TikaAndVisionVideo

Tika and Computer Vision for Videos

This page describes how to make use of Object (Visual) Recognition capability of Apache Tika for videos. TIKA-2322 introduced a new parser to perform object recognition on videos. Visit TIKA-2322 issue on Jira or pull request on Github to read the related conversation. Continue reading to get Tika up and running for object recognition in videos.

Currently, Tika utilizes Inception-V4 model from Tensorflow for recognizing objects in the videos.

Tika and Tensorflow Video Recognition

1. Tensorflow Using REST Server

This is the recommended way for utilizing visual recognition capability of Tika. This approach uses Tensorflow over REST API. To get this working, we are going to start a python flask based REST API server and tell tika to connect to it. All these dependencies and setup complexities are isolated in 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.

a. Using docker (Recommended)

Toggle line numbers

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

Once it is done, test the setup by visiting http://localhost:8764/inception/v4/classify/video?topn=5&min_confidence=0.015&url=http://dev.exiv2.org/attachments/344/MOV_0234.mp4&mode=fixed&ext=.m4v in your web browser.

Sample output from API:

```
{
        "confidence": [
                0.29454298615455626,
                0.2789864711463451,
                0.08245106576941907,
                0.07252519279718399,
                0.05011849589645863
        ],
        "classnames": [
                "screen, CRT screen",
                "hand-held computer, hand-held microcomputer",
                "oscilloscope, scope, cathode-ray oscilloscope, CRO",
                "monitor",
                "cellular telephone, cellular phone, cellphone, cell, mobile phone"
        ],
        "classids": [
                783,
                591,
                689,
                665,
                488
        ],
        "time": {
                "read": 0,
                "units": "ms",
                "classification": 20298
        }
}
```


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.

Below are the main requirements -

1. Tensorflow
2. OpenCV with python and video support
3. Flask and requests - pip install flask requests

Toggle line numbers

```
python tika-parsers/src/main/resources/org/apache/tika/parser/recognition/tf/inceptionapi.py  --port 8764
```

Step 2. Create a Tika-Config XML to enable Tensorflow Video 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-video-rest.xml

Here is an example:

Toggle line numbers

```
<properties>
    <parsers>
        <parser class="org.apache.tika.parser.recognition.ObjectRecognitionParser">
            <mime>video/mp4</mime>
            <params>
                <param name="topN" type="int">4</param>
                <param name="minConfidence" type="double">0.015</param>
                <param name="class" type="string">org.apache.tika.parser.recognition.tf.TensorflowRESTVideoRecogniser</param>
                <param name="healthUri" type="uri">http://localhost:8764/inception/v4/ping</param>
                <param name="apiUri" type="uri">http://localhost:8764/inception/v4/classify/video?mode=fixed</param>
            </params>
        </parser>
    </parsers>
</properties>
```

Description of parameters :

Param Name	Type	Meaning	Range	Example
topN	int	Number of object names to output	a non-zero positive integer	1 to receive top 1 object name.
minConfidence	double	Minimum confidence required to output the name of detected objects	[0.0 to 1.0] inclusive	0.9 for outputting object names iff at least 90% confident.
class	string	Class that implements object recognition functionality	constant string	org.apache.tika.parser.recognition.tf.TensorflowRESTVideoRecogniser
healthUri	uri	The URI link to the ping service to ensure that the Video classification service is up and running.		
apiUri	uri	The URI link to the Python OpenCV/ImageNet based Video classification service.		

Step 3: Demo

To use the vision capability via Tensorflow, just supply the above configuration to Tika.

For example, to use in Tika App (Assuming you have *tika-app* JAR and it is ready to run and you are running from the Tika src checkout dir):

```
$ java -jar tika-app/target/tika-app-1.15-SNAPSHOT.jar \
  --config=tika-parsers/src/test/resources/org/apache/tika/parser/recognition/tika-config-tflow-video-rest.xml \
  ./tika-parsers/src/test/resources/test-documents/testVideoMp4.mp4
```

The input video is from: Sample Videos.com.

