A Comparison of Approaches for Synchronizing Events in Video Streams Using Audio

Mohammad Norouzifard\(^1\), Ali Nemati\(^2\), Saeed Mollaee\(^1\), Hamid GholamHosseini\(^3\), Joanna Black\(^4\), Benjamin Thompson\(^4,5,6\), Jason Turuwhenua\(^1,4\), and on behalf of the hPOD Study Team

1 Auckland Bioengineering Institute, University of Auckland, Auckland, New Zealand
m.norouzifard@auckland.ac.nz
2 Institute for Health & Equity, Medical College of Wisconsin, Milwaukee, USA
3 School of Engineering, Computer, and Mathematical Sciences, Auckland University of Technology (AUT), Auckland, New Zealand
4 School of Optometry and Vision Science, University of Auckland, Auckland, New Zealand
5 School of Optometry and Vision Science, University of Waterloo, Waterloo, Canada
6 Centre for Eye and Vision Research, Hong Kong, China

Abstract. A common scenario found in experimentation is to synchronize events, such as breaks between visual stimulus, with the video record taken of an experiment made of participants as they undertake the task. In our case, we recently synchronized a protocol of stimulus presentations shown on a laptop display, with webcam video made of participants’ facial and eye movements as they were shown trials of stimulus containing moving dots (a random dot kinematogram or RDK). The purpose was to assess eye movements in response to these RDK stimulus as a part of a potential neurological assessment for children. The video contained audio signals such as “beeps” and musical interludes that indicated the start and end of trials, thereby providing a convenient opportunity to align these audio events with the timing of known events in the video record.

The process of alignment can be performed manually, but this is a tedious and time consuming task when considering, for example, large databases of videos. In this paper, we tested two alternate methods for synchronizing known audio events using: 1) a deep learning based model, and a 2) standard template matching algorithm. These methods were used to synchronize the known protocol of stimulus events in videos by processing the audio contents of the recording. The deep learning approach utilized simple mel-spectrum audio signal feature extraction, whilst we adopted a cross-correlation algorithm that detected an audio template in the time domain. We found that whilst correlation was not effective as a means of beep detection; but our machine learning-based technique was robust with 90% accuracy in the testing dataset and did not the same amount of remediation required of the correlation approach.
1 Introduction

Audio or visual cues can be used to indicate salient parts of a video record taken of an experiment. Recently, we used this approach in a visual assessment experiment that occurred as part of a larger study, called the hypoglycaemia Prevention with Oral Dextrose (hPOD) study [5,17]. In this study, babies at risk of hypoglycaemia who had been administered a dextrose gel at birth participated in a follow-up test of visual function at two years old. The primary aim of that assessment was to measure global motion coherence threshold (MCT), the ability to see overall (or global) motion as measured by a visual stimulus [11] called a random dot kinematogram (RDK).

The reader is directed elsewhere for details of the visual assessment procedure [18]. However, in brief, the participants (2-year-old children) viewed a laptop screen whilst they were shown trials of moving dots (including community and home settings). At the same time, video of their facial area and eyes were recorded using a webcam or in-built camera of a laptop. The video records were analysed post-hoc by experienced observers trained to assess the presence or absence of eye movements induced by the visual stimuli; a highly challenging and time intensive process. Furthermore, key events, such as the start or end of a trial in the experiment were only signalled by audible “beeps” in the video record, thereby presenting a difficulty: how to align the known stimulus on the display with the corresponding position in the video? This is a problem common to experiments of this kind, and arises when time stamping of video recordings cannot be performed during data collection.

Manual alignment is a particularly demanding and time consuming task when considering studies with large numbers of participants. The purpose of this paper is to present a comparison of two methods for determining the occurrence of key audio events or audio events in uncontrolled environments (i.e., in-the-wild): 1) a method based on deep learning of frequency space features (specifically, the mel spectrum features) and 2) a standard template matching approach. This paper is organized as follows: firstly, we describe the two major approaches that were used to detect beeps within the audio signal. Secondly, we describe the experimental methods that were used on a private database of videos, and finally, we discuss how the audio signal was synchronized to the timestamp data. We find that overall, the deep learning method is a robust solution that does not suffer from event-detection sensitivity to the same extent as template matching. However, we make recommendations here toward both improved template matching and ML (machine learning) model based trial event detection.

