Automatic Sleep Assessment from Eye Cues in Videos of Strongly Occluded Preterm Infants

Graduate School of Natural Sciences, Utrecht University

R. Horbach (Author, 6572626), R. Poppe (First Supervisor), A. Salah (Second Supervisor)

July 2023

Abstract

Monitoring the sleep of preterm infants provides valuable insights. Preterm infants are often strongly occluded, while the eyes are generally visible. Eye cues play an important role in manual sleep assessment of preterm infants. We exploit this correlation in an attempt to fully automate sleep assessment with (low-end) RGB cameras. We propose a framework to consistently extract eye regions in videos of occluded preterm infants. We show that convolutional neural networks (CNNs) can be trained on these regions to automatically identify eye states. We predict whether the eyes are opened or closed using a binary CNN, with a test accuracy of 96.3%. Using a sliding window and a binary 3D CNN, we also identify REMs, with a test accuracy up to 74.5%. We aggregate eye states per minute, and translate resulting features to sleep states with a random forest classifier. We manage to automatically discriminate sleep stages wake, active sleep and quiet sleep, with an accuracy of 92.2% - exclusively using eye cues. We discuss remaining issues and propose solutions to further improve the performance. Videos recorded at the neonatal intensive care unit of the University Medical Center Utrecht were used to construct labeled datasets.

1 Introduction

Preterm infants are born with a gestational age less than 37 weeks. Annually, roughly a tenth of births is preterm [1]. Preterm birth is the second largest cause of neonatal death, accounting for 35% [1]. Preterm birth also provokes long-term effects such as adverse neurological development and increased risks of several diseases [2]. Sleep during the early days after birth is key for the prospects [3].

Preterm infants spend their first days or weeks in a neonatal intensive care unit (NICU), in incubators. By monitoring the sleep of preterm infants, we can learn more about preterm sleep in general, make predictions on individual development [4] [5], and allow nurses to plan interventions more appropriately when the infant is awake. The latter importantly helps minimize sleep disturbance [6] [7].

Although experts are able to manually assess sleep states of preterm infants accurately - using methods such as BeSSPI [8], we prefer to automate it. Automating the process is advantageous, because we can monitor the infant continuously, and without subjective bias. Existing automatic methods are either obtrusive or not accurate enough. Polysomnography (PSG) is an accurate yet obtrusive method: it requires physical parts to be attached to the body, introducing risks for the fragile skin of preterm infants, and it can also potentially disturb sleep [5].

For this thesis, we introduce a camera-based approach to unobtrusively obtain visual information of a preterm infant. We put a (low-end) RGB camera somewhere around the incubator, aim it at the preterm infant, and then use the resulting video for analysis. This approach has been tried before [9] [10] [11] [12]. Their methods generally rely on finding body poses or facial landmarks, and are already able to detect various cues in infants. Two key issues are identified. First, datasets are needed to train classifiers, but are often limited due to privacy concerns of preterm infants [13]. Second, preterm infants are often strongly occluded in NICUs, causing important parts of the infants to be unobservable during analysis [14] [15] [13]. Given strong occlusions and little data, we are currently unable to reliably extract body poses and facial landmarks of preterm infants [13].

Large parts of the bodies and faces of preterm infants are often covered by blankets and medical equipment such as tubes. Extremely preterm infants are particularly strongly occluded. However, one or two eyes are generally visible. The eyes are also highly informative: eye cues play an important role in manual sleep assessment of preterm infants [16].

We present AVESSPIA (acronym for Automatic-Video-to-Eyes-to-Sleep-State-for-Preterm-Infants-Annotator), a new approach to fully automated sleep assessment, where we exploit the correlation between eye cues and sleep states. Figure 1 illustrates our approach.

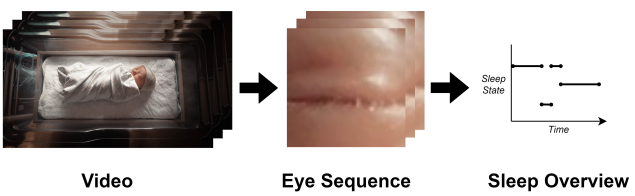


Fig. 1: High-level overview of our approach (eye image extracted with AVESSPIA).[1]

[1] Photo from Unsplash. URL: https://unsplash.com/photos/LtKoW6kh2eE.

1.1 Scope

We need video data of preterm infants. We position an arbitrary RGB camera outside the incubator, and make sure it captures the infant's face from the front. In dark rooms, camera sensors increase their sensitivity to pick up light, leading to noise in videos. The amount of noise varies strongly per camera. Noise makes it difficult to see eye cues, and should be a priority when considering cameras.

We consider eye states and sleep states as proposed by BeSSPI [8], a manual sleep stage classification system. We identify when eyes are opened (O) and closed (C), and when REMs occur. During REM, eyes can be opened (OR) and closed (CR). For sleep states, we consider W (wake), AS (active sleep) and QS (quiet sleep). We ignore IS (intermediate sleep), but generally when states have changed, we can assume IS has taken place in between.

We consider strongly occluded and extremely preterm infants. A gestational age of less than 37 weeks is defined as preterm; we consider infants down to a gestational age of 25 weeks. Extremely preterm infants have less frequent eye movements and a redder skin compared to preterm infants with higher gestational age [17]. Other than that, eye cues are similar in preterm infants of different gestational ages. Little is known about the differences between male and female preterm infants.

Preterm infants can be physically highly active. They may rotate their heads and move their arms and legs freely, but generally cannot change position on their own. Once in a while, a nurse or doctor may intervene for caregiving. We pause monitoring if both eyes of a preterm infant are occluded.

We also pause if the preterm infant is crying, as we need to interpret this separately [8]. Crying can be recognized through audio [18] [19].

In NICUs we have access to modalities such as heart rate information, but throughout this research we will limit ourselves to eye cues.

In order to train our classifiers, we have constructed labeled datasets from scratch with videos from the NICU of the University Medical Center Utrecht (UMCU).

1.2 Research Questions

Throughout the rest of this thesis, we will explore the following research questions:

I Can we extract eye regions in videos of preterm infants?

II Can we train a classifier to assess whether an eye is opened or closed, for preterm infants?

III Can we train a classifier to identify REMs, for preterm infants?

IV How well does our full pipeline perform in assessing sleep states of preterm infants?

A schematic overview of our research approach is given in Figure 2. For RQ I, we investigate how often the eye extractor manages to produce correct eye images. With RQ II, III and IV, we investigate the performance of predicting eye states and sleep states. We assigned labels for eye states ourselves; for sleep states, we compare with BeSSPI [8] annotations by human experts. We perform k-fold cross-validation to evaluate metrics. We consider accuracy, AUC, precision, recall and F1-score, and present confusion matrices.

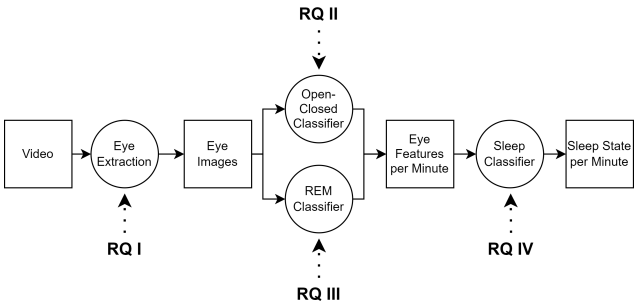


Fig. 2: Schematic overview of our research approach.

2 Literature Review

In this section, we discuss relevant literature. It is divided into three parts: sleep behaviour of preterm infants; methods for extracting eyes from images including further preprocessing techniques; different classification techniques.

2.1 Preterm Sleep

Preterm infants spend roughly 70% of their time asleep [5]; either in active sleep (AS) or quiet sleep (QS). They spend most of this time in AS, with approximately 60% [20] [21]. For preterm infants, uninterrupted periods of AS and QS generally last over 13 and 5 minutes respectively [3] [21]. For transitions between states, we use the term intermediate sleep (IS). The remaining time is defined as wake (W). Interestingly, the transitions of states can go in any direction at any moment, opposed to the fixed sleep cycles of adults [7]. If sleep states are only checked once in a while, interesting transitions may have been missed. Continuous monitoring is necessary to get a complete picture of a preterm infant's actual sleep.

AS can be confused with W and QS [8]. If a preterm infant is in QS, the infant is quiet with eyes closed. If they are awake, their eyes are opened (and moving [5]). During AS, next to appearing asleep, the infant can also have opened eyes during REMs [22] and be physically highly active suggesting W [5]. Of course, preterm infants can also shortly have closed eyes during W, through blinking for example. AS can be uniquely identified by rapid eye movements (REM) [3] [8] [5], with 20 second epochs of REM bursts [5]. Distinguishing opened eyes from closed eyes and recognizing REM is therefore key to assess the sleep state based on eyes alone. However, REM is not continuously visible: dur-

ing the intervals, AS may still be confused with QS. Prior experiments at UMCU indicated that REM occurs in approximately 53% of the minutes during AS. Sokoloff et al. [23] define individual REMs as eye movements that take less than one second, and found 0 to 15 REMs per minute for (term) infants during AS. We should be careful as eye twitches can look like REM for example [8]. IS can be characterized by a gradual change of cues from one state to another [3].

Behavioral sleep stage classification (BSSC) methods help researchers recognize sleep states [16]. They specify characteristic features per sleep stage, such as body and facial movements. A universal definition allows annotators to assess sleep states equally. One such method is BeSSPI, by De Groot et al. [8]. With this method, annotators assign sleep states to windows of one minute - they found that 30 seconds was too short to reliably register cues. Some researchers use a residual state in case of uncertainty; De Groot et al. propose instead to let annotators assign a confidence score to their annotations: 1 for 80-100% confidence; 0 for 50-80% confidence; -1 for 0-50% confidence.

2.2 Eye Extraction

We want to detect eye cues in videos. Our first step is to find the eye regions in frames, for which we consider body tracking and object detection. After we have extracted the eye regions, we consider image registration and color normalization for further preprocessing.

2.2.1 Body Tracking

Body tracking and facial landmark detection are two approaches to finding positions of specific body parts (respectively the entire body versus just the face, as the names suggest), among which eye and nose locations. As mentioned before, these approaches still do not work well for preterm infants [13]. However, at the UMCU they found that body tracking algorithm HigherHRNet [24] - despite not considering preterm infants specifically - can still reliably predict the locations of eyes and noses of preterm infants. It does not perform well for the other landmarks.

HigherHRNet is a CNN-based bottom up pose estimation model. In top down approaches - its counterpart, we first find the bounding box of one person, and then fit a pose within; in bottom up approaches, we first find keypoints with CNN-generated heatmaps for example, and then figure out best fitting poses [24]. It produces multiple poses if multiple bodies are visible, but we can simply extract the preterm infant by taking the pose with the highest confidence value. While a top down approach fails if it cannot find a bounding box [25], HigherHRNet can find the locations of the eyes and nose as long as these are sufficiently visible. HigherHRNet uses feature pyramids to deal with different body scales. The output of the locations are given in lower resolution than the input, so some precision is lost. Still, it provides a reliable base step for finding the rough eye and nose locations of preterm infants, even given strong occlusions. Pose detection techniques give confidence values per predicted body part, and can therefore also be used to predict whether body parts are visible [26].

2.2.2 Object Detection

Another approach to finding eyes is through object detection. We can use a region-based CNN (R-CNN) [27] to identify and find the bounding boxes (location, width and height) of objects. After extracting 2000 region proposals from an input image and warping these to square images, for each region, it uses a CNN to automatically extract features. Per proposal, with its features as input, the system then predicts a refined bounding box (through regression) and the presence of objects within. Computing this for 2000 images is a costly process. The authors introduced Fast R-CNN [28]. Here, they use the CNN once per image, to generate a feature map. They use this map to find 300 region proposals and to extract their corresponding features. Faster R-CNN [29] uses a Region Proposal Network (RPN) to find region proposals efficiently, speeding up the algorithm further. However, for finding faces of infants in videos, Li et al. [30] suggest to use Fast R-CNN, and simply find region proposals for subsequent frames by horizontally and vertically transposing the found bounding box of the previous frame. Mask R-CNN [31] introduces a third output - a mask, with the purpose of identifying the shapes of objects. Per object class, it trains per pixel within warped bounding boxes whether they are part of the object or not. Compared with Faster R-CNN, it takes approximately 20% overhead to calculate such masks for the 100 most promising bounding boxes [31]. Li et al. [32] use Faster R-CNN to find faces of infants and detect discomfort, also allowing occlusions to some extent.

