# Target Spotter using PSO and OpenCV

Dharaneesh A (19BPS1031), Manav J (19BCE1037)
Operating Systems CSE2005
School of Computer Science and Engineering
Vellore Institute of Technology, Chennai Campus

**Keywords:** Target Spotter, PSO, Optimization, OpenCV.

**ABSTRACT**
Finding the presence of a target in a video footage, whether it is live or recorded is one of the most common problems of all times. There are so many applications which has this problem, either as small part of the solution or as a soul requirement to solve one. It is worth to notice that, often there are limitations in the techniques which are already in existence, they may be flexibility in use of computational resource, portability of the software, heavy requirement of training data, and so on.In order to face these challenges we need newer techniques which are more efficient and reliable. The project is focused on spotting a target in a given footage, using an AI based algorithm, which is Particle Swarm Optimization (PSO), the approach is new and it is under research. The Algorithm is integrated with image processing techniques, machine learning techniques, and other multiprocessing techniques to identify the target in the footage.

## 1 INTRODUCTION

In the present world, we have camera's everywhere, which surely took the safety and well-being to the next level, as time goes on, its certain that their number goes up and a lot of human power will be required to keep the surveillance up. Observing long video footage is definitely a nerve-racking task, and a normal human brain is not designed for this. Trained people can carry out this task pretty well, but again getting trained to look at a screen for long time, with complete concentration, is damaging in long term. While the present technology is quite advanced and can surely be used to accomplish such tasks.In order to do this, we need a right technique to be followed so that the computational resources are used to their full potentials, and there can't be one technique which is the best for every problem.

The project is an approach to accomplish this task, it uses an AI based computing technique i.e Particle Swarm Optimization (PSO). This Computing technique, tries to give a possible best solution to a problem, through the position of the particle (can be said as 'eye' of the algorithm), the technique iteratively tries to find the best particle solution using the past and current data. An image processing technique is used to find the features of the image which is the ORB algorithm, then a machine learning algorithm for matching them, and Operating system related multiprocessing techniques are used for enabling the parallel computing for faster computation. All these techniques are integrated together to get the task done. It provides you a flexibility on the use of computational power based on the parameter values you give, which makes it useful for many applications.

## 2 LITERATURE

### 2.1 Video Frames

The need for recording and playing media has been since the day humans realized that "the past is something that can't be experienced again". For this solitary purpose videos came into existence. Videos are an electronic medium for recording, broadcasting and displaying the visual media. Video systems vary in display resolution, aspect ratio, refresh rate, color capabilities and other qualities. Basically, a video is a set of images that are displayed at a certain rate to show motion. The rate at which the images are displayed is termed as frame rate. The frame rate is thus the number of frames displayed per unit of time, and is measured in frames per second (fps) also measured in Hertz (Hz).

Let's say if we can capture frames from a video then a lot of stuff can be done in an effortless manner. This capturing of the frames in a video and finding some data in it, or matching it for some other purpose has been a research topic for not so long. This can be related to a lot of topics like detection of texts in a video, or matching something specific in a footage for an unidentifiable object and many other research based venture. In fact the space agencies like ISRO or NASA

### 2.2 Text Detection from video frames

Textual content appearing in videos represents an interesting index for semantic retrieval of videos (from archives), generation of alerts (live streams), as well as high level applications like opinion mining and content summarizing. The key components of such systems require detection and recognition of textual content which also make the subject of study. Detection of textual regions in video frames is carried out by fine-tuning deep neural networks based object detectors for the specific case of text detection. Script of the detected textual content is identified using convolutional neural networks.

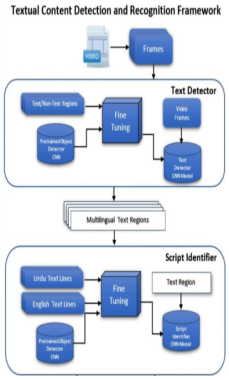


**Fig. 1.** Framework.

For detection of textual regions, state-of-the-art deep learning-based object detectors and fine-tuned them to detect text in multiple scripts have been deployed. Script of the detected textual regions is identified using CNNs in a classification framework. A key contribution of this study is the development UrduNet, a combination of CNN and bidirectional LSTMs which reports high recognition rates for the challenging video text in cursive Urdu.

