International Journal of Future Generation Communication and Networking
Vol. 13, No. 1s, (2020), pp. 156-- 164

Analyzing Video to Find Particular Object Timestamp

Sufiyan Sayyed[1], Sayali Patkar[2], Atharva Patil[3], Prof. Mahendra Patil[4]

[1,2,3,4] *Department of Computer Engineering, Atharva College of Engineering / Mumbai University, Mumbai, India*

Abstract

We all are surrounded by huge data. People upload, download, send and update video, audio, images, and documents from a variety of devices. We often need to find one particular item of data among these hundreds, thousands, millions or more of these data objects. In documents or on-web page we've all gotten the little magnifying glass to bring up the search field. So we type the keyword in the search field and immediately get a list of every time that word shows up. Nowadays we can also search in images by using object detection. But searching in videos is currently not feasible. As more and more information gets in a large amount of which is left unprocessed.

Video by itself is really hard to search. We can't find videos we want or browse for what we might like. That's a loss. So in our project, we provide a basic solution for it, so that we are able to find a particular object in videos and return the timestamp, where that object-related image was encountered. In the proposed system user will provide video as well as tag or image for the object to search. Then we process the video frame by frame and keep track of continuity of tag in frames. After the parsing of the entire video, we will provide a timestamp when particular tag related object encountered in the entire video.

Keywords *- Convolutional Neural Network, Image Processing, Object Timestamp, YOLO, Object detection*

1. Introduction

In today's world, we are used to being able to search just about anything. Be it documents, email or websites. This is fantastic, because if you can search it, it can be used as a resource. But same doesn't apply to videos. That's a problem because the video has already become a large part of the information resource [6]. Today if we want to search in video, video host searches it by manually by the entered information added on top of video file like title, its description, various tags attached to it, comments, etc. The actual content of video is not searched at all. If a part of the video isn't mentioned in these extra details then it is completely loss and totally un-findable by search, and even if it does find the video, the video host still can't tell you the specific and relevant moment, leaving us to hunt and peek in an entire timeline to try and find what we are looking for. As of today you can search on the internet or your mail or document or even inside document. With an increase in video content, many times we required to search particular things in the video, but we can't because no such technology is available and if available not feasible. No current feasible solution to search in the video, which leads to a lack of information processing when it comes to video. With the help of current technology, we can only classify an image or detect an object in real-time. But providing summarize results is not provided by any of the current solutions. In the world of today, the need for automated information processing in the video file has increased because it has a wide range of applications. So in this project, we are trying to approach this problem and provide a solution. Video can be said as collection of discrete images that are constantly changing to create a motion effect. The identification of objects in video files is often based on the concept of object detection in an image file. We will get the object name which is to be search in the video (Tag) from user and process the video frame by frame and keep track of continuity of Tag in each frame. After parsing the entire video we will provide timestamp from the video that gives exact moment when the object occurs in video and how many times it occurs. Our system combines the functionality of the existing system and the new functionality that we are going to implement. The first one is Object detection such as HOG, CNN, RCNN, fast RCNN, etc. As the name suggests it detect objects from the image but they are very slow and video

ISSN: 2233-7857 IJFGCN

processing required faster algorithm. To work on a video we used YOLO as it is a real- time object detection algorithm. It can detect objects accurately at the speed of almost 65fps[1]. The term Analyzing means we break the video into frames. We will be using python and its libraries like OpenCV to detect an object in the video because they support image processing and YOLO.

2. Literature Review

Object recognition is a technology that identifies various objects in digital images and videos. There are various methods for object detection, they can be classified into two groups, first is the classification based algorithm i.e. CNN and RNN. In "Simple Convolutional Neural Network on Image Classification" paper, Authors Tianmei G., Jiwen D., Henjian Li and Yunxing G. proposed a simple CNN which imposes a less computational cost. The CNN is a widely used model of deep learning and has shown high efficiency in the classification of images. They also analyzed different learning rate set methods and optimization algorithm based on CNN to solve the optimal influencing parameters on image classification. They have also tested that the shallow network has a fairly good recognition effect [2]. The limitation of this method is it has the slow real time prediction to implement since we have to run a prediction for every region selected. [4].

