

Hack Cambridge 2017 - Neural Fall Detection

Real-time Event-based Monitoring System for Seniors and Elderly using Neural Networks

Team: Cristian Axenie, Simeon Kostadinov, Mark Mathews

The problem

It has been estimated that 33% of people age 65 will fall. At around 80, that increases to 50%. In case of a fall, seniors who receive help within an hour have a better rate of survival and, the faster help arrives, the less likely an injury will lead to hospitalization or the need to move into a long-term care facility. In such cases fast visual detection of abnormal motion patterns is crucial.

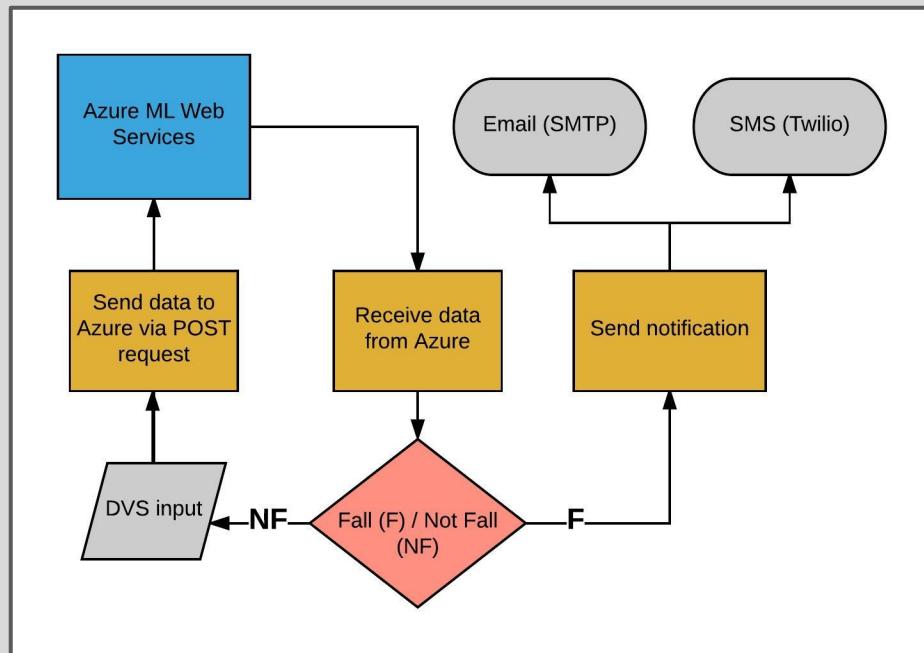
Our idea

In this project we propose the use of a novel embedded Dynamic Vision Sensor (eDVS) for the task of classifying falls. Opposite from standard cameras which provide a time sequenced stream of frames, the eDVS provides only relative changes in a scene, given by individual events at the pixel level. Using this different encoding scheme, the eDVS brings advantages over standard cameras. First, there is no redundancy in the data received from the sensor, only changes are reported. Second, as only events are considered the eDVS data rate is high. Third, the power consumption of the overall system is small, as just a low-end microcontroller is used to fetch events from the sensor and can ultimately run for long time periods in a battery powered setup. This project investigates how can we exploit the eDVS fast response time and low-redundancy in making decisions about elderly motion.

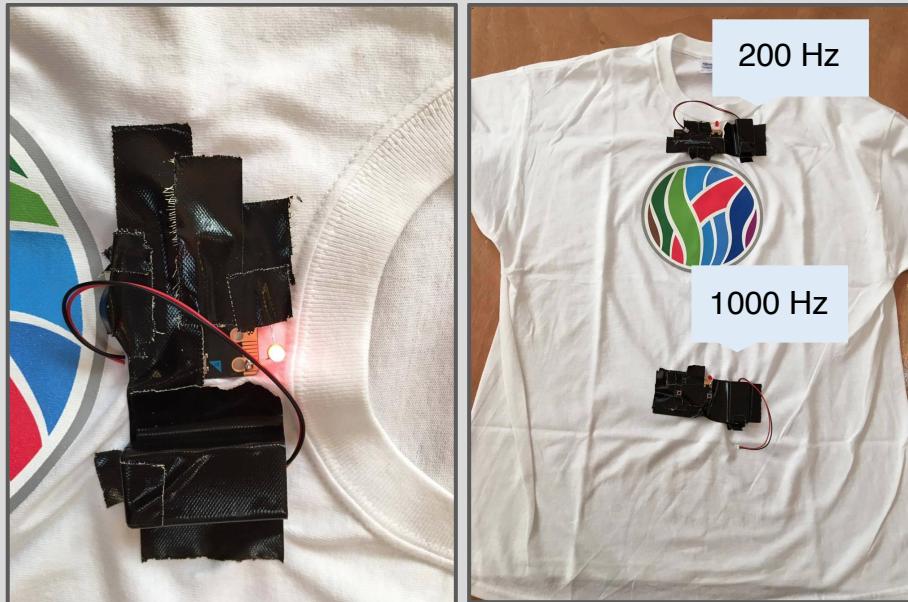
The computation backend will be realized with a neural network classifier to detect fall and filter outliers. The data will be provided from 3 stimuli (blinking LEDs at different frequencies) and will represent the actual position of the person wearing them. The changes in position of the stimuli will encode the possible positions corresponding to falls or normal cases.

