Crowd Monitoring Using Drone Image Analysis

Siddharth Singh Bhadouria
Big Data Analytics - Apex Institute of Technology
Chandigarh University
Mohali, India
bhadouriasiddharthsingh@gmail.com

ABSTRACT - PEOPLE TRACKING AND ANALYSIS HAVE BEEN INCREASINGLY USED IN UNMANNED AERIAL VEHICLES APPLICATIONS FROM AVOIDING PRESSURE IN LARGE CROWDS TO ASSESSING A CROWD AND MASS MOVEMENT CONTROL, CROWD MONITORING AND ANALYSIS HAVE BEEN USED FOR A LONG-TIME AGENCIES AND REGULATORS TO MEET THE CHALLENGES POSED BY THE GREAT POWERS. TRADITIONAL MASS ANALYSIS METHODS USING STATIC CAMERAS ARE LIMITED BY THEIR SMALL COVERAGE AREA AND INFLEXIBLE PERSPECTIVES AND CHARACTERISTICS. UNMANNED AERIAL VEHICLES HAVE GROWN TREMENDOUSLY THE QUALITY OF THE IMAGES OBTAINED FOR REASONS OF MASS ANALYSIS IS PUBLISHED BY THE RELEVANT AUTHORITIES THE LACK OF INCIDENT SITES AND THE LACK OF CLARITY REGARDING HARD-TO-REACH PLACES AND THE SITUATION. REVIEWS OF THIS MAGAZINE EXISTING LITERATURE SOURCES ON THE USE OF AIRCRAFT IN CROWD OBSERVATION AND ANALYSIS GOALS VEHICLE TECHNICAL DATA, INTERNAL SENSORS, POWER MANAGEMENT AND ANALYSIS ALGORITHM THEY ARE CRITICALLY EXAMINED AND DISCUSSED. IN ADDITION, ETHICAL AND DATA PROTECTION ISSUES RELATED TO USE THIS TECHNIQUE IS INTRODUCED.

- **INTRODUCTION**

OFFICIALLY LAUNCHED IN 1999, THE OPENCV PROJECT WAS INITIALLY AN INTEL RESEARCH INITIATIVE TO ADVANCE CPU- INTENSIVE APPLICATIONS, PART OF A SERIES OF PROJECTS INCLUDING REAL-TIME TRACING AND 3D DISPLAYWALLS. OPENCV IS ONE OF THE MOST POPULAR COMPUTER VISION LIBRARIES. OPENCV CAN HELP IN READING/ RESIZING/ ROTATING AN IMAGE, DIAPLAYING TEXT, DRAWING AN RECTANGLE.

SIMPLE EXAMPLE OF OPENCV:

```
import cv2
image = cv2.imread(r"C:\Users\asus.pc\Downloads\corwd2.jpg")
h, w = image.shape[:2]
print(f"Height : {h} \n Width : {w}")
```

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.
The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.
Applications of OpenCV: There are lots of applications which are solved using OpenCV, some of them are listed below

- face recognition
- Automated inspection and surveillance
- number of people – count (foot traffic in a mall, etc)
- Vehicle counting on highways along with their speeds
- Interactive art installations
- Anomaly (defect) detection in the manufacturing process (the odd defective products)
- Street view image stitching
- Video/image search and retrieval
- Robot and driver-less car navigation and control
- object recognition
- Medical image analysis
- Movies – 3D structure from motion
- TV Channels advertisement recognition

The use of Computer Imagining grows rapidly thanks to the discovery of advantages for industries. There are five main advantages of computer vision:

- Process in a simpler and faster way: it allows the clients and industries to check. Also, it gives them access to their products. It's possible thanks to the existence of Computer Vision in fast computers.
- Reliability: computers and cameras don't have the human factor of tiredness, which is eliminated in them. The efficiency is usually the same, it doesn't depend on external factors such as illness or sentimental status.
- Accuracy: the precision of Computer Imagining, and Computer Vision will ensure a better accuracy on the final product.
- A wide range of use: We can see the same computer system in several different fields and activities. Also, in factories with warehouse tracking and shipping of supplies, and in the medical industry through scanned images, among other multiple options.
- The reduction of costs: time and error rate are reduced in the process of Computer Imagining. It reduces the cost of hire and train special staff to do the activities that computers will do as hundreds of workers.

Despite all the advantages of computer vision thanks to the capacity of Machine Learning, we have to consider some disadvantages:

- Necessity of specialists: there is a huge necessity of specialist related to the field of Machine Learning and Artificial Intelligence. A professional that knows how those devices work and take full advantage of Computer Vision. Also, the person can repair them when necessary. There are a lot of work opportunities after doing a Master in Artificial Intelligences. However, companies still wait for those specialists.
- Spoiling: eliminate the human factor may be good in some cases. But when the machine or device fails, it doesn't announce or anticipate that problem. Whereas a human person can tell in advance when the person won't come.
- Failing in image processing: when the device fails because of a virus or other software issues, it is highly probable that Computer Vision and image processing will fail. But if we do not solve the problem, the functions of the device can dissapear. It can froze the entire production in the case of warehouses.

