Experiments in Automatic Gaze Estimation for the Lookit Developmental Research Platform

Jack Cook
cookj@mit.edu

Ria Das
riadas@mit.edu

Abstract

A recent MIT-developed crowdsourcing platform known as Lookit has significantly accelerated the rate at which developmental psychologists can collect video data of infant subjects. With Lookit, researchers can obtain footage of infants' gaze directions when shown different images and sounds, allowing them to draw important conclusions about early childhood understanding. Unfortunately, data collected from Lookit must be manually annotated with which direction (left, right, or away from screen) that infants are looking before inference can be performed. We experiment with CNN-based models that automatically classify infant gaze directions from Lookit videos to try to reduce the overhead caused by human annotation, and compare our results to an existing gaze detection module. We find that while custom gaze estimation methods perform better than existing out-of-the-box solutions, the accuracy does not meet the high threshold necessary to be immediately useful to researchers using Lookit.

1 Introduction

In developmental neuroscience, studies of preverbal infants' gazes when exposed to different audiovisual stimuli have led to great advances in answering many of the field's most pressing questions. By determining whether infants look longer at certain categories and concepts over others, researchers have discovered important insights about such topics as preverbal infant language comprehension, numerosity judgments, and even morality. Given its success, it is indisputable that gaze-based metrics will remain a critical line of attack for developmental psychology researchers in the future [1].

Unfortunately, current attempts at increasing the scale at which this mode of research is conducted are impeded by a common bottleneck in cognitive science experiments: the need for large volumes of human-annotated data. A new online crowdsourcing platform, known as Lookit, for gathering videos of infants responding to different cues allows scientists to obtain the raw gaze footage that they need. However, the collected video must be annotated or "coded" by humans with which direction the infant is looking before it can be analyzed. Specifically, annotators currently must mark whether an infant is looking to the left, right, or away from the screen in each video frame before any inference can be performed. Although there exist many machine-learning based methods for detecting general gaze direction from videos, none of these methods have met the necessarily high accuracy bars that these studies in developmental neuroscience require. In particular, generalized methods are not well-suited to making precise distinctions between subtly-different left and right gazes that may not match head orientation, a crucial separation needed by the studies and which humans can decide easily. In addition, infants in Lookit videos tend to move around frequently, making it difficult for models trained on more stable head positions to make accurate predictions. Finally, most existing gaze detection algorithms are trained on adult faces as opposed to baby faces, which often have entirely new elements such as pacifiers, making face extraction in itself difficult. The human annotation process takes roughly ten times as long as the recorded video itself due to the need for high agreement about frames, an overhead should be reduced [1].

In this paper, we tackle the problem of automatically detecting infant gaze direction in the specific context of Lookit videos by experimenting with different neural network architectures for classifying gaze direction as left, right, or away from screen. We design our models and their associated feature extraction methods with a focus on integrating Lookit-specific criteria that general gaze detection models do not take into account. We begin by evaluating a popular existing framework for gaze detection called OpenFace on Lookit videos, to use as a benchmark for comparing the effectiveness of our later models. We then experiment with our own models, summarized as follows: (1) models trained on frames with faces as features to make predictions on the same and new videos, (2) models trained on frames with eyes

extracted as features to predict on the same and new videos, and (3) models trained on a small starting proportion of a video's frame's faces before predicting the rest of the same video to the end.

2 Related Work

Gaze estimation is a popular problem in computer vision, with a number of existing solution attempts. A promising framework is the OpenFace library, an all-in-one tool that performs facial landmark detection, head pose estimation, facial action unit recognition, and eye-gaze estimation [2]. The OpenFace model forms the basis of a CNN-based gaze detection algorithm developed at the Max Planck Institute, which the developers of Lookit have been interested in trying on their own datasets [3]. Lookit developers have also spoken with the creators of such gaze detection modules as WebGazer, TurkerGaze, PyGaze, and tracking.js [1]. These preliminary conversations have suggested that the facial landmark detection step may be the more difficult technical challenge, due to the fact that infants tend to move around significantly during Lookit video and that infant faces differ from adult faces. Other important considerations are robustness to lighting, partial occlusion, and ethnicity. The diversity of video data that is collected from an open crowdsourcing platform such as Lookit suggests that, of the existing solutions, a more established and well-oiled video-handling system such as OpenFace would be a better model to use as a benchmark than more niche methods.

