Automated Gaze Coding for Infant Videos

Peng Cao
Massachusetts Institute of Technology
Cambridge, MA, USA
pengcao@mit.edu

Xincheng Tan
Harvard University
Cambridge, MA, USA
xinchengtan@g.harvard.edu

Abstract

Gaze-based studies are widely used in developmental science research for preverbal infants, but data collection has always been a big challenge for those studies. Although online platforms enable families to participate in studies via webcam, it is time- and energy- consuming to manually annotate gaze directions on the collected videos. Existing gaze coding algorithms on videos either suffer from low video quality or still require considerable manual effort. In this project, we propose a fully automated gaze coding framework for infant videos. In our framework, all possible faces in a video frame are first extracted by a face extractor, and then the infant face is selected by a infant selector; finally, the gaze direction is classified by a gaze estimator based on both the selected face and features of its bounding box. We evaluate the framework on a large-scale infant video dataset. The experimental results show the superiority of our framework in gaze estimation accuracy compared to the baseline framework.

Figure 1: An example of the preferential looking paradigm. A hand-clapping event and a drum-playing event are shown on the screen in front of an infant side-by-side while the sound of one of them is played at the same time. If the infant looks longer at the event projected in sound, it means it perceives the auditory-visual relationship.

1. Introduction

Gaze direction has become a cornerstone measure for developmental science research on preverbal infants over the past decades[3, 11, 16, 4]. For example, as shown in Figure 1, to assess the perception of auditory-visual relationships for preverbal infants, one can present two events (e.g. hand clapping and drum playing) side by side on a screen in front of an infant while playing the sound for one of them; if the infants' gaze focuses longer at the event projected in sound, it means they have perceived the relationship between the sound and the object. This paradigm is called preferential looking in developmental science[5].

However, lots of important questions about children's early abilities remain unanswered not because of their scientific difficulties but because of a practical challenge: it is notoriously troublesome to recruit an adequate number and variety of infants to conduct gaze-based studies. Although there are some commercial eye-tracking systems(e.g. Tobii Eye Tracker), researchers are constrained by their high prices and the corresponding special hardwares make it impossible for the studies to be conducted outside of the laboratory. A new online platform, Lookit[14], has been developed to allow families to participate in studies online via webcam. Nevertheless, it takes a human annotator twice to five times as long as the video itself to do gaze coding on the collected videos.

Existing algorithms for automated gaze direction estimation on videos are still premature for this task. Some of them rely on extracting eye features from the video[18, 17], which suffer from low video qualities and thus are not suitable for coding the videos collected online. Others apply deep learning models to predict the eye directions[1, 20], which require extracting facial landmarks(e.g. eyes, nose, mouth) on the video frame, but they are not guaranteed to work well for infant users since infants' actions like thumb-sucking make it much harder to detect those landmarks.

In fact, these methods need high-quality videos or fine-

grained features primarily because they attempt to estimate gaze directions at a very high resolution as a 3D vector. However, for studies using preferential looking paradigm, the question researchers really care about is whether the infant is looking at the left or the right of the screen, and a high-resolution gaze direction estimate is not necessary. A recent work, iCatcher[6], proposes a deep-learning-based gaze coding framework for infant videos, which considers gaze coding as a classification problem with three discrete categories of gaze directions: away, left and right. Though somewhat coarse, such resolution is sufficient for the preferential looking paradigm. iCatcher[6] is composed of an off-the-shelf face extractor and a deep-learning-based gaze estimator, with a manual or arbitrary infant face selection step. Such selection mechanism either incurs considerable manual effort or brings unwanted input noise to the gaze estimator. Moreoever, it only uses the face patch pixels as the input to the gaze estimator and discards the potentially useful positional information of the infant's face relative to the video frame.

In this project, we propose a fully automated gaze coding framework for infant videos based on iCathcer[6]. To address the problem of the face selection, we add an automatic infant selector to select the correct face from the extracted patches. And to avoid losing positional information, we augment the gaze estimator by including the spatial features of the selected bounding box in addition to the raw pixels as input. We implement and evaluate the whole framework on the Lookit dataset, which contains 96 10-minute infant videos collected by the Lookit[14] platform. Our framework outperforms iCatcher[6] in gaze direction estimation accuracy by around 2% and experimental results show that both of our two modifications are effective.

2. Related Work

Existing gaze estimation methods can be mainly classified into three categories: feature-based, model-based and appearance-based [7]. Feature-based methods[22, 23] leverage local characteristics around human eyes, such as pupil, limbus and corneal reflections, to identify gaze directions. These methods are commonly used in commercial eye trackers, which rely on expensive hardware devices and may impose specific procedural constraints. Model-based approaches usually fit a geometric 3D eyeball model based on a set of detected eye features, such as pupil, eyeball center and eye corners. Early work in this line requires high-resolution cameras and infrared lights[10, 19], while more recent works utilize machine learning models to detect eye features on webcam-based images [18, 17]. Those approaches tend to generate unsatisfying accuracy with low image quality, poor lighting condition or long sensing distance[20, 12].