2 Background

In this study, we were concerned with two approaches to identifying key events from video using audio: 1) a deep learning model and 2) a standard template-matching method.
matching algorithm. In the first approach, we employed a deep neural network to
detect and classify audio beep features in the frequency domain. We recognized
audible “beep” patterns using audio specific features (mel- spectrogram coeffi-
cients) which identified the start and end times of trials in the video footage. The
second approach employed template matching with a cross-correlation algorithm
to match times using the audio track of video streams. We have described both
approaches in Sect. 3. The rest of this section briefly explain algorithms which
were employed in our proposed pipeline. In both cases, the detected signals are
used to reconstruct the intervals in which visual stimulus was explained, and we
compare the two methods also from this perspective.

2.1 The Mel Spectrogram

The mel spectrogram is a representation of the audio content of a signal as it
varies over time. Whereas a typical spectrogram, determined using the Fourier
transform [2,8,10], is represented by a linear frequency scaling where each fre-
quency is spaced an equal number of Hertz apart, the Mel-frequency spectro-
gram is found by averaging the Fourier spectrogram values over mel-frequency
bands [1]. These bands are spaced according to a perceptual scale of pitches
judged by listeners to be equal in distance from one another [3].

2.2 Template Matching with Cross-Correlation

We utilized for comparison purposes a common template matching approach
for detecting the time of audio events from the soundtrack of the video: cross-
correlation in the time domain [16].

The approach depends on having a fixed kernel or template of sound data,
which slides along the incoming audio signal producing a correlation value at
each point along the signal. By sliding the kernel along the test signal we aim
to generate a time series of correlation values, and from it we locate point(s)
of maximum correlation. The cross-correlation, G which is defined by equality
(Eq. 1) which is specified by measuring the similarity between two waveforms, a
fixed kernel h and 1D/ 2D input signal data F. The cross-correlation equation
for 1D signal input data is as follows:

\[ G[i] = \sum_{u=-k}^{k} h[u] F[i + u] \] (1)

\( G[i] \) represents the output 1D array at index value I, as the correlation of kernel
h (template) and F, the input signal array.

3 Data Collection

The collection of these eye-tracking videos followed the tenets of the Declaration
of Helsinki. Written informed consent was obtained from all study participants.
Participants were young children who had been measured by laptop or computer
by an experienced assessor.
3.1 Database of Videos

Videos were collected as part of the hPOD study [5]. Experienced assessors performed the protocol which occurred either in assessments in an office environment at the University of Auckland, or in a home-based assessment in which a laptop and in-built camera/webcam were used to perform recording.

Videos contained the facial and eye areas of the participants as they viewed a sequence of 50 presentations. These presentations were shown in groups of five trials of RDK stimulus (6 s duration), followed by an animated movie. Before and after each trial were a potentially variable number of computer-generated beeps, with the trial start beep being different from the trial end beep. A short children’s animated movie (10 to 20 s) occurred after every five trials to encourage children to concentrate on the experimental test. We worked with a subset of five different participants’ videos (total of 32 min, 200 trials during which time a stimulus was shown to the participant). The recorded videos were roughly 6 min in duration, but the time length of each video varied according to factors such as attention of child and setup time. Two of the videos were employed for training and self validation of the deep learning model, and three unseen videos were utilized for testing purposes.

3.2 Manual Annotation for Training

We extracted mono audio of videos with Python scripting, and using Audacity(R) version 2.4.1 (https://www.audacityteam.org) recording and editing software to visualize the audio signal and beep annotation [13]. We manually labelled training data including beep events, and the time duration of trials using the “label” tool in the Audacity software application. This is referred to as manual annotation.

4 Methods

In this paper, we used two algorithms including deep learning and template matching which are described in this section.

4.1 Deep Learning Model

We utilized a pipeline for deep learning method shown by Fig. 1. Audio extracted mel-spectrum features were fed to the input layer in a deep MNN (multilayer neural network) [9,12]. We extracted mel-spectrum features in the frequency domain using a 4096 window size (of samples) and 512 hop length (the distance between two subsequent windows).

