Multi-modal Fusion for Flasher Detection in a Mobile Video Chat Application

Lei Tian[1], Rahat Rafiq[1], Shaosong Li[1], David Chu[2],
Richard Han[1], Qin Lv[1], Shivakant Mishra[1]
[1] Dept. of Computer Science, University of Colorado Boulder [2]Microsoft Research
[1] {lei.tian, rahat.rafiq, shaosong.li, richard.han, qin.lv, mishras}@colorado.edu
[2] davidchu@microsoft.com

ABSTRACT

This paper investigates the development of accurate and efficient classifiers to identify misbehaving users (i.e., "flashers") in a mobile video chat application. Our analysis is based on video session data collected from a mobile client that we built that connects to a popular random video chat service. We show that prior image-based classifiers designed for identifying normal and misbehaving users in online video chat systems perform poorly on mobile video chat data. We present an enhanced image-based classifier that improves classification performance on mobile data. More importantly, we demonstrate that incorporating multi-modal mobile sensor data from accelerometer and the camera state (front/back) along with audio can significantly improve the overall image-based classification accuracy. Our work also shows that leveraging multiple image-based predictions within a session (i.e., temporal modality) has the potential to further improve the classification performance. Finally, we show that the cost of classification in terms of running time can be significantly reduced by employing a multi-level cascaded classifier in which high-complexity features and further image-based predictions are not generated unless needed.

1. INTRODUCTION

The existence of "flashers", or misbehaving users who expose their private body parts, has been a serious problem in online video chat services which support random video chat among strangers. As the number of mobile users increases, this problem is also propagating to mobile video chat applications. A recent study has provided the first insights into user behavior at scale in a mobile video chat application, assessing a variety of correlative factors that occur between mobile sensor data and two types of user behavior, namely normal and flashing user behavior [23]. However, that study did not address the problem of detecting flashers with high accuracy and high efficiency. This paper describes the design, development and evaluation of flasher detection classifiers that uniquely incorporate multi-modal mobile sensor data to substantially improve the classification accuracy and efficiency of random mobile video chat users, based on a large scale data set derived from a popular video chat service.

In this work, we follow common definition in previous research [8, 1, 26, 7, 27] and define flashers (or misbehaviors) as users who expose their bodies and show obscene content to others such as males/females showing their lower body part or females exposing their chests.

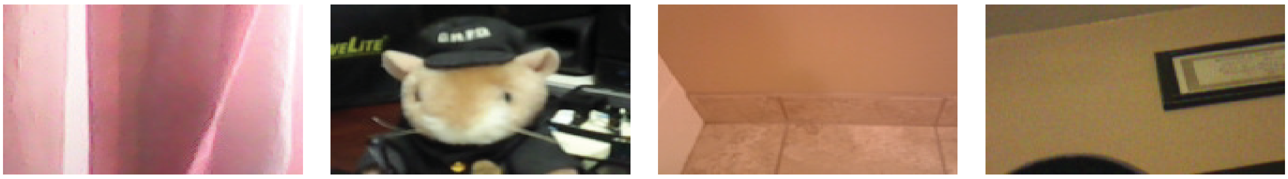
While prior research has described successful image-only classification for online video chat users [8, 1, 26, 7, 27], our focus in this paper is to investigate first whether these classifiers developed for online webcam-based video chat users are sufficiently accurate for mobile video chat users. The main property that was exploited in the design of previous classifiers based on online data was the relationship between a face and user behavior: online users who displayed a face were found to be correlated with normal behavior, whereas online users who did not display a face were found to be associated with flashing behavior.

In the new realm of mobile video chat, is the presence or absence of a face still as strong an indicator of a normal or misbehaving user? Prior work has shown that there is a wider diversity of scenes captured by the mobile video chat camera compared with a desktop webcam used by online video chat users [23]. As we will show, this greater diversity breaks down the relationship between the presence of a face and normal behavior. That is, in a mobile setting, we find that there are many more kinds of normal behavior that do not show a face yet not necessarily involve flashing behavior. In addition, there are other challenges introduced by the mobile environment, such as the appearance of partial faces and motion-blurred scenes that rarely occur in desktop milieus (see Figures 1 and 2 for some examples). We show later that previous face-centric classification achieves the lowest accuracy when operated over mobile video chat data among the various techniques explored in this paper.

Therefore, our focus in this work is to investigate whether additional sensing modalities can be leveraged to improve mobile flasher classification performance. Today's smartphones provide a wealth of contextual data from their mobile sensors, such as three-axis acceleration, orientation, light intensity, audio, front/back camera state, and image snapshots. We characterize which of these mobile sensors helps to meaningfully improve the accuracy of classification, and show that by fusing together classifiers based on images as well as key mobile sensing modalities like acceleration, camera state, and audio, the resulting fused classifier is substantially more accurate than the basic image-only face-centric classifier. Moreover, we demonstrate that by integrating multiple consecutive predictions within a session by our image-based multi-sensor classifier, accuracy can be further improved.

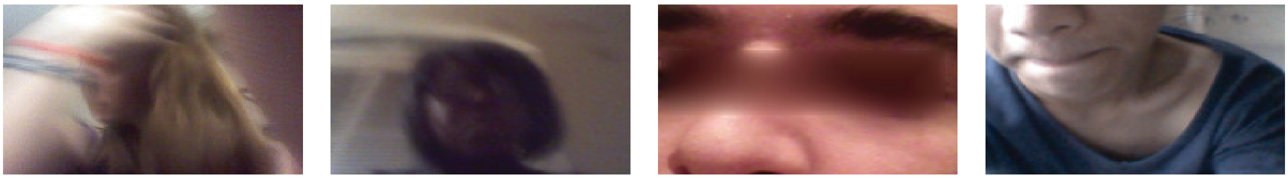
Besides accuracy, another key focus of this research is to improve the efficiency of classification, namely reducing the running time of the fused classifier. This is particularly helpful for mobile devices that are comparatively resource-constrained. We present in

MOBIQUITOUS 2014, December 02-05, London, Great Britain
Copyright © 2014 ICST 978-1-63190-039-6
DOI 10.4108/icst.mobiquitous.2014.257973



Curtain Doll Bathroom Wall

Figure 1: Examples of mobile phone-captured images that contain objects or background but no human.



Blurred Content 1 Blurred Content 2 Close Focus, Partial Face 1 Close Focus, Partial Face 2

Figure 2: Examples of low-quality images captured by mobile phones.

this paper a multi-level cascaded classifier that achieves efficient classification while largely preserving accuracy by carefully limiting execution of high-complexity classifiers and further imaged-based predictions within a session.

In summary, the major contributions of this paper are as follows. First, we show that prior image-only face-centric misbehavior classifier performs poorly on real-world mobile video chat data. Further, we demonstrate that an enhanced image-only classifier can meaningfully improve performance. Second, we show that a fused image-based multi-sensor classifier that incorporates the enhanced image classifier along with accelerometer, camera state, and audio can achieve significantly improvements on accuracy, precision and recall. Third, we present a session-based classifier which leverages multiple consecutive image-based multi-sensor predictions (i.e., temporal modality) and further improves classification performance. Finally, we demonstrate how cascaded classifier can be applied to our image-based and session-based classifiers and proposed a two-level cascaded classification to balance efficiency and accuracy. We quantify the tradeoffs between classification accuracy and efficiency.

