ADAMS

Advanced **D**ata mining **A**nd **M**achine learning **S**ystem

Module: adams-imaging-ext



Peter Reutemann

November 21, 2026

©2023



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato



Except where otherwise noted, this work is licensed under
http://creativecommons.org/licenses/by-sa/4.0/

Contents

1 Tools **5**

2 (**7**
2.1 DEXTR 7

3 (**9**
3.1 OPEX 9
3.2 DEXTR 10
3.3 SAM 11

Bibliography **13**

Chapter 1

Tools

This module represents *extended* imaging support that relies on *external* tools, like deep learning docker containers, to provide additional functionality.

Chapter 2

(

Image segmentation)

2.1 DEXTR

DEXTR[2] (Deep Extreme Cut) is a very useful tool for aiding human annotators in annotating images. The tool generates object masks using just four *extreme points* (x/y pairs) on the edges of the object that is to be annotated.

The *ImageSegmentationAnnotator* transformer offers a DEXTR tool in its toolbox. In order to use the tool, you need the following:

- A computer (preferably a Linux machine) with Redis[3] and Docker[4] running, this can be the local machine.
- Run the following flow to start a DEXTR container: *adams-imaging-ext_run-dextr.flow*.
- Run *adams-imaging-image_segmentation_annotation.flow* to annotate JPEG images with the help of DEXTR.
  - In the annotation interface, select the *DEXTR* tool.
  - Adjust any connection parameters if necessary, though defaults work with the above flow when run on the local machine, and click on the tick button to connect to the Redis instance. The cursor, when hovering over the image, should be cross-hair now.
  - With the left mouse button, perform four left-clicks to choose the four extreme points (use the right mouse button to remove the current points and start over) and hit *ENTER* to send image and points to the docker container for processing (on a CPU, this can take a couple of seconds).
  - If successful, the mask will be added to the current layer.

Chapter 3

(

Object detection)

3.1 OPEX

The *OPEX* tool, available through the interface of the *ImageObjectAnnotator* transformer, allows you to send the current image to a docker container that runs a deep learning model which returns object detection predictions in OPEX format[5].

In order to use the tool, you need the following:

- A computer (preferably a Linux machine) with Redis[3] and Docker[4] running, this can be the local machine.
- Run the following flow to start an OPEX-outputting container, e.g., using Yolov5:
  adams-imaging-ext_run-yolov5.flow.
- Run *adams-imaging-annotate_objects.flow* to annotate JPEG images with the help of OPEX.
  - In the annotation interface, select the *OPEX* tool.
  - Adjust any connection parameters if necessary, though defaults work with the above flow when run on the local machine, and click on the tick button to connect to the Redis instance. The cursor, when hovering over the image, should be cross-hair now.
  - Click anywhere on the image while holding the *SHIFT* key down, to send the image to the docker container for processing (even on a CPU, this will be very quick).
  - If successful, the predicted bounding boxes will be overlayed over the image.

3.2 DEXTR

DEXTR[2] (Deep Extreme Cut) is a very useful tool for aiding human annotators in annotating images. The tool generates object masks using just four *extreme points* (x/y pairs) on the edges of the object that is to be annotated. These masks can be converted into polygons and integrated into the annotations.

Both, the *ImageObjectAnnotator* and *ImageSegmentationAnnotator* transformers offer a DEXTR tool in their toolboxes. In order to use the tool, you need the following:

- A computer (preferably a Linux machine) with Redis[3] and Docker[4] running, this can be the local machine.
- Run the following flow to start a DEXTR container:
  adams-imaging-ext_run-dextr.flow.
- Run *adams-imaging-annotate_objects.flow* or *adams-imaging-image_segmentation_annotation.flow*
  to annotate JPEG images with the help of DEXTR.
  - In the annotation interface, select the *DEXTR* tool.
  - Adjust any connection parameters if necessary, though defaults work with the above flow when run on the local machine, and click on the tick button to connect to the Redis instance. The cursor, when hovering over the image, should be cross-hair now.
  - With the left mouse button, perform four left-clicks to choose the four extreme points (use the left mouse button while holding CTRL to remove the current points and start over) and hit *ENTER* to send image and points to the docker container for processing (on a CPU, depending on the image size, this can take a couple of seconds).
  - If successful, the annotation(s) will be added to the current ones.

3.3 SAM

SAM[6] (Segment Anything Model) is another very useful tool for aiding human annotators in annotating images. The tool generates object masks using one or more points as prompt within the object that is to be annotated. These masks can be converted into polygons and integrated into the annotations.

Both, the *ImageObjectAnnotator* and *ImageSegmentationAnnotator* transformers offer a SAM tool in their toolboxes. In order to use the tool, you need the following:

- A computer (preferably a Linux machine) with Redis[3] and Docker[4] running, this can be the local machine.
- Run the following flow to start a DEXTR container:
  adams-imaging-ext_run-sam.flow.
- Run *adams-imaging-annotate_objects.flow* or *adams-imaging-image_segmentation_annotation.flow* to annotate JPEG images with the help of DEXTR.
  - In the annotation interface, select the *SAM* tool.
  - Adjust any connection parameters if necessary, though defaults work with the above flow when run on the local machine, and click on the tick button to connect to the Redis instance. The cursor, when hovering over the image, should be cross-hair now.
  - With the left mouse button, perform one or more left-clicks to choose the prompt points (use the left mouse button while holding CTRL to remove the current points and start over) and hit *ENTER* to send image and points to the docker container for processing (on a CPU, depending on the model and image size, this can take up to a minute).
  - If successful, the annotation(s) will be added to the current ones.

Bibliography

[1] *ADAMS* – Advanced Data mining and Machine learning System
https://adams.cms.waikato.ac.nz/

[2] K.K. Maninis*, S. Caelles*, J. Pont-Tuset, and L. Van Gool. *Deep Extreme Cut: From Extreme Points to Object Segmentation*, Computer Vision and Pattern Recognition (CVPR), 2018.
https://cvlsegmentation.github.io/dextr/
https://github.com/Britefury/dextr

[3] *Redis* – an in-memory data structure store.
https://redis.io/

[4] *Docker* – offers OS-level virtualization of software using containers
https://www.docker.com/

[5] *OPEX* – Exchange format for predictions from object detection algorithms (JSON).
https://github.com/WaikatoLink2020/objdet-predictions-exchange-format

[6] Kirillov, Alexander and Mintun, Eric and Ravi, Nikhila and Mao, Hanzi and Rolland, Chloe and Gustafson, Laura and Xiao, Tete and Whitehead, Spencer and Berg, Alexander C. and Lo, Wan-Yen and Dollar, Piotr and Girshick, Ross. *Segment Anything*, 2023.
https://arxiv.org/abs/2304.02643 https://github.com/facebookresearch/segment-anything