3 Approach

We first evaluate the most promising framework according to previous outreach conducted by Lookit developers, the OpenFace platform. We use the performance of the OpenFace model, as well as observations about what types of input videos cause the model to perform poorly, to inform the models we construct later on. Our custom models fall into a few major categories:

1. **Models trained with faces as features**. In this category, we extract the infant's face from each frame of a Lookit video, and train AlexNet [4] on these facial images with the final classification layer mapping each face to a left, right, or away from screen label. We try several different hyperparameters as well as different architectures such as ResNet. We train the model on a subset of images from a single video to predict unseen frames from that video.

2. **Models trained with eyes as features**. This category of models mirrors that of the models trained on faces, with the infant's eyes extracted instead of the full face.

3. **Models trained on multiple videos to predict novel videos**. This category aims to find a model that would make it possible to automatically label novel videos without any labeling at all. We used face images in these experiments as opposed to eyes, due to the observed success of faces as features over eyes as described in the Experiments and Results sections.

3.1 Lookit Dataset

We briefly describe the Lookit dataset for context in the following sections. The dataset contains 15 webcam videos, each under 10 minutes long. In each video, an infant is shown images and played sounds. The infant is held by a parent who is supposed to be facing away from the screen at all times, but this constraint is not uniformly enforced, so that occasionally a parent's face can be seen for the first several seconds of a video. To eliminate this confounding factor when performing infant face extraction, we trim each video from the beginning so that the parent's face is never visible. Throughout the paper, the video files are identified by the last two characters of their file names for brevity.

Each video is associated with a single "coding file," which contains timestamps at which the infant's gaze changed direction (between left, right, and away), as determined by a human annotator. We construct a mapping between each frame of a video and the gaze direction labels in the coding file by determining the interval (enclosed by the timestamps at which the direction changed) into which the frame falls.

3.2 Model Tuning

We experimented substantially with our model architecture and hyperparameters. Specifically, we ran a brief hyperparameter search on an AWS server for two nights, and found that our best models used a batch size of 32, learning rate of 2.87×10^{-4}, and 19 epochs. We chose to use the AlexNet architecture because it trained substantially faster than alternative popular model architectures (e.g. VGG, ResNet, SqueezeNet, DenseNet), while yielding comparable accuracy.

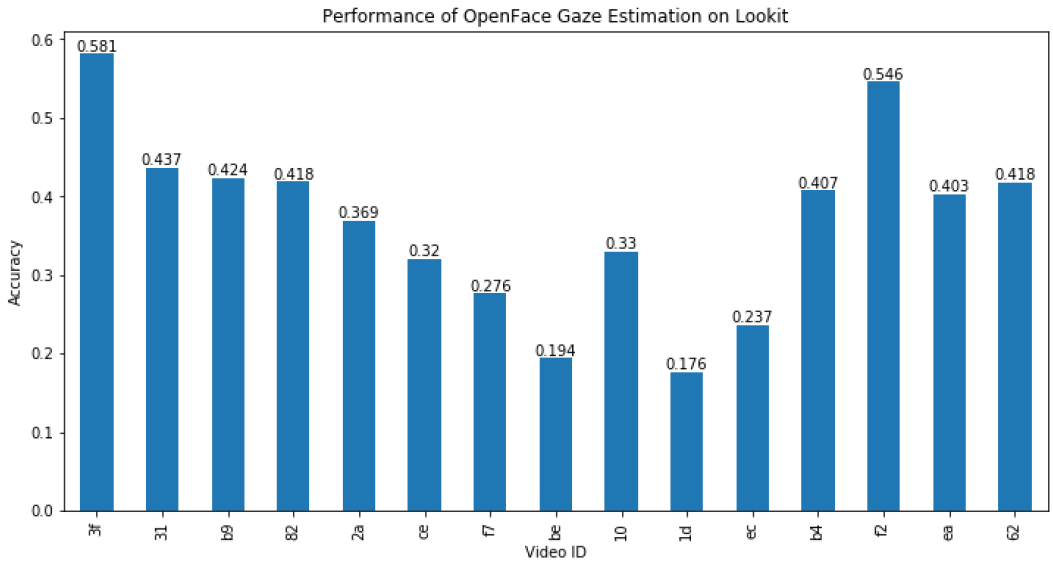


Figure 1: Accuracy of OpenFace left/right/away classifications on frames from each of the Lookit videos. Videos with low accuracy are of particular interest, because they suggest features that the OpenFace model may be failing to extract.

