Efficient Misbehaving User Detection in Online Video Chat Services

Hanqiang Cheng[1], Yu-Li Liang[2], Xinyu Xing[3], Xue Liu[1],Richard Han[2], Qin Lv[2], Shivakant Mishra[2]
[1]School of Computer Science, McGill University
[2]Department of Computer Science, University of Colorado at Boulder
[3]School of Computer Science, Georgia Institute of Technology
hanqiang.cheng@mail.mcgill.ca, yu-li.liang@colorado.edu, xxing8@gatech.edu
xueliu@cs.mcgill.ca, {rick.han, qin.lv, Shivakaht.Mishra}@colorado.edu

ABSTRACT

Online video chat services, such as Chatroulette, Omegle, and vChatter are becoming increasingly popular and have attracted millions of users. One critical problem encountered in such applications is the presence of misbehaving users ("flashers") and obscene content. Automatically filtering out obscene content from these systems in an *efficient* manner poses a difficult challenge. This paper presents a novel Fine-Grained Cascaded (FGC) classification solution that significantly speeds up the compute-intensive process of classifying misbehaving users by dividing image feature extraction into multiple stages and filtering out easily classified images in earlier stages, thus saving unnecessary computation costs of feature extraction in later stages. Our work is further enhanced by integrating new webcam-related contextual information (illumination and color) into the classification process, and a 2-stage soft margin SVM algorithm for combining multiple features. Evaluation results using real-world data set obtained from Chatroulette show that the proposed *FGC* based classification solution significantly outperforms state-of-the-art techniques.

Categories and Subject Descriptors

K.4.1 [**Computers and Society**]: Abuse and crime involving computers

General Terms

Algorithms, Performance, Design, Experimentation

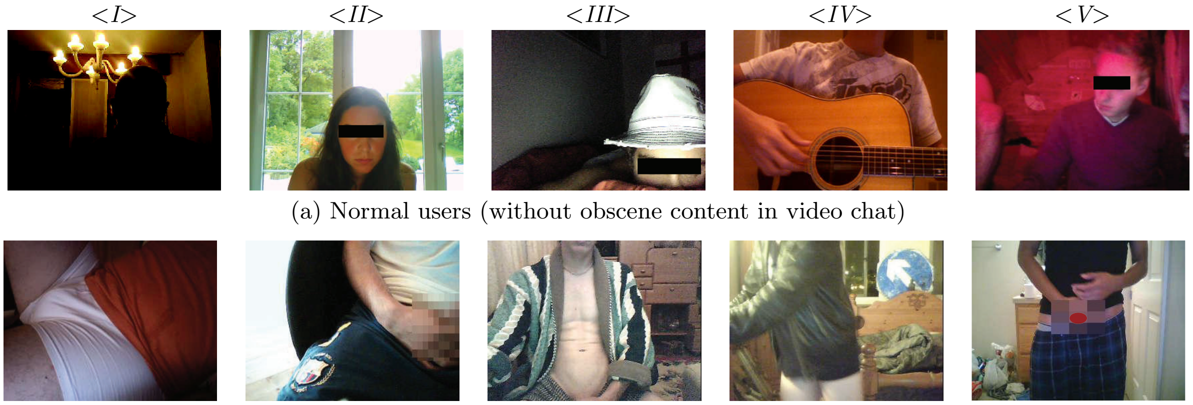
1. INTRODUCTION

Online video chatting has grown rapidly in popularity in recent years. The number of websites that provide an environment for users to chat online by video has expanded substantially since 2010, with websites such as Chatroulette [2], Myyearbook [22], Omegle [24], TinyChat [28], etc. experiencing aggressive membership growth. In an online video chat service, strangers from around the world are randomly paired together for webcam-based conversations. Visitors who are looking for entertainment go to the website and randomly begin a conversation (via video, audio, and text) with another visitor. Such websites are typically offered for free, and are easy to use, which enhances their popularity and implies a great market for advertisement. Online video chatting's appeal is based on providing a real-time video environment for a rich interactive experience via face-to-face conversations, gestures, and manners that extend beyond what mere text or audio chat can offer.