Appearance-based methods require only a single webcam and leverage deep learning models to map 2D input images to gaze directions. Since this family of approaches do not require explicit eye features, it can handle images with lower resolution or quality than the other two categories. A few successful toolkits, such as OpenFace[1] and OpenGaze[20], first detect the face and facial landmarks(e.g. eyes, nose, mouth) from the input image, then utilize the appearance-based gaze estimation model to predict the gaze direction in the camera coordinate system and finally convert it to the screen coordinate system [1, 21]. Although these methods have shown comparable estimation accuracy as human annotations, they are developed and evaluated on adult image datasets and thus are not guaranteed to generalize well on infant videos. Finetuning the models on infant datasets not only requires fine-grained annotations of gaze direction, which is hard to obtain from low-resolution webcam videos, but also needs granular labeling of the facial features, which might be impossible since infants' actions like sucking their thumbs or lying on their parents' shoulders with only part of their faces visible makes it much harder to detect those landmarks.

Erel *et al.* recently propose iCatcher[6], a deep-learning-based framework trained specifically on infant videos to estimate gaze directions as three discrete categories: away, left and right. Instead of capturing the facial landmarks for each video frame, it uses a pretrained face detector from OpenCV[2] to extract all possible faces in a video frame, and then manually selects the face of infant from the extracted faces or selects the lowest face since the infants' faces are usually placed lower than their parents in the videos. It then trains a convolutional neural network to predict the label of gaze direction based on the selected infant face. Our proposed framework is based on this work but has modified both the selection of infant face and the prediction of gaze direction.

3. Approach

Figure 2 shows the overview of our framework. It consists of three major components: a face extractor, an infant selector and a gaze estimator. The novelty of our framework compared to iCatcher[6] is two-fold, as highlighted in blue. First, we add the infant selector on top of the face extractor to ensure that the gaze estimator takes the correct infant's face to make the estimation. Second, in addition to the face of the infant, we also provide the gaze estimator with spatial features of the facial bounding box. In the following sections, we will introduce the three components in details.

3.1. Face Extractor

Given a video frame, the first step is to extract the face of the infant user. Inspired by iCatcher[6], we use a face extractor developed by OpenCV[2]. This model uses a Single Shot Multibox Detector (SSD) framework[13] and adopts a

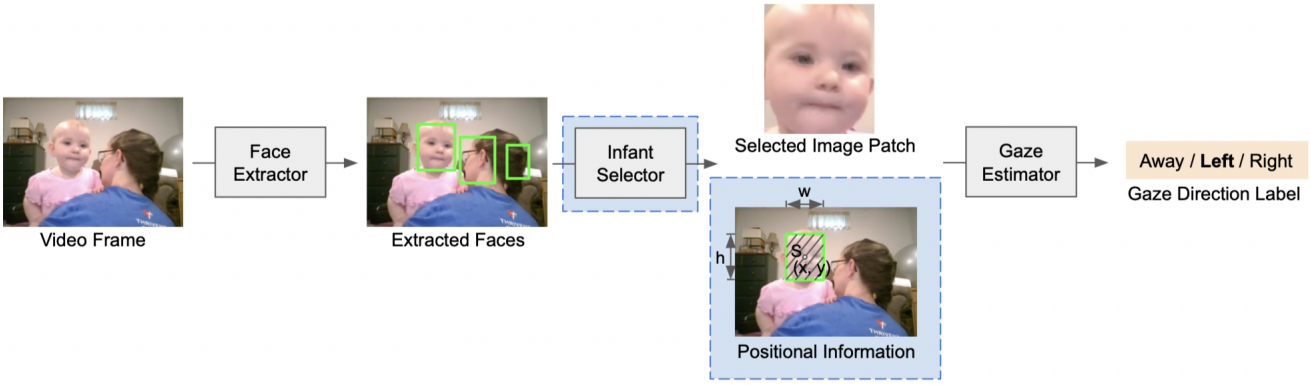


Figure 2: Overview of our framework. A face extractor first extracts all of the image patches that might bound human faces in the video frame. Afterwards, an infant selector picks the image patch that contains the infant's face. Finally, the selected image patch, together with the spatial features of its bounding box, is fed into a gaze estimator which estimates the gaze direction of the infant. The blue boxes highlight the novel aspects of our framework compared to iCatcher[6].

ResNet-10 like architecture as a backbone, except that it has much fewer channels in each convolutional layer compared to ResNet-10[8]. Its residual mechanism connects the original image with its hidden representation at different layers of the CNN, so that the final output is able to leverage the encoded features convoluted under different levels of complexity to determine a list of image patches that most likely contain human face. As shown in Figure 3, the face extractor also outputs a confidence score for each detected image patch, and we can filter these potential faces with a threshold of the confidence score. In our implementation, this threshold is set to 0.7.

