University of Arkansas – CSCE Department
Capstone I – Preliminary Proposal – Spring 2019

Vigilant

Haven Brown, Austin Kreulach, Ian Moncur, Lane Trobee, Zane Turner

Abstract

Many people with disabilities or health conditions(such as being prone to Grand Mal seizures) are at some risk to collapse suddenly. Additionally, often times slip and fall injuries can go unnoticed for a long period of time, as such, our goal is to use computer vision to recognize slip and fall. This would have the potential to reduce response times in key situations as well as reducing liability in slip and fall cases.

1.0 Problem

In public spaces, whether a public park, grocery store, or mall, oftentimes an injury (ala a seizure or heart attack) can go unnoticed for key minutes that are critical in making sure that treatment is delivered as soon as possible. Additionally, oftentimes these spaces are already monitored by security cameras that computer vision could use to recognize a potential injury.

Additionally often times in slip and fall cases where liability plays a role getting footage of a slip and fall can be complicated due to the amount of footage generated by an entity (i.e. Corporations or CCTV) and footage relevant to such a case may be deleted long before it is asked for, or deemed necessary in a relevant court case. If the slip and fall could be recognized, footage could be put into a longer term storage in the event it's deemed relevant to a case.

2.0 Objective

The objective of this project is to use computer vision to recognize a fall.

3.0 Background

3.1 Key Concepts

This project is fundamentally about applying computer vision to a particular use case. Computer vision is a topic in computer science relating to the application of various graphics algorithms to the task of interpreting images or video in various ways. The goal is to first determine if a human is in the frame using edge detection, second determine the pose and thus status of the human using motion history images.

Edge detection is a basic idea in computer vision that discovers boundaries between objects, edges, by detecting sudden changes in the gradient of the image. That is, it detects edges by looking for abrupt changes in the color between two pixels. This technique is essential in almost all computer vision applications that aim to detect particular shapes in the source image. In our case, we want to combine edge detection with an empirically generated template for what sort of edges imply ‘human.’

The next key technology in the project is motion history images. This is an idea in computer vision wherein a history of an object’s motion is stored in a grayscale image that is useful in comparing to current behavior in order to make a more information guess as to future behavior. This is useful in this project because if the human shape deviates from the recorded motion history image and the program detects that the human has also entered either the lying down or crawling state, this is a likely indicator of a fall.

A final key technology in Vigilant will be OpenCV, which is an open source computer vision library for C++ used in many computer vision projects. Vigilant may or may not make use of this system depending on whether the project is developed in C++ or another language. However it is a very common, easy to use, and highly popular computer vision solution.

3.2 Related Work

In 2007, a group of researchers at various universities in Montreal, Canada developed an algorithm with a very similar goal to the one proposed in this project [1]. Their system first detects abrupt motion and then detects if the final shape is in a common position for sitting or laying down or if the final shape is an appropriate shape for those activities. Otherwise their system suspects a fall. Their system did suffer from false positives when the user would aggressively sit down, and their system also used a very low frame rate, a single camera, and low resolution images. One possible solution is to define inactivity zones on chairs and other objects, but this could result in falls caused by a slip while sitting down to be missed by the program. The team will examine these possibilities.

Another group, also in 2007, of scholars working at various universities in Italy built a multi-camera computer vision model for fall detection [2]. This team developed a system for transferring information about the perceived action of a person between cameras in order to better inform the surveillance as the person passes between rooms. This was intended to prevent a person collapsing while at an awkward angle between cameras such that neither individually had enough information to suspect a problem. This system is more advanced than the one proposed in the original paper, presuming that the camera is able to detect falls and instead focusing on the infrastructure around the detected fall. The paper proposes that a detected fall should result in a text(SMS) being sent to an operator along with a short transcoded recording of the fall by all relevant cameras. The paper also spends a good deal of time focusing on how a multi-camera system can solve the basic problems in identifying a person in a potentially ‘busy’ room where their edges might be blocked by various bits of household clutter. The paper uses a probabilistic model based on prior readings of the size and shape of the person living in the household, so that if a head is detected the system can guess where the rest of the body is. This idea is quite liable to failure, especially when switching cameras. Another proposed solution to this problem offers much more promise. Yue et al. [3] showed a system of calibrating occluded objects using a separate camera and two synchronized tracks mapped from one to the other by a transformation matrix. This method allows a multi-camera system to coordinate both cameras' videos such that both are aware of the location in their field of view even if only one can truly see the object. Vigilant may incorporate this idea in the final product.

4.0 Design

4.1 Requirements and/or Use Cases and/or Design Goals

Requirements of the product:

1. Distinguish humans from non-human items
2. Determine if a human has fallen and is having difficulties getting up
3. Capture images/video of the incident and relay this to staff of medical assistants

Use cases of the product:

1. Hospitals
    a. Expediting the speed of response to patients that have fallen
2. Public malls, public parks, etc.
    a. Provide assistance to people that might get isolated in an area and are unable to contact others for assistance

4.2 High Level Architecture

The main focus of Vigilant is aid existing video surveillance systems and provide more value by allowing it to assist in providing help to people in need. The development of the product has the potential to save lives, since the time it takes for people to receive aid once they have taken a serious fall is critical to how well they recover and/or survive. There will need to be some software development requirements that will take plenty of critical thinking and knowledge of artificial intelligence in order to effectively teach the computer to know when someone has fallen and is in need of assistance.