Alternatively, Nagy et al. [14] use the accurate, fast and state-of-the-art YOLOv3 algorithm [33] to detect the faces and bodies of preterm infants. The hands, arms and caring artifacts are also recognized. YOLOv3 finds bounding boxes (at three different scales) and corresponding class labels (using multi-label classification) at once, making it a relatively fast object detection algorithm [33]. Salekin et al. [13] trained two separate YOLOv3 models for finding bodies and faces of infants, based on existing popular datasets WIDER FACE [34] and COCO [35] respectively.

2.2.3 Image Registration

The resulting sequences of eye images may be jittery, a potential problem for the later classification stage. To stabilize sequences of images, we can use image registration: find a transformation that aligns one image to a reference image. If the transformation is not rigid, shapes of the eyes may change.

Rigid transformations that align a face with a reference face are often based on locations of the eyes, and sometimes nose or mouth as well [36]. By transforming the image, they match the locations with those of

the reference image. The performance of alignment is limited to the precision as well as the resolution of the landmarks. Celona et al. [37] use five facial landmarks to align faces of infants. Sun et al. [38] simply rotate the faces of infants such that the landmarks of both eyes are horizontally aligned. There also exist feature-based methods [39]: instead of landmarks, they use salient structures in an image. Mahesh et al. [40] use a scale invariant feature transform (SIFT) algorithm to find such features for image registration. SIFT features can automatically be extracted from images, are invariant to position, scale and rotation, and are partially invariant to illumination and viewpoint.

To find a rigid transformation from one point set to another (reference) point set, we can use generalized Procrustes analysis (GPA) [41]. This technique minimizes the distances between two point sets: transpose both sets such that they are both centered at the origin, then scale both sets uniformly to unit size, and finally rotate one set such that the distances of corresponding points between both sets are minimized. Because of the last mentioned step, we can only align two point sets if we have a bijection between both sets. We can therefore not directly use this approach for feature-based methods such as SIFT. But, Eguizabal et al. [42] propose to combine GPA with dynamic time warping (DTW), to find correspondences for sets with different cardinalities. Alternatively, we can use methods based on the iterative closest point (ICP) algorithm [43]. ICP iteratively performs the following steps until convergence: for each point of the first set, select the nearest neighbours in the second point set (automatically finding correspondences); transpose both point sets such that their centroids (only including the selected points for the second point set) are aligned; rotate one point set such that the sum of distances between correspondences is minimized. Eguizabal et al. [42] claim to outperform ICP-based methods with GPA and DTW.

We can also perform image registration directly on images, without first extracting point sets. Such alternatives are expected to be more accurate for sequence registration where you want to fit frames to previous frames, as now the transformation is not limited to the precision of landmarks for example [36]. One example is the Lucas-Kanade (LK) algorithm [44], which is particularly popular for computing optical flow: given two subsequent frames, for every pixel in one image, find within a small window which pixel most likely corresponds in the other image. LK only works for small transformations, but could still be useful for sequence registration. Another example is Robust FFT [45], a Fourier-based method, and is able to estimate large translations, arbitrary rotations and scale factors up to 6. Fourier-based methods operate in the frequency domain, where we can use the phase differences between two images to find a transformation [46].

2.2.4 Color Normalization

In a NICU you continuously get different lighting conditions. It may be dark at any time. We want to disregard these conditions during classification. It is also desired to limit bias towards skin color. Hence, we are interested in normalizing the colors of our images. Additionally, contrast and brightness of images can affect the performance of a CNN [26]; color normalization can improve contrast and brightness.

Li et al. [47] use histogram equalization [48] to address variations in video quality of different infant videos, before feeding it to a classifier. Histogram equalization is a color normalization technique that operates on the histogram of an image's luminance channel. An RGB color space would have to be converted to a color space with a luminance channel, such as HSV. In histogram equalization you shift and merge the frequencies of values such that you get a uniform distribution. However, this does result in unnatural looking images [48].

Histogram specification is a generalized method of histogram equalization with which we can match a histogram to a reference histogram [48] [49]. With this, we can match the luminance of an image to that of a reference image.

There also exist techniques that preserve color information. Morovic et al. [50] use 3D image histograms. Here, each dimension represents a channel, where frequencies are stored at points in this space. Using the Earth Mover's Distance (EMD) metric, Morovic et al. find the transformation that minimizes the cost of moving from one distribution to another. This transformation can match the colors of an image to a reference image.

Reinhard et al. [51] mention the importance of the color space for matching colors to that of reference images. If a color space has negligible correlations between channels, you can normalize its channels separately. With this idea, they found success using the color space by Ruderman et al. [52]. They also propose methods to convert from and to this color space, given RGB images.

2.3 Classification

Previously, methods were described to extract eye sequences from videos. We want to automatically classify sleep states from these sequences. Below, we discuss different classification techniques, with a focus on CNNs.

The motion happening in the region of the eyes of preterm infants - our region of interest (ROI) - is complex. When we assess the eye state, we may use many cues or so-called features that are too complex to formalize manually. CNNs model the brain's visual cortex [53] - a part that handles sight - and are able to automatically learn complex features from images. We can connect these features - a 1D array of numbers - to a standard feedforward neural network (FNN), to learn relations between features and classes. We can read the predicted class from the last layer. Instead of FNNs, we can also use support vector machines (SVMs). An SVM is a binary linear classifier, and is for example used to predict the presence of pain in infants, given

extracted features [37] [54]. Weber et al. [55] use this approach to automatically assess whether a preterm infant can be monitored or not due to caregiving for example: they predict in a binary fashion whether a preterm infant is present and whether an adult is visible. Analogously, we could also train a model to predict whether an eye is closed or opened, given the image of the eye.

CNNs are also used for human pose estimation [9] [24]. Moccia et al. [9] go a step further and use a 3D CNN: they use a stack of frames as input, instead of a single frame. This way, temporal information is included, so that the CNN can also learn features across time. They stack 3 frames, covering 0.5 second, and found more robust pose estimations compared to using one frame.

To take many frames into account for video classification, Ng et al. [56] propose an adaptation of a CNN that can handle full length videos. They connect a long short-term memory (LSTM) [57] network to the CNN's feature output. An LSTM is a type of recurrent neural network (RNN), but performs better for long sequences. An RNN is similar to an FNN, but includes recurrent nodes: output of previous input is used as second inputs to recurrent nodes, allowing information to persist through time. However, training RNN models for longer sequences is difficult due to the vanishing and exploding gradient problem [58]. With LSTMs, we are able to control the importance of incoming information in a node using gates as well as its own memory. In turn, this allows us to effectively discover distant temporal relationships. Ng et al. [56] train their LSTM layers separately from the convolution layers, but they propose to integrate these for better results. Salekin et al. [13] use a CNN to find features in the faces and bodies of infants. They use an LSTM to capture temporal relations for features of consecutive frames. They work with video samples of 10 seconds with 5 frames per second, and are able to predict the presence of pain in infants.

RNNs and LSTMs are sensitive to overfitting, especially for smaller datasets [59]. We can combat this with regularization techniques [60]. Regularization techniques simplify neural networks to stimulate capturing generic relations over specific relations within data.

Ng et al. [56] were able to use samples of two minutes of video data for their predictions, with one frame per second. To capture more information related to motions, they mention the importance of optical flow. A so-called optical flow frame can be calculated from two subsequent frames - recall LK for example. The optical flow frame has two channels: one channel represents the direction (in degrees) of pixels from the first frame to the second frame, and the other channel represents the vector magnitudes. Optical flow frames are commonly used as input for CNNs when working with videos, to better include information of motion. As a downside, optical flow is a computationally expensive technique.

Our system needs to find subtle and complex motions. The task of recognizing micro-expressions is similar. Micro-expressions are spontaneous, brief and subtle facial movements. Kim et al. [61] found success recognizing micro-expressions, by using a combination of CNNs with LSTMs - similar to the aforementioned methods from Ng et al. [56] and Salekin et al. [13]. They use the CNN to learn spatial features, while using the LSTM to learn temporal features. As subtle motions need to be found, the CNN needs to find separating features for small spatial differences. Kim et al. introduce custom error functions that make the CNN find features that better discriminate different expression-states. As the authors needed more samples, they performed data augmentation on the available videos through horizontal flipping, rotation, translation, and scaling.

In case of subtle motions, to make differences between the frames larger, we can also use motion magnification. This technique has shown to improve accuracy of micro-expression recognition [62]. Eulerian Video Magnification (EVM) [63] is a popular motion magnification method and is able to magnify subtle changes between adjacent frames, even capable of clearly revealing blood flow through a human body. Xia et al. [64] use recurrent connections within their CNN as well as EVM to capture subtle motions for micro-expression recognition.

CNNs are able to automatically extract features from images. However, it needs relatively much data to train one [32]. To remedy the lack of footage of (preterm) infants, related works use pre-trained models on other datasets [13] [65] [37]. For example, they start with models that are trained on adults, and then fine-tune the weights on (preterm) infants [65]. A popular pre-trained model is VGG [66]. Salekin et al. [13] use a VGG model trained on the VGGFace2 dataset [67] - which includes adults - to automatically extract features from faces of infants, without fine-tuning the VGG layers to infants at all.

Next to CNN generated features, Celona et al. [37] also use histogram of oriented gradient (HOG) features for pain assessment in infants. A HOG descriptor [68] is a histogram that summarizes the gradients of pixels within a specific area. Sun et al. [38] use HOG features for detecting discomfort in infants, and call it an effective facial representation. Opposed to CNN features, HOG features are considered handcrafted features, and do not require a trained model [37].

Instead of using the features that a CNN generates, we can also train a CNN to detect so-called action units (AUs) in infants [11] [65]. AUs capture specific movements. For example, we can define a facial AU to capture whether the inner corner of an eye brow is raised.

We mentioned the use of CNNs to distinguish opened from closed eyes. Cabon et al. [10], given similar circumstances to our research, instead calculate the contours of preterm infants' eyes using image processing techniques. If they find a contour for an eye, they consider it opened, and otherwise they consider it closed. They manually indicate the location of the

eye, and track this region throughout the video with template matching. They also use image processing techniques to identify motion: they subtract a blurred previous frame from a blurred current frame (image differencing), threshold the result to black and white, and use the surfaces of white areas to identify frequent motion.

Finally, to distinguish opened from closed eyes, we could have also considered facial landmarks - if we were able to find them - by for example looking whether two predefined landmarks of an eye exceed a certain threshold [26].

3 Method

Our pipeline AVESSPIA converts videos of preterm infants to sleep annotations, by consecutively cropping eye regions from frames, classifying the eye regions with CNNs, and finally translating eye states to sleep states with random forest (RF).

3.1 Camera Setup

The camera should be placed at the height of the nose and eyes, as highlighted as a red circle in Figure 3. Not only does this expose the eyes and nose best and consistently, it also gives the opportunity to use the distance between the nose landmark and the center of the two eye landmarks as a stable reference for scale (and is invariant to rotations of the head), as will be useful in the following extraction step. The camera should also capture the front view of the infant's face.

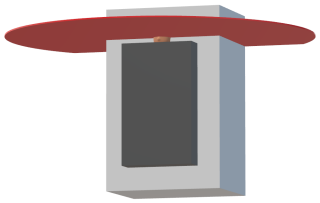


Fig. 3: Ideal camera location around incubator, illustrated in red.