The second algorithm is based upon regression which is YOLO. "You Only Look Once: Unified Real –Time Object Detection" by Joseph Redmond, Santosh D., Ross G. and Ali F. The prior work of these authors is on detecting objects using a regression algorithm. They proposed YOLO algorithm to obtain high precision and good prediction. At a single run of the algorithm they predict the classes and bounding boxes of the entire image and detect multiple objects using a single neural network. YOLO is fast compared to other algorithms for the classification. YOLO algorithm makes localization errors but predicts fewer false positives in background [1].

But all these techniques only detect objects in image and videos and don't provide summarized results. When we are going through the whole video just to find small object, it is very tedious task as we have to analyze entire video over self. Our system gives solution on it which gives summarized result of object present in a video means when object is entering and when it leaving the video.

3. Methodology

You only look once (YOLO) is an objects detection method intended for real time processing. YOLO divides the image data into the grid of S x S. Every cell of the grid predicts only one object. Each grid cell predicts a specified number of boundary boxes. Every boundary box has 5 elements :(x, y, w, h) and a confidence score for a box. The confidence score indicates how probable an item is in the box and how precise the boundary box is. In the bounding box' w' is width,' h' is height,' x' and y to the corresponding cell are offset. Therefore x, y, w and h are all 0 and 1. Every cell possesses 20 conditional class probabilities. The conditional class probability is the probability that the object being detected belongs to a certain class (one probability per category for each cell). The prediction of YOLO therefore has a form of (S, S, B + C)= (7, 7, 2+ 20) = (7, 7,30) [1].

YOLO's key idea is to create a CNN network to predict a tensor (7, 7, 30). Using a CNN network it reduces the spatial dimension to 7x7, with 1024 output channels at each location. A linear regression is performed by YOLO using two fully connected layers to make 7x7x2 boundary box prediction. We hold high box confidence scores as our preliminary predictions to make a final prediction. For each prediction box, the class confidence score is calculated as: box confidence score x conditional class probabilities. YOLO has 24 convolutional layers which are connected to 2 fully connected layers (FC). Alternatively, certain convolution layers use 1 x 1 reduction layers to reduce the depth of the feature map. It outputs a tensor with the shape of (7, 7, 1024) at the last convolution layer. Thus, the tensor is flattened. Two fully connected layers are used as a form of linear regression, which outputs 7×7×30 parameters and then it is reshaped to (7, 7, 30), i.e. 2 boundary box predictions per location. A quicker but less reliable version of YOLO, named as Fast YOLO, uses only 9 convolution layers with shallower feature maps [2].

ISSN: 2233-7857 IJFGCN

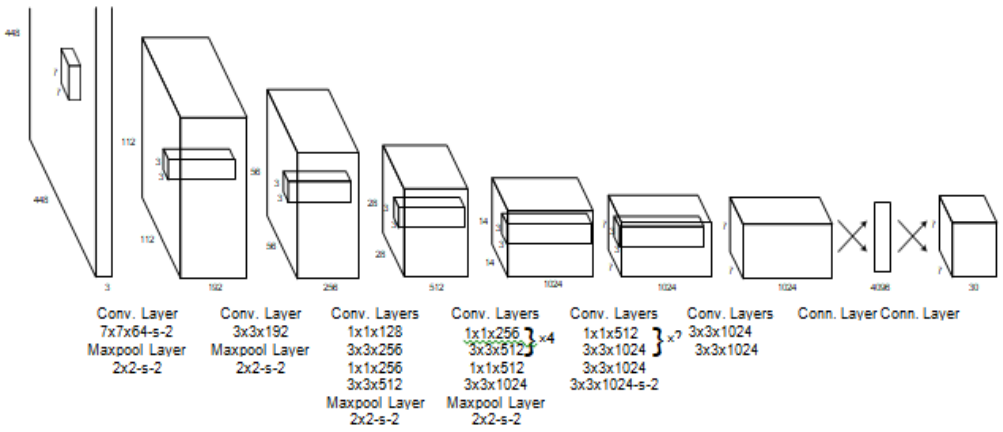

Figure 3.1: The Architecture

We have implemented this YOLO architecture in our project and for data set we used darknet's pre trained model for YOLO v3. The design of the project is relatively simple; it works in modules. Each module work on the input given from other module or input directly provided from the user. To keep the doors open for further improvement and adding more features. The proposed system is divided into four modules that are mentioned below,