The MNN consisted of four fully-connected (dense) layers, 256 neurons in the first layer, 128 neurons in the second layer, then a layer with 32 neurons, and finally two neurons in the last fully connected layer. We employed a ReLU activation function on the first three layers and the Softmax activation function
Fig. 1. Schematic diagram with deep neural network approach and mel-spectrum feature extraction in the frequency domain [7].

on the last layer (see Table 1). The task of the activation function was to express a final output for given inputs. The proposed model used the Adam optimizer and categorical cross-entropy as a loss function [4, 6].

We extracted \(\sim 10,000\) feature vectors from the mel-spectrogram data, which were assigned to either a beep class (1456 feature vectors), or a trial class (8654 features). The labelled features were used for training (two videos used to fit and tune the parameters of a classifier model) and validation (three new videos used). Because of the unbalance of data toward the trial class, we needed to tune the deep MNN model to achieve high-performance synchronization. The test dataset was used only to assess the performance of the final model fit on the training dataset [14]. The training feature dataset was divided into two parts, 80\% for the training set, 20\% for validation.

Training was performed by cycling through the total training dataset (or an epoch - which is one cycle). The training set was divided into batches of

| Layer     | Input-size | Output-size | Comment |
|-----------|------------|-------------|---------|
| Input     | 32         | 32          | –       |
| Dense     | 32         | 256         | –       |
| Activation| –          | –           | relu    |
| Dropout   | –          | –           | 0.5     |
| Dense     | 256        | 128         | –       |
| Activation| –          | –           | relu    |
| Dropout   | –          | –           | 0.2     |
| Dense     | 128        | 32          | –       |
| Activation| –          | –           | relu    |
| Dropout   | –          | –           | 0.1     |
| Dense     | 32         | 2           | –       |
| Activation| –          | –           | softmax |
samples, and an iteration was the number of batches. For example, we had 9984 sample data in the training dataset, the batch size was 32 sample, yielding 312 iterations. Therefore, batch is the number of training samples in one iteration.

We used 10-fold cross-validation technique to determine the reliability of the result. We dealt with the over-fitting and unbalanced dataset problems using a dropout technique without data augmentation [6,15].

4.2 Template Matching Model

Template matching was applied to the audio input signal in the time domain. We used an example of a beep extracted from the data. The cross-correlation with the signal was determined in the time domain using a kernel of 0.5 s duration. The choice of this kernel was based primarily on time efficiency considerations. Template matching was slowed (in its present form) considerably by the choice of window length. In any instance, prior to correlation we applied a filtering using the Savitzky-Golay algorithm (Fig. 2.b) to reduce the effect of noise. The polynomials order was 2 and the window size was 41 samples.

We applied a very simple proportional thresholding for indicating beep times in the correlation time series signal to determine the correlation peaks corresponding to beeps. We detected maxima in the signal, and then applied a proportional threshold which was 85% of the peak with maximum amplitude. An example of an input audio signal is illustrated in Fig. 2.a indicating the presence of beeps pattern with peaks in the time domain. It is noted that this process resulted in “clusters” of peaks around each beep event.

4.3 Re-creation of Trial and Evaluation Metric

We recreated trial times, the times when visual stimulus was shown to the participant, from the detected beep events. As mentioned, the experimental protocol consisted of five trials in which the participant concentrated on the experimental test, followed by a movie. A variable number of beeps often occurred between...
each trial, because they occurred during a waiting period that was user controlled and depended on the inattention of participants.

We evaluated our results’ accuracy by determining the average overlap between trial intervals using both manual annotation and machine learning results. In Fig. 7, we present an example of an annotation showing in green, the times in which visual stimulus was shown to the participant, versus, in red, times when it was not. The thicker red bars shown after every fifth green stripe corresponds to times when the movie was shown.

We employed an overlap threshold to evaluate machine learning (ML) and template matching (TM) results. An overall overlap threshold was calculated as the average of trial times differentiating between manual labelling and our results as a proportion.