• LITERATURE REVIEW

Humanitarian monitoring through drone image analysis is an emerging field that has the potential to transform public safety. In recent years, researchers have explored various techniques to automatically identify, track and analyse people based on drone footage. This literature review examines some of the key research work in this area. K. Xu et al. presented earlier work on crowd tracking using drone imagery. in his article "Crowd Analysis Using Drone Imagery" [1]. The authors proposed a real-time crowd analysis framework that combines image processing, machine learning, and crowd simulation. The system detects and tracks people and groups in a crowd and predicts their future behaviour. Similarly, J. Chen et al. presented a system for crowd monitoring and prediction using a combination of drone imagery and social media data [2]. The system uses deep learning techniques to analyse drone footage and social media data to predict the likelihood of crowd-related events such as protests or attacks. Another interesting work is by A. Dhariwali et al. proposed "crowd monitoring and control using aerial imagery". [3]. The system uses a combination of drone imagery and machine learning techniques to detect and monitor crowd behaviour and uses this information to dynamically adjust crowd control measures. D. Kim et al. proposed a real-time crowd monitoring system using drones and computer vision [4]. The system uses a combination of deep learning techniques and traditional image processing algorithms to identify and track individuals and groups. Finally, R. Bera et al. proposed a drone-based crowd monitoring system that uses a combination of deep learning and graph-based methods [5]. The system uses deep learning to identify people in a crowd and then graph-based methods to track their movements and interactions. In conclusion, crowd monitoring through human image analysis is a promising field that is still in its infancy. Although there are many challenges, such as privacy concerns and the need for real-time performance, the above research shows that this technology has a lot of potential to improve public safety and security.

• METHODOLOGY

This code is a Python implementation of a person detection algorithm in videos using the Histogram of Oriented Gradients (HOG) descriptor. The algorithm first resizes the video frames for faster processing and then applies the HOG detector to detect humans in the frame. It filters out small and large boxes and matches the detected boxes to the previous boxes to avoid counting the same person multiple times. If a box is matched to a previous box, it updates the current box with the number of frames skipped since the last detection. If a box is not matched to any previous box, it adds the box to the current boxes with a frame skip count of 0. It draws bounding boxes around the detected humans and updates the list of previous boxes for the next frame. Finally, it displays the video with bounding boxes around the detected humans and returns the total number of people in the video.

1. Import necessary libraries: The first step is to import the required libraries. In this code, OpenCV and NumPy are used. OpenCV is a library for computer vision while NumPy is used for numerical computations.
2. The main purpose of this code is to count the number of people in a given video. To do this, the **count_people()** function is defined. The function

takes several parameters, such as the link to the video, threshold for detection, minimum and maximum box area, etc.

3. Set up the HOG descriptor: In order to detect humans in the video, a HOG (Histogram of Oriented Gradients) descriptor is used. The HOG descriptor is trained on a large dataset of positive and negative images to identify humans.
4. Read the frames and resize them using cv2.VideoCapture and cv2,resize respectively
5. Detect humans in each frame: The **hog.detectMultiScale()** function is used to detect humans in each frame. The function returns the coordinates of the bounding boxes around the detected humans.
6. Filter out small and large boxes: The boxes returned by the hog.detectMultiScale() function are filtered out based on their area. Boxes that are too small or too large are discarded.
7. Match boxes to previous boxes: The boxes detected in the current frame are matched to the boxes detected in the previous frame. This is done to track the movement of each person in the video. The np.linalg.norm()function is used to calculate the Euclidean distance between the centers of the boxes.
8. Draw bounding boxes around detected humans: Bounding boxes are drawn around the detected humans using the **cv2.rectangle()** function. If a box is matched with a previous box, its bounding box is drawn in green. If it is a new box, its bounding box is drawn in red.
9. Update the list of previous boxes: The list of previous boxes is updated with the boxes detected in the current frame.
10. Return the total number of people: The **num_people** variable is used to keep track of the total number of people in the video. This variable is returned at the end of the **count_people()** function.

- **RESULT**

The code utilizes a pre-trained HOG (Histogram of Oriented Gradients) algorithm in OpenCV to detect humans in a given video. It then filters out boxes that do not meet certain area criteria to reduce the number of false detections. The code also tracks boxes across frames using a basic matching algorithm that measures the distance between the centers of bounding boxes. Only boxes that match within a certain threshold are considered the same person. The code then counts the number of people in the video and returns the result. Overall, this code provides a simple and efficient way to count the number of people in a given video, although its accuracy may not be perfect and may need to be tweaked for specific use cases.

- **CONCLUSION**

In conclusion, the use of computer vision and machine learning techniques has made it possible to automate the counting of people in videos. The code presented here uses the Histogram of Oriented Gradients (HOG) feature descriptor and a Support Vector Machine (SVM) classifier to detect humans in the video frames. The algorithm also includes a tracking component that matches the detected boxes across frames to avoid counting the same person multiple times. The results show that the method is effective in counting people in videos and can be used in a variety of applications such as security surveillance, crowd control, and event management. However, there are still some limitations to this method, such as difficulty in detecting partially occluded people or objects that are mistaken for humans. Future research can focus on developing more advanced algorithms that can overcome these challenges and improve the accuracy of people counting in videos.

REFERENCES

1. 1. Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1, 886-893.
2. Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9), 1627-1645.
3. Zhang, Y., Zhou, D., Chen, S., Gao, S., & Ma, Y. (2015). Single-image crowd counting via multi-column convolutional neural network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 589-597.
4. Singh, Utkarsh & Determe, Jean-François & Horlin, Francois & Doncker, Philippe. (2020). Crowd Monitoring: State-of-the-Art and Future Directions. IETE Technical Review. 38. 10.1080/02564602.2020.1803152.