3.2. Infant Selector

Since the face extractor outputs all image patches which contain not only infant faces, but also adult faces and face-like objects such as fists, ears and bricks, we need an infant selector to choose among them only the image patch of the infant's face. To do so, iCatcher[6] simply selects the image patch with the lowest bounding box on the video frame. We call it the *lowest-face* selection mechanism. However, the infant's face can be higher than other human faces or face-like objects in the videos, so the lowest-face selector will undoubtedly fail in such a scenario.

To ensure the downstream pipeline processes only infants' faces, we build a two-step infant selector component as shown in Figure 4. First, each image patch output by the face extractor is resized to 100×100 and fed into a VGG-16[15] classifier which classifies it as infant or non-infant. The classifier is trained on a small, manually-labeled subset of the image patches extracted by the face extractor. Once trained, our framework is free from any manual effort. Nonetheless, it is possible that multiple image patches are classified as infant in the same frame. For example, the

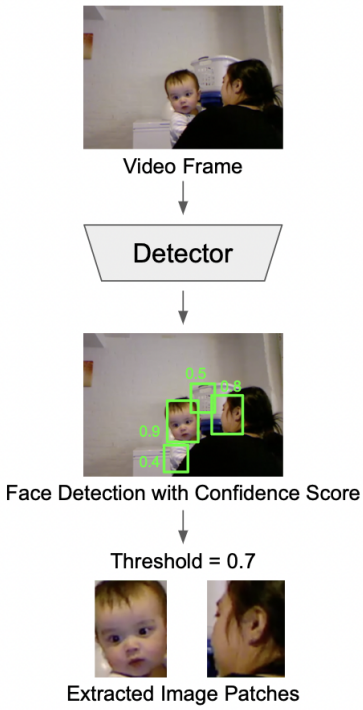


Figure 3: The face extractor. It first performs face detection with confidence score on the video frame, and then sets a threshold on the score to filter out non-facial patches.

siblings of the infant user may appear in the video, and his or her face is very likely to be classified as infant if they are young children. In this case, our infant selector executes a second step: among all image patches that are marked as infant, we select the one whose bounding box's center has the shortest Euclidean distance to the bounding box's center of the selected infant patch in the previous frame. This is

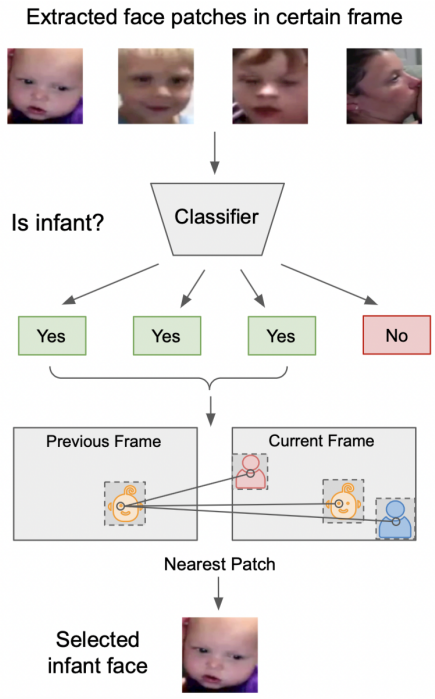
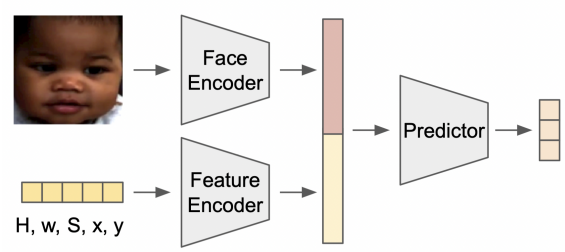
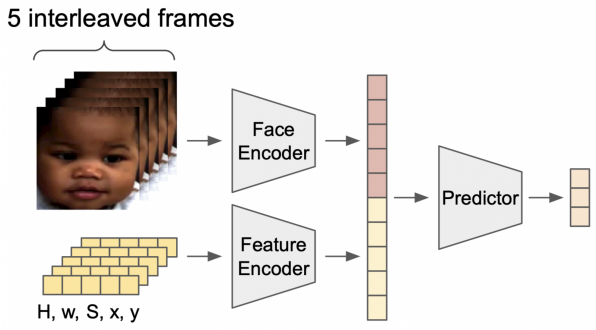


Figure 4: The infant selector. It first classifies each extracted image patch into infant or non-infant category. Then, among all the image patches that are classified as infant, it selects the one that is closest to the infant patch selected in the previous frame.



