

# ImageCaption

- [Tika and Computer Vision - Image Captioning](#)
  - [Tika and Tensorflow Image Captioning Using REST Server](#)
    - [Step 1. Setup REST Server](#)
      - [a. Using docker \(Recommended\)](#)
      - [b. Without Using docker](#)
    - [Step 2. Create a Tika-Config XML to enable Tensorflow parser.](#)
    - [Step 3. Demo](#)
  - [Questions / Suggestions / Improvements / Feedback ?](#)

## Tika and Computer Vision - Image Captioning

This page describes how to use the Image Captioning capability of Apache Tika. "Image captioning" or "describing the content of an image" is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. TIK-2262 introduced a new parser to perform captioning on images. Visit [TIKA-2262 issue on Jira](#) or [pull request on Github](#) to see the related conversations. Currently, Tika utilizes an implementation based on the paper [Show and Tell: A Neural Image Caption Generator](#) for captioning images. This paper presents a generative model based on a deep recurrent architecture that combines recent advances in computer vision and machine translation that can be used to generate natural sentences describing an image. Continue reading to get Tika up and running for image captioning.

## Tika and Tensorflow Image Captioning Using REST Server

We are going to start a python flask based REST API server and tell tika to connect to it. All the dependencies and setup complexities are isolated in the docker image.

Requirements :

Docker -- Visit [Docker.com](#) and install latest version of Docker. (Note: tested on docker v17.03.1)

### Step 1. Setup REST Server

You can either start the REST server in an isolated docker container or natively on the host that runs tensorflow v1.0

#### a. Using docker (Recommended)

[Toggle line numbers](#)

```
1 git clone https://github.com/USCDataScience/tika-dockers.git && cd tika-dockers
2 docker build -f Im2txtRestDockerfile -t uscdatascience/im2txt-rest-tika .
3 docker run -p 8764:8764 -it uscdatascience/im2txt-rest-tika
```

Once it is done, test the setup by visiting [http://localhost:8764/inception/v3/caption/image?url=https://upload.wikimedia.org/wikipedia/commons/thumb/1/1d/Marcus\\_Thames\\_Tigers\\_2007.jpg/1200px-Marcus\\_Thames\\_Tigers\\_2007.jpg](http://localhost:8764/inception/v3/caption/image?url=https://upload.wikimedia.org/wikipedia/commons/thumb/1/1d/Marcus_Thames_Tigers_2007.jpg/1200px-Marcus_Thames_Tigers_2007.jpg) in your web browser.

**Sample output from API:**

```
{
  "captions": [{
    "confidence": 0.010706611316269087,
    "sentence": "a baseball player swinging a bat at a ball"
  },
  {
    "confidence": 0.004686326913725872,
    "sentence": "a baseball player swinging a bat at a ball ."
  },
  {
    "confidence": 0.0041084865981657155,
    "sentence": "a baseball player swinging a bat on a field"
  }
],
  "beam_size": 3,
  "max_caption_length": 20,
  "time": {
    "read": 407,
    "captioning": 1632,
    "units": "ms"
  }
}
```

Note: MAC USERS:

If you are using an older version, say, 'Docker toolbox' instead of the newer 'Docker for Mac',

you need to add port forwarding rules in your Virtual Box default machine.

1. Open the Virtual Box Manager.
2. Select your Docker Machine Virtual Box image.
3. Open Settings -> Network -> Advanced -> Port Forwarding.
4. Add an appname, Host IP 127.0.0.1 and set both ports to 8764.

## b. Without Using docker

If you chose to setup REST server without a docker container, you are free to manually install all the required tools specified in the [docker file](#).

Note: docker file has setup instructions for Ubuntu, you will have to transform those commands for your environment.