However, the presence of misbehaving users ("flashers" who display obscene content) is a serious problem in these video chat services. Figure 1 shows some snapshot image samples that we have observed in one of the video chat websites (Chatroulette). We could see that misbehaving users display obscene content, such as genitalia or exhibit inappropriate behavior, while normal users do not. To make the problem even worse, a significant fraction of video chat participants are underage minors, whose exposure to such obscene content may cause legal problems. An instance is iChatr, an iphone application similar to Chatroulette that randomly pairs iphone users for video chat. iChatr was excluded from Apple's application store due to the transmission of obscene content.

In this paper, our goal is to investigate mechanisms that can *efficiently* detect obscene content in online video chat systems with high accuracy. We have experimented with a variety of approaches based on prior work to address the problem of misbehaving user detection, and found them to be insufficient. Prior work is ill-suited to the demanding real-world classification scenario posed by video chat streams. For example, the statistical skin-color model approach for pornography detection [12] does not work well with online video chat data, due to diverse illumination conditions. SafeVchat [33] is the first software that achieves reasonable performance for detecting obscene content in online video chat systems and is deployed in the world-largest online video chat website - Chatroulette. However, the software requires a large amount of computation resources and involves high computation latency, which makes some online video chat websites such as Omegle refuse its deployment. As a result, there is a strong motivation for us to improve its efficiency. In addition to these two approaches above, we further investigate the technique which uses Scale-Invariant Feature Transform (SIFT [20]) along with Bag-of-Visual-Words (BoVW [3, 6]) framework because SIFT is broadly used for pornographic content detection [5, 19] and achieve acceptable detection accuracy. Unfortunately, SIFT based solution is still not the answer for misbehaving user detec-

WSDM'12 February 8-12, 2012, Seattle, Washington, USA.



(a) Normal users (without obscene content in video chat)

(b) Misbehaving users (with obscene content in video chat)

Figure 1: Snapshot image samples from online video chat systems.

tion in online video chat systems. The detailed analysis and investigation of these techniques are presented in Section 2.

To detect misbehaving users in online video chat systems, this paper proposes a solution which (1) introduces a novel Fine-Grained Cascaded (FGC) classification which systematically generates the optimal combination of partial image features for classification and thus significantly improve classification efficiency without sacrificing classification precision/recall; and (2) achieves better classification performance in terms of precision and recall.

This paper makes the following contributions:

- We propose a novel *Fine-Grained Cascaded* (FGC) classification approach to significantly improve classification efficiency without sacrificing classification precision and recall. The FGC classification approach systematically evaluates the classification capacities of different features and automatically generates the optimal feature extraction order to allow for earlier classification with partial features.

- We identify two new contextual features which are significant for discriminating misbehaving users from normal ones and incorporate these new features into an enhanced combination algorithm for misbehaving user classification.

- We present key observations from over a half-year study of users on the Chatroulette system, the world's largest online video chat system. These long-term key observations inspire the design of our FGC classification approach.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 introduces the properties of online video chat systems, real-world experimental data, and the classification workflow of our proposed solution. Section 4 describes how our solution achieves high classification precision and recall by integrating two novel contextual features using a 2-stage soft margin SVM algorithm. Section 5 presents our innovative "Fine-Grained Cascaded (FGC)" classification approach for improved efficiency. Section 6 evaluates the precision, recall, and latency of our solution. Finally, Section 7 concludes the paper.

2. RELATED WORK

Over the past decade, a number of techniques have been introduced for detecting objectionable content, especially in the context of pornographic images. Recent work leverages different types of information such as text, image, and URL [9, 18]. In online video chat systems, only a small number of users use text for chatting, so detecting misbehaving users via text analysis is not effective. Furthermore, the large scale of video chat systems makes real-time video analysis infeasible. Any practical solution including ours for detecting misbehaving users needs to focus on individual snapshot images of each chatter. Therefore, we focus on image-based objectionable content filtering techniques, and summarize the ones that are most relevant to our proposed solution and their limitations under video chat scenarios.