We will use Microsoft Azure ML Studio to implement a MLP binary classifier for the 4 (2 stimuli x 2 Cartesian coordinates - (x, y) in the field of view) dimensional input. We labelled the data with Fall (F) and No Fall (NF).

Implementation



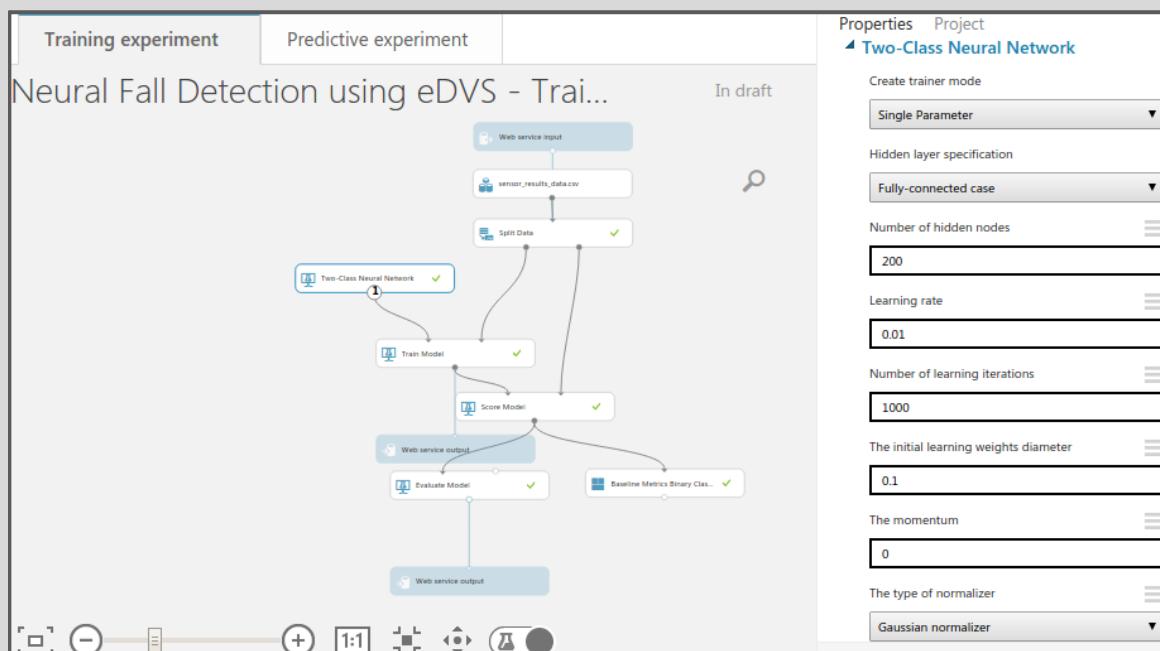
LEDs attached to a T-shirt flash at particular frequencies and are picked up by the DVS sensors.



Data from the DVS is processed using Python. The processed data is then fed into the Azure Machine Learning API via POST request, where a neural network that has been trained with supervised learning takes the 4D input and converts it into a binary output (F/NF).