1) Driver Module
2) Tag Module
3) Video Module
4) Time Module

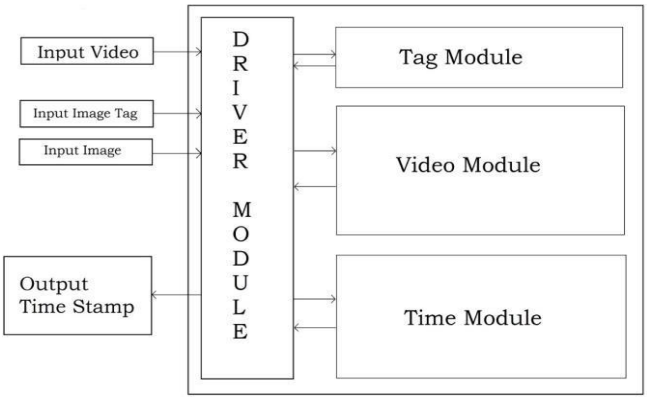

Figure 3.2: Block Diagrams of the System

1) **Driver Module:** As the name suggests Driver module is the main module that interacts with other modules. It takes all inputs, which includes the input that the user will be giving like Image, String Tag or Video. And other inputs are related to YOLO such as Class Labels, weights, etc. All the other modules are dependent on the driver module for any input(with one exception) and they give their output to the driver module and driver module process it further by passing it to other modules or showing to the user. The driver Module is responsible for passing required parameters to other modules such as an image to tag module or video to video module other than required module driver module also pass common parameters to modules that are related to YOLO such as Class Labels or YOLO object detector. The idea is to load all the resources in the driver module and just pass it as a parameter when required by other modules.
2) **Tag Module:** Tag modules extract tags from an image and store them in a dictionary. It takes an image and other YOLO variables as a parameter. After taking an image from the driver module

ISSN: 2233-7857 IJFGCN

it processes the entire image for object detection with YOLO Object detector. Once the entire image is processed the class IDs with confidence greater than 0.5 are considered and the label of the object with corresponding class ID is stored in the Dictionary as a value. Whereas Key for that particular value will be a counter variable starting from 1. For each new object that is stored in the dictionary, the counter variable will be increased. In the case where the same object with different confidence is detected then the object with the higher confidence will be considered and store. After all the entries are completed then the entire dictionary is printed with keys and values. If the dictionary is null then null is returned, or if dictionary has only one entry then that only value is returned, or if dictionary has more than one value then user has to enter key of the object to select the tag, once user entered the key to the tag that selected tag is returned to driver module.

3) **Video Module:** This module is the heart of the system. It is responsible for processing the entire video. This module work by considering video as a series of images and each image is processed individually. When the frame is captured YOLO object detector detects for the object, when the object is detected and Class ID of the object matches with the tag provided and confidence is greater than 0.5 that frame number is stored temporarily, once the frame number is stored it is kept under observation for a threshold of frame per second (FPS) of that particular video. If the object is continuously appearing in each frame till threshold count than the frame number is stored in the list and it is confirmed that the object was present from the time when we first temporarily stored the frame number. If the object disappears from the frame before threshold count we store the end frame number temporarily and consider it as False Negative for 5 frames if still, the object is not appearing in the frame then we considered that the object disappeared when we first noted the end frame. But if the object reappeared during the False Negative period we reset the False Negative variable again to 0. If the object crosses the threshold count we store the initial frame number on the list. And follow the above-mentioned procedure when the object disappears from the frame. After an entire video is processed the list of Start Frame and End Frame is returned to Driver Module.
4) **Time Module:** Time module is the module that calculates the exact timestamp of the object's appearance and disappearance. This module takes the Start Frame Number list and End Frame Number list and converts frame number to time in second with the formula mentioned below,

Time (in seconds) = Frame Number / FPS of video

The module work by iterating over Start Frame Number list and End Frame Number list together using only one loop and store time in seconds temporarily calculated by the above formula and then convert it into HH:MM: SS.MS format and store it into two different lists one for start time and one for the end time. Once both lists are completely iterated the resultant new lists are returned to Driver Module. Then Driver Module prints the time stamp in ascending order.

4. Result and Discussion

A System can be said complete until it is properly tested and confirmed that it is working as it was expected to work. The proposed system is no exception. To confirm this, the system was tested for various images and videos. Out of which results of three are mentioned below.

