2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

A CNN and LSTM Network for Eye-Blink Classification from MRI Scanner Monitoring Videos

Ronan Bennett[1], Shantanu H. Joshi[2,3]

Abstract— **Eye closure changes brain activity, so eye-blink tracking of subjects undergoing resting-state functional magnetic resonance imaging (fMRI) is relevant for identifying when a subject blinks, falls asleep, or keeps their eyes closed. Existing MRI eye-tracking solutions use commercially available MR-compatible video cameras with tracking software that can fail on low-quality videos. In this paper, we propose a two-stage convolutional recurrent neural network to classify open and closed eyes from frames of MRI eye-tracking videos under variable camera conditions. The model extracts visual features from each video frame using a convolutional neural network based on the Inception-v3 model, then uses a long short-term memory network to incorporate temporal information encoded in the sequence of visual features over time. Our model is implemented in Keras and demonstrated on a dataset of MRI eye-tracking videos from the Human Connectome Project. We manually labelled frames from the dataset for training and evaluation. The network was able to classify eye-blink states with a precision of 0.739 and recall of 0.835 on a previously unseen holdout dataset under varying camera conditions, eye position, and video quality.**

Clinical relevance— **Functional mapping studies in psychiatry and neuro-development which rely on a resting state fMRI protocol may yield divergent results depending on whether the subject keeps their eyes closed or open or whether the subject falls asleep. The clinical relevance of this work is to introduce the eye state (closed or open) in brain imaging studies as a prospective covariate, and as a feature that can potentially control for sleep state as a confounding factor.**

I. Introduction

Resting-state fMRI is a widely established imaging modality for inferring functional connectivity in the brain [1]. The resting functional MRI (rfMRI) paradigm typically involves the subject lying in the MRI (magnetic resonance imaging) scanner with eyes open and fixated on a dark screen with a cross-hair. The entire acquisition lasts anywhere from 5 to 15 minutes, often involving up to four scanning sessions. The longer sequences are designed for improving the signal to noise ratio, but may introduce subject motion because of the difficulty to stay still for a long time. Additionally, younger subjects (children) or elderly subjects may have difficulty keeping their eyes open and may drift in and out of sleep. This can potentially introduce variability in the scanning experiment. Also, studies that focus on sleep deprivation may make the subjects susceptible to falling asleep in the scanner. The state of the art in determining the sleep state or the awake state of the subject still involves manual monitoring of the subject by means of a video camera that captures a video of the eye and saves it along with the scan.

There have only been a few attempts to tackle eye videos originating from MRI scanner mounted cameras. Although there are various commercially available eye tracking solutions for such cameras, they usually solve a limited set of problems. Such methods for MR scan eye monitoring are typically integrated with the camera and annotate the pupil and track the movement. Such pupil-tracking solutions can fail on low-quality videos. These solutions are proprietary and it's often difficult to combine eye tracking results from several scanners or multiple sites.

Most existing methods attempting to capture the state of the eyes from MR scanner have made the use of the fMRI scan itself instead of depending on the acquired eye video [2], [3]. Methods including both classical machine learning approaches and deep learning based approaches that have directly used MR eye videos have largely focused on a frame-by-frame training of the static images [4]. For example, the approach in Yiu et al. [4] has focused on pupil segmentation and gaze estimation, which can be effectively solved on a frame-by-frame basis using a fully convolutional neural network.

The problem of eye-blink tracking in MRI is further confounded by the field of view (FOV), the angle of the mount, partial obstruction of the eye due to the scanner coil, as well as reduced illumination and ambient lighting in the MRI scanner bore. Figure 1 shows four frames from four different MR scanner mounted videos from the Human Connectome Project data [5]. It is observed that all videos exhibit different FOVs, show a large variation in the background illumination, and some demonstrate occlusion of the eye.

In this paper, we propose a novel application of a combined approach of using a convolutional neural network (CNN) for eye feature extraction followed by a long short-term memory network (LSTM) to model the temporal evolution of the eye blinking from MR scanner mounted videos. To our knowledge, this is the first time that a recurrent neural network approach, and the LSTM in particular, has been applied to this specific problem. We applied this algorithm to a dataset of MR eye videos collected across different scanners and different sites. The output of the model classifies each frame of the eye video with the label "eye open" or "eye closed".

