Objects Talk - Object detection and Pattern Tracking using TensorFlow

Rasika Phadnis
Department of Computer Engineering,
Vidyalankar Institute of Technology,
Mumbai, India
rphadnis8@gmail.com

Jaya Mishra
Department of Computer Engineering,
Vidyalankar Institute of Technology,
Mumbai, India
jayamishra07@outlook.com

Shruti Bendale
Department of Computer Engineering,
Vidyalankar Institute of Technology,
Mumbai, India
shrutibendale@gmail.com

Abstract—Objects in household that are frequently in use often follow certain patterns with respect to time and geographical movement. Analysing these patterns can help us keep better track of our objects and maximise efficiency by minimizing time wasted in forgetting or searching for them. In our project, we used TensorFlow, a relatively new library from Google, to model our neural network. The TensorFlow Object Detection API is used to detect multiple objects in real-time video streams. We then introduce an algorithm to detect patterns and alert the user if an anomaly is found. We consider the research presented by Laube et al., Finding REMO—detecting relative motion patterns in geospatial lifelines, 201–214, (2004)[1].

Keywords—object detection, TensorFlow, REMO, pattern recognition, SSD MobileNet

1. INTRODUCTION

With the introduction of GPS, it has become easy to track objects that employ a GPS tracker chip. However, once this chip is removed or is damaged, the tracking stops. Furthermore, there might be objects on which lodging a chip is not possible, for example- jewellery. Therefore, we have adapted an approach that does not require a GPS chip. This works because we have limited the scope of tracking trajectories to the scope of camera module of raspberry pi.

Neural networks are currently achieving things that no other machine learning algorithm can achieve. With massive datasets and computers capable of processing and optimizing against those massive datasets, neural networks have gained popularity in the recent years. They are incredible at learning from the datasets and creating models of the data. Our project uses TensorFlow, a framework provided for deep learning, to model our neural network. This API is used to detect multiple objects in real-time video streams. SSD MobileNet, a predefined model offered by TensorFlow is used as the base and fine-tuned to improve the accuracy and the range of objects that can be detected.[10] This model can be trained for any custom object that is required by the user to keep a track of that object.[9]

Once the system is well-equipped to detect objects, it is trained to track the object as long as it is in the range of camera. It is interesting to observe how even objects themselves can be a source of information that can be used to detect their behavioural patterns by tracking their movement. This analysis of the objects finds applications in home automation and security. 2D trajectories of the objects captured by the camera are fed as input to tracking and pattern detecting algorithm. Research presented in [1] regarding different patterns and REMO framework is used as a basis for developing a spacio-temporal pattern detecting algorithm. We aim to improve the productivity of the user by supporting his actions with memory and intelligence of machines.

2. LITERATURE SURVEY

2.1 Object Detection

We examined all the deep learning frameworks for key char
acteristics like their speed and classification accuracy.[7]

Authorized licensed use limited to: Malta College of ArtsScience&Tech. Downloaded on June 06,2023 at 19:38:18 UTC from IEEE Xplore. Restrictions apply.

		Caffe	TensorFlow	Torch	Theano
Language		C++, Python	Python	Lua	Python
Pretrained		Yes ++	Yes (Inception)	Yes ++	Yes (Lasagne)
GPU		CUDA, Opecl	CUDA	CUDA	CUDA, Opecl
Good at RNN		No	Yes (Best)	Mediocre	Yes

Advantages of TensorFlow over other frameworks:

1. Easy deployment (Python pip package manager deployed by TensorFlow facilitates easy installation).
2. Better support for GPUs as compared to other models.
3. It provides high level APIs for building models.
4. It is extremely easy to do unconventional and hard-core changes.

Considering all the above parameters and the requirements of our project, we used the TensorFlow framework to train our neural network. [8]

2.2 Pattern Detection Algorithms

Much research has been done on analysis of moving objects. We refer to this research and attempt to deploy the most efficient algorithm to achieve smooth tracking.
Majority of concepts have been defined in [1] which include *track, convergence, leadership.* We use these concepts in our algorithm by making changes according to our requirement. In *Efficient Detection of Patterns in* [5] it has also been demonstrated how algorithm can be designed to identify these patterns. It employs geographic data mining approach to detect generic aggregation patterns such as flocking behaviour and convergence in geospatial lifeline data. The REMO framework (RElative MOtion) developed by Laube and Imfeld [1] to detect certain behaviour in groups of entities. They define a collection of spatio-temporal patterns based on similar direction of motion or change of direction.
The input to pattern tracking algorithm is 2D trajectory captured by camera module. [2] improves efficiency of algorithms to find *flock, leadership, convergence and encounter* proposed by [1].
Similar approach is used to analyse patterns of game plan of football players on the field.
Our algorithm uses REMO framework to detect patterns by analysing 2D trajectories of objects defined by the customer.