### 2.3 Spotting sequence in Video streams

The idea of searching for a video clip or sequence that is already known within a video stream is not that common. A reliable and automatic recognition of video sequences has applications in a number of areas. One application is the spotting of a specific commercial within TV channel streams. One possibility would be to store the entire commercial and continually compare broadcast signals against the stored video sequences in the library through a frame by frame comparison. This solution is not very practical since it would require a prohibitive amount of memory and processing power to implement. Thus, instead of a frame by frame comparison, selecting specific frames in both the video sequence and the live video stream then comparing these corresponding frames would reduce the processing power, number of comparisons, and the size of the video sequence in question to be stored in the library.

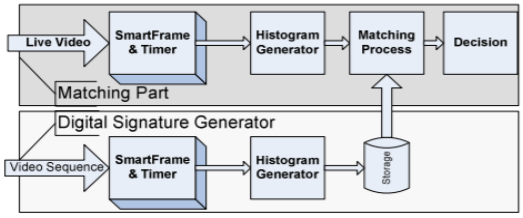


**Fig. 2.** Block Diagram.

The digital signature of a video sequence is composed of a sequence of one-dimensional Histograms of a limited number of frames in the video sequence. These frames have a particular significance, namely ”Cut”, ”Timeout”, or “Initial”, along with their relative timestamps. Such information is small in size, yet condensed in information. The use of tagged histograms, according to whether they belong to frames that are Cuts, Initial, or Timeouts, has considerably enhanced the efficiency of the matching process. The alert signal can take any form. It can be a visual or audible alert, or it can be used to take some actions such as increment a counter or recording a clip for later playback. The system presented here can load a number of signatures simultaneously, in which case the multiple incidences of the search algorithm are run on each signature in parallel. Also it may run on a multiple video stream at the same time.

## 3 PROPOSED METHOD

### 3.1 Apparatus

PSO- It is a meta heuristic algorithm that is appropriate to optimize nonlinear continuous functions. It is a very good technique for the optimization problems by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO is originally attributed to Kennedy, Eberhart and Shi and was first intended for simulating social behaviour,as a stylized representation of the movement of organisms in a bird flock or fish school. The algorithm was simplified and it was observed to be performing optimization. It is metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions.

NumPy- NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly inner loops, using NumPy.

Tkinter- It is the standard Python interface to the Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, as well as on Windows systems. In addition to the Tk interface module, tkinter includes a number of Python modules which are required for a perfect GUI. As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands, which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

OpenCV- Open Source Computer Vision Library is an open source computer vision and machine learning software library. It was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Computer Vision can be defined as a discipline that explains how to reconstruct, interrupt, and understand a 3D scene from its 2D images, in terms of the properties of the structure present in the scene. It deals with modeling and replicating human vision using computer software and hardware.

Multiprocessing- It supports spawning processes using an API similar to the threading module. It refers to a function that loads and executes a new child processes. The multiprocessing package offers both local and remote concurrency, effectively side-stepping the Global Interpreter Lock by using sub-processes instead of threads. The multiprocessing module also provides logging module to ensure that, if the logging package doesn't use locks function, the messages between processes mixed up during execution.