In the remainder of this paper, we first describe related work, then present our real world mobile data set and the system used to collect the data set. Next, we introduce the classification algorithms developed for each of the modalities, namely image (basic and enhanced), audio, and orientation (accelerometer + front/back camera state), and session-based classification algorithm as well as the cascaded classification algorithm. We then present a detailed evaluation of the accuracy, precision, and recall of different classifiers, and finally the tradeoffs between classification accuracy and efficiency in the cascaded classifiers.

2. RELATED WORK

Desktop-based random video chat services such as Chatroulette [6], Omegle [20] and MeetMe [13] recently succeeded in gaining popularity, with tens of thousands of people online at any time during the day. Previous works [26] [7] [27] [11] have extensively researched about detecting misbehaviors in such services. SafeVchat [26] fuses multiple facial evidences (faces, eyes, upper body, etc.) into a probabilistic model using Dempster-Shafer Theory to classify normal and misbehaving users. A fine-grained cascaded (FGC) classifier was proposed to speed up compute-intensive processing (such as Dense SIFT, HOG) for classification [7]. This cascaded classifier is limited to local optimization, as it cannot handle the huge combinatorial number of feature set permutations. Also FGC uses "normal" prediction as the default stopping condition without taking into account the confidence of predictions. In our work, we improve this approach by using global optimization over a compact set of key features and an improved stopping condition based on the confidence threshold at each stage. EMeralD [27] proposes a more coarse-grained two-stage classifier (a rule-based pre-classifier as the front stage and a high complexity binary logistic regression model as the back stage) to achieve low computation and high accuracy in misbehavior classification. There are also some works [4] [22] that aim to understand teenage usage in an online video chat application, while others focus on analyzing video chat usage within a family [1] [14] [21].

There are very limited studies on user behavior in mobile video chat. An outline of mobile video chat issues and challenges is presented in [9]. MVChat [23] has analyzed large scale data obtained from a random mobile video chat application to understand normal and misbehaving users. However, this study only reports demographic information and correlation statistics, and did not proceed to the next step of developing and evaluating behavioral classifiers operating on actual mobile video chat data.

Audio samples collected by mobile phones have been widely studied to provide phone contexts and to support new services. SoundSense [12] presents a scalable sound prediction architecture and applies it to an audio daily diary application and a music detector application. SwordFight [28] uses audio tones exchanged between phones to localize each other and support a real-time mobile motion game.

A layered probabilistic representation of Hidden Markov Models has been used to fuse multimodal sensing at multiple levels of temporal granularity to recognize office activities [19] [18] [17]. And to reduce the computation required for sensing and processing, researchers have conducted studies to explore some policies based on expected-value-of-information (EVI) for selective perception [18] [17].

We have obtained the code for the latest misbehavior classifier sent to Chatroulette. This state-of-the-art algorithm is summarized in Algorithm 1 and referred to as the "CR" algorithm. It is an enhanced version of flasher classifier combining works of SafeVChat [26], EMeralD [27] and FGC misbehavior classifier [7] to tradeoff efficiency and accuracy. Chatroulette reports that such an approach reduces their number of server instances by a factor of three from over 100 servers to somewhat over 30 servers. The CR algorithm extracts one facial descriptor at a time and then checks whether that feature satisfies any predefined normal user association rules related to the existence, number of instances, and spatial relations of specific features. For example, the FaceRule checks whether there are two or more faces detected or whether the face width is more than 2/7 of the longest diagonal between the face center and four image points; the EyeRule checks whether a pair of eyes has been detected, and whether both eyes are sufficiently small (less than 0.03 of image size) and close to each other. If one rule is not satisfied, the CR algorithm will generate another feature and examine it with a new rule until the feature set exhausts. These features and their association rules are ordered according to their relevance in identifying normal users, with the most relevant features/rules listed at the top. This algorithm produces up to 13 labels, with labels such as Face (1), Eyes (2), Upper Body (3), etc. Based on these labels, it is then up to Chatroulette to decide how to classify whether a user is normal or not. We are not privy to the exact split point employed by Chatroulette. However, we present later an exhaustive analysis of all possible split points and choose the best split point as our CR baseline classifier.

```
Input $img$: snapshot image to be classified;
Output $1 \sim 13$: "Normal user" prediction confidence from
high to low;

$faces \leftarrow$ FaceDetect ($img$);
if FaceRule ($faces$) then
    return 1
else
    $eyes \leftarrow$ EyeDetect ($img$);
    if EyeRule ($eyes$) then
        return 2
    else
        $upbs \leftarrow$ UpbDetect ($img$);
        if UpbRule ($upbs$) then
            return 3
        else
            $mouths \leftarrow$ MouthDetect ($img$);
            ⋮
        end
    end
end
```

Algorithm 1: CR algorithm: State-of-the-art classification algorithm for flasher detection in online video chat.

A variety of algorithms for fusing multiple classifiers have been studied in the literature and are available for use in the Weka toolkit [25]. These include the J48 Decision Tree [16] [24], Random Forest [3], AdaBoost [29], Bootstrap aggregating (Bagging) [2] and Naive Bayes [15] [24]. We analyze the performance of these fused classification algorithms in a later section.

3. DATA COLLECTION

In order to understand real-world normal vs. flashing behavior at scale and have better control over data collection, we built an Android based, Omegle compliant, random mobile video chat application and deployed it in the Google Play market. This application has enabled us to collect hundreds of gigabytes of user behavior data from thousands of real users.

To support our study, we collected multi-modal sensor data to better capture user behavior during mobile video chat. First, the application monitors user actions such as start/end of session, camera state changes, and text messages typed during a chat session. Also our system stores data periodically collected from sensors such as image snapshots, three-axis acceleration, gyroscope and audio. To avoid interfering with the video chat, we selected the following sample frequencies and duty cycles for data collection. The accelerometer and gyroscope data are sampled at 5Hz; periodic image snapshots are captured every 30 seconds with a resolution of 160x120 pixels; a 10-second audio is recorded for every 40 seconds at 8 kHz. Finally, in our system, each user is assigned a unique UserID and the system also generates a unique SessionID for each video chat session. All the behavioral and sensor data are logged along with the corresponding timestamp, UserID, and SessionID information. Institutional Review Board (IRB) approval was obtained for all data collected in this study. Users who downloaded our application were required to give their informed consent during installation.

4. FLASHER DETECTION ALGORITHMS

4.1 Image-based Classification

4.1.1 *CR Performance on Mobile Video Chat Data*

To detect flashers in mobile video chat, we choose as our baseline the CR algorithm, which is state-of-the-art for desktop-based video chat services. As mentioned earlier, CR generates 13 classification labels, and a split threshold is used to determine how many features need to be detected for a user to be classified as normal. Figure 3 shows the accuracy of CR using different split thresholds. There is a slight peak at 8, which achieves the best accuracy of ~63%.