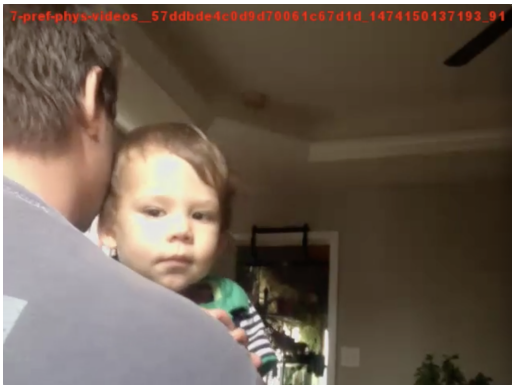
4 Experiments and Results

4.1 Evaluation of OpenFace

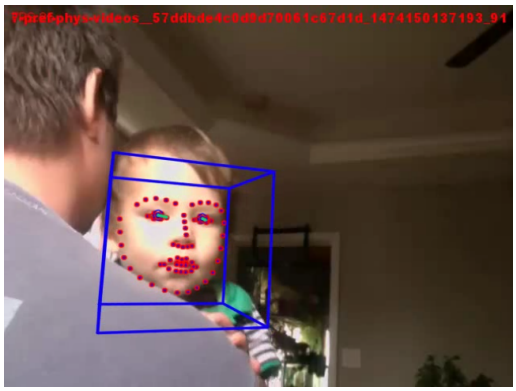
To evaluate OpenFace on the Lookit video dataset, we began by running the OpenFace gaze estimation module on each of the 15 Lookit videos. For each frame in each video, OpenFace produced a (1) gaze_angle_x and a (2) gaze_angle_y value in radians describing the left-right axis and the up-down axis of the detected infant's gaze (positive gaze_angle_x indicates the left direction, and positive gaze_angle_y indicates the up direction). We first converted each of these gaze angles into a left, right, or away classification, depending on the sign of gaze_angle_x. We classified all frames in which OpenFace was unable to detect a gaze as away. We compared these OpenFace classifications to the human-coded left/right/away classifications from the Lookit dataset for each video, and computed the percentage of matches. The results for each of the 15 videos are displayed in Figure 1, and the averaged accuracy over the videos was 37.8%.

As evident from the figure, there was a high degree of variability in the performance of OpenFace's left/right/away classification over the videos, with the highest accuracy being 58.1% and the lowest accuracy being 17.6%. Closer examination of the video with the lowest OpenFace accuracy revealed a likely reason for the low performance: The gaze detected by OpenFace was heavily influenced by the orientation of the infant's head, which remained steadily facing the right side of the screen through most of the video, though his eyes moved left and right frequently. The model consequently classified the majority of frames as right-gaze frames, though the Lookit data indicated that there were also many left-gaze frames. An example of this misclassification is displayed in Figure 2, where (a) shows the original video frame with the infant clearly looking left, while (b) shows OpenFace's misclassification of the gaze as right. In addition, examining the video also revealed that the position of the infant's head within the video, and in particular whether it was to the left or right side, might be a factor relevant to making left/right distinctions. The infant's head was far to the left of the video, making his gaze appear more central/straight-ahead when he looked at the left of the screen. This likely made the classifications determined using OpenFace less accurate as well.

Having determined this benchmark accuracy and some possible avenues via which to attempt to improve upon existing gaze detection models, we next describe the custom models we trained to classify the Lookit videos.



(a) Original Lookit Frame (Video **1d**)



(b) Frame with OpenFace Gaze Estimation

Figure 2: Example of OpenFace misclassifying infant gaze direction according to head orientation as opposed to pupil positions. From (a), it is clear that the infant is looking at the left of the screen, but in (b), the green vectors representing the detected gaze direction point to the right of the screen.

4.2 Evaluation of Custom Models

4.2.1 Face-Trained Models

We began by extracting faces from every frame of each video, using OpenCV's Haar cascade face classifier. If the classifier could not locate a face in a particular frame, we decided to use the rectangle enclosing the face in the previous frame as the current frame's face. We made this decision because we assumed that it was unlikely that the infant's head would move very significantly between individual frames of the video (all of which were 30 frames per second). We found in the face extraction process that a few of the videos were too blurry for the Haar cascade to consistently find faces, causing us to remove those videos from the rest of our analysis. We then associated each face image from the valid videos with the Lookit labels defined in the video's coding file.

We first trained AlexNet using the hyperparameters described in Section 3.2 on this set of labeled data for each individual valid video, splitting the frames into train/validation/test sets as 85%/5%/10%. We obtained the test accuracies described in Figure 2. The average accuracy of the left/right/away three-way classification across the videos was 70.6%, with a maximum accuracy of 88.2% and a miniumum accuracy of 50.4%. Interestingly, the video with the maximum accuracy (video 1d) was the video on which OpenFace performed most poorly, indicating that the gaze directions seemed to match the face images in some consistent way that could be learned.