5 Results and Discussions

We ran the deep learning and template matching algorithms to align two videos (stimulus and eye displacement videos) based on beep event detection. Both models were deployed on the hPOD private dataset and results were compared with manual annotation by an expert.

5.1 Deep Learning Outcome

The proposed deep learning pipeline was applied based on Mel spectrum features. The training-loss value was 0.12, and training accuracy was 93.98%. The model validation loss function value was 0.16. Figure 3 illustrates accuracy versus loss in the training and validation stages of the Deep MNN model. Figure 3.b illustrates the reduction in loss of training and validation with increasing numbers of epochs. Figure 3.a describes the accuracy trend on training and validation with increasing numbers of epochs. Training the deep learning model took about 10 s

![Fig. 3. Scores of accuracy and loss function for training and validation stages on deep neural network.](image_url)
with TPU and 20 s on the training dataset with a GPU processor based on the Google cloud and we only needed to do it one time.

The trained model was tested with test data on the hPOD dataset for checking the model. The result was 90.24% accuracy with test dataset in comparison with manual labelled dataset using the average overlap measure.

5.2 Template Matching Outcome

An example template matching approach results for a beep template is illustrated in Fig. 4.

The method was found to be very sensitive and reduced signal to noise segregation (see in Fig. 5 and Fig. 6) compared with the ML based method. The template matching was sensitive with mixed noise and value of the threshold. Our experience was that the cross-correlation algorithm was simpler to implement, but more difficult to generalize on all audio within the large dataset with a long recording time. We assessed an average overlap of 86% and 78% with the template matching model in corresponding trials using overlap thresholds of 80% and 90% respectively.

Even though, template-matching algorithm was simple to run. The template matching algorithm was time consuming for a long video stream; as an example: processing time was about 148 s with TPU and 272 s with a GPU processor based on the Google cloud to calculate the cross-collaboration module with an input audio signal consisting of 22899366 samples (8 min) and the kernel with 20505 samples (0.5 s).

As mentioned, in order to have a time efficient model we employed 0.5 s for both beep and movie kernels. This could have resulted in reduced discriminative performance. One of the solutions may use down-sampling to optimise processing time to avoid mis-classification of movie and beep kernels. It was also found that events such as a child’s laughter could be mistakenly identified as a beep, it is illustrated in Fig. 6, and there is another false detection in Fig. 5, which is related to movie start time.

![Fig. 4. Result of using template-matching algorithm with beep kernel.](image-url)
Fig. 5. The first five trials with movie false detection (shown as error). Beep detection (orange points) using a proportional threshold applied to the cross-correlation with false detection. (Color figure online)

Fig. 6. The second five trials containing a child’s laugh as a false detection (shown as error).

5.3 Reconstruction of Trial Times from Detected Events

In Fig. 7, we show reconstructions of the complete trials for machine learning, template matching, and manual annotation results. The results indicate a good reconstruction of the experimental record using deep learning in particular, over the simple template matching algorithm.

The results indicate many inaccuracies for the template-matching model in the long video stream. This appears to be due to poor detection of beep peaks. We determined an average overlap of 94% and 84% with the machine-learning model using overlap thresholds of 80% and 90% respectively.
Fig. 7. The reconstruction of trail times from detected events with both machine learning (ML) and template matching (TM) algorithms. Red one is noise including beep or movie, Green one is a period of visual stimulus (trial time). (Color figure online)

6 Conclusion

We studied an audio event pattern recognition technique to synchronize unseen video streams in frequency and time domains using a deep neural network and template matching algorithms. Mel-spectrum audio signal feature extraction and cross-correlation algorithms were employed to detect particular audio patterns in-the-wild.

We utilized roughly 10,000 training and validation feature vectors with optimization and tuning in the function of each layer of the deep neural network model to achieve accurate results. The machine-learning model we applied was more robust than the cross-correlation model to detect beep events.