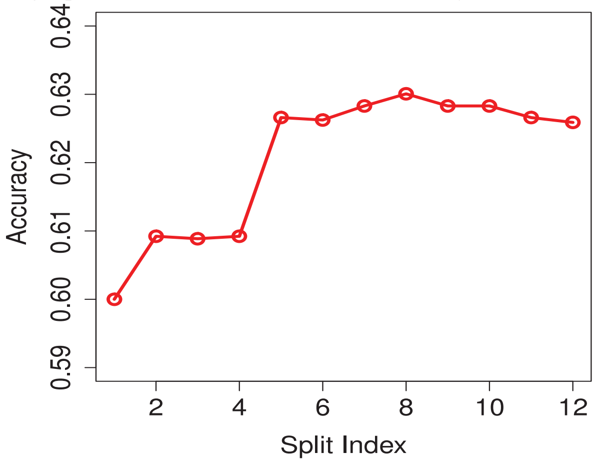


Figure 3: Classification accuracy of the CR algorithm on mobile video chat data using different split thresholds.

We posit the following reasons to explain the relatively limited performance of the CR algorithm:

(1) As shown in Figure 1, mobility results in much more diverse image content. The CR algorithm relies on facial features to predict whether a user is normal or misbehaving. In the online case, the absence of a face implies a misbehaving user with a very high confidence. However, in mobile video chat, we notice that there are a large number of video sessions that do not contain any faces and at the same time do not contain any objectionable content. Instead, they focus on the background or interesting objects around the users. An earlier research work showed that in a mobile video chat application, nearly 40% of video chat users show "others" type

of content which does not include a human [23]. Also, due to the lack of front camera on some low-end Android devices, some users cannot show their faces while chatting.

(2) Mobility also results in poor quality images that are blurred. Further, the distance between a user and the mobile camera is much shorter than that for desktop webcam-based users. Along with a limited wide angle camera, this results in many partial faces in the images (see Figure 2).

(3) Facial feature association rules defined in the CR algorithm are not as applicable on the mobile platform. For example, the EyeRule in the CR algorithm tries to detect a valid pair of eyes, where the distance between the eyes is fixed between 20 and 70 pixels. But since the mobile phone allows the user to put the camera close to their face, the faces can be much larger on the screen than typical webcams, resulting in larger distances between the eyes (Figure 2, third image). Also, the UpbRule in the CR algorithm looks for an upper body whose size is bigger than 40% of the image size. For the same reason of closeness between the mobile camera and the user, this rule is less applicable for mobile video chat images.

4.1.2 *Enhanced Image-based Classification*

Based on the observations above, we propose an enhanced image-based classifier, which improves upon the CR algorithm in two ways: 1) incorporates new features to detect images that do not contain any humans; and 2) improves the accuracy of facial feature detection on mobile video chat data. In particular, we incorporate the following five features:

Face size and proportion: Because of the short distance between user and the mobile camera, faces in mobile video chat tend to be much larger, occupying most of the screen space. So, we filter out all faces whose sizes are less than 1/6 of the image.

Pair of eyes: In low-quality mobile video chat images, it is difficult to detect a single eye. So, we focus on detecting only a pair of eyes that are close to each other and located in the same horizon.

Skin proportion: Skin proportion is a good feature to separate images containing humans from pure background content. We convert images to the YCbCr color space, which has been shown to be robust against large variations in lighting conditions and effective in skin detection [5].

Number and distribution of SIFT points: Scale-Invariant Feature Transform (SIFT) is well-known for object detection and was previously used for flasher detection [7]. Our analysis shows that images that do not contain humans tend to have either very few or a very large number of SIFT points and these points are typically scattered randomly. On the other hand, images that contain humans seem to have a medium number of SIFT points and those points are mostly concentrated around the facial area. We use the standard deviations of SIFT points' x and y positions to capture the distribution of SIFT points in an image.

Color histogram distribution: Images with only background content have very simple color hue, and generate sharp peaks and long tails in their color histograms. This pattern can be used to differentiate between images containing human from images containing only simple background. In our experiment, we use sixteen bins to measure R/G/B color histogram and calculate the standard deviations for the histogram distributions.

4.2 Sensor-based Classification

Mobile devices are equipped with a variety of sensors, such as accelerometer and gyroscope. In addition, newer smartphones are equipped with two cameras, both front and back. Those sensors can offer some useful contextual information about user behavior during a video chat. The question is which sensors and how they can be leveraged for flasher detection. Since many of our mobile video chat users have older Android phones that do not have gysoscope, our investigation focuses on the three-axis accelerometer data, which are available across all smartphone platforms.

Our preliminary analysis indicates that normal or misbehaving users' video chat content is highly correlated with the position that a mobile camera focuses on. This can be estimated by phone orientation along with active camera position (front or back). Furthermore, during a chat session, normal users tend to keep their phones stable, while flashers' phones have more slight vibrations.

In this work, for each snapshot image, we collect acceleration data for the two seconds before and two seconds after when the snapshot is captured. After applying a smoothing function to the four-second accelerometer data, we calculate the mean and standard deviation values along the three axes. These values are combined with the active camera (front or back) information to represent phone orientation and vibration during video chat.

4.3 Audio-based Classification

Besides image-based and sensor-based features, we also investigate the potential of using audio data to classify normal vs. misbehaving users in mobile video chat. We first labeled our audio data using six different categories: 1) Deep Breath; 2) Music; 3) TV; 4) Quiet; 5) Talk; and 6) Others (ambient noise with unrecognized audio). Figure 4 shows the number of normal and misbehaving users in the six different audio categories. We see that users who "Talk" are usually normal users while "Quiet" users are more likely to be misbehaving.

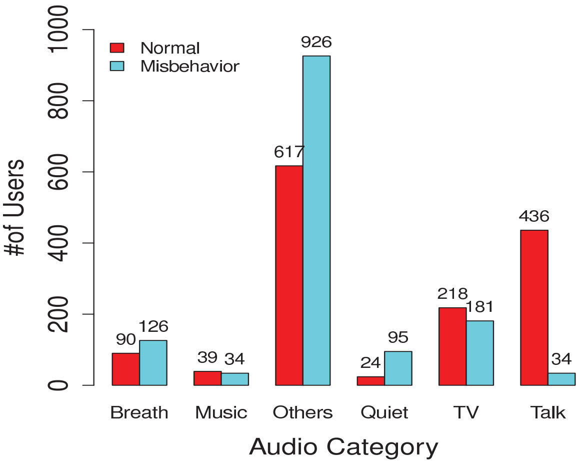


Figure 4: Number of normal vs. misbehaving users in different audio categories.

A lot of research has been done to predict audio categories by analyzing audio signals and has achieved very promising results [10] [12]. Our audio class prediction algorithm is based on these earlier works and consists of the following four steps.

The **framing** step segments each 10-second audio clip into non-overlapping 64-ms frames. In the **feature extraction** step, we extract the following features from each audio frame: (i) Root Mean Square (RMS) that captures the overall energy; (ii) Spectral entropy that indicates the frequency pattern of the audio frame: a high entropy resulting from flat spectrum strongly suggests silent audio; (iii) Zero Crossing Rate (ZCR) that measures sign change rate of a signal, which is effective in speech recognition and music information retrieval; (iv) Bandwidth: ambient sound typically has a small bandwidth and music consists of a wider mixture of frequencies; and (v) first 13 Mel-frequency Cepstral Coefficient

(MFCCs) that is a better approximation for human auditory system and has been proved to be effective to identify finer-grained audio categories. For **inter-frame feature extraction**, we consider n consecutive frames. We average the features extracted from the individual frames and calculate the standard deviation to measure feature changes among the frames. Finally, for **audio category prediction**, we feed the features into a J48 classifier to make a prediction for each frame window. The audio class that receives the majority vote among multiple frame windows is picked as the final prediction for an audio clip. The only exception is that, once an audio frame is determined to be in the "Talk" category, the whole audio clip is assigned to the "Talk" category.

