Goal

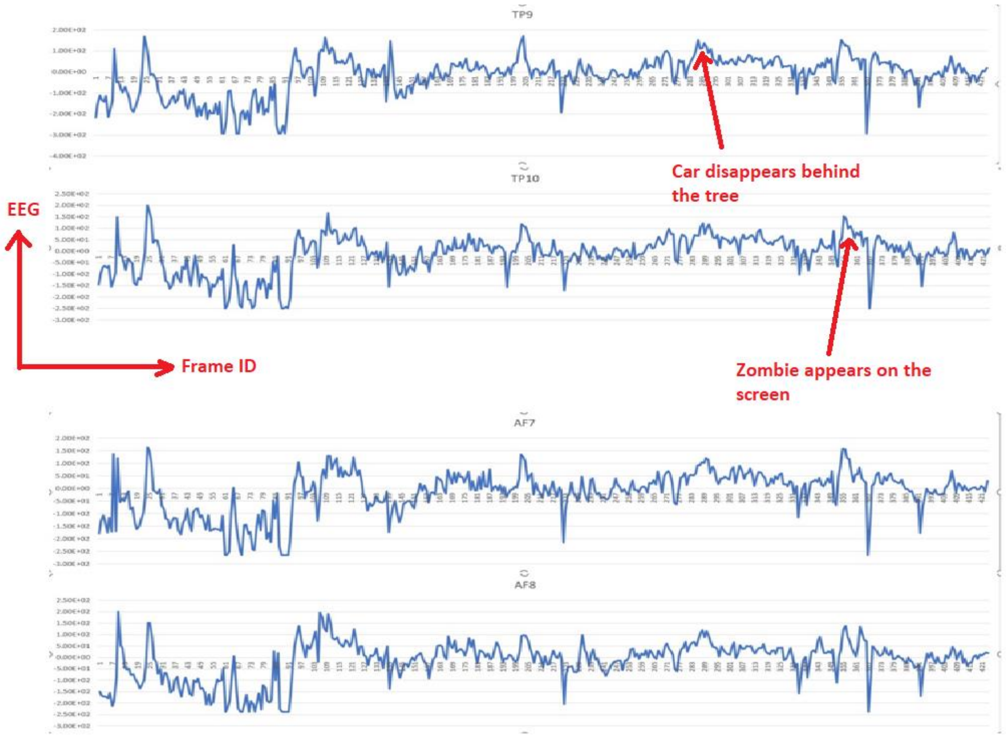
In this article, we are attempting to find *EEG correlates of surprise onset* incidences. We began with watching a video clip and recording the EEG data (while watching the clip for the first time). To mark the EEG data for *surprise onset*, we needed to synchronize the EEG and the video. In this article, we will introduce a method to synchronize the video and EEG. We will also introduce the EEG data extraction and labelling process in order to feed the data to machine learning algorithms, to find an *EEG surprise onset* pattern. These machine learning algorithms will create a model, which we will use over unseen video clips in order to detect *surprise onset* incidences.

Time Synchronization

In this article we are evaluating a method to synchronize video and EEG. This algorithm is based on video frames, which are rendered sequentially to display the video. The frames are rendered at a *variable frame rate.* We capture the local timestamp of all the frames. We log the “frame ID, local timestamp” in first Comma Separated Values (CSV) file. Here frame ID is an increasing integer. We also record the EEG in second CSV file in the format of “local timestamp, TP9, AF7, AF8, TP10”. Here TP9, AF7, AF8 and TP10 are the EEG potential across all four channel IDs, according to 10-20 electrode positioning system of Muse Headband. We then synchronize the EEG and Video with the following algorithm:

1. Pick the local timestamp from the CSV file across each frame in the first CSV file
2. Find the *same or immediate greater* local timestamp from the second CSV file (EEG recording)
3. For the timestamp found in step 2, log the frame number and the EEG potential across all four electrodes (TP9, AF7, AF8 and TP10) in a third CSV file.
4. Repeat steps 1, 2, 3 for all frames to have a complete log of frame ID and their corresponding EEG potential across all four channels.