For pornographic image detection, skin color based detection is the most widely used technique and achieves acceptable performance in this scenario [12, 7, 34, 16]. The basic idea of skin color based detection is to identify skin exposure regions in images based on a statistical color model that is trained from manually labeled data. Area size and shape of detected pixels are sometimes considered to further improve the skin color detection results. However, approaches based on statistical skin-color models, such as PicBlock [1], have been shown to be insufficient for identifying obscene content in video chat systems, due to the diverse quality and content of snapshot images captured from online video chat systems [33], A recent survey [13] concludes that skin detection methods may only be used as a preprocessor for obscene content detection, and other content types such as text [10] and motion analysis [11] may be incorporated to improve accuracy.

SafeVchat [33], an obscene content detection system we have recently developed for online video chat systems, is the first solution that addresses this difficult problem. It has been successfully deployed in the Chatroulette online video chat system. Unfortunately, this solution is highly compute-intensive, requiring over a hundred servers running continuously 24 × 7 to filter out misbehaving users. Moreover, the system's ability to classify misbehaving users is still weak. As a result, the software must be backed up by a second round of expensive human screeners. Together, these limitations make the SafeVchat solution too expensive at present for most online video chat services. For example, Omegle is not able to implement our solution because it is too heavyweight and costly for them. Thus, there is a strong need to improve upon the state of the art.

We further investigate Scale Invariant Feature Transform (SIFT) descriptors [20] along with the Bag-of-Visual-Words (BoVW) framework [3, 6], which was shown to work well in two recently-proposed pornography detectors [5, 19]. However, the classification performance of these two detectors is fairly poor when using them in the context of online video chat systems. There are several reasons behind the poor performance. First, though SIFT descriptors are keypoint descriptors that are good at describing salient regions, the salient regions of our snapshot images are not obvious in the context of online video chat systems. Second, SIFT descriptors are sparse and not uniformly distributed, which causes only few keypoints are extracted. Finally, the problem we face here is more difficult than the pornography classification problem due to the smaller inter-class distance between different categories. For example, the "difference", or visual distance, between pornography categories (e.g. fully-clothed and nude body trunk) is large. However, for our problem, the difference between misbehaving and normal users is not that clear. Figure 1 shows some snapshot image samples in our data set with obscene parts being blurred. As we can observe, the characteristics of misbehaving users are quite similar to those of normal users. For example, both misbehaving and normal users could be dressed, show their faces, and expose large skin area (normal users show their hands and upper chest while misbehaving users show their genitalia).

3. OVERVIEW

In this section, we start with an overview of online video chat systems and our real-world experimental data sets. We then present our key observations and the proposed Fine-Grained Cascaded (FGC) classification workflow.

3.1 Online Video Chat Systems and Experimental Data Sets

To support online video chat among a large number of users, all online video chat systems are designed to use a peer-to-peer architecture. As a result, it is infeasible to obtain continuous video chat streams between any two peers for analysis at a central server. To detect misbehaving users, online video chat systems utilize an HTTP polling method to obtain periodic snapshot images from users' video chat streams. For example, in Chatroulette, users' snapshots are captured every 30 seconds and forwarded to a central server. Such periodic snapshot images are thus what we can use for classifying misbehaving users.

To support our study, Andrey Ternovskiy, the founder of Chatroulette, has kindly provided us with these otherwise unobtainable internal data traces containing 20,000 users' snapshot images (with the resolution of 320 × 240 pixels) as well as Chatroulette computation resources. We conducted our experiments on the Chatroulette platform using these real-world data traces, in which 15% of users are misbehaving and the rest are normal. We have divided these data traces evenly into five sets – three sets for training, one for evaluation, and one for testing. To reduce variability, multiple rounds of cross-validation over the 20,000 users' data traces are performed in our experiments.