4.4 Session-based Classification

Our analysis also indicates that people tend to behave consistently during a video chat session, and seldom switch between normal and flashing behaviors. Motivated by this observation, we propose a session-based flasher detection mechanism that leverages the temporal modality and takes as input the classification results of multiple image snapshots and their corresponding sensor readings to generate a more reliable normal vs. misbehaving user prediction for a whole session.

Our session-based classification algorithm works as follows. For each snapshot image and its corresponding sensor data, our image-based multi-sensor classifier gives a binary prediction (Normal vs. Misbehaving) along with a confidence value. A value in the range of $[0, 0.5)$ (or $(0.5, 1]$) indicates misbehaving (or normal), and the lower (or higher) the value, the higher the likelihood that the user is misbehaving (or normal). We apply a 6-bin discretization on the binary prediction and confidence values, specifically,

$$\begin{array}{rcl} strong\_normal & : & normal + conf \in [0.75, 1] \\ medium\_normal & : & normal + conf \in (0.65, 0.75) \\ weak\_normal & : & normal + conf \in (0.5, 0.65] \\ weak\_mis & : & misbehaving + conf \in [0.4, 0.5) \\ medium\_mis & : & misbehaving + conf \in (0.25, 0.4) \\ strong\_mis & : & misbehaving + conf \in [0, 0.25] \end{array} \quad (1)$$

Then, for each session, we calculate the number of occurrences in each bin. We also measure the min, max, mean, and standard deviation of all the prediction confidence values in a session. All these features are fed into a Naive Bayes classifier to generate the final prediction for each session.

4.5 Cascaded Fusion Classifier

Given multiple features, one straightforward classification approach is to simply combine all features together. However, this is wasteful, since not all features are necessary when classifying a specific instance. For example, a user can be classified as normal with high confidence when a face is detected. Motivated by this observation, we propose cascaded classification. As illustrated in Figure 5, a cascaded classifier consists of a sequence of classifiers ordered by certain criteria (objective function F_{obj}) such as average acquisition time or accuracy. Samples that need to be classified pass through the classifiers in stages. At the i-th stage, a new feature f_i is extracted (acquisition time t_i) and used by classifier C_i (alone or with previously extracted features) for classification. If the classification confidence of C_i on a sample is above the confidence threshold σ_i, the classification process stops and the decision made by classifier C_i on the sample is accepted as the sample's final classification. Otherwise, the sample is passed to the next stage classifier C_{i+1} for further processing. Since samples classified with high confidence at earlier stages do not need to go through later stages, the overall classification time can be reduced, while still ensuring high classification accuracy. The challenge lies at the design of the objective function, which determines the ordering of the classifiers to be used at each stage.

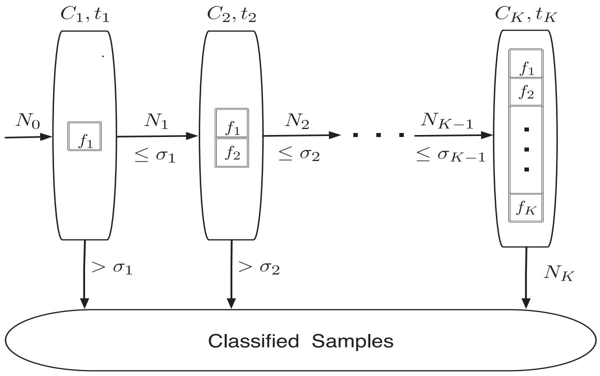


Figure 5: K-stage cascaded classifier.

4.5.1 Image-based Multi-sensor Cascading

Given each snapshot image and its corresponding sensor data, our image-based multi-sensor classifier fuses together multi-modal features in order to make a classification of normal vs. misbehaving users. Specifically, we aim to build a seven-stage cascaded classifier corresponding to the seven features (face, eye pair, skin proportion, SIFT number and distribution, color histogram, phone orientation, and audio category). Using cascaded fusion, the key is to utilize at earlier stages features that are efficient to compute and have high classification confidence, thereby reducing/avoiding more complex feature computations at later stages. The compact set of features and the relatively small number of permutations (7!) of the seven features make it possible for global optimization. To balance between classification accuracy and efficiency requirements, we design the following cascaded fusion classifier objective function:

$$F_{obj} = T_{save} + \alpha \cdot P_K \quad (2)$$

$$T_{save} = \frac{T_K \cdot N_K}{N_0} \quad (3)$$

$$T_i = (T_{i-1} - t_i) \cdot \frac{N_{i-1}}{N_i} \quad i \in [0, K-1] \quad (4)$$

$$T_0 = \sum_{i=1}^{K} t_i \quad (5)$$

$$\gamma = \frac{T_0}{T_0 - T_{save}} \quad (6)$$

Here, P_K is the overall classification accuracy after the K-th stage. $N_{i-1} (i \in [1, K])$ is the number of samples passed into the i-th stage for classification. Note that N_K equals N_{K-1} since the last stage classifies all remaining samples. t_i is the acquisition cost for feature f_i. Assuming at the beginning of the cascaded classification process, each of the N_0 samples is allowed T_0 amount of time for classification, i.e., going through all K stages. As more samples are filtered out and classified at earlier stages, the total amount of unused time $(T_{i-1} - t_i) \cdot N_{i-1}$ is divided over the remaining N_i samples, which is less than or equal to N_{i-1}. Therefore, T_{save} measures the amount of time saved per original sample, and γ measures the *speedup ratio* compared with the full-fusion process with all features. Finally, parameter α controls the importance

Table 1: Classification Quality Comparison of Different Image Features

	Accuracy	Normal Precision	Normal Recall	Misbehaving Precision	Misbehaving Recall
CR Algorithm (baseline)	0.630	0.694	0.518	0.608	0.765
Face+Histogram+Skin+SIFT+Eye	**0.689**	0.713	**0.646**	**0.669**	0.733
Face+Histogram+Skin+SIFT	0.680	0.700	0.644	0.663	0.717
Face+Histogram+Skin	0.665	0.681	0.637	0.651	0.693
Face+Histogram	0.633	0.644	0.616	0.623	0.651
Face	0.588	**0.962**	0.195	0.546	**0.992**
Eye	0.520	0.558	0.249	0.509	0.798
Histogram	0.594	0.600	0.591	0.587	0.596
Skin	0.565	0.570	0.573	0.560	0.557
SIFT	0.568	0.574	0.574	0.562	0.563

ratio between the acquisition cost requirement (average execution time) and the accuracy requirement.

4.5.2 Session-based Cascading

Given the temporal ordering of data in a video chat session, the sequence of snapshot images and their corresponding sensor data to be used for classification is fixed in a session. Thus the key issue for session-based cascading is determining the actual number of cascade stages to include in order to achieve a good balance between classification accuracy and efficiency. In other words, our goal is to explore the impact (or tradeoff) of different number of cascade stages on the overall session-based classification performance.