The contributions of this paper are as follows, i) application for the first time of a two-stage network combining eye

[1]Student in the Department of Computer Science at University of California, Los Angeles, USA ronanbennett@g.ucla.edu

[2]Ahmanson-Lovelace Brain Mapping Center, Department of Neurology, University of California, Los Angeles, USA, [3]Department of Bioengineering, University of California, Los Angeles, USA, s.joshi@g.ucla.edu

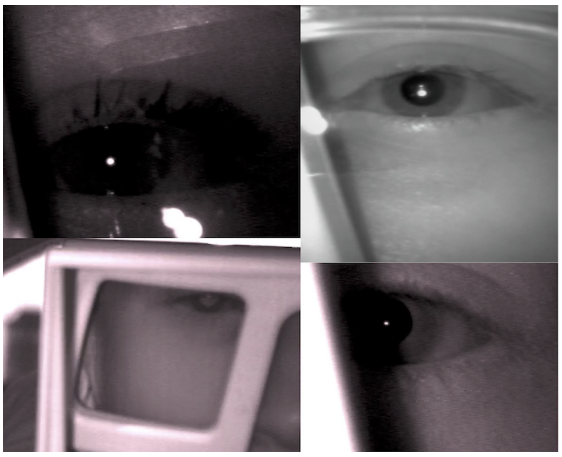

Fig. 1. Four static image frames of eye videos from different MR scanner mounted cameras across different sites.

features from a CNN and their dynamic state modelled by an LSTM for MR scanner mounted videos, ii) creation of a manually labeled dataset consisting of over $37K$ eye frames for training and testing, and iii) experimental evaluation of model parameters such as LSTM cell units and layers on a validation dataset, then evaluation on a holdout dataset yielding precision of 0.739, recall of 0.835, and 98.5% accuracy on videos from unknown camera models and settings. Experimental results show that the model can detect eye states while handling large differences in visual quality and eye position.

II. Data

The dataset consists of 30 MR scanner mounted eye videos from 30 distinct human subjects that underwent scanning using the Human Connectome Project protocols [5]. Each video represents a single fMRI scanning session at a site which varied across the United States. All videos consist of the subject blinking normally, with their eyes open most of the time. However, the blink rate varies widely between subjects. Each video is 6 minutes 30 seconds, 30 frames per second, and a resolution that varies across videos. The video camera model and settings are unknown. The aspect ratios of the videos are all close to 1:1, with side lengths between 500 and 800 pixels. All videos were converted to 299x299 pixels, with one grayscale color channel. Since some of the eyes are on the border of the frame, the images were resized to the new resolution without cropping, which introduces a slight stretching effect in videos with a non-square aspect ratio.

The dataset was split into a training set, validation set, and holdout set with a 60%, 20%, 20% ratio, yielding 18, 6, and 6 videos respectively. To eliminate bias, each labelled video is in exactly one of these three sets. We labelled the first 1100 frames of each video in the training set, and the first 1500 frames of each video in the validation and holdout sets. This corresponds to roughly 35 seconds of each video, thus yielding a labelled dataset of over $37K$ frames.

Since the dataset originally consisted of unlabelled videos, we labelled each frame in the training set with a ground truth value for the eye state. We implemented a Python program which allowed us to label each frame in a video, and store labelled frames in order. For our model to be maximally useful, we decided that it would only distinguish between the binary states of "eye open" and "eye closed". However, an eyelid can be in any position from fully open to fully closed, so we needed to decide a cutoff for an eye to be considered closed by our model. The most obvious choice is to label a frame as "eye closed" only when the eyelid completely covers the eye. However, this cutoff choice would result in frames with a very small amount of the eye visible being labelled as "eye open", and some blink events would not be labelled at all, since not every blink contains a frame with the eye fully closed. In this work, we define the "eye closed" class as any frame with approximately 10% or less of the iris being visible. This cutoff is roughly equivalent to the upper eyelashes beginning to overlap with the bottom eyelid. Of the 37800 manually labelled frames, 96.28% are in the "eye open" class.

III. Two-Stage Network for Eye-Blink Detection

Our model consists of a convolutional neural network followed by a unidirectional stacked long short-term memory network (LSTM) and is schematized in Figure 2. Since the eye location within frames varies between videos, and the eye is partially occluded for a few videos, we omitted an eye localizer and instead rely on the convolutional layer to extract the relevant features that have a sub-encoding of the eye-related features. The Inception-v3 CNN base model has moderate translation invariance due to the presence of maximum and average pooling operations, so we expected the classifier to perform adequately without a localizer, even though the eye position varies between videos. Table I outlines the parameters for the complete model.

A. First stage – Convolutional neural network

We used the Inception-v3 architecture for the convolutional neural network [6] with the Adam optimizer [7]. The base CNN model was pretrained on the 2012 ImageNet dataset [8]. These pretrained layers were frozen, and we used a transfer learning approach by adding additional trainable

TABLE I
Model parameters.

