Motion Event Recognition Using User Feedback
Popular Science Summary

Author: Hannes Jönsson

That a surveillance system can notify users when something happens, like triggering an alarm when motion is detected, is something we take for granted. But, of course, often we are not interested in everything that moves. How then can a surveillance system know which moving things we are interested in, and which we are not? Can a surveillance system learn from examples, provided by its user?

Today there are many ways in which motion detection in video surveillance can be tweaked, tuned and optimized to suit different needs. Filters can be applied, which tries their best to sort out things that are not of interest to the user. Things like small animals or tree branches swaying in the wind can be detected and ignored. But what about things that more closely resemble movement that *is* interesting? What if a user wants to be notified only when *some one else* arrives at their drive way, but not they themselves? How can a surveillance system learn to distinguish between two events, which seem so similar?

The use of machine learning techniques in visual domains, such as face recognition, object classification & tracking and action recognition has become ubiquitous. There are a lot of technologies were systems are exposed to huge amounts of training data for long periods of time, using extremely powerful computer systems, to learn to solve a specific task. This can mean something like learning to assign one of a certain number of possible labels to an image. Today such techniques can be used to aid in video surveillance, for example in order to recognize various objects such as cars, bikes and pedestrians. The question is, how well can the power of machine learning techniques be harnessed to create a system which can learn to distinguish between different events of motion.

In order for a user to be able to *train* a surveillance system based on examples of events the user wants to have recognized, either in order to ignore them or in order to raise an alarm because of them, techniques that use vast amounts of data and long training times must be discarded. Instead, a way to learn from just a few examples is needed. More over, a user should not have to present examples of things the user *doesn't* want recognized. A user wanting their own car to be recognized when entering their driveway should not have to present an example of every other car in the world to the system, but instead just show a few examples, or even a single one, of their own car, driving up on the drive way. This conditions combine the challenges from two groups of problems called *few-shot learning* and *one-class classification* respectively. Both of these groups of problems are relatively unexplored when compared to the more familiar case of assigning one of a number of labels to a picture.

To face the problem of working with a limited amount of data, a technique called *transfer learning* can be employed. With this technique, a neural network is trained for another task, one for which there exists a lot of data, in the hope that this will teach the net something that can be *transferred* to the new task as well. Solutions to the one-shot classification problem is harder to come by. In this thesis work two different approaches where

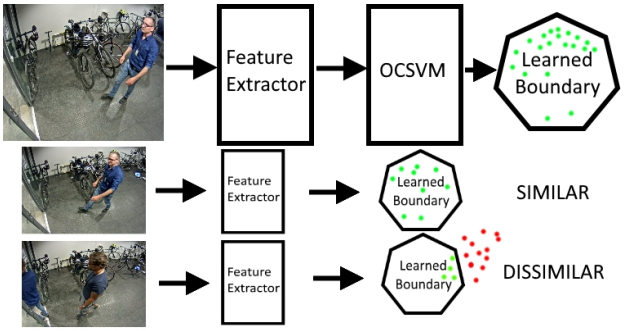

Fig. 1. Schematic overview of approach using One-Class Support Vector Machines to learn a representation of a type of motion event.

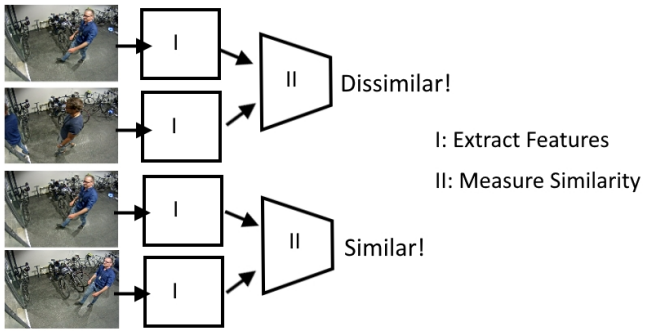

Fig. 2. Schematic overview of approach a neural network to measure similarities between events..

used. One using *One-Class Support Vector Machines* which try to learn a boundary around representations of the events we want to recognize, so that new events can be placed either inside or outside this boundary. The second approach instead relies more heavily on the transfer learning scheme, and uses a system which compares new events to old ones in a pairwise fashion, assigning a similarity score to each pair of events. In this way, a new event can be determined to be either similar or dissimilar to an event the user wants recognized, and has provided an example of. This second approach is implemented using *recurrent neural networks* to make use of the time aspect inherent to video sequences.

Experiments run both on large public datasets which tries to mimic scenarios found in the surveillance domain as well as on smaller datasets collected to the purpose of this thesis show that it is indeed possible to learn to recognize motion events in the form of video sequences. The greatest challenge comes in the form of the human-machine interaction aspect. How can the user of a such a system communicate their intent in a meaningful way, so that the system learns the *right* thing from a sequence portraying a motion event? And how can the system communicate what it has learned to the user? How can a user know how well the system will perform under unexpected circumstances? To answer these and related questions is the real challenge moving forward in this area of research.