3.2 Extracting Eyes

The extraction pipeline is summarized in Figure 4.

3.2.1 *Region Extraction*

When we have a video, we extract the eye regions per frame. While object detection techniques such as YOLO [33] are promising, it still requires a large number of labeled images with (rotated) bounding boxes around the eyes, which we do not have for preterm infants yet.

Instead, we integrated the publicly available HigherHRNet algorithm [24] to extract landmarks for the eyes and nose. As mentioned earlier, while it is not trained on preterm infants specifically, at UMCU it was found that eyes and nose landmarks perform well. The algorithm is designed to work with different body scales. Also, despite being a pose estimation algorithm, it can find eyes and noses even if all other landmarks are occluded. Altogether, it satisfies the exploratory goal of this research.

We calculate the angle α of the line crossing through both eye landmarks. We crop the eye regions with α. As an estimate for the scale of eyes, we multiply the distance between the nose landmark and the center of the two eye landmarks with 1.2. This is illustrated in Figure 5.

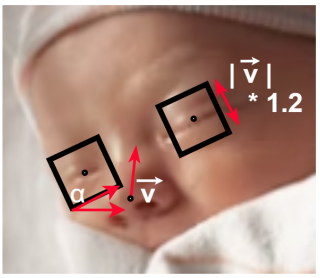


Fig. 5: Region extraction based on eyes and nose landmarks.[1]

Given an eye landmark, its angle α, and its scale, we can directly crop the eye region from the frame. We flip left eyes horizontally to appear as right eyes, to reduce variance. Figure 1 shows the typical output image for a given input frame.

HigherHRNet produces confidence scores, and we have found that landmarks are acceptable if the score is higher than 0.1. With a lower score, the landmark is usually far off from its target. If any of the three landmarks does not pass the threshold, we stop processing the frame.

We convert the eye image to grayscale. This way, the system is consistent with videos recorded in darkness. It is also expected to treat different skin colors more equally. We apply histogram equalization to bring out details, and to address variations in video quality of different videos as argued before [47]. Finally, to prepare the images for CNNs, we further normalize the pixels to the range of $[0, 1]$, and resize the images to a fixed resolution of 56 by 56 (generally, we found the eye region to be approximately of this resolution).

3.2.2 *Visibility Model*

Even though HigherHRNet produces confidence scores for landmarks, it can still produce a high confidence if it is (partially) occluded. If only the nose and one eye are visible, it can predict where the other eye should be. While this is an advantage, as we can deal with occlusions, we cannot reliably use the confidence scores to tell whether the eyes are occluded and which eye is visible best.

We trained a CNN model that can discriminate visible eyes from (partially) occluded eyes. We use a small architecture (see Table 1), due to the limited size of our dataset and the relatively low complexity of the classification task. With strides and max pooling layers, we

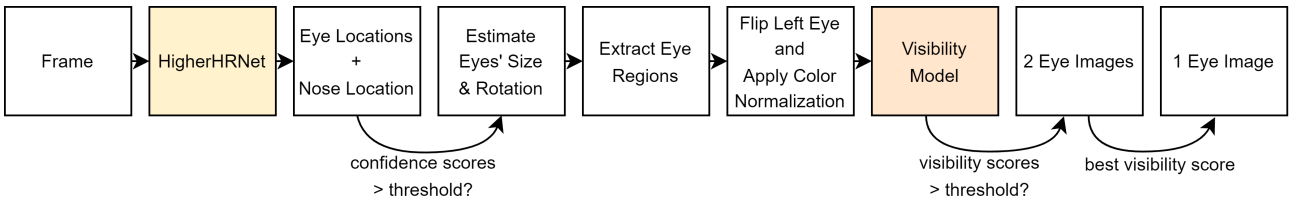


Fig. 4: Eye extraction pipeline.

control receptive field such that the first convolutional layer captures low-level features and the second convolutional layer captures high-level features. We reduce overfitting with dropout layers.

Layer	Params	Output
Input		56 x 56 x 1
Conv2D	5 x 5, stride 2 x 2, valid	26 x 26 x 6
MaxPooling2D	2 x 2, stride 2 x 2, same	13 x 13 x 6
Dropout	50%	
Conv2D	3 x 3, stride 1 x 1, valid	11 x 11 x 16
MaxPooling2D	2 x 2, stride 2 x 2, same	6 x 6 x 16
Dropout	50%	
Flatten		576
Dense		64
Dropout	50%	
Dense		1

Tab. 1: CNN architecture for visibility.

We suggest a batch size of 1 for the visibility model, to directly update the model after seeing a new sample. The negative class has high variance (the eyes can be occluded by anything), while we have few samples. We found that the model generalizes better to exceptions with a smaller batch size.

If the HigherHRNet landmarks of the eyes and nose are confident (they pass the predefined threshold), we employ the visibility model. If both eyes are considered visible (both pass a threshold of 0.5), we choose the eye with the best probability. We stop processing the frame if neither is visible.

3.2.3 *Constraints*

Ultimately, the eye extraction pipeline accepts a sample if following conditions apply:

1. Scores of eyes and nose landmarks are big enough.
2. One eye has a sufficiently high visibility score (no (partial) occlusions).
3. Throughout 1 second, the landmarks of the eyes are not too far apart (due to body or head movements). This will be discussed further in Section 3.3.2.

3.3 Classifying Eye States

The classification pipeline is summarized in Figure 6.

3.3.1 *Predicting Open-Closed*

Given the final eye images, we can now use CNNs to predict the states of eyes. We pass the eye images to a second CNN model we trained, to predict whether eyes are opened or closed. We use the same CNN architecture as for visibility (see Table 2). However, there are two changes. First, after investigating the output of color normalization on the grayscale eye images, we found that it becomes difficult to distinguish slightly opened eyes from closed eyes. A typical example of this is illustrated in Figure 7. Hence, we use the original colored eye image, without color normalization, exclusively for the open-closed model. Secondly, for the open-closed model, we use a batch size of 8, so we also apply batch normalization.

Layer	Params	Output
Input		56 x 56 x 3
Conv2D	5 x 5, stride 2 x 2, valid	26 x 26 x 6
MaxPooling2D	2 x 2, stride 2 x 2, same	13 x 13 x 6
BatchNormalization		
Dropout	50%	
Conv2D	3 x 3, stride 1 x 1, valid	11 x 11 x 16
MaxPooling2D	2 x 2, stride 2 x 2, same	6 x 6 x 16
BatchNormalization		
Dropout	50%	
Flatten		576
Dense		64
Dropout	50%	
Dense		1

Tab. 2: CNN architecture for open-closed.

(a) Original colors. (b) Normalized colors.

Fig. 7: Example image with original colors versus normalized colors, demonstrating difficulties for the open-closed model.

3.3.2 *Identifying REMs*

For detecting REMs, we use a 3D CNN (see Table 3), so that we can capture motions. The architecture has a third convolutional layer and more fully connected nodes, as the classification task is more complex than

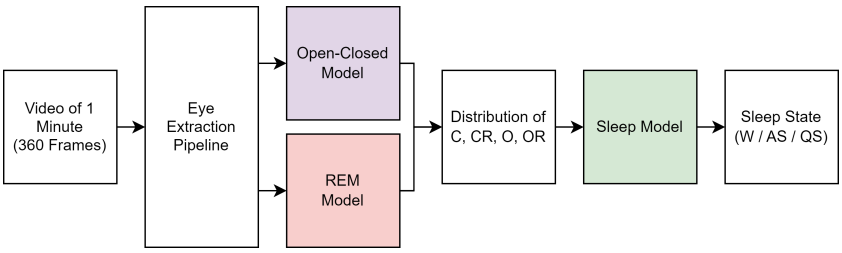


Fig. 6: Classification pipeline.

the previous two CNNs (low inter-class variance and high intra-class variance).

For input, we stack 6 frames, covering 1 second in total (a single REM takes around 1 second). We use a sliding window over the entire video, predicting REMs for every timestamp, as illustrated in Figure 8. If a window misses an eye image, we skip the sample. If both eyes are eligible, we use the eye with the best average visibility score. If the REM model then outputs a probability greater than a predefined threshold, we count the sample as a REM. We use a threshold of 0.9 to strongly reduce false positives and only count confident REMs. Since REMs come in bursts, we can afford to miss a few.

Layer	Params	Output
Input		6 x 56 x 56 x 1
Conv3D	1 x 5 x 5, stride 1 x 2 x 2, same	6 x 28 x 28 x 6
MaxPooling3D	1 x 2 x 2, stride 1 x 2 x 2, same	6 x 14 x 14 x 6
BatchNormalization		
Dropout	50%	
Conv3D	3 x 3 x 3, stride 2 x 1 x 1, same	3 x 14 x 14 x 16
Conv3D	3 x 3 x 3, stride 1 x 1 x 1, valid	1 x 12 x 12 x 16
MaxPooling3D	1 x 2 x 2, stride 1 x 2 x 2, same	1 x 6 x 6 x 16
BatchNormalization		
Dropout	50%	
Flatten		576
Dense		128
Dropout	50%	
Dense		1

Tab. 3: CNN architecture for REM.

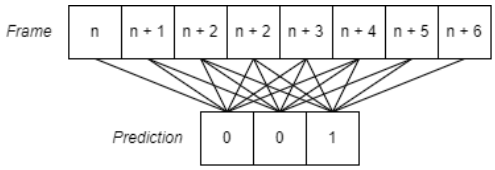


Fig. 8: Identifying REMs with sliding window.

To solve jitter in the stacks of 6 images, we could consider applying image registration as extra preprocessing step. However, we propose a simpler solution. We can exploit that the eye is approximately at the same position over 1 second: we average the eyes and nose landmarks of the frames within the window (note that this indirectly averages rotation and scale too), and use the three average landmarks to crop 6 eye images for both eyes.

During this process, if an individual eye landmark is too far away from the average eye landmark (more than half the estimated width of the eye), we reject the sample. It indicates movement of the body or head, and the eye would not be sufficiently visible throughout the 1 second.

3.4 Classifying Sleep States

3.4.1 *Eye Features*

Per timestamp, we predict whether the eye is opened and whether a REM occurs, indicating C, CR, O or OR. Per minute, we calculate the distribution of these eye states (for example $f_C = \frac{|C|}{|C|+|CR|+|O|+|OR|}$), resulting in four values: f_C, f_{CR}, f_O, f_{OR}.

3.4.2 *Predicting Sleep States*

With the sliding window, we consider 360 timestamps ($60 seconds \times 6 frames$) per minute. If the eye extraction pipeline fails for more than 50% of these frames, we skip the minute, as the distribution is likely unrepresentative of the entire minute. This also automatically prevents sleep annotations during caregiving.

Otherwise, we translate the four eye features to sleep states. We use RF as classifier, due to the simplicity of the task and the lack of training data. With RF, we automatically construct 300 small decision trees (typically depth of 2), and use majority voting to make predictions. While 300 trees is the default setting, we suspect that only a few trees are needed. RF finds thresholds per eye feature to discriminate the sleep states. A typical decision tree is visualized in Figure 9.

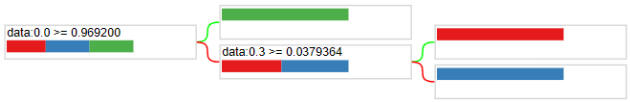


Fig. 9: Example of a decision tree generated by AVESSPIA (0.0 refers to f_C and 0.3 refers to f_{OR}).

4 Experimental Protocol

AVESSPIA uses four classifiers, which need to be trained and evaluated. This section, we first describe our datasets. Then, we report the used parameter values for training. Finally, we describe how we evaluate the models. We present the results in the next section.

4.1 Datasets and Annotation

We used a total of 24 videos to construct four datasets: one for each classifier. Samples for the three CNN models are frame-based. The sleep model works with minute-based samples.