### 3.2 Procedure

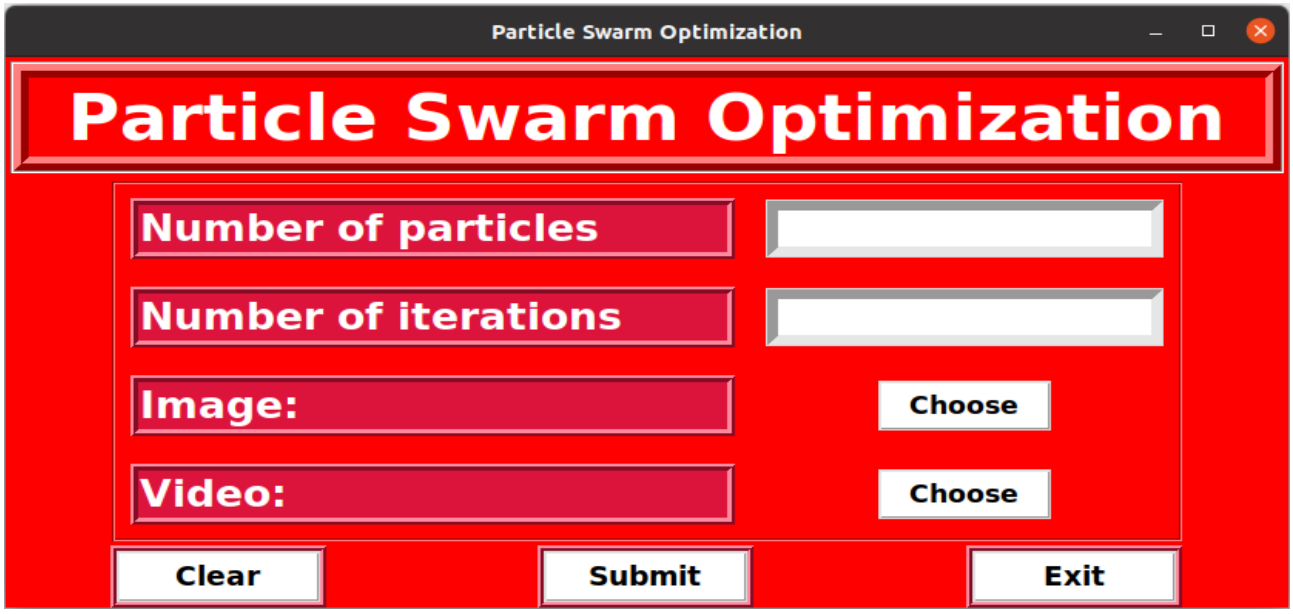


**Fig. 3.** User Interface.

**Fig. 4.** Target.

Once the Inputs are taken from the User which are the Video footage and the Target image, the program starts 2 child processes lets name them as Parent Process(PP) the main one, Algorithm Process(AP) first child, Output Process (OP) second child. TO ensure communication between them 2 pipes are created in advance and let them be Parent-Algorithm Pipe (PA pipe), and Algorithm-Output Pipe (AO Pipe), as their name suggests the PA pipe is used for communication between Parent Process and Algorithm Process, same for AO Pipe.