5. EVALUATION

In this section, using real-world data collected from our mobile video chat system, we first evaluate classifier performance on individual features. We show that by fusing together features obtained from image, audio, and multiple sensors, we can significantly improve the classification accuracy, compared with the baseline CR algorithm, which is state-of-the-art for online video chat systems. Also, we show that session-based classification further improves the classification accuracy by leveraging the temporal modality containing multiple classification results within a session. Finally, we measure the execution time of different features, and correspondingly the tradeoff between accuracy and efficiency using image-based multi-sensor cascading and session-based cascading methods.

Let R_n and R_m be the sets of normal and misbehaving users identified by a given classifier, respectively. Let I_n and I_m be the sets of true normal and true misbehaving users. We consider the following five different classification quality metrics:

$$\begin{aligned}
Accuracy &= \frac{|R_n \cap I_n| + |R_m \cap I_m|}{|R_n| + |R_m|} \\
Normal\ Precision &= |R_n \cap I_n| / |R_n| \\
Normal\ Recall &= |R_n \cap I_n| / |I_n| \\
Misbehaving\ Precision &= |R_m \cap I_m| / |R_m| \\
Misbehaving\ Recall &= |R_m \cap I_m| / |I_m|
\end{aligned}$$

5.1 Dataset

According to a previous study on mobile video chat [23], most video chat sessions are short because users keep requesting the next random user pairing until they have found someone interesting to chat for a longer session. In our analysis, we focus on these “meaningful" sessions whose durations are more than 90 seconds, which allow us to collect at least 4 snapshot images and 3 audio clips for each session and are sufficient for session-based classification evaluation. From our dataset, we obtain nearly 350 labeled misbehaving sessions and 1450 labeled normal sessions. The ratio is approximately 1 to 4, which is consistent with the finding in [23]. To deal with this skewed distribution and avoid over-training for the normal category, we then pick a balanced dataset containing 348 misbehaving sessions and 357 normal sessions. For all these sessions, we only consider the first 90 seconds. And the four snapshot images contained in each 90-second session are split into four subsets. In all image-based classifier evaluation, the four subsets of data are evaluated separately using ten-fold cross validation and the average is reported as the overall performance.

When labeling images as normal or misbehaving, we follow a procedure that is similar to the one used in [23]. Misbehaving users were identified as displaying naked lower bodies for males and naked lower and/or upper bodies for females. Two people labeled the same data set. If an image received conflicting labels, then the two labelers would meet to resolve the conflict.

Audio labeling was also performed by two people with a similar procedure. Apart from our test audio dataset collected from the mobile video chat clients, we also captured a 10-minute training dataset for each of the six predefined audio categories described earlier. After experimenting with different frame sizes, we found that when window size $n = 16$, our predictor provides the best performance. This results in each frame window to be approximately 1 second in length ($16 \times 0.064 = 1.024s$).

5.2 Image-based Classifier Performance

We begin by examining the best classification performance that could be achieved using image-only features. The baseline classifier we use is the CR algorithm, which is a face-centric, image-only algorithm that is currently used by Chatroulette, and is considered state-of-the-art for flasher detection in online video chat systems. In our enhanced image-based classifier, we consider five features (number of faces, existence of eye pair, color histogram statistics, SIFT feature vectors, and skin proportion). We report the classification performance using each individual features, as well as the fused features using Random Forest[1].

Table 1 summarizes the classification performance when different image-based features are used. In particular, we find that all proposed features are important factors in improving the accuracy of the image-based classifier. Compared with the 0.630 accuracy achieved by the baseline CR algorithm, our enhanced image-based classifier, which combines all five features (face, eye, histogram, skin, SIFT), achieved an improved accuracy of 0.689.

5.3 Audio Category Classifier Performance

Table 3 shows the confusion matrix of our audio category clas-

[1] We evaluated different fusion techniques and Random Forest achieves the highest accuracy.

Table 2: Classification Quality Comparison of Multi-sensor Fusion

	Accuracy	Normal Precision	Normal Recall	Misbehaving Precision	Misbehaving Recall
Enhanced Image	0.689	0.713	0.646	0.669	0.773
Audio	0.606	0.675	0.431	0.574	0.787
Acc. + Camera Position	0.769	0.764	0.787	0.775	0.750
Acc. + Camera Position + Enhanced Image	0.804	0.810	0.807	0.803	0.805
Acc. + Camera Position + Enhanced Image + Audio	**0.820**	**0.822**	**0.821**	**0.817**	**0.818**

sifier, i.e., the number of audio instances in each category that are (mis-)classified into other audio categories. We see that Deep Breath are often misclassified as Others due to ambient noise in the background, and the TV category is likely to be misclassified as Talk, Music or Others since TV audio could contain variants of these types of sound as well. Overall, our audio classifier achieves an accuracy of 0.70, and performs well for the two categories that are most effective for flasher detection (0.95 accuracy for the Quiet category and 0.73 accuracy for the Talk category).

Table 3: Confusion Matrix of Audio Category Prediction

	Deep Breath	Music	Others	Quiet	TV	Talk
Deep Breath	**52**	6	131	2	6	19
Music	2	**35**	5	0	19	12
Others	67	27	**1193**	104	49	93
Quiet	0	0	7	**122**	0	0
TV	6	61	55	4	**200**	73
Talk	10	14	60	2	40	**344**

5.4 Multi-sensor Fusion Classifier Performance

A major goal of this paper is to understand whether and to what extent multi-modality mobile sensor data can help improve the flasher detection performance, compared with previous face-centric image-only classification. Here, we evaluate the classification quality using multiple sensors. In particular, we examine the following three modalities: (1) **image:** the enhanced image-only classifier developed earlier that combines face, eye, skin proportion, sift and histogram distributions; (2) **orientation:** the phone orientation-related features processed from 3-axis accelerometers and camera position (front/back); and (3) **audio:** the predicted major audio category as well as a vector containing the predicted occurrences of each audio category.

Table 2 shows the classification quality of different sensor modalities as well as the fused results using Random Forest. We find that the mobile accelerometer and camera sensors used by the Orientation classifier result in a strong gain in accuracy to 0.769 compared to our previous best enhanced image-only classifier of 0.689. Moreover, we find that when fusing the orientation and enhanced image classifiers together, we can achieve an additional gain in accuracy to 0.804. Finally, when we fuse all three modalities (mobile sensor + enhanced image + audio), we observe a final overall accuracy gain to 0.820. In addition, we see that normal and misbehavior precision and recall values are all improved up to 0.82 as we fuse together more sensing modalities. This demonstrates that combining contextual information from multiple sensing modalities substantially improves the overall classification performance of flasher detection on mobile video chat data.

We also evaluate our multi-sensor fusion classifier's performance over different fusion algorithms. We choose to compare five different fusion algorithms (J48 Decision Tree, Random Forest, AdaBoost, Bootstrap Aggregating (Bagging) and Naive Bayes) on our dataset, using default parameter settings from the Weka toolkit. While other fusion algorithms' quality values are mostly below 0.80 and/or unbalanced between precision and recall, Random Forest achieves the highest accuracy of 0.820 and balanced results over precision and recall for normal and misbehaving users.