We plotted the graph of EEG potential across all four channels verses the frame ID with the above algorithm. This graph is for the 20 sec video of Ghost Car [K-Fee - Auto (2004, Deutschland) - YouTube](). It looks like below.



Based on my observations of all four channels, they all display quite similar characteristics throughout the video. A positive deflection occurs when I am expecting a surprise ahead (the car disappears) and when the surprise is actually present (The zombie appears on the screen). This deflection occurs across all four channels.

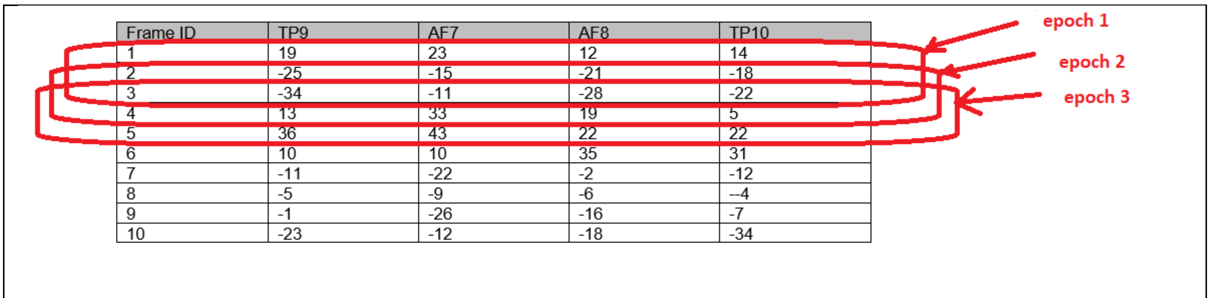
Data Collection and Training of the Model

At the end of time synchronization, we get a CSV file with frame ID and EEG potential (in micro volts) at the frame timestamp. If we put it in a tabular format, it looks somewhat like below.

Table 1 Data representation as an outcome of synchronization

Frame ID	TP9	AF7	AF8	TP10
1	19	23	12	14
2	-25	-15	-21	-18
3	-34	-11	-28	-22
4	13	33	19	5
5	36	43	22	22
6	10	10	35	31
7	-11	-22	-2	-12
8	-5	-9	-6	--4
9	-1	-26	-16	-7
10	-23	-12	-18	-34

We will slide over the frames data and create epochs as shown in Figure 1.



Frame ID	TP9	AF7	AF8	TP10
1	19	23	12	14
2	-25	-15	-21	-18
3	-34	-11	-28	-22
4	13	33	19	5
5	36	43	22	22
6	10	10	35	31
7	-11	-22	-2	-12
8	-5	-9	-6	--4
9	-1	-26	-16	-7
10	-23	-12	-18	-34

Figure 1 Epoching of EEG Vs Frame data

The frames data is grouped into epochs as shown in the Figure 1. Each epoch data is flattened across the designated rows above. The last column represents the epoch label in the Table 2. 0 indicates that there is no surprise and 1 indicate that there exists surprise in the epoch data. The labels are applied by manually observing the data. For Example: Let us say, in the ghost car video, the ghost appears at frame 4 in the data (by manual observation). This makes epoch 2 and epoch 3 both to be "surprise epochs", because both of them include frame 4.

Table 2 Epoch Data generated by sliding the window over the synchronized data

e1	19	23	12	14	-25	-15	-21	-18	-34	-11	-28	-22	0
e2	-25	-15	-21	-18	-34	-11	-28	-22	13	33	19	5	1
e3	-34	-11	-28	-22	13	33	19	5	38	43	22	22	1

This epoch data along with the labels, is further fed to the CNN to train the model.

Detect surprise over Unseen Data

Unseen data is divided into epochs in real time. These epochs are fed to the model sequentially. If the model detects a surprise for a given epoch, the corresponding frame is highlighted while rendering the video.

Further Requirements

1. Data generation: More videos and consequent frames for “surprises” as well as “no surprises”