3. METHODOLOGY

3.1 Object Detection and Recognition

Our neural network is trained using supervised classification learning in two basic steps:
1) A dataset of around 150 images of each object is created.
2) The existing model is fine-tuned by training the object detector using the above dataset.
The above steps are repeated for all the objects that need to be detected. Fine tuning results in a much more efficient model that can detect more objects with improved efficiency. [6]

A brief overview of fine-tuning the pre-existing model with respect to custom objects is given below: [5]

1. Prepare a training dataset. This data is a large collection of images resized as per the requirement.
2. Crete an XML file that describes the objects in the pictures. This is called as labelling.
3. Convert all the XML files to TFRecord files.
4. Setting up the configuration file for the model.
5. Train the model by using the TFRecord files of the images and the configuration file.

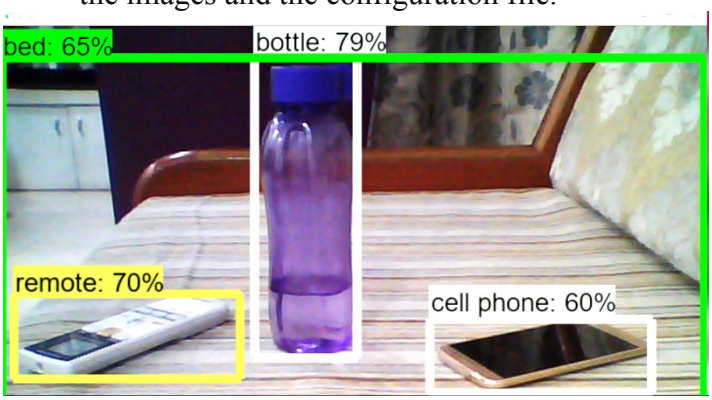

Figure 1: The system detects everyday objects

3.2 Pattern Detection

As defined in [1] the following have been considered and modified to match the specifications of our system.

i. Flock – More than one object is in the defined radius ‘r’.

Authorized licensed use limited to: Malta College of ArtsScience&Tech. Downloaded on June 06,2023 at 19:38:18 UTC from IEEE Xplore. Restrictions apply.

ii. Leadership - More than one object is in the defined radius ‘r’ and one of them was heading for the direction before anyone else.
iii. Convergence – Assuming that moving objects don’t change direction, predict that more than one object will pass through a radius ‘r’.
iv. Encounter - Assuming that moving objects don’t change direction and speed, predict that their paths will intersect at some point.

For the proposed system, we consider the user to always be the leader and use this assumption to predict location of other objects (using concepts of leadership, encounter and convergence).

When custom object is in the defined radius along with its predefined reference object or if it follows the leader (in this case, user), it will declare this state to be a stable state. Failing this, system goes into an unstable state and user gets an alert via mobile application (using concept of flock).

For instance, let us assume that custom object is a precious trophy and its reference object is a glass show case. Now if the trophy stays in the showcase or if it is carried by the user, state of the system is stable. Any other conditions will result in user receiving an alert.

Another application can be seen in the form of tracking habits of the user. If user carries his watch and wallet (custom objects) every day for 3 consecutive days from his table (reference) at 8 am then the system detects this as a pattern (as shown in figure 2). If he fails to do so on a certain day then he receives an alert.