The main scope of the product is an add-on to an existing video surveillance system with a server that can handle running computer vision algorithms and connect to medical assistance to aid people in the case that they have fallen and need help. From here, the product will take the video of the person falling and send that to medical assistance as well as the individual's location to expedite the medical aid. The artificial intelligence and algorithms that determine if a person has fallen will need to be conditioned to properly detect when someone has fallen and needs help. It will take countless tweaks to prevent as many false - positives as possible, but it will be even more important to prevent false - negatives. A false alarm looks a lot better than dismissing a real situation. It will also need to be able to be integrated with existing surveillance systems, as if it is too difficult to implement this into an existing system, it will likely be disregarded.

The hardware required will be surveillance server system that presently hosts surveillance cameras and has enough processing power to maintain its current workload and run our product. The main focus of the product is to be a add-in to existing systems as it would possibly help aging systems that can have server upgrades but not rewire existing camera fixtures. This will mean that we will need to experiment and see the hardware requirements of the server and camera quality as both will have and impact on the efficiency of the products overall performance.

4.3 Risks

Risk	Risk Reduction
False - Positive False - negative	Optimize/ train fall detection algorithms with edge case scenarios
Adding Server Overhead	Optimize processes to use as little processing power as possible or determining a minimum hardware requirement

4.4 Tasks –

1. Research on computer vision models.
2. Selection of best model given research. Determining specific needs for the remainder of the project. (What type of data will we need? Will this be a supervised or unsupervised model? Will we keep a database? How much of an interface do we plan to build for users?)
3. Data collection and labeling.
4. Data manipulation and preparation.
5. Implementation of computer vision model.
6. Iterate upon model.
7. Interface for receiving model results.

4.5 Schedule –

Tasks	Dates

1. Research on computer vision models.	08/26-09/06
2. Selection of best model given research.	09/07-09/13
3. Data collection and labeling.	09/14-09/27
4. Data manipulation and preparation.	09/28-10/11
5. Implementation of computer vision model.	10/12-11/1
6. Iterate upon model.	11/2-11/15
7. Interface for receiving results.	11/16-11/29
8. Presentation prep and final report.	11/30-12/12

4.6 Deliverables –

- Design Document: A description of the components of our project.
- Model and components: If this ends up being a supervised model, any data used to train the model will be included in the final project submission.
- An interface for using the model (if time allows).
- Final Report

[The final deliverable to the instructor is a zip file containing all reports and code. Also, all results from the project are posted on your website (except, optionally, any proprietary code).]

5.0 Key Personnel

Austin Kreulach – Junior computer science and mathematics major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed

Artificial Intelligence, Algorithms, Formal Languages, and Numerical Linear Algebra. Austin has developed an artificial intelligence that composes music, an artificial intelligence that detects malicious activity on a server, and a celestial navigation system for unmanned aerial drones. Austin is responsible for the Python A.I. architecture(maybe?), and literature review(background research).

Zane Turner - Junior computer science major at the University of Arkansas. He has completed machine learning and data mining which are somewhat relevant to our project. He had an internship with Cerner where he did machine learning related work. Responsible for helping build the algorithm and collecting some of the data.

Haven Brown – Brown is a junior Computer Science major in the Computer Science and Computer Engineering Department and Pure Mathematics major in the Mathematics Department at the University of Arkansas. She has completed Artificial Intelligence and Algorithms. She also completed an internship at Tyson Foods, Inc. in data science and machine learning. She will be responsible for gathering research on and developing a computer vision model capable of detecting falls in public spaces.

Lane Trobee - Senior, Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Computer Networks and Programming Paradigms which will prove to be of use in the project. He is interning with JB Hunt with a primary focus on front end development and user experience design. He will be responsible for user interface and user experience design.

Ian Moncur - Senior computer science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He interned at JB Hunt and is also pursuing a degree in Applied Mathematics. He has completed Algorithms, Numerical Analysis, and Numerical Linear Algebra. He will be responsible for gathering research and sample data.

5.0 Facilities and Equipment

No facilities or equipment will be required for this project.

7.0 References

[1] Caroline Rougier, Jean Meunier, Alain St-Arnaud, Jacqueline Rousseau, "Fall Detection from Human Shape and Motion History using Video Surveillance," *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, Niagara Falls, Ont., pp. 875-880, 2007.
[2] Rita Cucchiara, Andrea Prati, Roberto Vezzani, "A Multi-Camera Vision System for Fall Detection and Alarm Generation," Expert Systems, Wiley, 2007.

[3] Yue, Z., S. K. Zhou, R. Chellappa, “Robust Two-Camera Tracking Using Homography,” Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004.