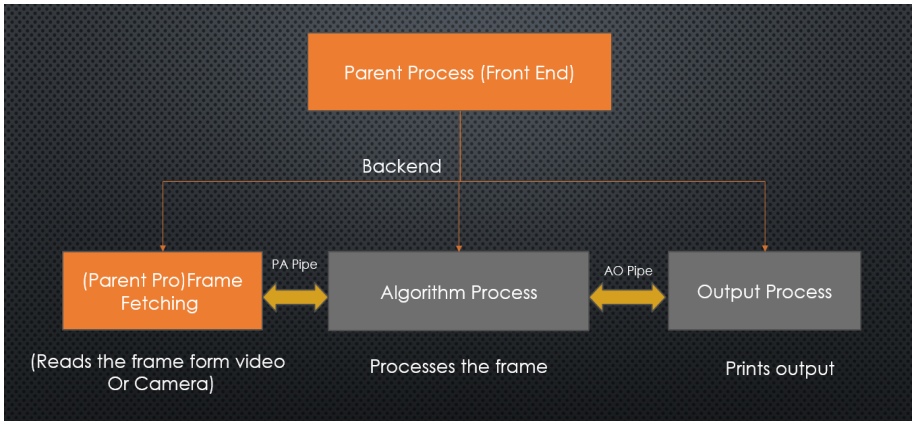


**Fig. 5.** Block diagram.

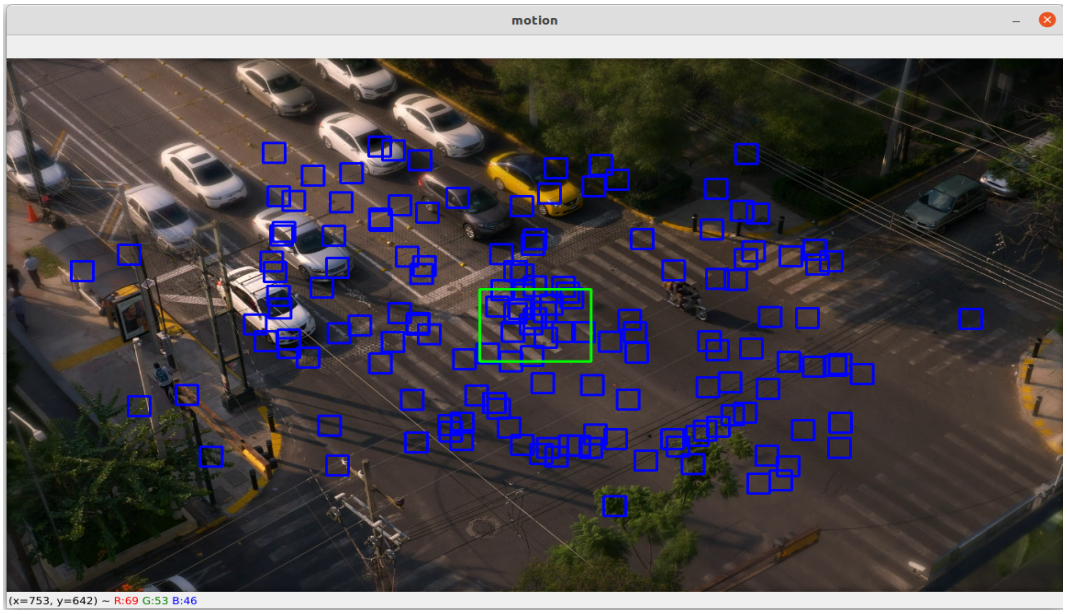


**Fig. 6.** Particles.

Once all the required objects are created, the Parent Process reads the frames from the video footage sequentially and send them to the Algorithm process through the PA pipe, this is the only thing the parent process will be doing from now until all the frames are read.

The Algorithm Process has 2 functions, one function runs the Particle Swarm Optimization Algo, where it creates the no of particles (Every particle has a co-ordinate where it lies , a coordinate of its personal best result, and its personal best score and a function to update its personal best variables) required based in the inputs given by the user, to each particle it gives a random co-ordinate in the space of the frame ensuring that its personal frame of exploration is not going outside the frame given for the algorithm to process we can do this by limiting the space in which the particle can lie. Once all the particles are created and given their coordinates, we append all of them into a list for referencing. We create 2 variables which store the global best results (you'll get to know in further reading) and the co-ordinates of the global best frame which is considered as the target and are common for all particles. From now we use these particles to look into the frames of given footage, for every particle, its coordinate is used to determine its frame, and once its frame is determined the present frame and the target photo is sent to a function (the other function in this process), in this function Orb classifier is used to generate the key points in the both the frames and their descriptors are generated at the same time together known as features, then we find the matches between the features using a machine learning algorithm (KNN) and the no of good matches (good matches are determined based the difference between the matched matches) are returned, and lets call this returned value as fitness Value, this value is then compared with its present