3.2 Key Observations

In conjunction with Chatroulette's human moderation team, we conducted a half-year study of the world's largest online video chat system – Chatroulette. Through this long-term study, we have made a number of key observations, which significantly inspired our solution for classifying misbehaving users in online video chat systems. We summarize our key observations as follows.

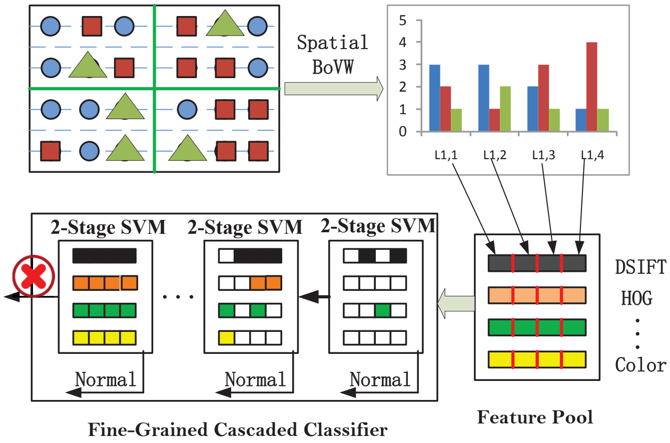


Figure 2: Fine-Grained Cascaded (FGC) classification workflow for efficient misbehavior detection.

(1) The fundamental observation from online video chat systems is that misbehaving users expose themselves in front of webcams while normal users are fully-clothed. In addition, the Chatroulette moderation team observed that normal users on their system usually maintain a stable posture while misbehaving users do not.

(2) Misbehaving users usually attempt to point their webcams downwards, while normal users typically point their webcams upwards. The reason is that webcams usually have a very narrow field of vision and users cannot present their full body in their video. This motivates us to identify the orientation of users' webcams and classify a user to be normal (misbehaving) if his/her webcam points upward (downward). The webcam orientation usually correlates strongly with the illumination context of a snapshot image. For normal users, a bright illumination area usually appears either on the top (from the ceiling light) or the side area (from open windows) of the snapshot image when users point their webcams upwards (e.g., <I> in Figure 1(a)). These bright illumination areas can be used to infer that the webcam is oriented upwards and thus the users are normal. In contrast, we observed that some misbehaving users typically stay in a dark environment but present themselves in a centered bright area (e.g., <I> in Figure 1(b)).

(3) Misbehaving users generally do not expose themselves in an outdoor environment. The color characteristics of an outdoor scene is significantly different from that of an indoor scene. As shown in <II> of Figure 1(a), an outdoor scene usually includes green trees – green area in the top of a snapshot image.

(4) Another surprising observation illustrates that users in a room with dark red lighting (e.g., in a bar) are unlikely to be misbehaving users (<V> in Figure 1(a)). This is because misbehaving users usually expose themselves in their bedrooms, and bedrooms typically have white or yellow lighting instead of dark red lighting.

3.3 Classification Workflow

Figure 2 illustrates the workflow of our Fine-Grained Cascaded (FGC) classification solution for efficient detection of misbehaving users. It consists of four key steps.

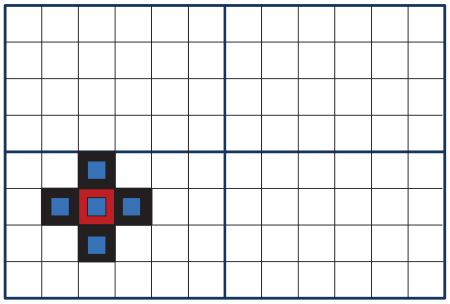


Figure 3: Illumination feature extraction. The dense grid is also used for the calculation of other descriptors.