Table 4: Session-based Classifier Quality Comparison

First x Image Predictions Used	1	2	3	4
Major Voting	0.815	0.823	0.850	0.848
Normal Dominated	0.815	0.823	0.812	0.807
Misbehavior Dominated	0.815	0.817	0.803	0.780
Average Confidence	0.815	0.838	**0.855**	0.845
Naive Bayes (our approach)	0.815	**0.843**	0.854	**0.859**

5.5 Session-based Classifier Performance

Here, we evaluate the classification quality of our session-based classifier. We consider the first $x (x = 1, 2, 3, 4)$ image predictions in each session, and different policies to combine the prediction results. The policies include Major Voting, Normal (Misbehaving) Dominated which predicts the user to be Normal (Misbehaving) when at least one image prediction is Normal (Misbehaving), average confidence, and Naive Bayes (used by our proposed session-based method).

The results are summarized in Table 4. When only the first image prediction is used, an accuracy of 0.815 is achieved. Using more image predictions generally improves the classification accuracy. The most gain is achieved with Naive Bayes (our approach) and all 4 image predictions, which resulted in an accuracy of 0.859. We also find policies such as Normal/Misbehavior Dominated perform worse with more image predictions, since they use partial result (ignore confidence) and focus on local information.

5.6 Classifier Efficiency on Mobile Devices

Flasher detection is a computation intensive task. For example, using the CR algorithm, Chatroulette needed 30–40 servers running 24/7 to identify misbehaving users in a timely fashion. For mobile video chat, one option is sending all sensor data back to central servers for classification, but that incurs significant network traffic and delay as well as extra workload on the central servers. One key question that we want to address here is whether our multi-sensor flasher detection method is feasible for mobile devices.

We have implemented all our sensor feature extraction functionalities using the C language on Android phones and used JNI to call them. During our experiments, we ran our mobile video chat application and maintained an active video chat session in the foreground to emulate a practical execution scenario. In the background, we executed all feature extraction operations with the same frequency as our image sample rate (one snapshot image every 30 seconds) to mimic real-world conditions. We also built a lightweight resource measurement Android service that runs continuously in the background to monitor phone resource usage. We conducted experiments on two different types of mobile phones: 1) **HTC One :** an advanced quad-core 1.7 GHz Android phone with 2 GB of memory; and 2) **Galaxy Nexus:** a medium range dual-core 1.2

Table 5: Acquisition Time Comparison of Different Features

Feature	HTC One	Galaxy Nexus
Face	1.014s	2.373s
Eye	0.404s	1.665s
SIFT	0.226s	0.335s
Skin	0.033s	0.040s
Histogram	0.032s	0.040s
Audio	0.469s	1.673s
Acc. + Camera State	0.003s	0.002s
Total	2.181s	6.128s

GHz Android phone with 1 GB of memory, representative of many phones with similar capabilities in the market nowadays.

Table 5 shows the average acquisition time for extracting different features on mobile devices. Note that the feature acquisition time dominates the overall classification time since classifiers are pre-trained and takes minimum time to execute while features need to extracted at runtime. We notice that face detection is the most computationally intensive task, followed by audio and eye feature extraction. On the other hand, extracting features such as acc.+camera state, skin proportion, and histogram has only negligibly impact the overall running time on both dual and quad core phones. Overall, the fused classifier takes almost three times longer to run on a dual-core over a quad-core phone. The most important finding here is that our classifier runs reasonably efficiently on both phones (2–6 seconds). Further, we could not detect any noticeable impact on the quality of the video chat service that runs concurrently with the classifier. This shows that it is feasible to run our classifier on the phone itself, thus taking off significant burden from the servers. We also measure the energy usage of our fusion classifier. Compared with the large battery drain cost by the mobile video chat application, the energy used by our classifier is negligible.

5.7 Cascaded Classifier Performance

Given the execution time measurements obtained above for individual features, we can now evaluate to what extent cascaded fusion can help to further reduce the classifier's running time while maintaining certain accuracy requirements. We conduct this evaluation for both cascaded fusion scenarios: (1) image-based multi-sensor cascading which is based on a single snapshot image and its corresponding sensor data; and (2) session-based cascading which utilizes multiple images-based multi-sensor predictions in a session.

5.7.1 Image-based Multi-sensor Cascading Performance

Given the measured feature acquisition time t_i ($i \in [1, K]$) shown in Table 5, we can calculate T_0, which is the total amount of time needed per sample to go through all K stages. In our remaining evaluations, we use the running time measured on HTC One and $T_0 = 2.181s$ as the all-stage execution time per sample.

Our first experiment explores the influence of the confidence threshold σ on the performance of the cascaded classifier. We are especially interested in two special cases for optimizing the objective function: (i) the case where we seek an ordering of the cascade that maximizes efficiency or time saved ($\alpha = 0$), which we refer to as "best average saving time" $T_{save}(BAST)$; and (ii) the case where we seek a cascade ordering that maximizes accuracy ($\alpha = \infty$), which we refer to as "best final accuracy" P (BFA).

For all our experiments, we use the same confidence threshold for every stage. And as mentioned before, Weka generates a bidirectional confidence distribution for binary classification. For example, Figure 6 indicates that misbehaving prediction has confidence between 0 and 0.5 and the the lower the value, the more confident the prediction; while normal prediction has confidence values between 0.5 and 1 and the higher the value, the more confident the prediction. Because of this, we need two distinct confidence thresholds σ_1 and σ_2 for misbehaving and normal user classification respectively. Since the bidirectional confidence distribution is nearly symmetric, in our study, for simplicity we define a new factor named "confidence cutting threshold" ρ which is derived from the confidence threshold σ:

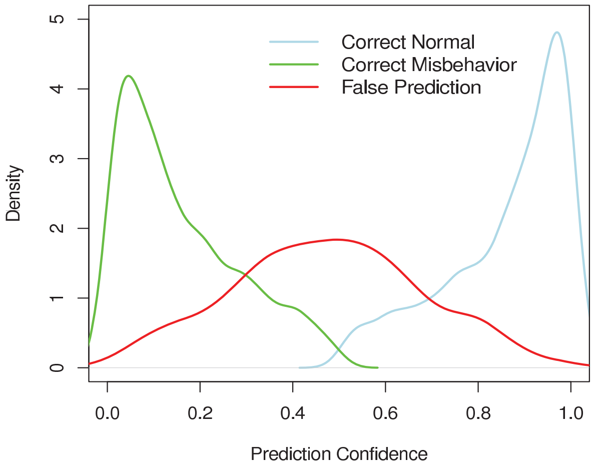


Figure 6: Bidirectional confidence distribution generated by Weka for binary classification.

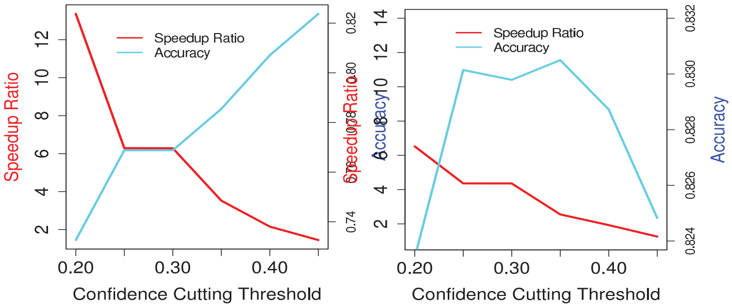


Figure 7: Optimal cascaded classifier performance by the (L) BAST criterion and by the (R) BFA criterion.

$$\rho \quad = \quad |\, \sigma - 0.5 \,| \tag{7}$$

Then ρ can somehow represent the value for both σ_1 for misbehaving users and σ_2 for normal users such that when $\rho = 0.3$, $\sigma_1 = 0.2$ and $\sigma_2 = 0.8$.