4.1.1 Video Selection

We selected the 24 videos (each with a different infant) from the SLAPI dataset, a set of videos recorded at the NICU of the UMCU. Due to the limited data, we generally accepted videos as soon as one eye was clearly visible. We excluded a video if:

1. Camera setup used deviates strongly from the setup described in Section 3.1, and for example only records the side view of the face.
2. Both eyes are covered by masks throughout the entire video.
3. Video is recorded in darkness, and contains too much noise.

Due to the limited accessibility to video data and the limited scope of this research, each video used concerned a Caucasian infant.

We manually rotated each video with right angles, until the face of the infant is approximately upright (a single rotation per video). A face that is recorded upside down, for example, may perform worse in HigherHRNet [24]. This would need to be further investigated in the future.

4.1.2 Eye Samples

We trained three CNN models for AVESSPIA. Each model required its own labeled dataset, which had to be made from scratch each. We manually labeled the samples ourselves, after receiving extensive training at the UMCU.

For the REM model, we carefully picked 280 typical 1 second fragments across 22 videos, balancing over C, CR, O and OR. We generated samples using the extraction pipeline, but excluding the visibility model steps: while picking the fragments, we manually encoded which eye is visible best to make sure the correct eye is chosen. 7 (2.5%) fragments had to be dropped due to low landmark scores and 19 (6.8%) fragments had to be dropped due to excessive motion. We ended up with 254 samples for our REM dataset.

Each sample from the REM dataset takes 6 eye images, resulting in 1524 eye images in total. We used these eye images for our open-closed model. We arbitrarily picked 658 eye images over 21 videos. The labels had to be checked manually, as O samples of 1 second could still contain blinking eyes for example.

For the visibility model, we again take the 1524 eye images. In many of the 1 second fragments, one eye is partially or fully occluded; for these fragments, we labeled the best visible eye as positive, and the (partially) occluded eye as negative, yielding 244 samples over 12 videos. By including REM images, the shape of eyes during REM is also learned.

Classes in the datasets are balanced with a ratio of around 55% to 45%.

For every single video, we tried to include samples of each label. This is to ensure that the classifiers do not simply learn to identify videos. To illustrate, if a video only captures opened eyes, and has a unique color in the eye images, then that color could be used to classify the images as opened.

4.1.3 Sleep Samples

Out of 24 videos, 7 videos are annotated by experts from the UMCU with sleep states according to BeSSPI [8]. Per video, we arbitrarily picked up to 6 minutes per present class (W, AS, QS). When AVESSPIA left a minute blank due to intervention of nurses for example (criteria from Section 3.4.2), we tried other minutes. We ended up with 20 to 22 samples per class, and 64 samples in total. Along the way, 28 minutes were left blank.

4.2 Training Settings

We use the Adam optimizer and the binary cross entropy loss function to update the weights of the models. For the output layers, we use the sigmoid activation function. For the other layers, we use ReLU.

We initialize weights of layers with the Glorot uniform initializer; we initialize the biases with zeros. We initialize the weights and biases of the visibility model further by first pre-training it to separate opened from closed eyes. This is to ensure that the visibility model learns what a visible eye looks like.

For the visibility model and the open-closed model, we use a fixed learning rate of respectively 0.001 and 0.01. We use a lower learning rate for the visibility model, as it has a smaller batch size and updates more often. For the REM model, we start with a learning rate of 0.01. For each CNN model, we use early stopping on the validation loss with a patience of 10 epochs. When the REM model is finished, we halve its learning rate. We halve the learning rate four times. We halve learning rates to guide the exploration, as the REM model is expected to have a relatively complex search space. Typically, the models run 50 to 100 epochs in total.

For the visibility model, the open-closed model and the REM model, we found batch sizes of respectively 1, 8 and 8 to work best, after having tried batch sizes of 1, 8, 16 and 32.

4.3 Evaluation Protocol

We use k-fold cross-validation to measure the performance for each of the three CNN models: we split the videos into k groups, pick one group of videos as test set, use the remaining videos to train a model on, and then repeat this until each group is tested. So, samples of one infant are always in the same fold. 20% of the training samples are used for validation.

For the visibility model, we have samples from 12 videos, and each video is balanced in positive and negative labels. Thus, we can use 12 folds: a separate fold for every video (leave-one-subject-out cross-validation). For the other two models, labels are not balanced per video. Some videos, eyes of the infant do not open for example. For REM and open-closed, we put videos together - respectively in 7 and 5 folds. We ensured that each fold is balanced in labels and cardinalities of videos. A more detailed overview of fold selection can be found in Appendix A.

We first train the visibility model on the open-closed dataset, as part of initialization. To prevent bias, we exclude the test videos from the open-closed dataset during this step.

We have few samples for the sleep classifier. We employ leave-one-out cross-validation and use a separate fold for each sample. We have 64 samples, so we repeatedly test on 1 minute after training on 63 minutes. To obtain the four eye features of a minute, for input, we exclude the concerning test video while training AVESSPIA on the other 23 videos. So, the features are always extracted on unseen videos.

4.4 Performance Metrics

For the (binary) CNN models, we compute accuracy, AUC (area under the ROC-curve), precision, recall and F1-score. We report the average and standard deviation over all folds.

When computing precision, recall and F1-score, we consider visible eyes, opened eyes and presence of REMs as positives.

We include and investigate confusion matrices to support the metrics.

We have few samples for the sleep classifier. We consider reporting both accuracy and confusion matrix sufficient.

5 Results

In this section, results of the four classifiers from Section 3 are reported, according to the experimental protocol described in Section 4.

5.1 Visibility Model

The performance of the visibility model is presented in Table 4. The summed confusion matrix is shown in Table 5. The high standard deviation of accuracy suggests that the performance differs significantly per video. It makes mistakes for specific videos. This can be a serious problem, as false positives for fully occluded eyes would lead to arbitrary annotations in later stages of the pipeline.

	Accuracy	*AUC*	*Precision*	*Recall*	*F1*
Baseline (Average)	0.844	0.951	0.851	0.914	0.863
Standard Deviation	0.145	0.085	0.174	0.148	0.128

Tab. 4: Baseline performance on test set (visibility).

	Occluded (Predicted)	*Visible (Predicted)*
Occluded (Label)	86	28
Visible (Label)	10	120

Tab. 5: Baseline confusion matrix on test set (visibility).

However, the AUC value is almost perfect and shows that the model is well able to separate (true) positives from (true) negatives. Thus, we can improve the predictions by finding a better decision threshold. The optimal threshold t^* can be found by taking the threshold for which the location in the ROC-curve - the curve over false positive rate (FPR) and true positive rate (TPR) - is closest to $(0, 1)$.

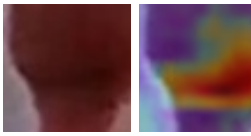
$$t^* = \arg\min_t \sqrt{FPR(t)^2 + (1 - TPR(t))^2} \quad (1)$$

We evaluated the model again with the equation above applied to the test folds. We show in Table 6 the impact of the threshold on accuracy, precision, recall and F1-score. While recall is similar, accuracy and precision are strongly improved. So, when it predicts an eye as visible, it is now more likely to be correct. This indicates the optimal threshold to be higher than 0.5.

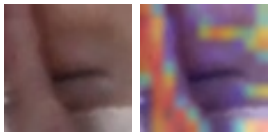
	Accuracy	*Precision*	*Recall*	*F1*
Baseline (Average)	0.948	0.962	0.923	0.939
Standard Deviation	0.093	0.070	0.157	0.119

Tab. 6: Baseline performance on test set, if we had optimal thresholds (visibility).

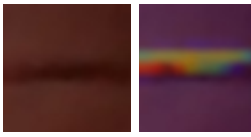
With Grad-CAM [69], we can highlight (in red) which features a model uses for making a prediction. The model particularly struggles with partially occluded eyes: the model uses the visible part of the eye to classify the eye as visible, as demonstrated in Figure 10a.



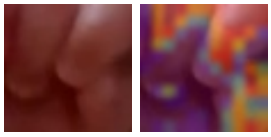
(a) Sticker occludes the bottom left. Predicted as visible.



(b) Arm occludes the left side. Predicted as occluded.



(c) No occlusions. Predicted as visible.



(d) Hand covers eye completely. Predicted as occluded.

Fig. 10: Four Grad-CAM examples (visibility).

5.2 Open-Closed Model

The measured performance and summed confusion matrix of the open-closed model are shown in respectively

Table 7 and Table 8. The results suggest strong generalization over unseen videos. We observed the mistakes, and found that it occasionally confuses slightly opened eyes, for both O and OR samples.

	Accuracy	*AUC*	*Precision*	*Recall*	*F1*
Baseline (Average)	0.963	0.982	1.0	0.917	0.956
Standard Deviation	0.021	0.023	0.0	0.047	0.026

Tab. 7: Baseline performance on test set (open-closed).

	Closed (Predicted)	*Opened (Predicted)*
Closed (Label)	368	0
Opened (Label)	24	266

Tab. 8: Baseline confusion matrix on test set (open-closed).

In Figure 11, in a second run, we visualize how features within the open-closed model scatter opened eyes (green/lime) and closed eyes (red/orange), for an arbitrary test fold. After passing a test sample through the trained model, we extract the output of the second to last dense layer, giving 64 features. So, for each sample, we obtain a point in a 64-dimensional feature space. With t-SNE [70], we reduce the features to 2 dimensions. We use different colors for C, CR, O and OR samples, to expose how the model treats samples during REM versus non-REM. With numbers, we expose patterns related to source videos.

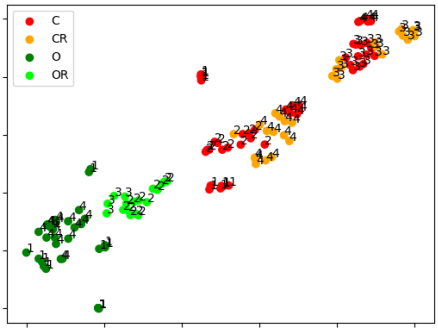


Fig. 11: Open-closed model visualized with t-SNE [70] for an arbitrary test fold. The numbers identify the source videos.

Overall, we see clear separation of opened and closed eyes. More locally, we find many clusters of samples from same videos. These samples are likely to be more similar, as these samples are from the same infant, with similar lighting conditions, and equal camera setup. O and OR samples are also clearly separated, with OR samples closer to closed eyes. OR samples often involve slightly opened eyes rather than fully opened eyes. C and CR samples are similar without temporal information.

5.3 REM Model

The results of our REM model are shown in Table 9 and Table 10. REMs uniquely identify AS. False positive REMs during other sleep states may pose a problem therefore. Unfortunately, as can be seen in the confusion matrix, we get many false positives. If we increased our threshold to get 100% precision (no false positives), we could have only identified 13.3% of all REMs, as measured by the *Recall*@100% metric. Considering the high standard deviation as well, it shows that false positives are unavoidable for certain videos with our model.

	Accuracy	*AUC*	*Precision*	*Recall*	*F1*	*Recall@100%*
Baseline (Average)	0.723	0.759	0.710	0.661	0.676	0.133
Standard Deviation	0.089	0.104	0.115	0.135	0.090	0.214

Tab. 9: Baseline performance on test set (REM).

	No REM (Predicted)	*REM (Predicted)*
No REM (Label)	111	34
REM (Label)	37	72

Tab. 10: Baseline confusion matrix on test set (REM).

We suspect that we have too little data to generalize the complex task of detecting subtle cues for REMs. Even more, the REM samples we have are of low quality: we included videos with poor image quality, inadequate camera positions, and occlusions around the eyes. Some samples, stickers close to the eyes are predicted as landmarks, leading to images where the eyes are not centered. Gathering more and better samples is a challenge, however: next to limited access to video data of preterm infants, it is also a time consuming process to find typical REMs in videos.