Neural Fall Detection using eDVS - Trainin... > Score Model > Scored dataset

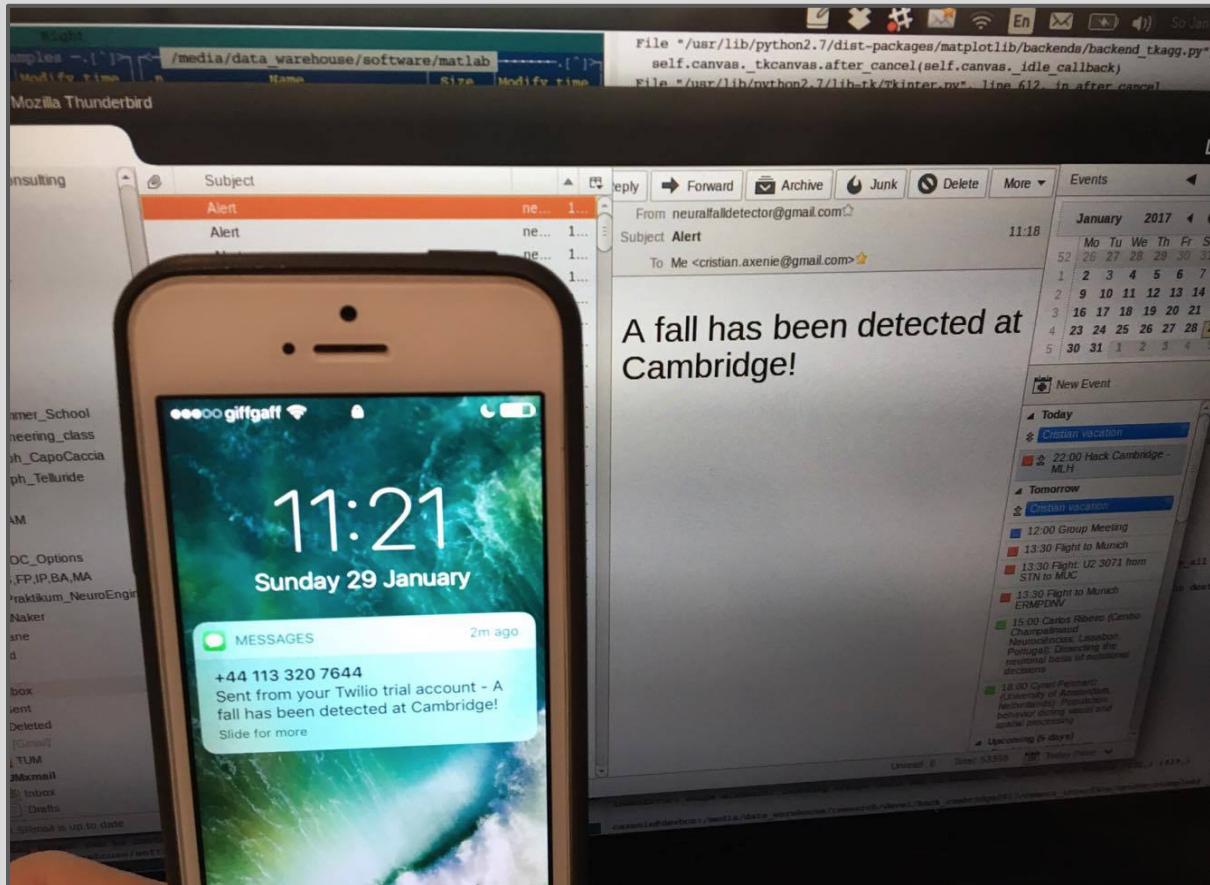
rows	columns						
35	7						
view as	xt	yt	xb	yb	state	Scored Labels	Scored Probabilities
0	0	0	0	0	F	F	0.272266
0.129176	-0.207166	0	0	0	NF	F	0.214065
0.75066	-0.358744	0	0	0	F	F	0.458239
0.785358	-0.337077	0	0	0	F	F	0.497197
0.493971	-0.526845	0	0	0	NF	F	0.20858
0.447774	-0.439014	0.127316	-0.315914	F	NF		0.532335
0.506557	-0.184636	0.491844	-0.18127	F	F		0.27882
0.184636	-0.506557	0.18127	-0.491844	NF	NF		0.516667
-0.664486	0.699149	0.138996	-0.498565	NF	NF		0.960255



Neural Fall Detection using eDVS - Training > Baseline Metrics Binary Classifier - Copy > Dataset

rows	columns	
21	3	
view as	F	NF
Accuracy	0.8	0.8
Precision	0.889	0
Recall	0.889	0
F1	0.889	
MajorityClassAccuracy	0.9	0.9
MajorityClassPrecision	0.9	0
MajorityClassRecall	1	0
MajorityClassF1	0.947	0
Kappa	-0.111	-0.111

This then decides the program flow: continue processing or send a notification through SMS and email in case someone has fallen. The notification will also communicate the location at which the fall has taken place.



Basic principle