Figure 7 shows the results of our experiments. From the figure, we make several important observations:

- For the BAST criterion, increasing the confidence cutting threshold for each stage of the cascaded classifier results in an optimal cascaded classifier with lower saving time but higher accuracy. This is because a high confidence threshold adds a high stopping requirement at each stage, which makes the overall cascaded classifier more conservative and causes more samples to proceed deeper into the cascade's stages. While for the BFA criterion, there is a same trend found for speed up ratio but the accuracy goes down when confidence cutting threshold is very high ($\rho = 0.4$ and $\rho = 0.45$). This indicates for some samples, too many indicators might make negative contributions for the predictions.

Table 6: Structure and Performance of Optimal Cascaded Classifier for BAST and BFA when $\rho = 0.75$

Stage i	BAST		BFA	
	New Features	N_i	New Features	N_i
1	Skin	1810	Acc+Cam	1203
2	Acc+Cam	793	Sift	1203
3	Histogram	568	Skin	848
4	Audio	422	Histogram	651
5	Eye	345	Face	520
6	Face	297	Audio	393
7	Sift	0	Eye	0

- For the BAST criterion, with a reasonable confidence cutting threshold ($\rho = 0.25$ or $\rho = 0.3$), the optimal cascaded classifier can achieve a final accuracy of about 0.77 and meanwhile reduce average running time by a factor of 6.3. Table 6's columns 2 and 3 illustrate the optimal ordering of the stages and the flow of samples through this cascaded classifier, where again N_{i-1} means the number of samples passed into the i-th stage, and each run starts with a balanced data set of 2820 total samples from 705 sessions that are then processed through the cascaded classifier.
- For the BFA criterion, with the same threshold ($\rho = 0.25$ or $\rho = 0.3$), the optimal cascaded classifier can maintain even better performance (0.83 accuracy) than our fusion classifier while achieving a modest 4.4 time speedup in execution time. Table 6 columns 4 and 5 show the optimal ordering of stages to maximize accuracy, and the sample flow through the cascaded classifier.

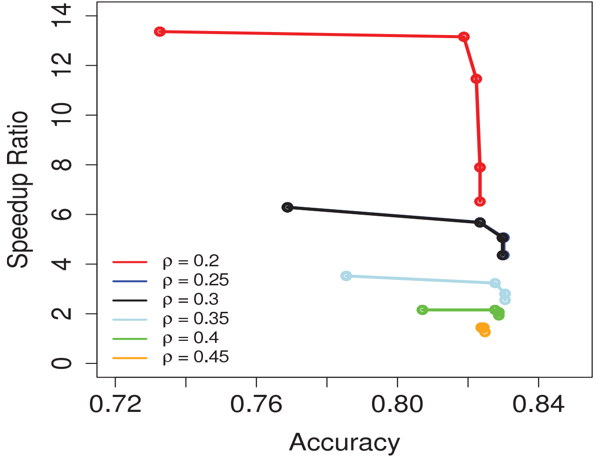


Figure 8: Optimal cascaded classifier performance on different control balance and confidence threshold.

Our second experiment seeks to understand the tradeoff between efficient and accurate classifications for a wider range of values α of the defined objective function F_{obj}, not just the two extrema of BAST and BFA. We let α range across the values {0 (BAST), 0.001, 0.01, 0.1, 1 (equal weight), 10, 100, 1000, ∞ (BFA)}. Figure 8 shows the tradeoff function between speedup ratio and accuracy for the optimal cascaded classifier across a range of α, and also shows different curves that correspond to different confidence cutting thresholds ρ. Based on this figure, we make the following observations:

- Each function (ρ constant) exhibits a similar shape wherein the low α values are clustered together on the upper left, followed by an inflection point farther down and right that represents the balance point $\alpha = 1$, followed by a clustering of the high α values even further down and to the right.
- Overall, execution time speedups range from a factor of 1.25 to 13.36 times while the final accuracy correspondingly decreases from 0.829 to 0.732.
- Similar to our previous observation for just the two extrema, fixing α across our large range still results in the trend wherein a lower confidence threshold ρ generates an optimal cascaded classifier with a higher factor of time savings and generally a lower final accuracy only except when ρ is pretty high (0.95).

Table 7: Session-based Cascaded Classifier Performance

Cascade start image id	1	2	3	4
Accuracy	0.848	0.854	0.856	0.859
Aver Image Used for Prediction	1.29	2.17	3.11	4
Speedup Ratio	3.10	1.84	1.29	1

Figure 8 helps to quantify the general tradeoff between efficiency and accuracy for our cascaded classifier on different requirements. If we wish to achieve a certain efficiency speedup target, then we can assess by precisely how much the accuracy will be sacrificed. Whereas, if we wish to boost the accuracy to a given level (higher ρ), then we will be able to determine the amount of reduction in the speedup factor of the execution time. For example, if we desire to push the edge on speed, then we can achieve an approximately 13X speed gain, with nearly the same accuracy at about 0.819. If our goal is to push the edge on accuracy, then we can obtain a better accuracy of 0.831 - the best that even the non-cascaded multi-sensor fusion classifier wasn't able to achieve - at the cost of earning only a 3X speed gain.

5.7.2 *Session-based Cascading Performance*

For our session-based classifier, features are derived from the prediction results generated by our image-based multi-sensor fusion classifier. Since the image sequence is fixed for a session, a full stage (4 stages in our study) session-based cascaded classifier have only one possible ordering. So in our evaluation, we focus on: 1) what performance our full stage cascaded classifier could achieve; 2) how accuracy and efficiency change when the cascade starts at later predictions namely a partial stage cascaded classifier.

Table 7 indicates with a full stage cascaded session-based classifier, we can achieve more than 3X speed gain at the cost of only 1.1% accuracy reduction. Also it indicates by waiting a little longer to start the cascaded prediction, classification can reach nearly the same accuracy as non-cascaded classifier at the cost of executing very small amount of image predictions. For example, if we decide to start the cascaded classifier after the second image-based prediction is generated, the overall session-based classifier can reach 0.854 accuracy by only requiring 2.17 image predictions which could achieve more than 2X speed gain.

5.7.3 *Multi-level Cascading Performance*

Finally, we combine our image-based multi-sensor fusion cascaded classifier with session-based cascaded classifier together, building a two-level cascaded misbehavior classification. In image-based fusion cascaded classifier, we choose $\rho = 0.3$ and $\alpha = 1$ and the classifier achieves 13.15X speed gain with accuracy equal to 0.819. In session-based cascaded classifier, the cascade classifier chooses to start once the second prediction is generated. In total, our multi-level cascaded classifier reaches 0.843 accuracy while taking only

0.408 seconds for a session prediction which achieves a significantly 21X speed up compared with non-cascaded session-based classification (which takes $2.181 * 4 = 8.724$ seconds).