In a second run, we measure how often O and C samples are classified as non-REM, and how often CR and OR samples are predicted as REM. We report average accuracy over all test folds, see Table 11. We are curious if performance differs for opened and closed eyes. We find that OR samples are confused relatively often, with only 52.4% accuracy. OR samples are most underrepresented in the dataset (see Appendix A), so weights of the model are trained more frequently to predict on C, O and CR samples. The model seems inclined to predict samples with opened eyes as non-REM, explaining the relatively high accuracy for O samples.

	C	*O*	*CR*	*OR*
Accuracy (Average)	0.703	0.798	0.698	0.524

Tab. 11: Performance of assessing REMs on test set, isolated per eye state.

In Figure 12, in a third run, we visualize how features within the REM model scatter REMs (green/lime) and non-REMs (red/orange), given an arbitrary test fold. We use the same approach as for open-closed.

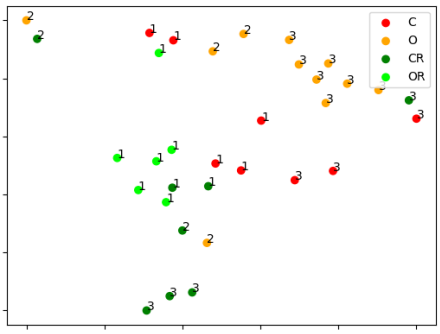


Fig. 12: REM model visualized with t-SNE [70] for an arbitrary test fold. The numbers identify the source videos.

We see that samples of the same color are close to each other, with some exceptions. The only task is to discriminate REMs and non-REMs, but the model shows to also consider the openness of eyes. We also find separations based on source video. As expected, samples with same color and same number are often closest, similar to the open-closed visualization (Figure 11). Overall, REMs and non-REMs are clearly separated.

The results in Table 9 should be considered as a conservative estimate. The samples are manually picked and mostly tackle challenging scenarios such as blinking eyes that look like REMs. Throughout a video, the model is able to predict most parts correctly.

In Appendix B, learning behaviour is presented for each CNN model.

We performed an ablation study to further investigate the REM model. Results are presented in Figure 12.

	Accuracy	*AUC*	*Precision*	*Recall*	*F1*
Baseline	0.723	0.759	0.710	0.661	0.676
3 *Frames*	0.745	0.780	0.701	0.735	0.711
12 *Frames*	0.671	0.741	0.623	0.646	0.615
3 *Seconds*	0.660	0.777	0.677	0.691	0.665
Larger Crops	0.650	0.725	0.676	0.587	0.595
Open-Closed Features	0.681	0.765	0.665	0.609	0.619
Two-Stream	0.669	0.727	0.663	0.474	0.548
Two Tasks	0.656	0.746	0.617	0.622	0.599
Reduced Training Data	0.625	0.684	0.576	0.642	0.576

Tab. 12: Ablation study results on test set, presenting averages per metric (REM).

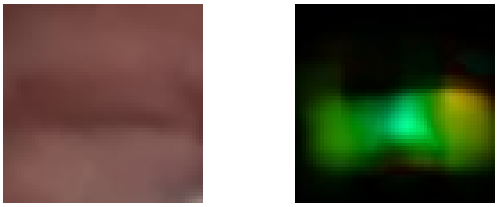
3 *Frames & 12 Frames.* While all other changes led to worse results, reducing the number of frames per sample from 6 to 3 led to a small improvement. Firstly, for a CNN it is generally easier to find relations in smaller input samples. Secondly, it may capture motions more generically than the subtle motions you capture with 12 frames for example, making it generalize better to unseen data. To support 3 frames, we changed the stride of the second convolutional layer from $2 \times 1 \times 1$ to $1 \times 1 \times 1$. For 12 frames, we allow temporal features in the first convolutional layer: we change its kernel size from $1 \times 5 \times 5$ to $5 \times 5 \times 5$, and change the kernel size of the first max pooling layer from $1 \times 2 \times 2$ to $2 \times 2 \times 2$.

3 *Seconds.* We show that using samples of 1 second (so 6 images per second) is superior to using samples of 3 seconds (so 2 images per second). With samples of 1 second, characteristic short movements can be captured and are therefore more concise than samples of 3 seconds. Conducting this experiment required a new dataset with samples of 3 seconds. Approximately the same timestamps were used as for the samples of 1 second. Sometimes, there was too much motion over 3 seconds, which we compensated by manually finding new samples in our videos.

Larger Crops. In the baseline, the height and width of the eye regions are 1.2 times the distance between the average nose landmark and the center of the two average eye landmarks. These eye regions strictly capture the eye, and do not include eyebrows for example. A factor of 2.0 instead of 1.2, taking larger crops, leads to worse performance. We suspect two reasons. Firstly, more occlusions of medical equipment are included in the samples. Secondly, the outer parts may be more misleading than being informative and discriminative.

Open-Closed Features. Initializing the model by making it predict open-closed labels first (similar to pre-training of the visibility model) is shown to have no added value.

Two-Stream. For two-stream, we added an input branch to the REM model that takes optical flow samples. We fuse the two branches at the flatten layer (see Table 3). We use the same architecture for this second branch until the flatten layer. For every eye image used in the original sample, we generate its optical flow image with respect to the eye image from the very next frame in the original video, assuming 30 frames per second. An example output image is given in Figure 13. From the results, it seems that two-stream performs worse. Optical flow information may complicate the classification task. To illustrate its challenges: movement of pupils can indicate both O and OR; and movement of a closed eye can indicate both C (when the head is moving) and CR.



(a) Original. (b) Optical flow.

Fig. 13: Example of optical flow during REM.

Two Tasks. We also tried adding a second output layer, and let our REM model discriminate opened and closed eyes as extra task. We stimulate the REM model to look at the shape of the eyes. However, O and OR samples can include closed eyes in some of their images, so this task may be confusing. The results indicate that the extension does not help the main task.

Reduced Training Data. Finally, we want to illustrate the impact of dataset size on the performance.

We test the performance with 3 folds instead of 6 folds for training (and still 1 fold for testing). We completely ignore our last 3 folds during this test. We find that training with more data strongly improves the performance. Expanding the dataset in the future has the potential to improve the model further. However, it could be possible that the first four folds were more difficult than the last three, by coincidence.

5.4 Sleep Assessment

AVESSPIA ultimately predicts sleep states per minute. We have ground truth sleep annotations for 64 minutes over 7 videos. The results are presented as a confusion matrix in Table 13. We find a promising accuracy of 92.2% over 3 classes. The eye features are well able to differentiate the sleep stages. AVESSPIA misclassifies 5 out of 64 minutes. We investigated these minutes.

	W (Predicted)	*AS (Predicted)*	*QS (Predicted)*
W (Label)	20	2	0
AS (Label)	1	20	1
QS (Label)	0	1	19

Tab. 13: Confusion matrix of AVESSPIA on test set.

One minute of AS is predicted as QS, because the extraction pipeline rejected eye images during REM. In the video, only one eye is visible (even the nose is occluded by a mask), with a sticker close to it. AVESSPIA could not find landmarks when the eyes moved during the REM.

Another minute, a small head movement during QS caused a false positive REM. We found that the frequent eye movements during W are also often misclassified as REMs. Movements in the eye region have higher variance than still samples, and we suspect that the REM model has not yet robustly generalized to this temporal variance.

However, it does not affect performance of sleep assessment much. First, during QS, there is generally little movement in the eye region, resulting in much fewer REM predictions than during AS. Secondly, eyes are mostly opened during W, so the sleep classifier can correct REM misclassifications by looking at open-closed predictions instead. In other words, the random forest classifier can branch off on f_C and f_O to separate AS and W, and branch off on f_{CR} and f_{OR} to separate AS and QS.

In between two subsequent minutes during W, eyes are closed extraordinarily long, including movements that resemble CR. Both minutes were incorrectly predicted as AS. Another minute, during AS, eyes were still and slightly opened for a while, causing the minute to be predicted as W. We also found these three minutes difficult to annotate.

There are no confusions between W and QS. While eyes may open during AS due to REM, eyes do not open during QS. So, W and QS can be separated based on whether eyes open during a minute. This is expected to work well given the performance of the open-closed model.

5.4.1 *Annotating Full-Length Videos*

We ran AVESSPIA on 4 arbitrary full-length videos, see Figure 14. The videos are completely unseen while training AVESSPIA. There is no ground truth of these videos. In BeSSPI [8], sleep stage is assessed over 60 seconds. We also include predictions over windows of 15 seconds and 30 seconds. A shorter window time is beneficial during live monitoring, as it reduces delay of present sleep stage.

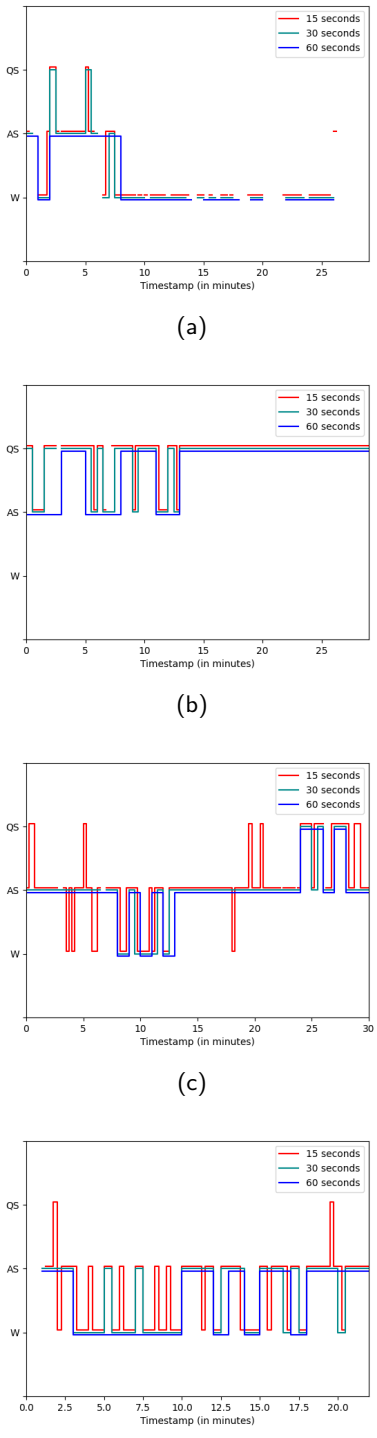


Fig. 14: Automatic sleep annotations by AVESSPIA on 4 arbitrary videos.

While we do not have formal ground truth per minute, the video of Figure 14a is used in UMCU to illustrate the transition from AS to W in preterm infants. According to the human expert, the infant wakes up in the second minute and falls asleep later that minute. They observed REMs while the infant is asleep. The infant wakes up again after eight minutes and then stays wake for the rest of the video. With a window of 60 seconds, AVESSPIA predicts the same. With 15 seconds, it records more precisely that the infant falls asleep late in the second minute. However, the shorter window is also more sensitive to mistakes, such as increased chances of missing REMs in a window of AS. It also predicts AS during W, when the infant closes the eyes while shortly crying. Some minutes are left blank, as the infant is too physically active here. The last three minutes, a human intervention takes place.

We also included a video with a transition to QS, see Figure 14b. The video starts with occasional REMs. In the second part, there is hardly any movement in the eye region. The graph reflects this observation.

Figure 14c is also in line with our observations. REMs with closed eyes occur in the first few minutes. Then, the eyes alternately open and close for a couple of minutes, seemingly without REM, and together with yawns and stretches. Then, the eyes stay closed, and with REMs again. In the end, the infant is quiet, but occasionally with REM.

The video of Figure 14d starts with two minutes of human intervention. The graph is chaotic, but reflects the nature of the video: there are alternating periods of open and closed eyes. However, we suspect that the eyes are opened due to REM and that the infant does not actually wake up each time, contrary to what the graph suggests. It is a difficult video and may require an improvement of the REM classifier or more training samples for RF.

6 Discussion

In this section, we discuss the results and answer the research questions.