4.2.2 Eye-Trained Models

We chose to experiment with extracting the infants' eyes as features to feed into the model instead of just the full faces due to the observed bias towards head orientation in the predictions made by OpenFace. We hoped that narrowing our extracted images to just the region around the eyes would limit the influence of head orientation, since we expected that such smaller, more focused images would encode less information that could be infered about head orientation.

Unfortunately, it proved to be very difficult to extract infant eye features using OpenCV's Haar cascade eye classifier on the Lookit video frames. Frequently, a face could be extracted from a frame, but no eyes could be detected due to the relatively poor resolution of the webcam video. Training models on a smaller number of frames where eyes were successfully detected produced models that performed either comparably or worse than the models that were trained on face images. This may have been due to a variety of reasons, but we suspect that the model is still getting information about the head orientation in these images due to lighting and shadows, which doesn't entirely remove the problem. In addition, the infants' heads tend to occupy small portions of each video frame. This means that the eyes tend to be extremely pixelated, and it is often difficult to tell where the infant is looking without extra information.

4.2.3 Multi-Video Models

In addition, we tried to train multiple models that utilized data from many different training videos, as opposed to the frames within a single video. However, none of these models were ultimately able to

apply what it learned in one video or collection of videos toward classifying the looking directions in a novel video. All of these models performed at chance or worse. A representative example of this poor generalization of the multi-video trained models to unseen videos is given in Table 1. There, the accuracy of a model trained on three of the videos that had high accuracy scores with single-video models is shown to be very high on frames from those videos, but very low on frames from an unseen video that also had a high single-model accuracy. For this reason, we largely focused on training on one video with partially labeled data, because these yielded much better results. However, we think that this may be a good area to explore in future work.

4.2.4 Models trained on Small Starting Proportion of Video

Finally, we experimented with a last idea that extends on our single-video faces-trained models that are described in Section 4.2.1. Although it is a laborious process for humans to annotate the entire length of a Lookit video with the gaze directions of an infant, it would still be useful if there existed a model that could be trained on a small subset of the labeled video's frames, and could accurately annotate the rest of the video given those labels. In practice, this would mean that a human annotator would have to label just a short clip of a video, and have the model automatically label the rest, which would still reduce the time needed to obtain annotations. To explore this idea, we trained a model on frames from different percentages of the video starting at time 0. We then tested the trained model on the remainder of the frames from the video. We conducted this analysis on each video, and plotted the average, minimum, and maximum accuracies that resulted for each training proportion in Figure 4. The accuracy of this model plateaus around when 40% of the video is used for training, which means that if implemented into Lookit, this model could save human labelers lots of time. Rather than labeling 100% of every video, in the future, after labeling about 40% of a given video, the rest could be done automatically.

5 Conclusion

We determined that the gaze estimation module OpenFace performs poorly on videos in which the head orientation of an infant does not align with the infant's gaze direction. This observation that OpenFace, and likely other gaze estimation models, may