In this paper, the application of deep neural networks in audio signal processing and pattern recognition was studied, in comparison to a standard template matching approach. Template matching has the benefit of conceptual simplicity, but in our work, we found that it can suffer when the input signal is degraded by noise and in long video streams. However, in comparison the deep learning approach had good performance and can detect trial time accurately to synchronize audio signal in-the-wild with an unseen video stream. Contributions of this paper are as follows:

- We employed feature extraction with a template matching and a deep neural network model. The template matching was not effective over the dataset we tested, and could not be generalized, at least in its present form.
- The indicate that mel-spectrum features employed in a machine learning algorithm enables a consistent and generalized model. At least in this small dataset.

Overall, this research points toward an appropriate path for two video alignment using a deep learning algorithm.

References

1. Anden, J., Mallat, S.: Deep scattering spectrum. IEEE Trans. Signal Process. 62(16), 4114–4128 (2014). https://doi.org/10.1109/TSP.2014.2326991
2. Bailey, D.H., Swarztrauber, P.N.: A fast method for the numerical evaluation of continuous Fourier and Laplace transforms. SIAM J. Sci. Comput. 15(5), 1105–1110 (1994). https://doi.org/10.1137/0915067

3. Byrne, C.L.: Signal Processing: A Mathematical Approach. CRC Press, Boca Raton (2014). https://doi.org/10.1201/b17672

4. Golovin, D., Solnik, B., Moitra, S., Kochanski, G., Karro, J., Sculley, D.: Google vizier: a service for black-box optimization. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1487–1495 (2017). https://doi.org/10.1145/3097983.3098043

5. Griffith, R., et al.: Two-year outcomes after dextrose gel prophylaxis for neonatal hypoglycaemia. Arch. Dis. Child.-Fetal Neonatal Ed. (2020). https://doi.org/10.1136/archdischild-2020-320305

6. Hamad, R.A., Kimura, M., Lundström, J.: Efficacy of imbalanced data handling methods on deep learning for smart homes environments. SN Comput. Sci. 1(4), 1–10 (2020). https://doi.org/10.1007/s42979-020-00211-1

7. LeNail, A.: NN-SVG: publication-ready neural network architecture schematics. J. Open Source Softw. 4(33), 747 (2019). https://doi.org/10.21105/joss.00747

8. Lenssen, N.: Applications of Fourier analysis to audio signal processing: an investigation of chord detection algorithms (2013)

9. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E.: A survey of deep neural network architectures and their applications. Neurocomputing 234, 11–26 (2017). https://doi.org/10.1016/j.neucom.2016.12.038

10. Madisetti, V.: Video, Speech, and Audio Signal Processing and Associated Standards. CRC Press, Boca Raton (2018). https://doi.org/10.1201/9781315216065

11. USPTO. report, Head and eye tracking patent application. https://uspto.report/patent/app/20200069179. Accessed 10 Jan 2021

12. Norouzifard, M., Nemati, A., Abdul-Rahman, A., GholamHosseini, H., Klette, R.: A comparison of transfer learning techniques, deep convolutional neural network and multilayer neural network methods for the diagnosis of glaucomatous optic neuropathy. In: Chang, C.-Y., Lin, C.-C., Lin, H.-H. (eds.) ICS 2018. CCIS, vol. 1013, pp. 627–635. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-9190-3_69

13. Release Notes 2.4.2. Audacity Wiki, 26 June 2020. https://wiki.audacityteam.org/wiki/Release_Notes_2_4_2. Accessed 20 Nov 2020

14. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge (2007)

15. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15(1), 1929–1958 (2014)

16. Zahid, S., Hussain, F., Rashid, M., Yousaq, M.H., Habib, H.A.: Optimized audio classification and segmentation algorithm by using ensemble methods. Math. Probl. Eng.(2015). https://doi.org/10.1155/2015/209814

17. McKinlay, C.J., et al.: Neonatal glycemia and neurodevelopmental outcomes at 2 years. N. Engl. J. Med. 373(16), 1507–1518 (2015)

18. Yu, T.Y., Jacobs, R.J., Anstice, N.S., Paudel, N., Harding, J.E., Thompson, B.: Global motion perception in 2-year-old children: a method for psychophysical assessment and relationships with clinical measures of visual function. Invest. Ophthalmol. Vis. Sci. 54(13), 8408–8419 (2013)