6.1 Eye Extraction (RQ I)

We have found that HigherHRNet [24] is able to extract eyes and nose landmarks of preterm infants. We use these landmarks to crop out images of the eyes. With a CNN that filters out the (partially) occluded eyes with a 95.1% AUC, our extraction pipeline manages to consistently extract eye images from frames.

If the extraction pipeline returns eye images that do not resemble eyes at all, the errors would propagate and yield wrong eye features. Accuracy of sleep assessment would be low. However, we find a high accuracy in sleep assessment, suggesting that the extraction pipeline does well in practice.

Only 2.5% of our REM samples, across 22 videos (of lower quality), had to be dropped due to low landmark scores. 6.8% of the samples were dropped due to excessive movement of eye landmarks.

Some minutes, AVESSPIA is unable to extract sufficient eye images to assess sleep stage. We tried 92 arbitrary minutes to get sleep predictions for 64 minutes. Causes include stickers and masks around the nose and eyes, and arms moving in front of the eyes. Still, we were able to predict most minutes of full-length videos. As an advantage, minutes are automatically skipped during interventions.

We particularly observed successful extraction results for videos, if following conditions are met:

1. The image quality is acceptable (negligible noise).
2. The camera is positioned properly (see Figure 3).
3. At least one eye is clearly visible.
4. The infant is not extremely active physically.
5. There are no distracting stickers near the eye regions.
6. There is enough visual reference to identify the nose location.

6.2 Eye State Classification (RQ II & III)

We trained one CNN to discriminate opened eyes and closed eyes, and one 3D CNN to identify REMs.

We found that the open-closed model can almost perfectly generalize to unseen videos, with 96.3% accuracy. Slightly opened eyes are occasionally predicted as closed. We achieve similar performance to Cabon et al. [10] (see Section 2.3). However, their method is semi-automatic and requires user interactions, such as manually relocalizing the eye region after certain pose changes, and selecting thresholds of luminosity per video.

Our REM model performs worse, compared to our open-closed model, with 72.3% test accuracy. Its task is more complex due to lower inter-class variance and higher intra-class variance. We suggest to use 3 frames instead of 6 frames per REM sample, as it improved the accuracy to 74.5%.

We show with t-SNE [70] that features within the REM model are able to separate REMs and non-REMs of unseen videos. The found patterns are a promising sign that the CNN is able to find typical distinguishing features.

Predicting over whole videos, we observed false positive REMs especially during small head movements or eye movements. We found that the REM model mostly confuses OR samples (52.4% accuracy). These are underrepresented in the dataset.

The REM dataset is small (254 samples) and of low quality (noise in videos; suboptimal camera angles; stickers near the eyes; occluded noses; relatively few OR samples). We suspect that performance can be strongly improved, by improving the REM dataset. We also showed a positive correlation between amount of training data and performance.

6.3 Sleep Classification (RQ IV)

AVESSPIA predicts sleep states at the minute level with a test accuracy of 92.2%. It shows that our RF classifier is able to deal with the limitations of the REM model. Particularly, it uses open-closed annotations to discriminate AS and W, and the number of REMs to discriminate AS and QS. We find confusions when eyes are closed exceptionally long during W or opened exceptionally long during AS. Additionally, REMs can be missing during minutes of AS, and false positive REMs can cause confusions during QS.

Annotating 4 full-length videos, we find patterns in sleep stages, indicating periods of W, AS and QS. One video captures the transition from AS to W, and is perfectly annotated by AVESSPIA. For the other videos, we do not have ground truth at all, but the predictions reflect our own observations of the videos. We do particularly doubt the predictions in Figure 14d, but overall the graphs tell a lot about what is happening throughout the videos. The predictions over windows shorter than 60 seconds are less stable, with many unlikely transitions. On the other hand, it gives a more detailed estimate of what is happening. The predictions on shorter windows may improve by training separate random forest models for each. However, reliability remains limited: with short windows, the eye features are less representative of sleep stage.

As far as we know, other than Cabon et al. [10], automating sleep assessment based on videos of preterm infants has not been tried before. Cabon et al. use three features: whether an eye is opened, body movements and vocalizations. They are able to confidently predict active alert and quiet alert stages. But, they considered separating AS and QS still a difficult task, as they found AS and QS to have no statistical difference in terms of their features. Our REM feature is a new and promising addition, to help separate AS and QS.

We do not know the upper bound performance of assessing sleep states exclusively on eye cues. There may always exist situations where eye cues are not enough to determine sleep stage. For one, a minute of AS does not always involve REMs.

7 Limitations & Future Work

This section, we discuss limitations of our work, and provide considerations for the future.

Limited Modality. We cannot avoid occlusions during intervention of nurses, or when infants move their arms in front of the eyes. Monitoring the eye region during gross body and head movements is currently also not possible. If this happens exactly during REM, we may classify a minute as QS instead of AS. Besides, not every minute of AS includes REM. Additional modalities, such as EEG [71], vital parameters [71] and body movements [10] [72], could be used to cover these situations. Generally, it will be interesting to combine eye features with other features.

Finding Landmarks (and Fast). Another limitation of AVESSPIA is finding landmarks in difficult videos. In the future, it may be considered to replace the eye extraction pipeline by an object detection algorithm such as YOLO [33]. It has the potential to find eyes in even stronger occluded frames.

HigherHRNet is also computationally expensive (processing around 2.35 frames per second on a Quadro RTX 6000, for our videos). Finding landmarks in real-time, for live monitoring, would require modifications. A simple solution would be to increase the step size of the sliding window. We could also consider to only compute landmarks every few frames, and interpolate for the frames in between. YOLO, on the other hand, is known for being fast. With our current approach, we need to process 6 frames per second; with YOLO this would be no issue [33]. The three CNNs we trained for AVESSPIA are all relatively small, and should not be an obstacle to running the system in real-time.

However, we also expect challenges and downsides for using object detection. First, there are no public datasets available of (preterm) infants labeled with (rotated) bounding boxes around the eyes, as far as we know, and eyes of adults are different. Secondly, input data can be challenging with low contrast around the eyes (see Figure 1 and Figure 7a), due to camera quality, the distance of the camera to the infant, and the size of preterm infants. Thirdly, patterns in the images may look like eyes to object detection, such as patterns on blankets, and pose information would not be considered anymore to prevent this.

We would also need to extend object detection - which predicts rectangles - with the task of estimating rotations. However, this double classification task could be easily achieved with automatic data augmentation: rotate the images arbitrarily with θ degrees, and store the rotated images together with θ as extra label.

REM Model. The next limitation of AVESSPIA is the performance of the REM model. False positive REMs with closed eyes can make AVESSPIA predict AS instead of QS. In the future, we need an efficient method to expand and improve the REM dataset.

As a potential solution to expand and improve the REM dataset, we propose active learning [73]. Essentially, given a new video, the algorithm would pick random samples, and if the old REM model is unsure about the annotation of a sample, we are asked to label the sample manually. As REMs are sparse in our videos, we propose to pick samples from the underrepresented class in the dataset [74]. The active learning system could be made user-friendly, such that the REM model can continuously be improved in the future, without involving technical knowledge.

Random Forest. To illustrate a potential flaw of random forest in our application, f_C could be used as feature to identify W, as a low percentage is expected. However, if f_C is low because of f_{CR} being extraordinarily high, we expect AS instead of W. For applications of AVESSPIA, we suggest to limit the number of decision trees in random forest, so each decision tree can be inspected and verified for such flaws.

Sleep Assessment. A limitation of our results is that we only have few sleep annotations at the minute level. We would like to test sleep assessment with AVESSPIA on many more videos, to get a more complete image of its performance in practice.

Population. In the future, we also need to involve a more diverse population of preterm infants. While we do apply color normalization for our visibility and REM model, we cannot tell yet if this is a sufficient generalization, and whether other complications may arise. For our open-closed model, we still need to look into implementing color normalization for three channels, and ensure it does not suffer from the issue illustrated in Figure 7.

Upper Bound. To get a better understanding of the upper bound performance of assessing sleep states with eye cues, we could test performance with eye cues annotated by human experts. With BeSSPI [8], experts already annotate eye states along sleep states, at the minute level. However, they also look at other modalities, so they may miss important eye cues. The upper bound would be weak and unrepresentative. We suggest to construct a dataset where human experts exclusively look at eye cues.

Implementation. Finally, we can make many implementation specific improvements, and further apply domain knowledge. We could for example use that OR generally occurs directly after eyes are closed, and that REMs happen in bursts.

8 Conclusion

We have introduced AVESSPIA, a pipeline to automatically assess sleep states in videos of preterm infants. We focus on using eye cues, mainly for two reasons: it is one of the only regions that is not regularly occluded, and it is a highly informative source in sleep assessment of preterm infants. In our pipeline, we first extract eye regions. Then, we predict eye states with CNNs. Finally, we translate eye states to sleep states with random forest.

Some minutes, AVESSPIA is unable to extract sufficient eye images to assess sleep stage. We leave these minutes blank. Annotating full-length videos, we get predictions most minutes when an eye is visible. Out of 280 manually picked REM samples, AVESSPIA successfully found the eye regions for 97.5%. We specified when videos are difficult, and which videos are likely to succeed.

AVESSPIA discriminates minutes of wake, active sleep and quiet sleep with 92.2% test accuracy, exclusively using eye cues. It is able to reveal patterns and transitions of sleep stages in the automatic annotations of full-length videos.

We show that eye cues are promising features for automatic sleep assessment of preterm infants. It would be interesting to combine eye cues with other modalities. Next to assessing sleep states, eye cues may also help in detecting discomfort, among others.

References

[1] Hannah Blencowe, Simon Cousens, Mikkel Z Oestergaard, Doris Chou, Ann-Beth Moller, Rajesh Narwal, Alma Adler, Claudia Vera Garcia, Sarah Rohde, Lale Say, and Joy E Lawn. "National, regional, and worldwide estimates of preterm birth rates in the year 2010 with time trends since 1990 for selected countries: a systematic analysis and implications". In: *The Lancet* 379.9832 (2012), pp. 2162–2172. ISSN: 0140-6736. DOI: https://doi.org/10.1016/S0140-6736(12)60820-4. URL: https://www.sciencedirect.com/science/article/pii/S0140673612608204.

[2] Joann R. Petrini, Todd Dias, Marie C. McCormick, Maria L. Massolo, Nancy S. Green, and Gabriel J. Escobar. "Increased Risk of Adverse Neurological Development for Late Preterm Infants". In: *The Journal of Pediatrics* 154.2 (2009), 169–176.e3. ISSN: 0022-3476. DOI: https://doi.org/10.1016/j.jpeds.2008.08.020. URL: https://www.sciencedirect.com/science/article/pii/S0022347608006999.

[3] Emilie Bourel-Ponchel, Danièle Hasaerts, Marie-Josèphe Challamel, and Marie-Dominique Lamblin. "Behavioral-state development and sleep-state differentiation during early ontogenesis". In: *Neurophysiologie Clinique* 51.1 (2021), pp. 89–98. ISSN: 0987-7053. DOI: https://doi.org/10.1016/j.neucli.2020.10.003. URL: https://www.sciencedirect.com/science/article/pii/S0987705320301088.

[4] Omri Weisman, Reuma Magori-Cohen, Yoram Louzoun, Arthur I. Eidelman, and Ruth Feldman. "Sleep-Wake Transitions in Premature Neonates Predict Early Development". In: *Pediatrics* 128.4 (Oct. 2011), pp. 706–714. ISSN: 0031-4005. DOI: 10.1542/peds.2011-0047. eprint: https://publications.aap.org/pediatrics/article-pdf/128/4/706/1056012/zpe01011000706.pdf. URL: https://doi.org/10.1542/peds.2011-0047.

[5] Jan Werth, Louis Atallah, Peter Andriessen, Xi Long, Elly Zwartkruis-Pelgrim, and Ronald M. Aarts. "Unobtrusive sleep state measurements in preterm infants – A review". In: *Sleep Medicine Reviews* 32 (2017), pp. 109–122. ISSN: 1087-0792. DOI: https://doi.org/10.1016/j.smrv.2016.03.005. URL: https://www.sciencedirect.com/science/article/pii/S1087079216300065.

