



# Java API Example

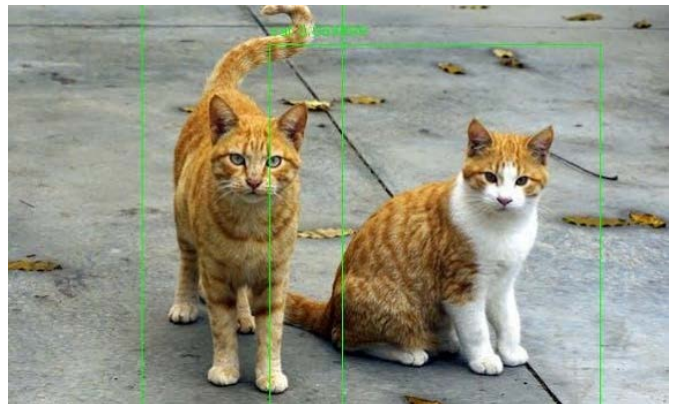
This sample application was created to present an example for TensorFlow Java API usage. It uses YOLOv2 model for object detection and Gradle as build and dependency management tool. TensorFlow Java API is a new opportunity to use TensorFlow from Java applications. It is well-suited to load models created in Python and execute them within a Java application, however it is still under development and it is not covered by the [Tensorflow API stability guarantees](#).

For more information about the installation please read the [official TensorFlow site](#).

## Tensorflow Java Example application

On the pictures below you can see how the application works. You can also try it out as a live demo on the following links:

- Deployed to Google Cloud with YOLOv9000 model: <http://35.229.93.105:8080/>
- Deployed to Heroku with *tiny-yolo* model: <https://still-crag-64816.herokuapp.com/>



TensorFlow example Java API - cats

TensorFlow example Java API - cats detected

1. cat - 0.94341564
2. cat - 0.864999

These demo applications are using Spring Boot and Spring MVC to create a web user interface, where you can upload images and detect the objects on them. The application is able to detect 20 classes of the Pascal VOC dataset: *aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, diningtable, dog, horse, motorbike, person, pottedplant, sheep, sofa, train, tvmonitor*. If you are interested in training YOLO models, please visit the following page: [Complete guide to train YOLO](#). If you would like to build a server application like our live demo, please check the [TensorFlow Java API with Spring](#) page, what presents you a more sophisticated example.

At this time we show you a more simple object detection project. The source code is available on the Github here: <https://github.com/szaza/tensorflow-example-java>. This project contains a [HelloTensorFlow](#) Java class to check if your configuration works well, beside this an [ObjectDetector](#) class is present, what reads the YOLO protobuff file and executes it by using the TensorFlow Java API. It communicates through Java Native Interface (JNI) with your installed TensorFlow instance.

## Installation of the TensorFlow Java API example project

Preconditions:

- Java JDK 1.8 or greater;
- TensorFlow 1.6 or grater;
- Git version control system;

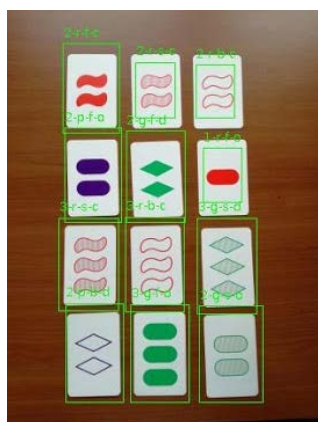
Strongly recommended to install:

- [nVidia CUDA Toolkit 8.0](#) or higher version;
- [nVidia cuDNN](#) GPU accelerated deep learning framework;

To compile and run this TensorFlow Java example application you have to do the following steps:

1. Clone the repository from the Github link: <https://github.com/szaza/tensorflow-example-java.git>;
2. Download the protobuff and label files from this link:  
<https://drive.google.com/drive/folders/1GfS1Yle7Xari1tRUEi2EDYedFteAOaoN>;

3. Place them under the `src/main/resources/YOLO` directory;
4. Copy your images into the `/sample` directory and modify the [configuration](#) file;
5. Build the application with Gradle by running the following command from the root directory:  
`./gradlew clean build;`
6. Now, you can start the application with: `./gradlew run;`
7. In the `/out/images` folder you can see the result;



SET cards were detected with YOLO using TensorFlow Java API.

Of course the model can be retrained for another classes and objects as well. It can be retrained either by using [Darknet](#) or [DarkFlow](#) frameworks. On the picture at the left side you can see that we retrained this model for specific cards. Because we didn't have any dataset with these specific cards we were needed to generate one as it is described on the [Dataset generator page](#). Probably, you also will need to generate a large set of training data to retrain your model.

Please follow this guide if you are interested to retrain your model for custom dataset: [Complete guide to train YOLO](#). In this tutorial we retrain the YOLO model for Winnie-the-Pooh (as visible on the picture at the right side) with DarkNet and DarkFlow frameworks.



Winnie-the-Pooh detected with YOLO.