Step 1 (*Upper Left*): Each user's snapshot image is first partitioned into several spatial regions ($2 \times 2 = 4$ regions in the example). We then use the Bag-of-Visual-Words framework [3, 6] to extract visual words (e.g., $\circ$, $\triangle$, and $\square$) in each region for each feature descriptor. We also propose two new contextual descriptors (illumination and color) for better classification accuracy (Section 4).

Step 2 (*Upper Right*): For each region, we generate one histogram based on the frequencies of different visual words.

Step 3 (*Lower Right*): Each histogram is a "*regional feature*" for a specific descriptor. If we use four descriptors to describe a snapshot image with four regions, then the snapshot image has $4 \times 4 = 16$ *regional features* (histograms).

Step 4 (*Lower Left*): The proposed Fine-Grained Cascaded (FGC) classifier is utilized to distinguish misbehaving users from normal ones. FGC selects multiple sets of appropriate *regional features*, combines features in each set using a 2-stage soft margin SVM algorithm (Section 4), and computes the optimal ordering of classification stages (each stage uses one set of combined features) in order to achieve high quality and high efficiency for classification (Section 5).

4. BOOSTING CLASSIFICATION PERFORMANCE

We first focus on how to boost classification performance, i.e., classify misbehaving users with high precision and high recall. We propose two new contextual feature descriptors based on illumination and color, and a 2-stage soft margin SVM algorithm to combine these features. The efficiency and effectiveness of the feature descriptors as well as our combination algorithm are validated in Section 6.

4.1 Feature Descriptors

A significant number of feature descriptors used in computer vision and image processing have been introduced for the purpose of object detection. Based on our observations in Section 3.2, we select the following three feature descriptors for classifying misbehaving users.

1. Dense SIFT (DSIFT). Dense SIFT [29] is one of the most effective descriptors used for a wide variety of object/scene classification applications. Compared with traditional SIFT [20], it is more efficient and can achieve higher classification performance.

2. Histogram of oriented gradient (HOG). Based on our observations described in Section 3.2, normal chatters habitually maintain a stable posture (i.e., sitting in front of a webcam). We choose HOG since it has been used successfully for capturing stable posture or shape [4].

3. Local Binary Patterns (LBP). The fundamental difference between misbehaving and normal users is that normal users are generally fully-clothed, while misbehaving ones are not. The texture between large skin exposure and clothes is discriminative. Therefore LBP is chosen to capture texture information.

In addition to these three feature descriptors, we introduce two new contextual descriptors that can obviously boost the classification performance. These two feature descriptors represent illumination context and color context.

4.1.1 Illumination Contextual Descriptor

Our observations in Section 3.2 illustrate that the orientation of users' webcams can be used to discriminate normal users from misbehaving ones, and the webcam orientation usually correlates strongly with the illumination context of a snapshot image. Therefore, we propose a new illumination contextual descriptor that can be used for discriminating normal users from misbehaving ones.

To reduce noise in a snapshot image caused by low-quality webcams, we first smooth the snapshot image by averaging its value with the neighboring pixels (i.e., on a 3×3 patch). After smoothing, we compute illumination contextual descriptors on a dense grid of uniformly spaced cells. Each illumination contextual descriptor is defined as the brightest light intensity value on a smaller patch (shown as blue patches in Figure 3) in a grid cell. Mathematically, the illumination descriptor on patch K (Figure 3: blue patch in the red cell) $I(K)$ is computed as follows:

$$I(K) = \alpha \cdot I_0(K) + (1-\alpha) \cdot \frac{I_{neighbor}(K)}{4} \tag{1}$$