6. CONCLUSIONS

This paper presents a multi-modal fusion framework for accurate and efficient flasher detection in a mobile video chat application. First, we show that traditional face-centric image-based classification developed for online video chat users achieves only 0.63 accuracy when applied to a balanced real world data set of normal and misbehaving mobile video chat users. We further show that our enhanced image classifier improves overall accuracy to 0.69. Second, we demonstrate that an image-based multi-sensor fusion classifier that integrates mobile accelerometer data along with front/back mobile camera position and audio category can substantially improve the overall accuracy to 0.82. Third, we explore the temporal modality and by leveraging four image-based predictions within a session, our session-based classifier achieves a further improvement to 0.86 accuracy. Fourth, we demonstrate the feasibility of running the fused multi-modal misbehavior classifier on mobile devices, and then design and evaluate multi-level cascaded classifier to quantify the tradeoff between efficiency and accuracy. We show that with a certain configuration, our classifier could achieve 21X speedup gain with a reasonable 0.84 accuracy.

7. ACKNOWLEDGMENTS

We would like to acknowledge the support of NSF grants CSR-1162614 and CIC-1048298. We would like to thank Leif K-Brooks, the CEO of Omegle, for helping to enable our MVChat client to be compliant with Omegle's server, and in providing data for our research. We would also like to thank Xinyu Xing, Hanqiang Chen and Yuli Liang for providing their state-of-art CR algorithm and Chatroulette for sharing the performance of CR algorithm on their system. We also sincerely thank the anonymous reviewers for their comments and suggestions.

8. REFERENCES

[1] Ames, M. G., Go, J., Kaye, J. J., and Spasojevic, M. Making love in the network closet: the benefits and work of family videochat. In *Proc. of the 2010 ACM conf. on Computer supported cooperative work*, CSCW '10 (2010).

[2] Breiman, L. Bagging predictors. *Mach. Learn. 24*, 2 (1996).

[3] Breiman, L. Random forests. *Mach. Learn. 45*, 1 (2001).

[4] Buhler, T., Neustaedter, C., and Hillman, S. How and why teenagers use video chat. In *Proc. of the 2013 conf. on Computer supported cooperative work*, CSCW '13 (2013).

[5] Chai, D., and Ngan, K. N. Face segmentation using skin-color map in videophone applications. *IEEE Trans. Cir. and Sys. for Video Technol. 9*, 4 (1999).

[6] Chatroulette web site. http://www.chatroulette.com/.

[7] Cheng, H., Liang, Y.-L., Xing, X., Liu, X., Han, R., Lv, Q., and Mishra, S. Efficient misbehaving user detection in online video chat services. In *WSDM '12* (2012).

[8] Inkpen, K., Du, H., Roseway, A., Hoff, A., and Johns, P. Video kids: augmenting close friendships with asynchronous video conversations in videopal. CHI '12 (2012).

[9] Jana, S., Pande, A., Chan, A., and Mohapatra, P. Mobile video chat: Issues and challenges. *IEEE Communications Magazine 51*, 6 (2013).

[10] Li, D., Sethi, I. K., Dimitrova, N., and McGee, T. Classification of general audio data for content-based retrieval. *Pattern Recognition Letters 22* (2001).

[11] Lopes, A. P. B., Avila, R. E. F. D., Peixoto, A. N. A., Oliveira, R. S., and Araújo, A. D. A. A bag-of-features approach based on hue-sift descriptor for nude detection. In *Proc. of the XVII European Signal Processing Conf.* (2009).

[12] Lu, H., Pan, W., Lane, N. D., Choudhury, T., and Campbell, A. T. Soundsense: Scalable sound sensing for people-centric sensing applications on mobile phones. In *Proc. of 7th Intl. ACM Conf. on Mobile Systems, Applications, and Services*, MobiSys '09 (2009).

[13] MeetMe web site. http://www.meetme.com/.

[14] Milliken, M., ODonnell, S., Gibson, K., and Daniels, B. Older adults and video communications: A case study. *The Journal of Community Informatics 8*, 1 (2012).

[15] Naive bayes classifier wikipage. http://en.wikipedia.org/wiki/Naive_Bayes_classifier.

[16] Neeraj Bhargava, Girja Sharma, R. B. M. M. Decision tree analysis on J48 algorithm for data mining. *Intl. Journal of Advanced Research in Computer Science and Software Engineering 3*, 6 (2013).

[17] Oliver, N., and Horvitz, E. Selective perception policies for guiding sensing and computation in multimodal systems: A comparative analysis. In *Proc. of the 5th Intl. Conf. on Multimodal Interfaces*, ICMI '03 (2003).

[18] Oliver, N., and Horvitz, E. S-seer: Selective perception in a multimodal office activity recognition system. In *Machine Learning for Multimodal Interaction*. Springer, 2005.

[19] Oliver, N., Horvitz, E., and Garg, A. Layered representations for human activity recognition. In *Proc. of the 4th IEEE Intl. Conf. on Multimodal Interfaces*, ICMI '02 (2002).

[20] Omegle web site. http://www.omegle.com/.

[21] Raffle, H., Revelle, G., Mori, K., Ballagas, R., Buza, K., Horii, H., Kaye, J., Cook, K., Freed, N., Go, J., and Spasojevic, M. Hello, is grandma there? let's read! storyvisit: family video chat and connected e-books. CHI '11 (2011).

[22] Study: 37% of U.S. teens now use video chat, 27% upload videos. http://techcrunch.com/2012/05/03/study-37-of-u-s-teens-now-use-video-chat-27-upload-videos/.

[23] Tian, L., Li, S., Ahn, J., Chu, D., Han, R., Lv, Q., and Mishra, S. Understanding user behavior at scale in a mobile video chat application. In *Proc. of the 2013 ACM Intl. Joint Conf. on Pervasive and Ubiquitous Computing* (2013).

[24] Tina R. Patil, M. S. S. S. Performance analysis of naive bayes and J48 classification algorithm for data classification. *Intl. Journal of Computer Science and Applications 6*, 2 (2013).

[25] Weka website. http://www.cs.waikato.ac.nz/ml/weka/.

[26] Xing, X., Liang, Y.-L., Cheng, H., Dang, J., Huang, S., Han, R., Liu, X., Lv, Q., and Mishra, S. Safevchat: Detecting obscene content and misbehaving users in online video chat services. In *Proc. of the 20th intl. conf. on World Wide Web*, WWW '11 (2011).

[27] Xing, X., Liang, Y.-l., Huang, S., Cheng, H., Han, R., Lv, Q., Liu, X., Mishra, S., and Zhu, Y. Scalable misbehavior detection in online video chat services. In *Proc. of the 18th ACM SIGKDD conf. on Knowledge discoery and data mining*, KDD '12 (2012).

[28] Zhang, Z., Chu, D., Chen, X., and Moscibroda, T. Swordfight: Enabling a new class of phone-to-phone action games on commodity phones. In *Proc. of the 10th Intl. Conf. on Mobile Systems, Applications, and Services* (2012).

[29] Zhu, J., Rosset, S., Zou, H., and Hastie, T. Multi-class adaboost. Tech. rep., 2005.