[6] Susan M. Ludington-Hoe, Mark W. Johnson, Kathy Morgan, Tina Lewis, Judy Gutman, P. David Wilson, and Mark S. Scher. "Neurophysiologic Assessment of Neonatal Sleep Organization: Preliminary Results of a Randomized, Controlled Trial of Skin Contact With Preterm Infants". In: *Pediatrics* 117.5 (May 2006), e909–e923. ISSN: 0031-4005. DOI: 10.1542/peds.2004-1422. eprint: https://publications.aap.org/pediatrics/article-pdf/117/5/e909/1070358/zpe0050600e909.pdf. URL: https://doi.org/10.1542/peds.2004-1422.

[7] Jinhee Park. "Sleep Promotion for Preterm Infants in the NICU". In: *Nursing for Women's Health* 24.1 (2020), pp. 24–35. ISSN: 1751-4851. DOI: https://doi.org/10.1016/j.nwh.2019.11.004. URL: https://www.sciencedirect.com/science/article/pii/S1751485119302302.

[8] E.R. de Groot, A. Bik, C. Sam, X. Wang, R.A. Shellhaas, T. Austin, M.L. Tataranno, M.J.N.L. Benders, A. van den Hoogen, and J. Dudink. "Creating an optimal observational sleep stage classification system for very and extremely preterm infants". In: *Sleep Medicine* 90 (2022), pp. 167–175. ISSN: 1389-9457. DOI: https://doi.org/10.1016/j.sleep.2022.01.020. URL: https://www.sciencedirect.com/science/article/pii/S1389945722000272.

[9] Sara Moccia, Lucia Migliorelli, Virgilio Carnielli, and Emanuele Frontoni. "Preterm Infants' Pose Estimation With Spatio-Temporal Features". In: *IEEE Transactions on Biomedical Engineering* 67.8 (Aug. 2020), pp. 2370–2380. DOI: 10.1109/tbme.2019.2961448. URL: https://doi.org/10.1109%5C%2Ftbme.2019.2961448.

[10] S. Cabon, F. Porée, A. Simon, B. Met-Montot, P. Pladys, O. Rosec, N. Nardi, and G. Carrault. "Audio- and video-based estimation of the sleep stages of newborns in Neonatal Intensive Care Unit". In: *Biomedical Signal Processing and Control* 52 (2019), pp. 362–370. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2019.04.011. URL: https://www.sciencedirect.com/science/article/pii/S1746809419301089.

[11] Zakia Hammal, Wen-Sheng Chu, Jeffrey F. Cohn, Carrie Heike, and Matthew L. Speltz. "Automatic action unit detection in infants using convolutional neural network". In: *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. 2017, pp. 216–221. DOI: 10.1109/ACII.2017.8273603.

[12] Marco Leo, Giuseppe Massimo Bernava, Pierluigi Carcagnì, and Cosimo Distante. "Video-Based Automatic Baby Motion Analysis for Early Neurological Disorder Diagnosis: State of the Art and Future Directions". In: *Sensors* 22.3 (2022). ISSN: 1424-8220. DOI: 10.3390/s22030866. URL: https://www.mdpi.com/1424-8220/22/3/866.

[13] Md Sirajus Salekin, Ghada Zamzmi, Dmitry Goldgof, Rangachar Kasturi, Thao Ho, and Yu Sun. "Multi-Channel Neural Network for Assessing Neonatal Pain from Videos". In: Oct. 2019, pp. 1551–1556. DOI: 10.1109/SMC.2019.8914537.

[14] Ádám Nagy, Péter Földesy, Imre Jánoki, Dániel Terbe, Máté Siket, Miklós Szabó, Judit Varga, and Ákos Zarándy. "Continuous Camera-Based Premature-Infant Monitoring Algorithms for NICU". In: *Applied Sciences* 11.16 (2021). ISSN: 2076-3417. DOI: 10.3390/app11167215. URL: https://www.mdpi.com/2076-3417/11/16/7215.

[15] Nikolas Hesse, Sergi Pujades, Michael J. Black, Michael Arens, Ulrich G. Hofmann, and A. Sebastian Schroeder. *Learning and Tracking the 3D Body Shape of Freely Moving Infants from RGB-D sequences*. 2018. DOI: 10.48550/ARXIV.1810.07538. URL: https://arxiv.org/abs/1810.07538.

[16] Anne Bik, Chanel Sam, Eline R. de Groot, Simone S.M. Visser, Xiaowan Wang, Maria Luisa Tataranno, Manon J.N.L. Benders, Agnes van den Hoogen, and Jeroen Dudink. "A scoping review of behavioral sleep stage classification methods for preterm infants". In: *Sleep Medicine* 90 (2022), pp. 74–82. ISSN: 1389-9457. DOI: https://doi.org/10.1016/j.sleep.2022.01.006. URL: https://www.sciencedirect.com/science/article/pii/S1389945722000077.

[17] Nicholas Hoque. *Neonatology At A Glance*. Aug. 2015. ISBN: 978-1-118-76743-6.

[18] Aomar Osmani, Massinissa Hamidi, and Abdelghani Chibani. "Machine Learning Approach for Infant Cry Interpretation". In: *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. 2017, pp. 182–186. DOI: 10.1109/ICTAI.2017.00038.

[19] S.E. Barajas-Montiel and C.A. Reyes-Garcia. "Identifying Pain and Hunger in Infant Cry with Classifiers Ensembles". In: *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*. Vol. 2. 2005, pp. 770–775. DOI: 10.1109/CIMCA.2005.1631561.

[20] Lilia Curzi-Dascalova, Patricio Peirano, and Françoise Morel-Kahn. "Development of sleep states in normal premature and full-term newborns". In: *Developmental Psychobiology* 21.5 (1988), pp. 431–444. DOI: https://doi.org/10.1002/dev.420210503. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/dev.420210503. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/dev.420210503.

[21] Lilia Curzi-Dascalova, JM Figueroa, M Eiselt, E Christova, A Virassamy, AM d'Allest, H Guimaraes, Christine Gaultier, and M Dehan. "Sleep state organization in premature infants of less than 35 weeks' gestational age". In: *Pediatric research* 34.5 (1993), pp. 624–628.

[22] MJ Hayes, MR Akilesh, M Fukumizu, AA Gilles, BA Sallinen, M Troese, and JA Paul. "Apneic preterms and methylxanthines: arousal deficits, sleep fragmentation and suppressed spontaneous movements". In: *Journal of Perinatology* 27.12 (2007), pp. 782–789.

[23] Greta Sokoloff, James C. Dooley, Ryan M. Glanz, Rebecca Y. Wen, Meredith M. Hickerson, Laura G. Evans, Haley M. Laughlin, Keith S. Apfelbaum, and Mark S. Blumberg. "Twitches emerge postnatally during quiet sleep in human infants and are synchronized with sleep spindles". In: *Current Biology* 31.15 (2021), 3426–3432.e4. ISSN: 0960-9822. DOI: https://doi.org/10.1016/j.cub.2021.05.038. URL: https://www.sciencedirect.com/science/article/pii/S0960982221007363.

[24] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. "HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[25] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. *Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*. 2016. DOI: 10.48550/ARXIV.1611.08050. URL: https://arxiv.org/abs/1611.08050.

[26] Tareq Khan. "An Intelligent Baby Monitor with Automatic Sleeping Posture Detection and Notification". In: *AI* 2 (June 2021), pp. 290–306. DOI: 10.3390/ai2020018.

[27] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2014.

[28] Ross Girshick. "Fast R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

[30] Cheng Li, Arash Pourtaherian, Lonneke Onzenoort, W. Ten, and Peter With. "Infant Facial Expression Analysis: Towards A Real-time Video Monitoring System Using R-CNN and HMM". In: *IEEE Journal of Biomedical and Health Informatics* PP (Nov. 2020), pp. 1–1. DOI: 10.1109/JBHI.2020.3037031.

[31] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. *Mask R-CNN*. 2017. DOI: 10.48550/ARXIV.1703.06870. URL: https://arxiv.org/abs/1703.06870.

[32] C. Li, A. Pourtaherian, W. E. Tjon a Ten, and P. H. N. de With. "Infant Monitoring System for Real-Time and Remote Discomfort Detection". In: *2020 IEEE International Conference on Consumer Electronics (ICCE)*. 2020, pp. 1–2. DOI: 10.1109/ICCE46568.2020.9043065.

[33] Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement*. 2018. DOI: 10.48550/ARXIV.1804.02767. URL: https://arxiv.org/abs/1804.02767.

[34] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. *WIDER FACE: A Face Detection Benchmark*. 2015. DOI: 10.48550/ARXIV.1511.06523. URL: https://arxiv.org/abs/1511.06523.

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. *Microsoft COCO: Common Objects in Context*. 2014. DOI: 10.48550/ARXIV.1405.0312. URL: https://arxiv.org/abs/1405.0312.

[36] Evangelos Sariyanidi, Hatice Gunes, and Andrea Cavallaro. "Automatic Analysis of Facial Affect: A Survey of Registration, Representation, and Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.6 (2015), pp. 1113–1133. DOI: 10.1109/TPAMI.2014.2366127.

[37] Luigi Celona and Luca Manoni. "Neonatal Facial Pain Assessment Combining Hand-Crafted and Deep Features". In: Dec. 2017, pp. 197–204. ISBN: 978-3-319-70741-9. DOI: 10.1007/978-3-319-70742-6_19.

[38] Yue Sun, Caifeng Shan, Tao Tan, Xi Long, Arash Pourtaherian, Sveta Zinger, and Peter With. "Video-based discomfort detection for infants". In: *Machine Vision and Applications* 30 (July 2019), pp. 933–944. DOI: 10.1007/s00138-018-0968-1.

[39] Barbara Zitová and Jan Flusser. "Image Registration Methods: A Survey". In: *Image and Vision Computing* 21 (Oct. 2003), pp. 977–1000. DOI: 10.1016/S0262-8856(03)00137-9.

[40] Mahesh and M .V. Subramanyam. "Automatic feature based image registration using SIFT algorithm". In: *2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12)*. 2012, pp. 1–5. DOI: 10.1109/ICCCNT.2012.6396024.

[41] John C Gower. "Generalized procrustes analysis". In: *Psychometrika* 40 (1975), pp. 33–51.

[42] Alma Eguizabal, Peter J Schreier, and Jürgen Schmidt. "Procrustes registration of two-dimensional statistical shape models without correspondences". In: *arXiv preprint arXiv:1911.11431* (2019).

[43] Paul J Besl and Neil D McKay. "Method for registration of 3-D shapes". In: *Sensor fusion IV: control paradigms and data structures*. Vol. 1611. Spie. 1992, pp. 586–606.

[44] Simon Baker and Iain Matthews. "Lucas-kanade 20 years on: A unifying framework". In: *International journal of computer vision* 56 (2004), pp. 221–255.

[45] Georgios Tzimiropoulos, Vasileios Argyriou, Stefanos Zafeiriou, and Tania Stathaki. "Robust FFT-based scale-invariant image registration with image gradients". In: *IEEE transactions on pattern analysis and machine intelligence* 32.10 (2010), pp. 1899–1906.

[46] B Srinivasa Reddy and Biswanath N Chatterji. "An FFT-based technique for translation, rotation, and scale-invariant image registration". In: *IEEE transactions on image processing* 5.8 (1996), pp. 1266–1271.

[47] Cheng Li, Arash Pourtaherian, Lonneke Onzenoort, W. Ten, and Peter With. "Infant Facial Expression Analysis: Towards A Real-time Video Monitoring System Using R-CNN and HMM". In: *IEEE Journal of Biomedical and Health Informatics* PP (Nov. 2020), pp. 1–1. DOI: 10.1109/JBHI.2020.3037031.