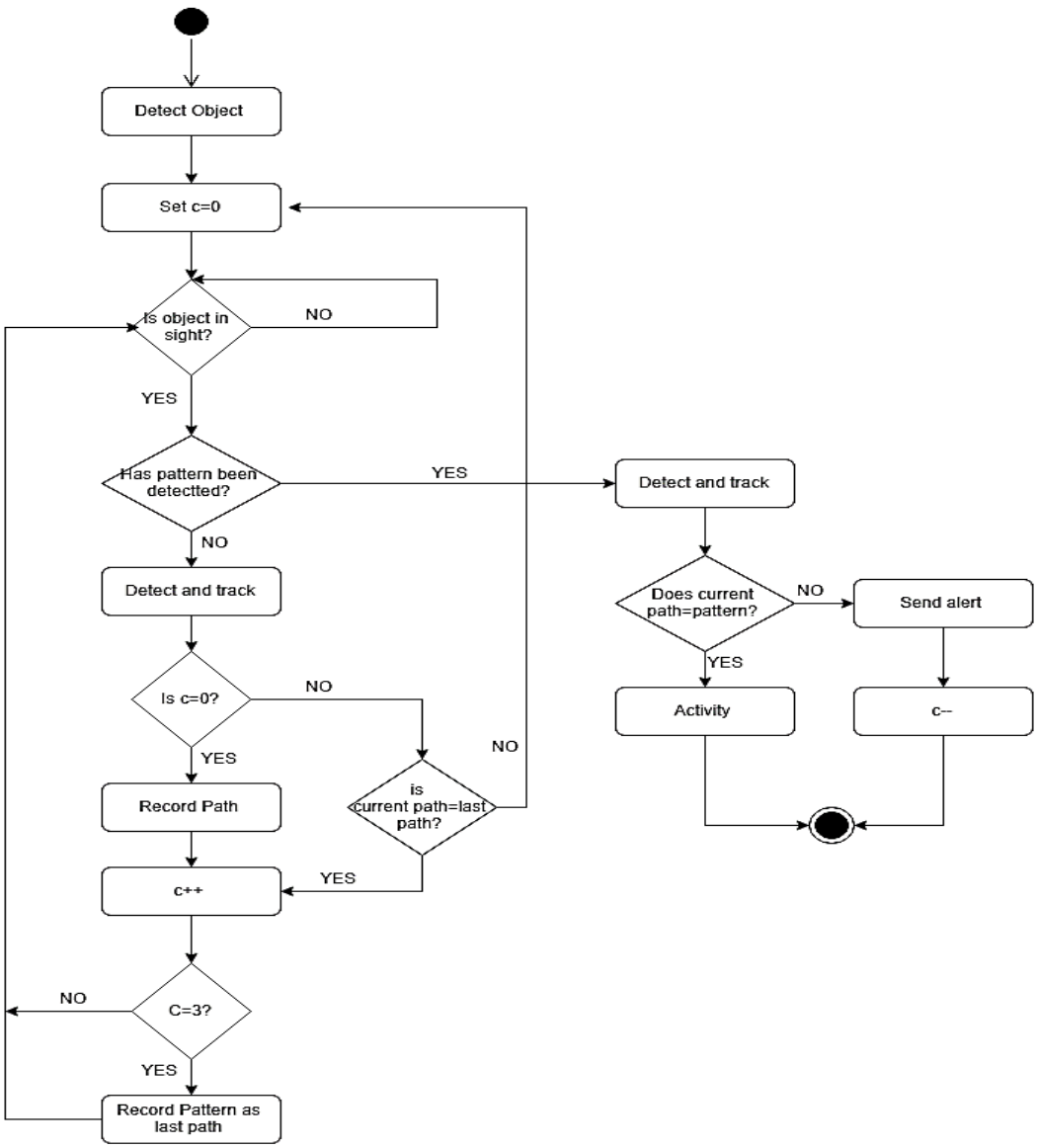

Figure 2

Authorized licensed use limited to: Malta College of ArtsScience&Tech. Downloaded on June 06,2023 at 19:38:18 UTC from IEEE Xplore. Restrictions apply.

4. CONCLUSION

The system attempts to eliminate the need for setting alarms or reminders. It aims to create a futuristic environment where all the actions of a person are taken into account and help him to maximise his efficiency. It not only will bring order to the way things are done but anomalies will be scrutinised and analysed to give a better output in the form of suggestions, reminders or emergency alerts.

5. FUTURE SCOPE

The scope of the system can be further extended to train it to recognize symptoms of a user (symptoms specific to him) and advise him to rest, suggest exercises or visit a doctor. It is possible to incorporate voice recognition and voice commands to the system to shut it down when user does not wish to be tracked. It may be argued that system invades privacy of the user up to certain extent, therefore the voice commands can be a quick way to shut it down when not required.

7. REFERENCES

[1] Patrick Laube, Marc van Kreveld, Stephan Imfeld, "Finding REMO - Detecting Relative Motion Patterns in Geospatial Lifelines", 11th International Symposium on Spatial Data Handling, Leicester, London, 201–214

[2] Joachim Gudmundsson, Marc van Kreveld, Bettina Speckmann, "Efficient Detection of Patterns in 2D Trajectories of Moving Points", Geoinformatica (2007), 11:195-215.

[3] Xiaoliang Geng, Takeaki Uno, Hiroki Arimura1, "Trajectory Pattern Mining in Practice: Algorithms for Mining FlockPatterns from Trajectories".

[4] Gordon Forbes, "The Automatic Detection of Patterns in People's Movements" Submitted to the Department of Electrical Engineering, University of Cape Town, in fulfilment of the requirements for the degree of Master of Science in Engineering. March 2002.

[5] Tracking Custom Objects Intro - Tensorflow Object Detection API Tutorial. Retrieved from https://pythonprogramming.net/custom-objects-tracking-tensorflow-object-detection-api-tutorial/

[6] Mustamo P., "Object detection in sports: TensorFlow Object Detection API case study", University of Oulu, Degree Programme in Mathematical Sciences. Bachelor's Thesis, 2018, 43.

[7] Popular Deep Learning Tools – a review. Retrieved from www.kdnuggets.com/2015/06/popular-deep-learning-tools.html

[8] Tensorflow API Documenation using Python. Retreived from www.tensorflow.org/api_docs/python/

[9] How to train your own Object Detector with TensorFlow's Object Detector API. Reterived from https://towardsdatascience.com/how-to-train-your-own-object-detector-with-tensorflows-object-detector-api-bec72ecfe1d9

[10] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C & Berg AC, " Ssd: Single shot multibox detector", European conference on computer vision, Springer: 21-37.

Authorized licensed use limited to: Malta College of ArtsScience&Tech. Downloaded on June 06,2023 at 19:38:18 UTC from IEEE Xplore. Restrictions apply.