**Fig. 7.** Target in frame.

personal best values(along with the personal best coordinates) ,or some thresholds, and gets updated if it is more (more no of matches), and then the fitness value is compared with the global best value , or some threshold, and updates the global best value and the coordinates if the thresholds satisfy, once this is done , then we determine the direction and jump magnitude of the particle for the next iteration, this movement along x and y direction comes from the sum of 3 components - previous direction, movement towards personal best and movement towards the global best. the formula goes like:

$$Vx = (wt * R * Vx) + (pc * R * (Px - X)) + (gc * R * (Gx - X)) \tag{1}$$

$$Vy = (wt * R * Vy) + (pc * R * (Py - Y)) + (gc * R * (Gy - Y)) \tag{2}$$

Here, 'wt' is the magnitude at which the particle will tend to move in its previous direction and magnitude, 'pc' is the magnitude at which the particle will tend to move towards the personal best co-ordinates, 'gc' is the magnitude at which the particle will tend to move towards the global best co-ordinates, 'R' is the random number between 0 and 1 , we need it to have a randomness in the magnitude of the movements so the they don't settle at a place faster and we have large and favourable space for more exploration, once Vx and Vy are determined we add these values to the co-ordinates of the particle and ensure that it is not going out of the frame with some conditions, this is all what happens in an algorithm iteration of a particle, the no of iterations to be performed on each frame is determined by the user, the co-ordinates of each particle are remembered in video frame so there more chance that the particle is near its target in the next video frame, and make's it easier to track the target. Once all the iterations for all the particles is done the global best co-ordinates are used to draw a green frame around their location, which says it is best result they found in that frame and this output frame is sent to Output process through the AO Pipe.

The Output Process waits for the frame at the other end of AO Pipe once it gets a frame it then prints the frame and comes back to wait for the next frame, this is all what the Output Process does.

As the user we'll only see the output printed by the Output Process but Algorithm process will always process the next frame, and the Parent Process is reading the future frame (the one after the next frame) at the same time,hence makes it possible to execute at an optimum speed.

## 4 RESULTS

The program can spot the target when it comes into program's sight, basically when the particles come close to the target, it can find the presence very quickly compared to the traditional brute force method, but there is chance that it can fail if the particles couldn't come close to the target, as they move in random directions when they don't find anything. Therefore, a lot of care has to be taken while choosing the parameters, as the program involves randomness, when the particles and number of iterations are given as high values this problem can be easily tackled, but uses very high computational power. It is found that in most cases even if one particle finds the target or comes close to it, the program can accurately catch its location, and even the no of particles can be decreased, as more no of particles can overlap one another, and this can efficiently decrease the computation power required.

## 5 DISCUSSION

### 5.1 Analysis

The approach taken in this project towards spotting a target in a footage is unprecedented. It uses Particle Swarm Optimization to move its particles, can be said as eyes, in the footage so they find the target, instead of trying to find it by scanning the whole frame. It is programmed using the multi-programming techniques, so that that it uses the advantage of multi-processing and gives an optimised performance.