Parameter Name	Parameter
CNN Base Model	Inception v3
CNN Pretraining Dataset	ImageNet
CNN Loss function	Categorical cross-entropy
CNN Regularization	L2
CNN Output dimension	2048
CNN and LSTM Optimizer	Adam
Number of Trainable CNN Weights	4196352
Number of LSTM units	128
Depth of Stacked LSTM	3
LSTM loss function	Binary cross-entropy
Classification Threshold on Softmax	0.65

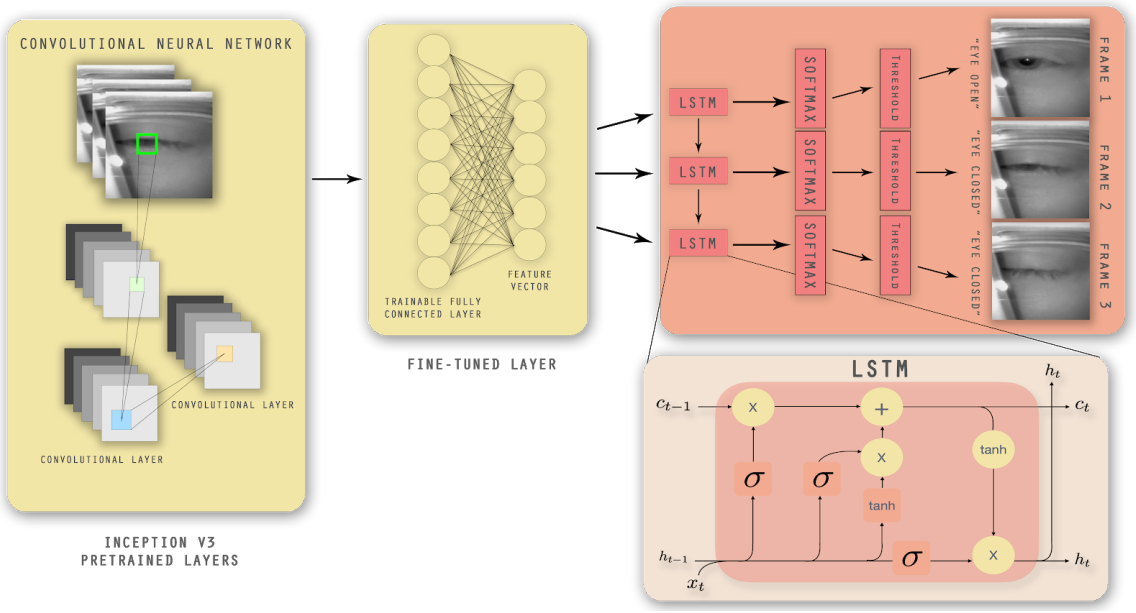

Fig. 2. Schematic of the two-stage framework consisting of the CNN as a feature extractor and the LSTM as a sequence modeler for MR eye videos.

layers to the CNN for this eye-blink problem. After the 2048-d output of the Inception-v3 model, we added a dropout layer, then a 2048-d to 2048-d fully connected layer. A 2048-d to 2-d fully connected layer was added to the end for CNN training only. We used categorical cross-entropy as the loss function, as given by:

$$\mathcal{L} = -\sum_{c=1}^{M} b_{i,c} \log(p_{i,c}), \tag{1}$$

where $b_{i,c}$ is 1 if the i^{th} observation belongs to class c, and $p_{i,c}$ is the predicted probability of the i^{th} observation for the class c, and $M = 2$ is the number of classes.

B. Second stage – Long Short-Term Memory Network

Unlike previous approaches [4] that have used CNNs for prediction of eye states, in this paper, we added a second stage consisting of a recurrent neural network. The output of the CNN consists of a $2048-$dimensional feature vector for each frame, which is a representation of the visual information in a single frame which is relevant to the eye state. Each CNN feature vector does not leverage the information stored in the ordering of those frames over time, such as the fact that each frame's eye state is likely to be the same as neighboring frames. We introduce temporal modelling of the state of the frames in the video sequence to effectively help us to classify eye-blinks with a Long Short-Term Memory network (LSTM) [9], [10]. LSTMs avoid the long-term dependency issue in traditional recurrent neural networks that make it prohibitive to learn with increasing sequence lengths. Each unit of the LSTM is a cell which manages its state. The input to each LSTM cell is the feature vector produced by the CNN, and the output is a 256 dimensional hidden state vector. Each cell also inputs the hidden state from the previous frame's cell. We utilized a stacked LSTM with 3 layers, then added a time-distributed softmax layer with two outputs, and finally a classification threshold which gives a binary output for each frame.

C. Model Training and Output

We trained the final layer of the CNN on our training data, treating each frame independently. The pretrained CNN base layers were frozen during training. Next, we evaluated each image in our training set with the CNN, yielding an ordered sequence of 1100 feature vectors for each video. We then trained the LSTM on these feature vector sequences, with frame ordering maintained during LSTM training. We used a validation set to tune the model hyperparameters, then evaluated the model on a previously unseen holdout set.