1. **Person Test:** The first test was done on a video where a woman enters a frame takes a photograph and exits the frame. The video is of 14 seconds, in which woman enters the frame after 2 seconds and exits the frame after 13 seconds but before video ends. The tag for searching was given through an image which had a horse and a person in it [7].

ISSN: 2233-7857 IJFGCN

Figure 4.1: Input Image for Tag

The above image was provided to extract tag system and gave the expected result.

```
E:\AVOFPOT>python AVOFPOT.py
Enter video path:E:\AVOFPOT\videos\person_test.mp4

Select the method of giving TAG input for searching:

1-Via String
2-Via Image

Enter choice:2

Enter Image path:E:\AVOFPOT\images\person_img.jpg

Index    Tag(Confidence %)
  1   -   horse ( 99.75 %)
  2   -   person ( 99.99 %)

Enter Index no to select tag:2
```

Figure 4.4: Tag Extracted From Input Image

Figure 4.2: Person's Entry in Frame

Figure 4.3: Person's exit from Frame

After selecting person tag and process the entire video we get the result as,

```
Tag is "person"

Processing: 100.0%
Processing completed!!

" person " occured in video at
Occurence       Start Time       End Time

  1             00:00:02.88      00:00:13.56
```

Figure 4.5: Output of Person_test.mp4

The output which is provided by the system is correct although we cannot confirm the time in milliseconds but the results in second is correct.

2. **Dog Test:** In dog test, the dog passes by two people whose only leg is visible. The video is of 14 seconds. In the video dog enters in the frame after 2 seconds and exits the frame after 8 seconds. But the problem is after 6 seconds complete dog is not visible to the system because part of its head get covered by one of the person's leg. To test the second functionality of taking tag input we provided tag as string, below is the start and end frame of dog followed by the output.[8]

ISSN: 2233-7857 IJFGCN

Figure 4.6: Dog's Entry in Frame

Figure 4.7: Dog's Exit from Frame

Even though we can recognize the dog in end frame but the system is unable to recognize as its head get merged with the person's leg.

```
E:\AVOFPOT>python AVOFPOT.py
Enter video path:E:\AVOFPOT\videos\dog_test.mp4

Select the method of giving TAG input for searching:

1-Via String
2-Via Image

Enter choice:1

Enter Tag:Dog

Tag is "dog"

Processing: 100.0%
Processing completed!!

" dog " occured in video at
Occurence       Start Time      End Time

  1             00:00:02.97     00:00:06.50
```

Figure 4.8: Output of Dog Test

3-<u>Traffic Light Test:</u> Third and the final test video is a time lapse of a car drive. Where the car is traveling on the streets in the night and encountering various signals. The video is of 55 seconds. In total car stopped at more than 15 signals, but at most signal it didn't wait for more than 1 second so that signal is not count (as threshold is of 1 second). There were only few signals were car stopped for more than 1 second. To provide tag an image was given as input.[9]

Figure 4.9: Image Provided for Tag

As car stops at signal so we have considered that start frame and end frame is same,

ISSN: 2233-7857 IJFGCN

Figure 4.10: Car Stopping at Signal for More than 1 Second

The output of the above test video is,

```
E:\AVOFPOT>python AVOFPOT.py
Enter video path:E:\AVOFPOT\videos\traffic_light_test.mp4

Select the method of giving TAG input for searching:

1-Via String
2-Via Image

Enter choice:2

Enter Image path:E:\AVOFPOT\images\traffic_img.jpg

Index    Tag(Confidence %)
  1   -   car ( 99.5 %)
  2   -   traffic light ( 99.76 %)

Enter Index no to select tag:2

Tag is "traffic light"

Processing: 100.0%
Processing completed!!

" traffic light " occured in video at
Occurence       Start Time      End Time

  1             00:00:13.11     00:00:14.65
  2             00:00:21.52     00:00:22.99
  3             00:00:25.83     00:00:27.59
  4             00:00:29.56     00:00:30.96
```

Figure 4.11: Output of Car Stopping at Single More than 1 second

The table mentioned below is summarized the result of all the three cases their respective method of tag input and obtained output.

