**A blue text on a white background

Description automatically generated**

**Object Detection Extension**

**Guidelines**

**Contents**

1. **Introduction** ...............................................................................................................................1
2. **Installation & Setup** ....................................................................................................................1
   1. **Object Detection Extension Dependencies**...............................................................................
   2. **Setup of Python Environments**................................................................................................3
   3. **Setup of Extension JAR Files**
3. **Using Object Detection Extension** ...............................................................................................5
   1. **Using Fine Tune Object Detection Model Operator**...............................................................5
   2. **Using Object Detection Model Inference**
4. **Introduction**

The Object Detection Extension for RapidMiner allows users to fine tune a pre-trained object

detection model on a new set of image data and do inferencing using the fine-tuned model on

any new image. The extension has support for FASTER-RCNN, FCOS,SSD, SSDLite , RetinaNet model. The extension comes with two operators,

* **FineTuneObjectDetectionModel**: Used for fine-tuning a pre-trained model on a new set

of images.

* **ObjectDetectionModelInference**: Used for inferencing on a new image using the fine

tuned model.

1. **Installation & Setup**

The following paragraphs describe the requirements of the Object Detection Extension. Please

make sure that you provide the package manager, the exact Python environments, and are

using the correct versions as described below since otherwise the operators in this extension

will likely not work.

* 1. **Object Detection Extension Dependencies**

This extension depends on following extensions which need to be installed before this

one can be used:

**- Custom Operators >= 1.1.0**

**- Operator Toolbox >= 2.17.0**

**- Image Handling >= 0.2.1**

**- Python Scripting >= 10.0.1**

For RapidMiner Studio. Please go to the menu *Extensions -> Marketplace* and install above

mentioned extensions before you proceed with installing this extension. For RapidMiner AI Hub

please follow the instructions in the AI Hub documentation for installing all three extensions

there.

The Python scripting extension needs to be properly configured, i.e., in RM settings please

specify a working Python installation together with a default environment which needs to

contain at least the Pandas package.

A screenshot of a computer

Description automatically generated

* 1. **Setup of Python Environments**

This extension will install missing packages into the current environment automatically. Those

packages will be installed into the currently selected environment (which is a parameter for all

the operators). So, if you do not want to make changes to an environment used elsewhere, we

do recommend generating a new environment specifically for usage with this extension.

Also, and because of this, you will need to use Conda as the package manager which needs to

be installed first. Please refer to

https://docs.conda.io/projects/miniconda/en/latest/miniconda-install.html for a how to in case

you have not installed Conda yet.

After installing Conda, we advise creating a new environment with Conda which only will be

used by this extension. This is also recommended since you will need very specific package

versions to ensure that everything is working. You must be using the following Python versions

and basic packages with the specified versions for this package:

**- Python 3.10.8**

**- Numpy 1.23.2**

**- Pandas 1.5.2**

Using Conda in a command line shell, please create a new environment called

**“rm\_obj\_detection”:**

conda create -n rm\_obj\_detection python=3.10.8 numpy=1.23.2 pandas=1.5.2 -c conda-forge

You will also need to install Pytorch. You will need to define a different version of it depending

on if you have access to a CUDA-enabled Graphics Processor Unit (GPU) on the system or not. It

will also depend on the used operating system.

**CPU Computation**

*If you do NOT have a CUDA-enabled GPU, the extension will run on CPUs, and you will need to run the following*

*commands in a command shell after installing conda. Please use the correct command for your system:*

Windows/Linux:

conda activate rm\_obj\_detection

pip install torch torchvision torchaudio

**GPU Computation**

*If you DO have a CUDA-enabled GPU in your system and want to run the operators in this extension on a GPU, you*

*will need to execute the following commands in a command shell after installing conda:*

conda activate rm\_obj\_detection

pip install torch torchvision torchaudio --index-url <https://download.pytorch.org/whl/cu118>

Refer https://pytorch.org/get-started/locally/ for more information.