[48] Wilhelm Burger and Mark J. Burge. *Principles of Digital Image Processing: Fundamental Techniques*. 1st ed. Springer Publishing Company, Incorporated, 2009. ISBN: 1848001908.

[49] Dinu Coltuc, Philippe Bolon, and J-M Chassery. "Exact histogram specification". In: *IEEE Transactions on Image processing* 15.5 (2006), pp. 1143–1152.

[50] Ján Morovic and Pei-Li Sun. "Accurate 3D image colour histogram transformation". In: *Pattern Recognition Letters* 24.11 (2003). Colour Image Processing and Analysis. First European Conference on Colour in Graphics, Imaging, and Vision (CGIV 2002), pp. 1725–1735. ISSN: 0167-8655. DOI: https://doi.org/10.1016/S0167-8655(02)00328-8. URL: https://www.sciencedirect.com/science/article/pii/S0167865502003288.

[51] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. "Color Transfer between Images". In: *IEEE Computer Graphics and Applications* 21 (Oct. 2001), pp. 34–41. DOI: 10.1109/38.946629.

[52] Daniel L. Ruderman, Thomas W. Cronin, and Chuan-Chin Chiao. "Statistics of cone responses to natural images: implications for visual coding". In: *J. Opt. Soc. Am. A* 15.8 (Aug. 1998), pp. 2036–2045. DOI: 10.1364/JOSAA.15.002036. URL: https://opg.optica.org/josaa/abstract.cfm?URI=josaa-15-8-2036.

[53] Yann LeCun, Yoshua Bengio, et al. "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[54] Ghada Zamzmi, Rangachar Kasturi, Dmitry Goldgof, Ruicong Zhi, Terri Ashmeade, and Yu Sun. "A Review of Automated Pain Assessment in Infants: Features, Classification Tasks, and Databases". In: *IEEE Reviews in Biomedical Engineering* PP (Nov. 2017), pp. 1–1. DOI: 10.1109/RBME.2017.2777907.

[55] Raphaël Weber, Sandie Cabon, Antoine Simon, Fabienne Poree, and Guy Carrault. "Preterm Newborn Presence Detection in Incubator and Open Bed Using Deep Transfer Learning". In: *IEEE journal of biomedical and health informatics* PP (Mar. 2021). DOI: 10.1109/JBHI.2021.3062617.

[56] Joe Yue-Hei Ng, Matthew J. Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. "Beyond Short Snippets: Deep Networks for Video Classification". In: *CoRR* abs/1503.08909 (2015). arXiv: 1503.08909. URL: http://arxiv.org/abs/1503.08909.

[57] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[58] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. *On the difficulty of training Recurrent Neural Networks*. 2012. DOI: 10.48550/ARXIV.1211.5063. URL: https://arxiv.org/abs/1211.5063.

[59] Allen Chang, Lauren Klein, Marcelo R. Rosales, Weiyang Deng, Beth A. Smith, and Maja J. Matarić. *Evaluating Temporal Patterns in Applied Infant Affect Recognition*. 2022. DOI: 10.48550/ARXIV.2209.03496. URL: https://arxiv.org/abs/2209.03496.

[60] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. *Recurrent Neural Network Regularization*. 2014. DOI: 10.48550/ARXIV.1409.2329. URL: https://arxiv.org/abs/1409.2329.

[61] Dae Kim, Wissam Baddar, and Yong Ro. "Micro-Expression Recognition with Expression-State Constrained Spatio-Temporal Feature Representations". In: Oct. 2016, pp. 382–386. DOI: 10.1145/2964284.2967247.

[62] Yuchi Liu, Heming Du, Liang Zheng, and Tom Gedeon. "A Neural Micro-Expression Recognizer". In: *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. 2019, pp. 1–4. DOI: 10.1109/FG.2019.8756583.

[63] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William Freeman. "Eulerian Video Magnification for Revealing Subtle Changes in the World". In: *ACM Transactions on Graphics - TOG* 31 (July 2012). DOI: 10.1145/2185520.2185561.

[64] Zhaoqiang Xia, Xiaopeng Hong, Xingyu Gao, Xiaoyi Feng, and Guoying Zhao. *Spatiotemporal Recurrent Convolutional Networks for Recognizing Spontaneous Micro-expressions*. 2019. DOI: 10.48550/ARXIV.1901.04656. URL: https://arxiv.org/abs/1901.04656.

[65] Itir Onal Ertugrul, Yeojin Ahn, Maneesh Bilalpur, Daniel Messinger, Matthew Speltz, and Jeffrey Cohn. "Infant AFAR: Automated facial action recognition in infants". In: *Behavior Research Methods* (May 2022). DOI: 10.3758/s13428-022-01863-y.

[66] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[67] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. *VGGFace2: A dataset for recognising faces across pose and age*. 2018. arXiv: 1710.08092 [cs.CV].

[68] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee. 2005, pp. 886–893.

[69] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *International Journal of Computer Vision* 128.2 (Oct. 2019), pp. 336–359. DOI: 10.1007/s11263-019-01228-7. URL: https://doi.org/10.1007%5C%2Fs11263-019-01228-7.

[70] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of Machine Learning Research* 9 (Nov. 2008), pp. 2579–2605.

[71] Thom Sentner, Xiaowan Wang, Eline Groot, Lieke Schaijk, Maria Tataranno, Daniel Vijlbrief, Manon Benders, Richard Bartels, and Jeroen Dudink. "The Sleep Well Baby project: an automated real-time sleep-wake state prediction algorithm in preterm infants". In: *Sleep* 45 (June 2022). DOI: 10.1093/sleep/zsac143.

[72] Xi Long, Renée Otte, Eric van der Sanden, Jan Werth, and Tao Tan. "Video-Based Actigraphy for Monitoring Wake and Sleep in Healthy Infants: A Laboratory Study". In: *Sensors* 19.5 (2019). ISSN: 1424-8220. DOI: 10.3390/s19051075. URL: https://www.mdpi.com/1424-8220/19/5/1075.

[73] Burr Settles. "Active learning literature survey". In: (2009).

[74] Umang Aggarwal, Adrian Popescu, and Céline Hudelot. "Active learning for imbalanced datasets". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 1428–1437.

Appendix A: Dataset Distributions

C	O	CR	OR
2	1	1	2
0	0	3	0
6	16	2	8
2	2	4	2
5	0	7	0
2	9	1	1
3	0	5	2
3	0	5	3
7	10	1	0
3	7	4	0
5	0	2	6
0	4	2	0
4	1	0	3
2	5	0	0
5	0	9	3
2	0	1	1
4	3	4	1
5	4	4	3
8	0	5	0
5	3	5	0
4	0	2	2
3	0	5	0

C	O
7	10
2	0
2	1
3	0
5	0
2	4
4	0
3	6
6	10
2	0
5	3
3	0
8	0
2	7
1	1
2	2
3	0
5	4
4	3
2	0
0	4

O	V
3	3
2	2
2	2
1	2
3	3
1	2
2	3
1	2
3	2
1	2
1	1
2	2

Tab. 14: Distributions of respectively REM folds, open-closed folds, and visibility folds, with rows representing videos. Negative labels are indicated in red; positive labels in green. The numbers indicate how many samples of 6 images were used, so the number of actual samples used for open-closed and visibility are higher than shown here.

W	AS	QS
0	5	4
0	2	5
6	4	0
6	0	0
6	6	0
4	3	5
0	2	6

Tab. 15: Distribution of sleep state minutes, with rows representing videos.

Appendix B: Learning Behaviour of Models

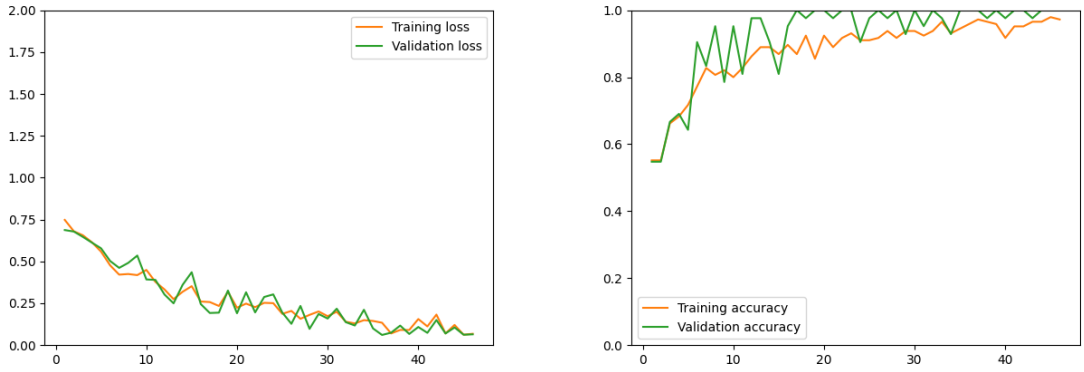


Fig. 15: Typical example of training and validation performance per epoch when training the visibility model. The validation set outperforms the training set due to dropout and batch normalization.

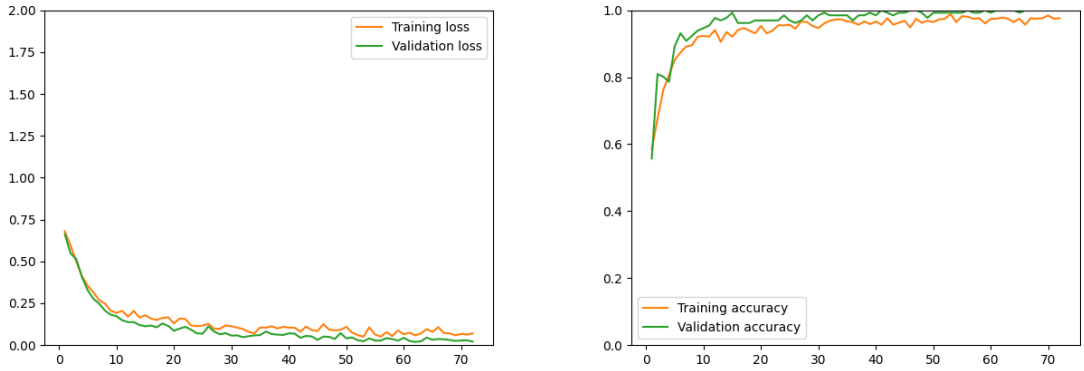


Fig. 16: Typical example of training and validation performance per epoch when training the open-closed model. The validation set outperforms the training set due to dropout and batch normalization.

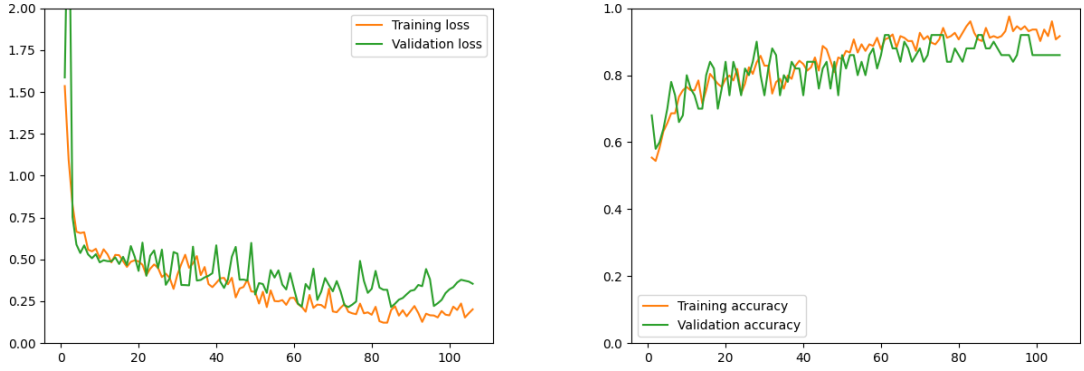


Fig. 17: Typical example of training and validation performance per epoch when training the REM model. The model shows to have some difficulties generalizing to the validation set.