### Toggle line numbers

```
1 python tika-parsers/src/main/resources/org/apache/tika/parser/captioning/tf/im2txtapi.py
```

## Step 2. Create a Tika-Config XML to enable Tensorflow parser.

A sample config can be found in Tika source code at [tika-parsers/src/test/resources/org/apache/tika/parser/recognition/tika-config-tflow-rest.xml](#)

Here is an example:

```
<properties>
  <parsers>
    <parser class="org.apache.tika.parser.recognition.ObjectRecognitionParser">
      <mime>image/jpeg</mime>
      <mime>image/png</mime>
      <mime>image/gif</mime>
      <params>
        <param name="apiBaseUri" type="uri">http://localhost:8764/inception/v3</param>
        <param name="captions" type="int">5</param>
        <param name="maxCaptionLength" type="int">15</param>
        <param name="class" type="string">org.apache.tika.parser.captioning.tf.TensorflowRESTCaptioner<
/param>
      </params>
    </parser>
  </parsers>
</properties>
```

Description of parameters :

Param Name	Type	Meaning	Range	Example
apiBaseUri	uri	HTTP URL that will be used to create apiUri & healthUri	any HTTP URL	<a href="http://localhost:8764/inception/v3">http://localhost:8764/inception/v3</a>

captions	int	Number of captions to output	a non-zero positive integer	3 to receive 3 captions
maxCaptionLength	int	Maximum length of a caption	a non-zero positive integer(recommended >=15)	for 15 the sentence length of a caption won't be greater than 15
class	string	Name of class that Implements Object recognition Contract	constant string	<a href="#">org.apache.tika.parser.recognition.tf.TensorflowRESTCaptioner</a>

### Step 3. Demo

```
$ java -jar tika-app/target/tika-app-1.17-SNAPSHOT.jar \
  --config=tika-parsers/src/test/resources/org/apache/tika/parser/recognition/tika-config-tflow-im2txt-
rest.xml \
  https://upload.wikimedia.org/wikipedia/commons/f/f6/Working_Dogs%
2C_Handlers_Share_Special_Bond_DVIDS124942.jpg
```

The input image is:

[blocked URL](#)

And, the output is

[Toggle line numbers](#)

```
1 ...
2
3 INFO Available = true, API Status = HTTP/1.0 200 OK
4 INFO Captions = 5, MaxCaptionLength = 15
5 INFO Recogniser = org.apache.tika.parser.captioning.tf.TensorflowRESTCaptioner
6 INFO Recogniser Available = true
7 INFO minConfidence = 0.05, topN=2
8 INFO Time taken 1779ms
9 <?xml version="1.0" encoding="UTF-8"?><html xmlns="http://www.w3.org/1999/xhtml">
10 <head>
11 <meta name="org.apache.tika.parser.recognition.object.rec.impl" content="org.apache.tika.parser.captioning.
tf.TensorflowRESTCaptioner"/>
12 <meta name="X-Parsed-By" content="org.apache.tika.parser.CompositeParser"/>
13 <meta name="X-Parsed-By" content="org.apache.tika.parser.recognition.ObjectRecognitionParser"/>
14 <meta name="resourceName" content="Working_Dogs%2C_Handlers_Share_Special_Bond_DVIDS124942.jpg"/>
15 <meta name="Content-Length" content="295937"/>
16 <meta name="CAPTION" content="a man standing next to a dog on a leash . (0.00017)"/>
17 <meta name="CAPTION" content="a man standing next to a dog on a bench . (0.00017)"/>
18 <meta name="CAPTION" content="a man and a dog are sitting on a bench . (0.00014)"/>
19 <meta name="CAPTION" content="a man and a dog sitting on a bench . (0.00013)"/>
20 <meta name="CAPTION" content="a man and a dog are sitting on a bench (0.00009)"/>
21 <meta name="Content-Type" content="image/jpeg"/>
22 <title/>
23 </head>
24 <body><ol id="captions">
25 <li id="0"> a man standing next to a dog on a leash . [en](confidence =
0.000167)</li>
26 <li id="1"> a man standing next to a dog on a bench . [en](confidence = 0.000167)</li>
27 <li id="2"> a man and a dog are sitting on a bench . [en](confidence = 0.000138)</li>
28 <li id="3"> a man and a dog sitting on a bench . [en](confidence = 0.000131)</li>
29 <li id="4"> a man and a dog are sitting on a bench [en](confidence = 0.000092)</li>
30 </ol>
31 </body></html>
32 $
```

### Questions / Suggestions / Improvements / Feedback ?

1. If it was useful, let us know on twitter by mentioning [@ApacheTika](#)
2. If you have questions, let us know by [using Mailing Lists](#)
3. If you find any bugs, [use Jira to report them](#)