Also, after the above installations please install the following packages inside rm\_obj\_detection

environment.

pip install pillow matplotlib chardet

* 1. **Setup of Extension JAR Files**

Place the provided .jar file inside .RapidMiner\extensions folder of the user directory and restart

RapidMiner

Example,

C:\Users\user\_name\.RapidMiner\extensions

**3.Using Object Detection Extension**

After the required dependencies are installed & .jar files are placed correctly, you can see the extension

inside Extension folder of Operators pane in RapidMiner Studio,

A screenshot of a computer

Description automatically generated

**3.1 Using Fine Tune Object Detection Model Operator**

This operator is responsible for fine tuning a pre-trained object detection model

on a new set of images. User needs to provide the images directory, annotations

directory & results directory for storing the results. Apart from directories, user

can also provide hyperparameters values for fine tuning the model.

A screenshot of a computer

Description automatically generated

After the fine-tuning process is completed, user will get following files inside the

provided results directory,

A screenshot of a computer

Description automatically generated

* Class\_list\_df.csv : This file contains the class labels on which the model was trained.
* model\_state\_dict.pth: State dictionary of fine-tuned model.

These files will be used when inferencing is done on the fine-tuned model.

**3.2 Using Object Detection Model Inference Operator**

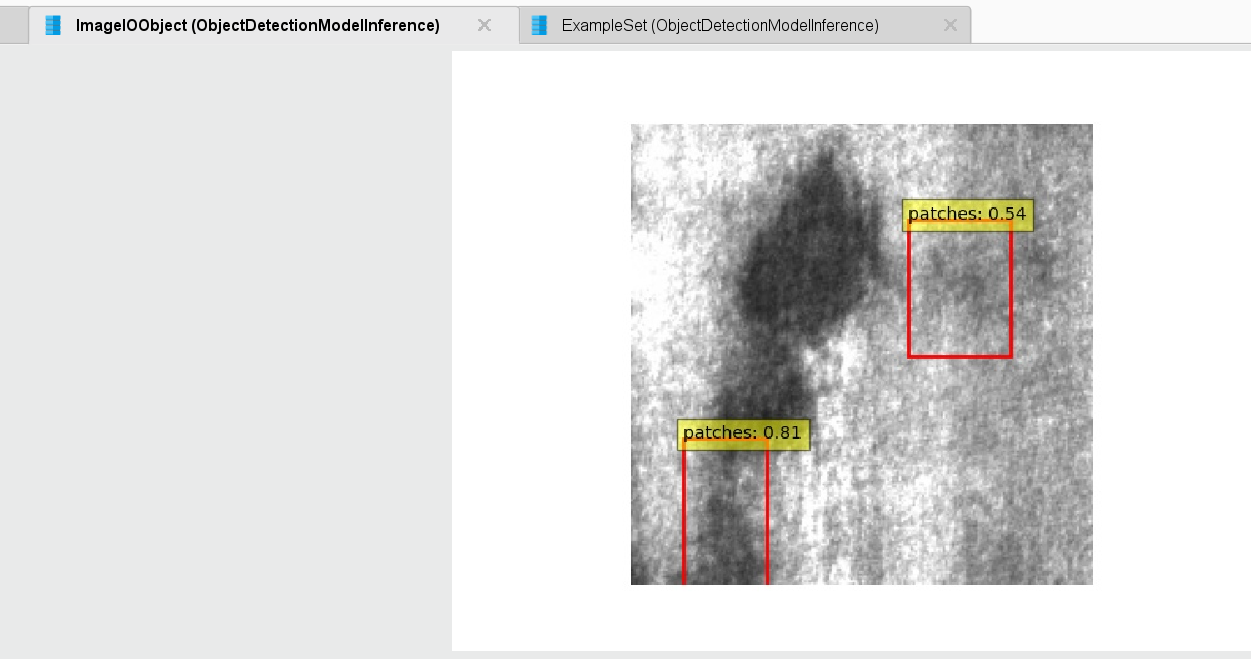
This operator is responsible for inferencing a new image using the fine-tuned model & class table. User

needs to provide the model path, image path & class table as an input for using this operator.

A screenshot of a computer

Description automatically generated

This operator will output the image with detected objects/labels al and a table with all the labels with their corresponding scores

 A screenshot of a computer

Description automatically generated