$$I_0(K) = \max(\max(v_R(K)), \max(v_G(K)), \max(v_B(K))) \tag{2}$$

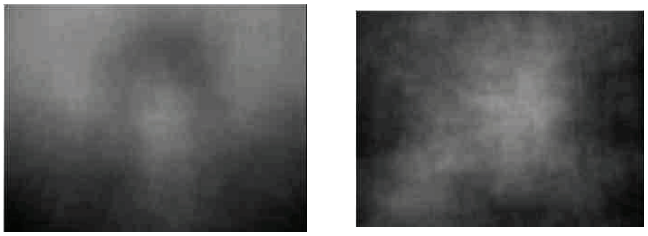
$$I_{neighbor}(K) = \sum_{i \in \{Up, Down, Left, Right\}} I_0(i) \tag{3}$$

where α is a scaling factor for further noise reduction, and $v_R(K)$, $v_G(K)$, $v_B(K)$ denote the R,G,B values on patch K, respectively. $I_{neighbor}(K)$ represents the sum of the four neighbor descriptors' I_0 (Figure 3: blue patches in black cells). Figure 4(a) and Figure 4(b) show the average illumination contextual descriptors of normal users' snapshots and that of misbehaving users' snapshots, respectively [1]. It is obvious that our illumination contextual descriptors clearly capture the illumination difference between normal users and misbehaving users, which is also consistent with our observation in Section 3.2.

4.1.2 Color Contextual Descriptor

Since the observations in Section 3.2 also indicate that the color presented in a snapshot image can be harnessed to identify normal and misbehaving users, we propose a color contextual descriptor. Similar to the illumination contextual descriptor, we sample color descriptors on a dense grid of uniformly spaced cells. Since illumination intensity can significantly affect the color appearance, we only utilize Hue

[1] The illumination values have been scaled to illustrate the contrast of the illumination contextual descriptors.



(a) Normal users (b) Misbehaving users

Figure 4: Comparison of average illumination contextual descriptors of normal and misbehaving users.

and Saturation in the HSV color space. Each color descriptor therefore is represented as a two-element vector (hue and saturation). The color contextual descriptors are further quantized (mapped) to the most similar major colors, which are generated using k-means clustering algorithm.

4.1.3 Usage of Feature Descriptors

Bag-of-Visual-Words framework is used to represent each type of descriptors [3, 6]. Since the spatial information is useful for boosting classification performance, a 3-level Spatial Pyramid Matching [15] is used to process the feature descriptors above[2].

A 3-level Spatial Pyramid is composed of level-0, level-1, and level-2. In each level, a given image is evenly partitioned into several blocks. The number of blocks for each level is: level-0: (1 × 1 = 1 block), level-1: (2 × 2 = 4 blocks), and level-2: (4 × 4 = 16 blocks). In Figure 2, the upper left diagram shows an image with level-1 partition (2 × 2 = 4 blocks). For each block, one histogram of visual word frequency is created. For example, the upper right diagram in Figure 2 describes four histograms for DSIFT descriptor. For a 3-level spatial pyramid, there are totally 21 histograms from all 3 levels (i.e., 1(level-0)+4(level-1)+16(level-2)= 21). Therefore, for an image which is partitioned using a 3-level spatial pyramid, each type of feature descriptor can be projected to 21 histograms.

4.2 Feature Combination

Given the five different types of features described above, our goal is to combine multiple features and achieve better performance for the classification of misbehaving users. This is accomplished through a 2-stage soft margin SVM classifier, which is inspired by a recently proposed multiple feature combination algorithm, Linear Programming Boosting algorithm (LPBoost) [8] .

LPBoost was originally designed for very high dimensional heterogeneous visual feature combination and has demonstrated better performance than multiple kernel learning (MKL) [8]. Since *norm-1* constrained variables tend to have a sparse optimal solution [27], LPBoost searches for a sparse optimal set of combination coefficients, which improves the interpretability of features for a classification problem. However, when applying LPBoost to combine both weak and much stronger features (e.g., DSIFT is much stronger in misbehaving users detection than other four features), the combination coefficients of the strong feature(s) can carry much larger weights than that of the weak feature(s), thus limiting the contributions of the weak features in terms of boosting the classification performance. To address this problem, we extend LPBoost to *norm-2* and reformulate LPBoost with ℓ_2-norm constraint on combination coefficients [3], which is inspired by l_p-norm multiple kernel learning [14]. In this paper, we refer to the original LPBoost and the 2-stage soft margin SVM as LPBoost-ℓ_1 and 2-Stage SVM, respectively.