(a) Single-frame Gaze Estimator



(b) Multi-frame Gaze Estimator

Figure 5: Two types of gaze estimator. The single-frame gaze estimator takes the selected image patch and its corresponding positional features from a single frame as input, while the multi-frame gaze estimator takes those from 5 interleaved frames. The embedding(s) of the image patch(es) from the face encoder and the embedding(s) of the feature vector(s) are concatenated together as input to a CNN-based predictor that infers the gaze direction label.

based on the fact that infants are unlikely to drastically shift their head position within the interval of two consecutive frames[9].

3.3. Gaze Estimator

Once we obtain the correct face, the gaze estimator needs to infer the gaze direction among "away", "left" and "right". Unlike iCatcher[6] which makes the inference solely based on the raw pixels of the image patches, we feed into the gaze estimator the following spatial features of the bounding box in addition to the pixels: the height h, the width w, the area S and the coordinate of box center (x, y). In order to unify different sizes of the video screens, we normalize these spatial features between 0 and 1, with h and y divided by the height of the video frame, w and x by the width of the video frame, and S by the area of the video frame. The coordinates of the box center provide the information of relative location of the infant, while the height, width and area of the box provide the information about how far the infant is from the screen.

As shown in Figure 5, following iCatcher[6], we experimented with two kinds of gaze estimators. The first one, single-frame gaze estimator, takes the selected image patch and the spatial features of the corresponding bounding box in a single video frame as input. On one hand, the selected image patch is resized to 100×100 and the face encoder maps it to a 256-dimensional embedding. We use a ResNet-18[8] as the face encoder. On the other hand, the features of bounding box, i.e. the 5-dimensional vector $[h, w, S, x, y]$ is fed into a feature encoder which also outputs a 256-dimensional embedding. We use a two-layer fully connected neural network as the feature encoder. Afterwards, the embedding of the image patch and the embedding of the features are concatenated together and passed to a predictor. The predictor, for which we use a three-layer fully connected neural network, outputs a 3-dimensional vector for the three gaze directions. During training, the cross entropy loss is calculated based on the output of the predictor and the ground truth of the frame. During test, the estimated gaze direction of each frame corresponds to the the dimension with the highest score in the output vector.

Instead of relying on a single frame, the multi-frame gaze estimator leverages two preceding and two subsequent interleaved frames to make the estimation. The time in-

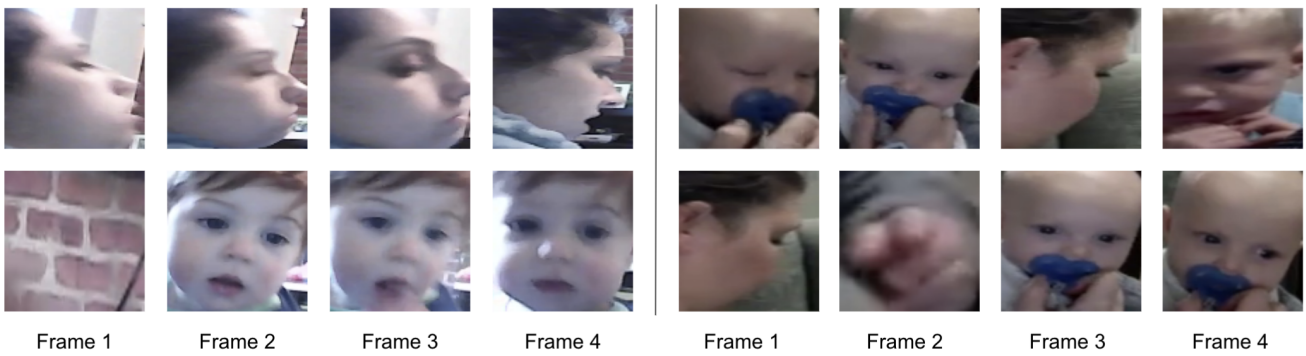


Figure 6: Image patches extracted by the face extractor from several example frames from two videos. Most of the patches are exactly human faces while there are still some non-face patches. The bottom line will be selected by iCatcher[6]'s lowest-face mechanism as infant faces. The lowest-face mechanism will make mistakes when none of the patches extracted from a frame is infant face or when the adult face's or the face-like object's position is lower than the infant's.

terval between each frame is $\frac{1}{15}$ second, thus the five interleaved frames span $\frac{1}{3}$ second, which is still negligible compared to infant's head shifting speed. The five frames are input to the shared face encoder, and the feature vectors for the five frames are also fed into the shared feature encoder. The predictor takes as input the concatenation of the five 256-dimensional embeddings of the images and the five 256-dimensional embeddings of the features to predict a 3-dimensional score vector for the three gaze direction classes. Other details are same as those of the single-frame gaze estimator. In addition to the shared ResNet-18 encoder, we also trained a 3D CNN as the face encoder of the multiple frames. Although we have tried our best to tune the hyperparameters, the performance of this design is not as good as the shared encoders design. Also, it takes much longer to train the 3D CNNs, so we use the shared encoders design in our final experiments.