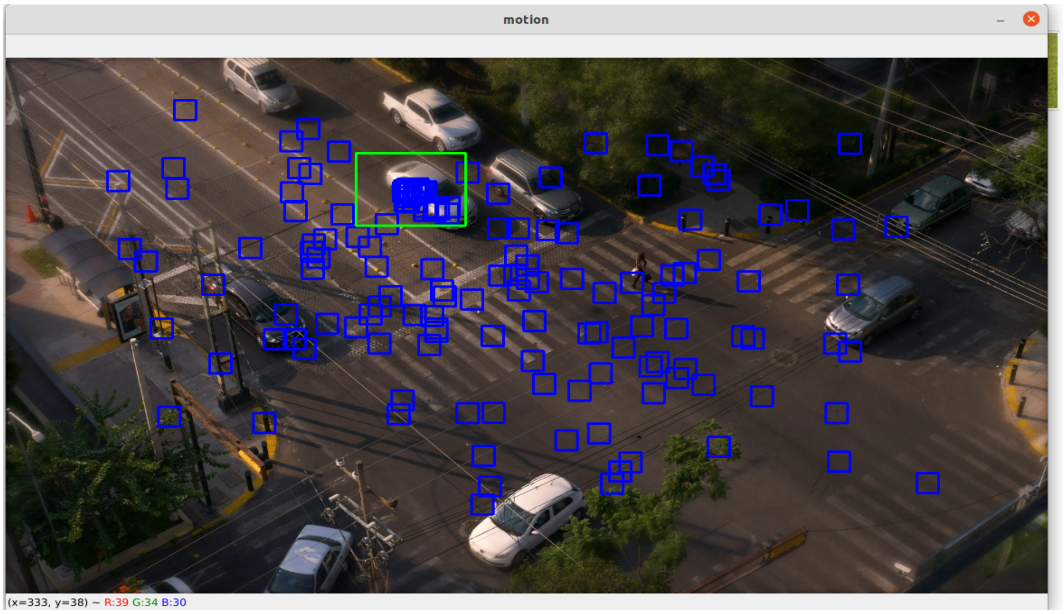


**Fig. 8.** Target caught.

| Number of particles | Number of iterations | With IPC | Without IPC | Difference |
|---|---|---|---|---|
| 1 | 1 | 11.98 | 8.23 | 3.75 |
| 1 | 3 | 12.38 | 8.54 | 3.84 |
| 1 | 10 | 16.78 | 12.84 | 3.94 |
| 20 | 1 | 24.67 | 24.38 | 0.29 |
| 30 | 2 | 60.56 | 58.70 | 1.86 |
| 30 | 3 | 92.94 | 86.5 | 6.44 |
| 40 | 1 | 45.36 | 43.9 | 1.46 |
| 40 | 2 | 81.96 | 73.35 | 8.61 |
| 40 | 3 | 116.3 | 115.5 | 0.8 |
| 50 | 1 | 152.93 | 131.45 | 21.48 |
| 50 | 2 | 347.47 | 335.66 | 11.81 |
| 75 | 3 | 646.34 | 628.04 | 18.3 |
| 90 | 3 | 766.11 | 663.98 | 102.13 |
| 100 | 3 | 684.58 | 844.97 | -160.39 |
| 150 | 3 | 996.27 | 1004.59 | -8.32 |

**Fig. 9.** Block Diagram.

We performed a thorough research on this approach by comparing the time for execution for the main process with IPC and without IPC. The table above shows the complete data of this analysis. We can observe here, that if the no of particles and number of iterations values are low which means low computation power, we don't find a lot of difference but if they are high which means high computation power is required then IPC performs it pretty good comparatively, and that's what the optimisation achieved.

## 6 CONCLUSION

Advancement in techniques used for programs which can find a target in a footage, is always a requirement, and the present project is a new approach to solve this problem, it uses AI and ML based techniques to get the best results, it might not show its capabilities when the load is low but when the load is high, there is a drastic change in the time required for completing the task compared to the program without multi-programming. This improvement in the performance is at the greatest priority in the computer science industry and the present project contributes a lot to it!

## ACKNOWLEDGEMENTS

We wish to express our sincere thanks and deep sense of gratitude to our project guide, Dr. Phrangboklang Lyngton Thangkhiew, Assistant Professor, School of Computer Science Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work. We express our thanks to our Head of the Department Dr. Justus S for his support throughout the course of this project. We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course. We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

## REFERENCES

[1]Ali Mirza, Ossama Zeshan, Muhammad Atif & Imran Siddiqi, Detection and recognition of cursive text from video frames. In: EURASIP Journal on Image and Video processing 2020, Article number: 34

[2]Nael Hirzallah, A Fast Method to Spot a Video Sequence within a Live Stream. In: JOURNAL OF MULTIMEDIA, VOL. 6, NO. 2, APRIL 2011

[3]Kennedy J., and Eberhart R.C., ”Particle swarm optimization”, Proceeding of the 1995 IEEE International Conference on Neural Networks, Perth, Australia, pp. IV: 1942-1948, 1995.

[4]Parsopoulos K.E. and Vrahatis M.N., ”Initializing the particle swarm optimizer using the nonlinear simplex method”, A. Grmela and N.B. Mastorakis (eds.), Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, pp. 216-221, WSEAS Press, 2002.

[5]Aakanksha Mahajan, Sushil Kumar and Rohit Bansal, ”Diagnosis of diabetes mellitus using PSO and KNN classifier”, Diabetes Mellitus, KNN, Pima Indians Diabetes DatasetDate of Conference: 12-14 Oct. 2017.