The final output of the LSTM is a softmax value for each class "eye closed" and "eye open" as floating-point values that add to 1. We convert these softmax values to a binary prediction with a classification threshold. We settled on a classification threshold of 0.65 on the "eye closed" softmax value for the model. We justify this choice in Section IV.

IV. Results and Conclusion

Table II outlines the experimental results on the validation set, which was used for hyperparameter tuning. In the experiments, we varied the LSTM inner cell dimension and stacked LSTM depth. We also varied the classification threshold from 0 to 1 in increments of 0.05 for each test. In each row of Table II, we searched all thresholds from 0 to 1 in increments of 0.05, and only report metrics for the threshold corresponding to the largest F1-score.

When deciding on the best model from the table, we considered the value of the F1-score, along with the area under the precision-recall curve. A larger F1-score corresponds to

Fig. 3. An example sequence of eye frames with ground truth labels and the output of the LSTM softmax layer for the "eye closed" class. Only the second image is manually labelled as having a ground truth of "eye closed".

a better tradeoff between precision and recall. Based on both these considerations, we decided the best performing validation set model was the 128 unit LSTM with 3 stacked layers with a threshold of 0.65 on the "eye closed" softmax value. Table III outlines the experimental results for our fixed

TABLE II

PERFORMANCE METRICS ON THE VALIDATION DATASET.

LSTM Hyperparameters	Precision	Recall	F1-score
32 units, 1 layer	0.853	0.922	0.887
64 units, 1 layer	0.834	0.938	0.883
128 units, 1 layer	0.836	0.931	0.881
32 units, 3 layers	0.746	0.690	0.717
64 units, 3 layers	0.791	0.899	0.841
128 units, 3 layers	*0.829*	*0.939*	*0.881*

final model on the previously unseen holdout set. Figure 3 shows an example sequence of frames along with the ground truth labels and the model softmax value for the eye being closed. Frames 1 and 3 are examples of model failure cases.

In this paper, we presented an application of a two-stage network that concatenated a CNN feature extraction stage and a dynamic sequence modeling LSTM stage to yield eye state classification from MR scanner mounted videos. We also created (to our knowledge) a large labelled dataset of over $37K$ eye frames for training and testing.

The high accuracy for the holdout set but lower precision-recall is due to the unbalanced dataset with many "eye open" states correctly classified. Future work could test the model on eye-blink detection benchmark datasets, such as HUST-LEBW [11]. The LSTM in the two-stage network could be replaced with a newer, attention-based model to yield higher performance. In fMRI systems with eye cameras, our model could be used for greater confidence in eye state detection. For existing analyses of fMRI data, our model could potentially allow the eye-blink state to be controlled for as a confounding variable, whereas future studies could use our model to explore whether eye-blink properties such as frequency and duration are indicative of psychiatric disorders.

TABLE III

PERFORMANCE METRICS ON THE HOLDOUT SET FOR THE FINAL MODEL.

	Accuracy	Precision	Recall	F1-score
Final Model	0.985	0.739	0.835	0.784

ACKNOWLEDGMENT

The eye tracking videos were obtained from the Human Connectome Project, "Mapping the Human Connectome During Typical Development" and their access was facilitated to the authors after a UCLA Institutional Review Board approval. No other identifying human subject data was available to the authors.

REFERENCES

[1] Stephen M Smith, Diego Vidaurre, Christian F Beckmann, Matthew F Glasser, Mark Jenkinson, Karla L Miller, Thomas E Nichols, Emma C Robinson, Gholamreza Salimi-Khorshidi, Mark W Woolrich, et al., "Functional connectomics from resting-state fMRI," *Trends in Cognitive Sciences*, vol. 17, no. 12, pp. 666–682, 2013.

[2] Michael S Beauchamp, "Detection of eye movements from fmri data," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 49, no. 2, pp. 376–380, 2003.

[3] Stefan Brodoehl, Otto W Witte, and Carsten M Klingner, "Measuring eye states in functional mri," *BMC neuroscience*, vol. 17, no. 1, pp. 1–10, 2016.

[4] Yuk-Hoi Yiu, Moustafa Aboulatta, Theresa Raiser, Leoni Ophey, Virginia L Flanagin, Peter zu Eulenburg, and Seyed-Ahmad Ahmadi, "DeepVOG: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning," *Journal of neuroscience methods*, vol. 324, pp. 108307, 2019.

[5] David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, Wu-Minn HCP Consortium, et al., "The wu-minn human connectome project: an overview," *Neuroimage*, vol. 80, pp. 62–79, 2013.

[6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[7] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[8] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[9] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[11] Guilei Hu, Yang Xiao, Zhiguo Cao, Lubin Meng, Zhiwen Fang, Joey Tianyi Zhou, and Junsong Yuan, "Towards real-time eyeblink detection in the wild: Dataset, theory and practices," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2194–2208, 2019.