4. Experimental Results

We implement our proposed framework in Python and evaluate it on the Lookit dataset. The code is available here. The dataset contains 96 videos with gaze labels from the same human annotator. 39 of the videos are labeled by a second human annotator. We use the 39 videos with two sets of annotations as test set and the rest 57 videos as training set. The length of each video is around 10 minutes and the frame rate is 30 fps. After filtering out the frames that do not have the gaze annotations or do not have infant faces, there are around $600,000$ frames in the training set and around $400,000$ frames in the test set. Figure 7 shows an example of video frame in the dataset for each class.

We evaluate the performance by calculating the gaze direction classification accuracy, which is defined as the percentage of agreement with the first human annotator. For the second annotator, the overall agreement with the first annotator is 92.92% over the test set videos. Figure 8 shows the confusion matrix, which indicates that the accuracy of each class is around 93% and the confusion between each two classes is similar.

Figure 7: Examples video frames in which the infant's gaze direction is away, left and right, respectively.

In the following, we will present the experimental results of each component in our framework.

4.1. Face Extractor

We load the pretrained weights of the detector in the face extractor from OpenCV[2] and perform the face extraction

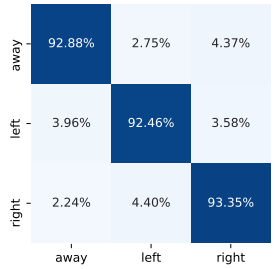


Figure 8: Confusion matrix of the second human annotator. The confusions between classes are balanced.

process on our dataset. Figure 6 shows a few image patches outputted by the face extractor for several frames in two videos, where the lowest-face selection mechanism used by iCatcher[6] considers the bottom row as infant faces. Although most of the extracted image patches are indeed human faces, there are still a few cases where non-face objects, such as fist and bricks, are detected as human face. On the other hand, the lowest-face selection mechanism will make mistakes in some cases if no manual correction is conducted. For example, in the first column of Figure 6, neither of the two extracted image patches is a infant face, but the bricks image patch is falsely considered as the target infant face. For another example, in the first example of the second video, the parent's face is falsely chosen since it is lower the infant's face in this frame.

4.2. Infant Selector

To train the infant selector, we randomly selected and manually labeled a small subset of the image patches extracted by the face extractor. Note that we only select the image patches from the video frames where more than one image patch is extracted. This small subset includes a variety of head poses for infants and adults, as well as the typical falsely detected face-like objects. It contains 600 image patches in total with half under the infant class and half under the non-infant class. 400 of them come from the training set and 200 of them come from the test set, and we use them as the corresponding training set and test set for the infant selector.

We leverage the existing architectures in the VGG[15] and ResNet[8] family for the infant classifier, and tune the corresponding hyperparameters. Each model is trained for 40 epochs with data augmentations of Random Rotation, Horizontal Flip, Color Jitter and Random Erasing. These augmentations correspond to the different head positions, lighting conditions and potential occlusions in the training data. For each model, we have explored different learning rates and optimizers. The learning rate is selected from $\{0.001, 0.003, 0.01, 0.03, 0.1\}$, the optimizer is selected from Adam, Adagrad, SGD with momentum 0.9 with and without weight decay 0.0001 and the batch size is set to 8. We use the cross entropy loss with the ground truth label to make the final classification.

In Table 1, we report the best test accuracy as well as the test accuracy in the last training epoch.

Infant Classifier	Last-epoch Accuracy	Best Accuracy
Lowest-face	60.5%	
VGG-11	92.5%	94.5%
VGG-16	**95.0%**	**95.0%**
ResNet-18	94.0%	94.0%
ResNet-34	91.5%	94.5%
Wide ResNet	87.5%	92.0%

Table 1: Infant face classifier accuracies of different architectures. Higher is better. Each row represents the best model after its hyperparameters are tuned. VGG-16 is the best model with the highest and stable test accuracy.

Compared with iCatcher[6]'s lowest-face selection mechanism, our selectors have obtained a considerable improvement. On the test dataset, our best infant classifier achieves an accuracy of 95.0%, while the lowest-face method achieves 60.5% accuracy, which is only slightly better than random guessing. Even though the images misclassified by the lowest-face selection mechanism are not very frequent among the entire set of extracted image patches, our infant selector can reduce the noise of the input to the downstream gaze estimator without manual effort.

However, our best model is not perfect. In the test set of 200 images, it fails to detect 7 infant images while falsely labels 3 non-infant images. The misclassified images are listed in Figure 9. Most of the errors in undetected infant images occur when the child is looking away from the screen, typically with some extreme head poses. This type of error is tolerable because when no infant is detected by the infant selector, the gaze direction of the frame will be consider as "away" by default, which coincides with the infant's actual gaze direction most of the times. On the other hand, when non-infant objects are falsely classified as infants, our selector is able to reduce these errors by the second step, i.e. selecting the face patch closest to the infant patch detected in the previous frame, ignoring these false positive patches. In fact, all of the three false positives here are indeed corrected by the second step in our experiment.

