Smart City Monitor: Motion Detection

DS 620 - Group Project

Members:

Metin Senturk
Pooja Umathe
Janam Dalal

Index:

Introduction:

In this project, we will investigate the use of robustly computable motion features can be used directly as a means of recognition. We will design, implement, and test a general framework for detecting and recognizing both distributed motion activity based on temporal texture, and complexly moving, compact objects based on their activity which will alert the user of motion in the area or perform certain tasks.

This recognition approach contrasts with the reconstructive approach that has typified most prior work on motion. The motivation is the observation that in many instances, it is easier to detect and identify objects when they are moving than when they are stationary. Specifically, in the case of temporal texture, the investigators will extract statistical spatial and temporal features from approximations to the motion field and use techniques analogous to those developed for gray-scale texture analysis to classify regional activities such as windblown trees, ripples on water, or chaotic fluid flow, that are characterized by complex, non-rigid motion.

For action identification, we will use an independent motion detector to locate and track moving objects, and then used the spatial and temporal arrangement of motion features in conjunction with Fourier image analysis to flag and identify any objects that moved periodically. The work has practical applications in monitoring and surveillance, and as a component of a sophisticated visual system.

Tools and Techniques:

For this project we are planning to use the following tools and techniques.

- Data analysis and Preprocessing (Pandas, Numpy, Scipy)
- Machine learning. (imutils, sklearn, tensorflow)
- computer vision, image processing, (OpenCV)

For motion detection, we will be using the following methods.

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LCA)
- Linear Regression Models (Simple, Multi, Ridge, Lasso)

Team members:

1. Metin Senturk
2. Pooja Umathe
3. Janam Dalal

For the later project, we might combine our project with image recognition smart monitor team Anika Patel and Nandhini Kathod.

Project Timeline:

The project will consist of the following phases.

1. Preliminary research
2. Data preprocessing
3. Data analysis and feature selection
4. Motion detection in a video
5. Motion detection in a live camera
6. Motion tracking

According to this plan, we will follow the time schedule in below.

Start	Preliminary Research	Data Processing	Data Analysis	Training on Video Datasets	Testing on Live Video Implementation	Motion Tracking
	2 Days	1 Week	3 Days	1 Week	1.5 Week	3 Days

Phase 1:

Involves in research of motion detection case studies and researches. We will be focusing on motion detection models and following code base in order to implement it.

Phase 2:

Data collection method will be determined in this step. We will apply cleaning and feature operations at this step.

Phase 3:

The features gathered in Phase 2 will be tested on the model basis by its usefulness in terms of motion detection. We will move into experimentation of various combinations of the target devices and their algorithms.

Phase 4:

Motion models will be applied to a dataset of video files. The model will be trained based on the videos.

Phase 5:

Motion models will be applied to a live camera. The model will be evaluated based on the accuracy of detection.

Phase 6:

Motion in the video will be continuously detected.

Discussion:

Based on our findings we argue that motion recognition could be widely used in our daily life. So we will combine this technology with one of the most widely used other machine learning applications like image processing.

We are planning to merge our project with other team members who are doing image processing. The merged application will be able to differentiate objects from humans, and will also be able to detect motion. Additionally, facial recognition and motion detection based on facial expressions can be added in order to maximize user experience.

Application areas could be gesture detection, automatic doors, motion-based human-computer interfaces like virtual keyboards and etc.