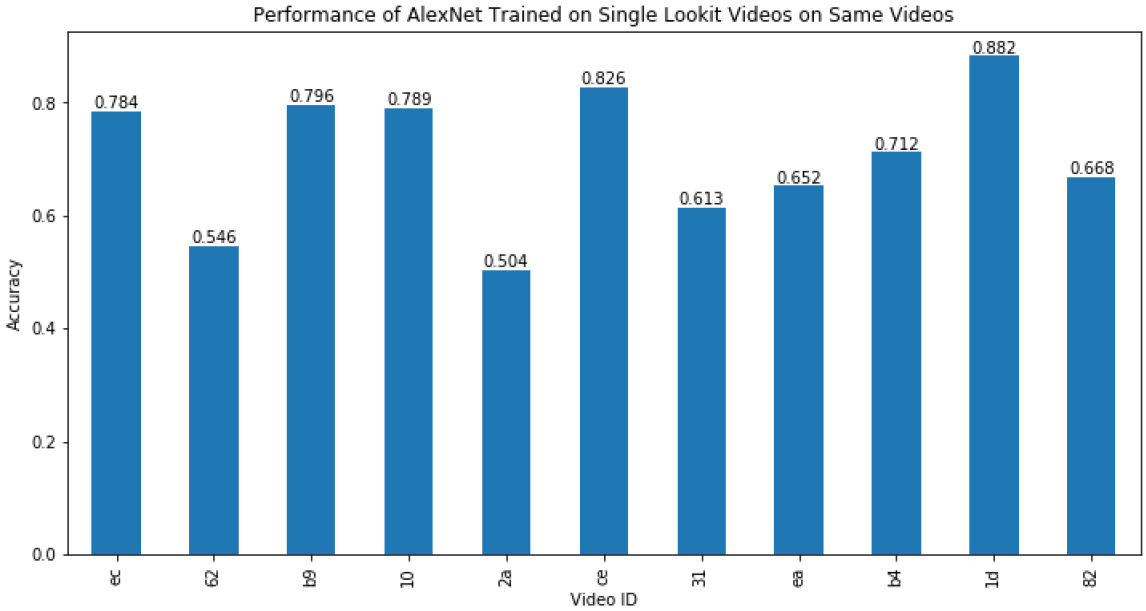


Figure 3: Accuracy of AlexNet trained on training frames from individual videos on test frames from the same videos.

Test Set	Accuracy
Frames from Videos ec, b9, 10	80.09
Frames from Videos ce	28.17

Table 1: Accuracy of AlexNet trained on training frames from individual videos on test frames from the same videos.

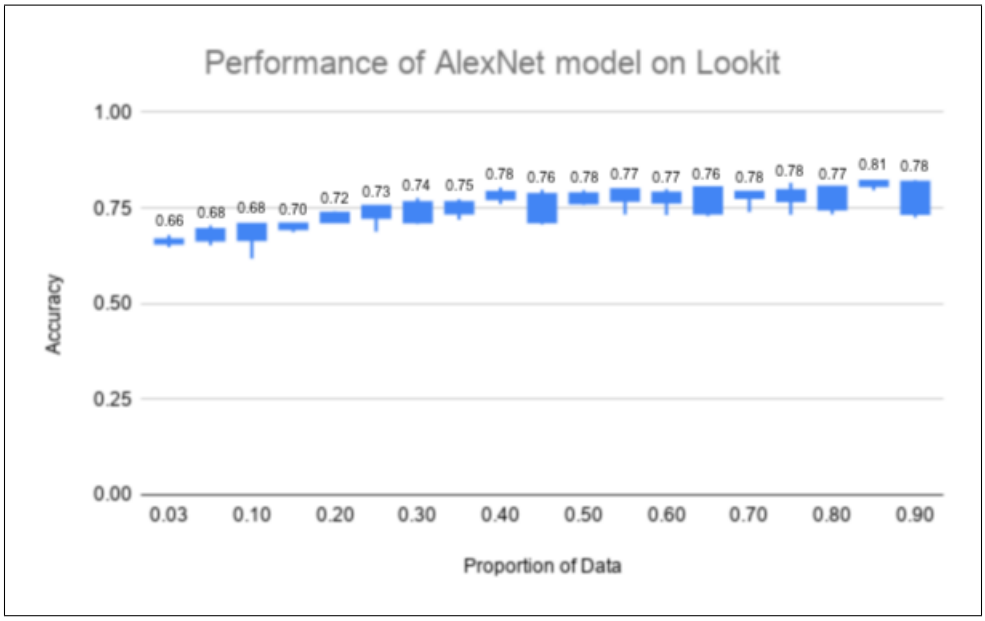


Figure 4: Accuracy of AlexNet trained on training frames from individual videos on test frames from the same videos. Labels above each candlestick depict averages for each proportion.

be biased towards predicting gaze direction to match head orientation should inform future attempts at improving automatic gaze detection in Lookit videos. The high mobility of infants in Lookit videos makes such scenarios common, and hence a priority in terms of improving automatic gaze estimation performance.

Moreover, we found that though training CNNs on faces from video frames worked on individual videos, more work needs to be done in order to label entirely novel videos from scratch.

6 Individual Contributions

Ria worked on assessing the OpenFace gaze estimation module on the Lookit dataset, as well as on preparing the Lookit dataset and extracting features for model training. Jack also worked on preparing data and extracting features for training, and further trained the models and tuned parameters to search for the best performance. Both authors contributed equally to the writing of the paper.

References

[1] K. Scott , and L. Schulz. Lookit (Part 1): A new online platform for developmental research. *Open Mind: Discoveries in Cognitive Science*, 1(1), 4–14. doi:10.1162/opmi_a_00002, 2017.

[2] T. Baltrušaitis, A. Zadeh, Y. Lim, and Louis-Philippe Morency. OpenFace 2.0: Facial Behavior Analysis Toolkit. *IEEE International Conference on Automatic Face and Gesture Recognition*, 2018.

[3] X. Zhang, Y. Sugano, M. Fritz and A. Bulling (201). *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW), 2017.

[4] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural net- works. In NIPS, 2012.