4.3. Gaze Estimator

We train the single-frame gaze estimator and multi-frame gaze estimator on the $600,000$-frame training set with a SGD optimizer with momentum 0.9 and weight decay 0.0001 for 20 epochs. Learning rate is set to be 0.01 for epochs 1-10, 0.001 for epochs 11-15 and 0.0001 for epochs 16-20. The batch size is set to 128. Color Jitter and Ran-

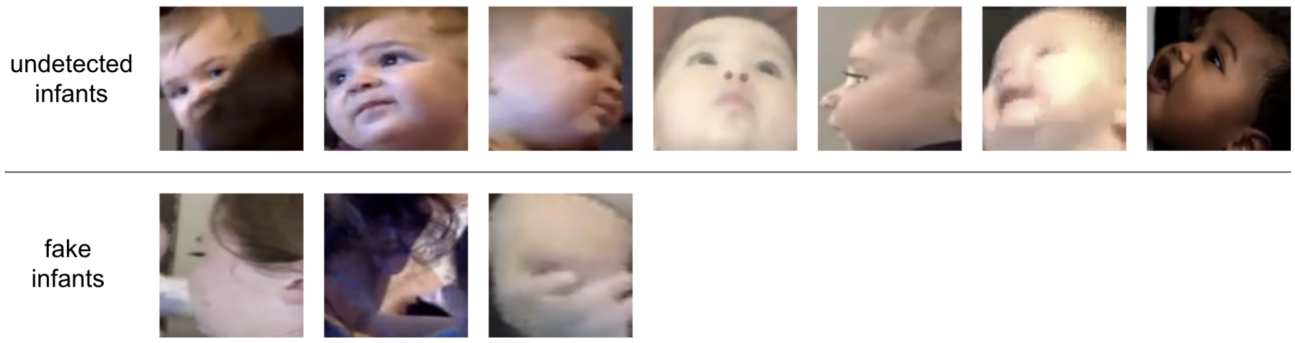


Figure 9: Failure cases of the infant selector. Most of them are undetected infants and 6 out of 7 of the undetected infants are looking away from the screen, which is tolerable since the gaze direction label of those frames will be considered as "away". The fake infants can be corrected by the second step of the infant selector.

dom Erasing are applied as data augmentations during training. The test accuracy is calculated for the model of the last training epoch on the $400,000$-frame test set.

As described before, the multi-frame gaze estimator takes $\frac{1}{3}$ second of data as one datapoint to make predictions. However, it is possible that the infant changes the gaze direction within the $\frac{1}{3}$ second. We call such datapoint a *transition* datapoint. Since even human annotators are unlikely to be perfect in classifying gaze direction of these transition frames, and the transition datapoints may also be confusing for the deep models, we add a set of experiments to train and test the multi-frame gaze estimator after eliminating all the transition datapoints.

Gaze Estimator Input	Lowest-face	Ours
Single-frame w/o Pos. Info.	82.14%	83.58%
Single-frame w/ Pos. Info.	82.23%	**84.20%**
Multi-frame w/o Pos. Info.	84.25%	85.61%
Multi-frame w/ Pos. Info.	84.65%	**85.95%**
Multi-frame(E) w/o Pos. Info.	86.23%	88.11%
Multi-frame(E) w/ Pos. Info.	86.98%	**88.58%**

Table 2: Gaze direction classification accuracies of different gaze estimators with different infant selectors. Higher is better. Multi-frame(E) stands for the experiments that have eliminated all the transition datapoints for the multi-frame gaze estimator. The gaze estimators with positional information consistently outperform those without. Estimators using our infant selector achieve higher accuracies than those using lowest-face infant selector.

Table 2 shows the gaze direction classification accuracies of three sets of experiments: single-frame, multi-frame and multi-frame(E), where Multi-frame(E) represents the experiments in which the transition datapoints are eliminated for the multi-frame gaze estimator. In each set of the experiments, we train and test the gaze estimator with or without the positional information, i.e. the spatial features of the bounding boxes, as well as using the lowest-face selection mechanism or using our infant selector. From the table, we can see that adding positional information, though slightly, consistently improves the performance from only using the image patch(es) for both selectors in all three sets of experiments. On the other hand, compared with the lowest-face selector, our infant selector delivers higher gaze direction classification accuracies for both with and without positional information in all three sets of experiments.

Figure 10 shows the confusion matrices for all the experiments. Unlike the human annotators, all of the models make more mistakes when the ground truth label of the frame is "away" and are more capable of distinguishing between "left" and "right". The potential reason is that the left or right gaze directions have more perceptible features to the models, such as the position of pupils, and have smaller intra-class sample variance. However, the "away" class contains a variety of circumstances and has a less fixed pattern. In addition, since the boundary of the screen is not known by the algorithm, the frames in which the infant is looking somewhere left(right) to but outside of the screen is likely to be classified into the "left"("right") class.

