebooks (Artifact_technology_classification.ipynb and Artifact_instruction_event_classification_integrated_model.ipynb), please install two libraries. Alternatively, download the environment file env.yml and run

```
coda env create -f environment_jointDeepLearning.yml
```

in the terminal or Anaconda Prompt to create a new environment.

The UNet model that we trained for artficat segmentation relies on the Fastai **v2** . Before runing the notebook, please install Fastai v2 library. Alternatively, run

```
coda env create -f environment_segmentation.yml
```

in the terminal or Anaconda Prompt

The Yolo that we trained for the detection of design elements relies on iceVision library. Before runing the notebook, please run

```
coda env create -f environment_icevision.yml
```

to install the icevision framework and the pretrained models of object detection.

The CNN classifier that we trained to code the material structuring principles of artifact relies on the Fastai **v2** library . before running the notebook, please install fastai v2 library. Alternatively, run

```
coda env create -f environment_classification.yml
```

in the terminal or Anaconda Prompt to create a new environment.

Data

We provided a few data samples for each training task in the folder data. In the design element analysis (segmentation & detection), we utilized labelme, a graphical image annotation tool, to collect the regions of the design elements.