Big Rabbit coming out of his house

And, the top 4 detections are:

Toggle line numbers

```
...
<meta name="OBJECT" content="king penguin, Aptenodytes patagonica (0.19076)"/>
<meta name="OBJECT" content="hare (0.13538)"/>
<meta name="OBJECT" content="wallaby, brush kangaroo (0.09441)"/>
<meta name="OBJECT" content="ice bear, polar bear, Ursus Maritimus, Thalarctos maritimus (0.09350)"/>
...
```

Changing the default topN, API port or URL

To change the defaults, update the parameters in config XML file accordingly. Below is a description of parameters accepted by Inception REST API, they can be passed through apiUri in config file.

Param Name	Type	Description	Default Value
mode	string	Available Modes of frame extraction	center
		"center" - Just one frame in center	

		"interval" - Extracts frames after fixed interval	
		"fixed" - Extract fixed number of frames	
frame-interval	int	Interval for frame extraction to be used with INTERVAL mode. If frame_interval=10 then every 10th frame will be extracted	10
num-frame	int	Number of frames to be extracted from video while using FIXED model. If num_frame=10 then 10 frames equally distant from each other will be extracted	10

If you want to run video recognition by extracting labels from every 100th image you can make an apiUri like http://localhost:8764/inception/v4/classify/video?mode=interval&frame-interval=100

Here is an example scenario:

Run REST API on port 3030, and get top 4 object names if the confidence is above 10%. You may also change host to something else than 'localhost' if required.

Example Config File

```
<properties>
    <parsers>
        <parser class="org.apache.tika.parser.recognition.ObjectRecognitionParser">
            <mime>image/jpeg</mime>
            <params>
                <param name="topN" type="int">4</param>
                <param name="minConfidence" type="double">0.1</param>
                <param name="class" type="string">org.apache.tika.parser.recognition.tf.TensorflowRESTRecogniser<
/param>
                <param name="healthUri" type="uri">http://localhost:3030/inception/v4/ping</param>
                <param name="apiUri" type="uri">http://localhost:3030/inception/v4/classify/video?topk=4</param>
            </params>
        </parser>
    </parsers>
</properties>
```

To Start the service on port 3030:

Using Docker:

```
docker run -it -p 3030:8764 uscdatascience/inception-video-rest-tika
```

Without Using Docker:

```
python tika-parsers/src/main/resources/org/apache/tika/parser/recognition/tf/inceptionapi.py  --port 3030
```

Manual Installation guide

Installing Tensorflow

To install tensorflow, follow the instructions on the official site here for your environment. Unless you know what you are doing, you are recommended to follow pip installation.

Then clone the repository tensorflow/models or download the zip file.

```
git clone https://github.com/tensorflow/models.git
```

Add 'models/slim' folder to the environment variable, PYTHONPATH.

```
$ export PYTHONPATH="$PYTHONPATH:/path/to/models/slim"
```

To test the readiness of your environment :

```
$ python -c 'import tensorflow, numpy, dataset; print("OK")'
```

If the above command prints the message "OK", then the requirements are satisfied.

Installing OpenCV

OpenCV is a very popular computer vision library. We are using it to extract still images from video file. For this to work you need to install OpenCV with video support and python end point. Installation will vary from platform to platform so if you don't have enough experience with installing C packages it's a good idea to use docker build.

- On Mac you can use homebrew to install OpenCV and make sure cv2.py is in your python path.

```
brew tap homebrew/science
brew install opencv
```

- docker commands on ubuntu

To verify if your setup is correct you can run below commands and they should print "SUCCESS"

Toggle line numbers

```
1 curl -Lo testVideoMp4.mp4  "https://github.com/apache/tika/blob/e141640891cd7adcfc1848b351c0db7eab00a2d2/tika-parsers/src/test/resources/test-documents/testVideoMp4.mp4?raw=true"
2 python -c "import cv2; cap = cv2.VideoCapture('testVideoMp4.mp4'); print 'SUCCESS' if cap.isOpened() else 'FAILURE'"
```

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