Meanwhile, even though the overall accuracies of the multi-frame models are higher than the single-frame models, the accuracies of the "away" class under the multi-frame models are mostly worse than the single-frame counterparts. This might be caused by the transition datapoints. For example, when the gaze direction of the infant shifts from the left *on* the screen to the left *to* the screen, it is hard for the model to categorize it into "left" or "away". With the transition datapoints eliminated, the multi-frame models achieve higher accuracies of the "away" class.

Another finding from the confusion matrices is that the improvement of the infant selector compared to the lowest-

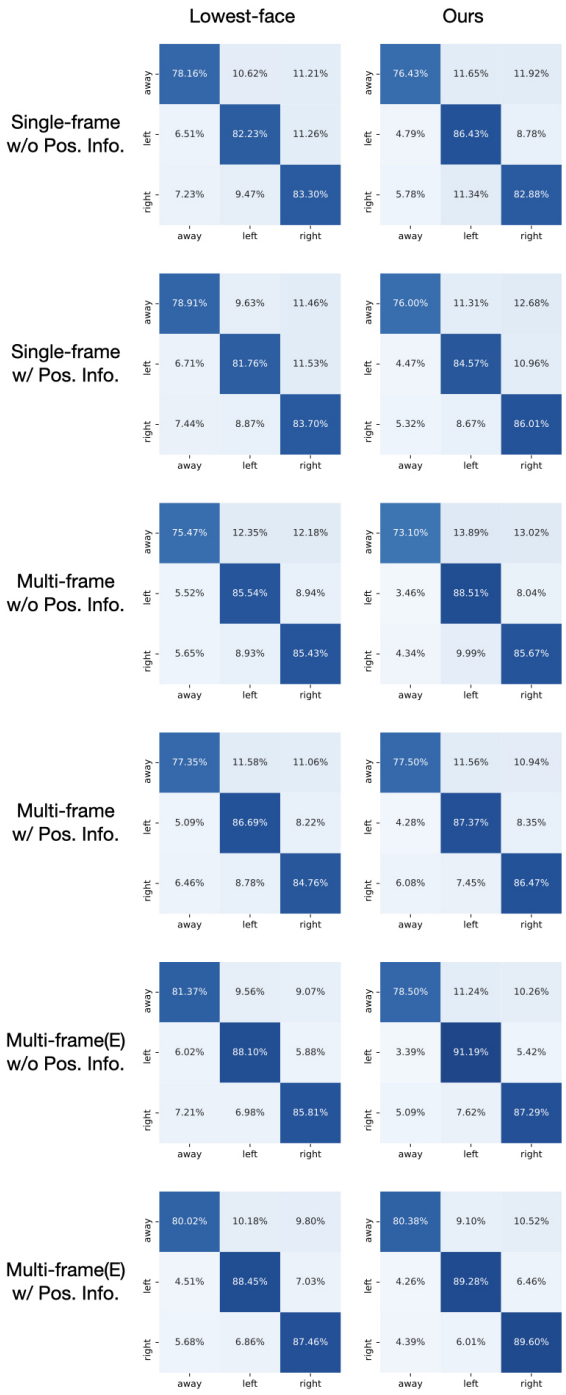


Figure 10: Confusion matrices for gaze estimation experiments. The accuracies of the "away" class is lower than that of other classes for all models. Compared to the lowest-face selector, our infant selector mainly improves the accuracy of the "left" and "right" classes. Eliminating the transition datapoints leads to higher accuracy of the "away" class for the multi-frame gaze estimator.

face selection mechanism is mainly in the "left" and "right" classes. A possible reason is that the gaze estimator can hardly output the correct direction of left or right when the lowest-face selector picks the wrong image patch that does not contain the infant's face, but it can predict the direction as "away" even if the given image patch is not the infant's face, e.g. the parent's face which is also looking away from the screen. Since our infant selector outputs more correct infant faces, we obtain higher accuracies on the "left" and "right" classes.

5. Discussion

5.1. Limitation

Although our framework achieves promising performance in terms of gaze direction classification accuracy, there are still several limitations. One of the intrinsic limitation of our framework is that it requires face extraction, which is sometimes hard for infant users due to occlusions. For example, in the first image patch of the undetected infants in Figure 9, most part of the infant's face is occluded by his parent, causing it to be falsely classified as non-infant. Moreover, its ground truth gaze label is "right" instead of "away", but our current selector cannot avoid this type of error.

In addition, our infant selector is trained on a manually curated dataset with only 300 image patches, which is too small for deep CNN models. The classifiers trained solely on this dataset may tend to overfit and may not generalize well on new data. Additional evaluation and enhancement needs to be conducted to ensure its generalizability.

5.2. Future Work

Given the intrinsic limitation of the framework, one direction of future work lies in developing an automated gaze coding framework for infant videos without the face extraction step. One possible solution can be extracting image patches of eyes instead of faces for the first step.