Table 1: Summarized Result of Object Detection in Videos

Video	Object	Method of giving tags	No of times object appeared	Entry of object in video	Exit of object in video
Video 1	Person	Image	1	00:00:02.88	00:00:13.56
Video 2	Dog	String	1	00:00:02.97	00:00:06.50
Video 3	Traffic Signal	Image	4	00:00:13.11	00:00:14.65
				00:00:21.52	00:00:22.99
				00:00:25.83	00:00:27.59
				00:00:29.56	00:00:30.96

5. Future Enhancement

Every project has infinite ways, in which it can be developed and implemented in the future. The same is applicable to the above proposed system. Small changes in the base system can create

ISSN: 2233-7857 IJFGCN

many variations of a system with completely different purposes and applications. [4] Some of the possible future developments are mentioned below,

- **All Object Analysis**: In the proposed system we take a tag and search for the object in the video that represents the given tag and return time stamp. In All Object Search, the entire video will be parsed and timestamp of each object that occurred in the video will be noted individually and will be displayed or stored. This will eliminate the requirement of a tag to find the object and will also save time if we want to search for multiple objects in the same video.
- **Specific Search:** A Situation might occur when we have to find an object but with specific constraints. Like "Red Car" or "Person with Blue Shirt" but it is not possible with the current proposed system as it searches for the object in general. But with the help of some improvement and changes in the system, we can achieve it. This will help us to extract even more details from the video.
- **Time-Bound Search:** This might not be a big improvement but can be very useful. If the video is long and we need to find the object's timestamp but within given porting of the video. For example, we need to find "car" but between the first 5 to 13 seconds of the video. So the system will eliminate the first 5 seconds of the video and video after 13 seconds and process just 8 seconds from the entire video to find "car".
- **Real-Time Object Analysis:** Real-time video footage coming from CCTV cameras or webcam or from any other video footage device can be processed at the time when the video was captured, and the results can be stored in some database. So that when we need toextract some information or find the timestamp of any object from these recorded videos, we don't have to process the entire video again. We can simply fire some queries in the database to find the timestamp of the required object. This can be implemented for single object detection like the proposed system or can be implemented for "All object analysis" which is mentioned above.

As mentioned above possibilities are endless, and we can make varieties of different systems that can be focused on a specific operation or can be more like a general-purpose system.

6. Conclusion

In this paper, we provide solution to find a particular object in videos and return the timestamp. For that we have used YOLO algorithm that is much more efficient and useful in real time and can be directly trained on a complete image. YOLO as well as module of proposed system were discussed in depth, which provided us with the insight of how YOLO as well as proposed system works. The proposed system search for particular Tag in entire video which can be provided as string or with the help of image, after getting the tag we parse entire video to find timestamp and provide summarized result. For that we used python libraries like OpenCV.

References

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi," You Only Look Once: Unified, Real- Time Object Detection", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA.

[2] TianmeiGuo, Jiwen Dong, Henjian Li, YunxingGao, "Simple Convolutional Neural Network on Image Classification ",2017 IEEE 2nd International Conference on Big Data Analysis. Beijing, China.

[3] Cuong Nguyen, DmirityShashev, "Methods and Algorithm for detecting objects in video files", MATEC Web of Conferences.

[4] Geethapriya.S, N.Duraimurugan, S.P.Chokkalingam ,"Object Detection with YOLO" 2019 International Journal of Engineering and advanced technology.

[5] Shrish Trivedi, "What are the applications of moving object detection?", Quora, (2017) May 30,

ISSN: 2233-7857 IJFGCN

Retrieved from https://www.quora.com/What-are-the-applications-of-moving-object-detection.

[6] Jeff Schultz, "How Much Data is Created on the Internet Each Day?", Micro Focus Blog, (2019) June 8, Retrieved from https://blog.microfocus.com/how-much-data-is-created-on-the-internet-each-day.

[7]"Photographer Beach Photography Camera Woman Girl free stock video footage. "YouTube, uploaded by Finding Footage,(2016) June 21, Retrieved from https://www.youtube.com/watch?v=Zb-fWTNIVXQ

[8]"Dog Running Walking The Dog Park Animal Pet Fast free footage Free stock footage" YouTube, uploaded by Finding Footage,(2016) May 22, Retrieved from https://www.youtube.com/watch?v=TIlcin94HwE.

[9]"4K Free stock video: Traffic Night Street Cars Speed Signal Lights - Free footage" YouTube, uploaded by Finding Footage,(2016) July 15, Retrieved from https://www.youtube.com/watch?v=n5y5BIsE-Cw.

ISSN: 2233-7857 IJFGCN