Moreover, another future direction is improving the generalizability and accuracy of infant face extractor. For example, we can enrich the curated dataset for the infant classifier by including more face images from and outside the Lookit study. Meanwhile, we can apply transfer learning techniques, such as training the latter half of a pretrained face detection model.

Furthermore, adding calibration steps to the framework can also be helpful to the gaze estimator. For example, we can ask the infant to look at something that moves around the whole boundary of the screen at the beginning of the video. By doing this, we can know where the boundary of the screen is from the view of the webcam, which may contribute to improving the accuracy of the "away" class.

6. Conclusion

In this project, we propose an automated gaze coding framework for infant videos. The framework is composed of a face extractor, a infant selector and a gaze estimator. Given a video frame, the face extractor extracts all image patches that may contain faces, and then the infant selector selects the patch that has the face of the infant user from the image patches. The gaze estimator finally approximates the gaze direction of the infant in the frame based on both the selected image patch and its positional information with three categories: away, left and right. The two main modifications of this framework to iCatcher[6] are both proved to be effective by empirical evaluations. Our framework achieves accuracies of 84.20%, 85.95%, 88.58% for the three-class gaze direction classification when using single frame, multiple frames and multiple frames without transition datapoints respectively. We believe this framework can serve as an assistant for gaze coding, especially for infant studies using preferential looking paradigms.

7. Individual Contributions

Peng and Xincheng contributed equally to defining the topic, communicating with the Lookit researchers, designing the proposed framework, doing the project presentation and writing the report. The data processing pipeline is implemented by Peng. The infant classifier is developed by Xincheng. The experiments of the gaze estimators are conducted by Peng.

References

[1] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016. 1, 2

[2] G. BRADSKI. The opencv library. *Dr Dobb's J. Software Tools*, 25:120–125, 2000. 2, 5

[3] Rechele Brooks and Andrew N Meltzoff. The development of gaze following and its relation to language. *Developmental science*, 8(6):535–543, 2005. 1

[4] Virginia Chow, Diane Poulin-Dubois, and Jessica Lewis. To see or not to see: Infants prefer to follow the gaze of a reliable looker. *Developmental science*, 11(5):761–770, 2008. 1

[5] Leslie B Cohen and Cara H Cashon. Infant perception and cognition. *Handbook of psychology*, pages 63–89, 2003. 1

[6] Y. Erel, C. Potter, S. Jaffe-Dax, C. Lew-Williams, and A. Bermano. Automatic, realtime coding of looking-while-listening videos using neural networks. Poster presented. 2, 3, 4, 5, 6, 9

[7] Dan Witzner Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):478–500, 2009. 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 3, 4, 6

[9] Bruce M. Hood and Janette Atkinson. Disengaging visual attention in the infant and adult. *Infant Behavior and Development*, 16(4):405–422, 1993. 4

[10] Takahiro Ishikawa. Passive driver gaze tracking with active appearance models. 2004. 2

[11] Susan Johnson, Virginia Slaughter, and Susan Carey. Whose gaze will infants follow? the elicitation of gaze-following in 12-month-olds. *Developmental Science*, 1(2):233–238, 1998. 1

[12] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016. 2

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, page 21–37, 2016. 2

[14] Kimberly Scott and Laura Schulz. Lookit (part 1): A new online platform for developmental research. *Open Mind*, 1(1):4–14, 2017. 1, 2

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 3, 6

[16] Jochen Triesch, Christof Teuscher, Gedeon O Deák, and Eric Carlson. Gaze following: Why (not) learn it? *Developmental science*, 9(2):125–147, 2006. 1

[17] Roberto Valenti, Nicu Sebe, and Theo Gevers. Combining head pose and eye location information for gaze estimation. *IEEE Transactions on Image Processing*, 21(2):802–815, 2012. 1, 2

[18] Erroll Wood and Andreas Bulling. Eyetab: Model-based gaze estimation on unmodified tablet computers. pages 207–210, 03 2014. 1, 2

[19] Hirotake Yamazoe, Akira Utsumi, Tomoko Yonezawa, and Shinji Abe. Remote gaze estimation with a single camera based on facial-feature tracking without special calibration actions. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 245–250, 2008. 2

[20] Xucong Zhang, Yusuke Sugano, and Andreas Bulling. Evaluation of appearance-based methods and implications for gaze-based applications. *CoRR*, abs/1901.10906, 2019. 1, 2

[21] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 41(1):162–175, 2019. 2

[22] Zhiwei Zhu and Qiang Ji. Eye gaze tracking under natural head movements. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 918–923. IEEE, 2005. 2

[23] Zhiwei Zhu, Qiang Ji, and Kristin P Bennett. Nonlinear eye gaze mapping function estimation via support vector regression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 1, pages 1132–1135. IEEE, 2006. 2