Similar to LPBoost-ℓ_1, our 2-stage SVM also contains two separate steps. In the first step, classifier f_m for each type of feature is trained in a labeled sample set $D = \{(x_{i,m}, y_i)\}_{i=1,2,...,N}$ (where $x_{i,m}$ is the m^{th} type of feature for snapshot i and y_i is the corresponding label – misbehaving or normal user). Subsequently we optimize over combination coefficients $\beta = \{\beta_1, \beta_2, ..., \beta_F\}$ as follows:

$$\min_{\beta,\xi} \sum_{m=1}^{F} \beta_m^2 + \frac{1}{\nu N} \sum_{i=1}^{N} \xi_i \tag{4}$$

$$sb.t. \ y_i(\sum_{i=1}^{F} \beta_m f_m(x_i) + b) + \xi_i \geq 1, \quad i = 1, 2, ..., N \tag{5}$$

$$\|\beta\|_2^2 \leq 1, \quad \beta_m \geq 0, \qquad \xi_i \geq 0, \tag{6}$$

with $\xi = \{\xi_1, \xi_2, ..., \xi_N\}$ being slack variables. Here, ν denotes the parameter that controls the tradeoff between the margin of the boosting classifier and misclassification penalty, and $f_m(x_i)$ represents the output of the m^{th} individual classifier for the i^{th} training sample. Different from LPBoost-ℓ_1 which employs regularizers of the form $\|\beta\|_1$ to promote sparse combination, our 2-stage SVM uses smooth convex regularizers of the form $\|\beta\|_2$, allowing for non-sparse solutions.

5. REDUCING CLASSIFICATION LATENCY

We now focus on the problem of reducing the classification latency, which is particularly important given the large scale of online video chat services. To answer this question, we revisit and analyze the classification technique that we proposed in Section 4. Based on our analysis, we propose a Fine-Grained Cascaded (FGC) classification scheme to reduce the computation latency that our misbehaving user classification technique involves.

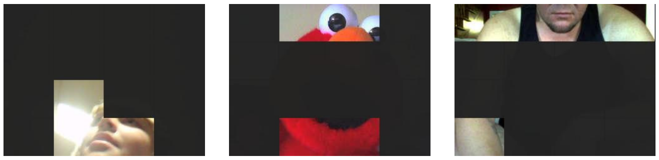
5.1 Classification Analysis

In general, there are three steps for classifying images using the Bag-of-Visual-Words framework: 1) descriptor extraction, 2) descriptor projection, and 3) feature classification. In the first step, certain types of descriptors (e.g., DSIFT, HOG, etc.) are computed from an image. In the second step, the Bag-of-Visual-Words framework projects the extracted descriptors to corresponding visual words which are further used to generate corresponding features. Based on the features, the final step uses a classification algorithm to classify images into different classes, in our case, misbehaving or normal users.

A number of research efforts have been made to accelerate the procedure above. Vedaldi and Zisserman introduce an explicit feature map approximating non-linear kernels (e.g., intersection kernel and chi kernel) by a linear kernel [32], which reduces the feature classification latency for those widely used kernel based classification algorithms (such as SVM and LPBoost). Furthermore, the feature map generally does not have negative impacts on classification performance in terms of precision and recall. Another pioneering

[2] [15] indicates the L-level spatial pyramid can achieve good classification performance when $L = 3$.

[3] *norm-2* is a quadratic optimization problem, which is easy to solve.



(a) (b) (c)

Figure 5: Using partial blocks of snapshots to discriminate normal users from misbehaving ones.

work uses a vocabulary tree to reduce the computation latency for descriptor projection [25]. Uijlings et al. designed fast SIFT descriptors, which significantly reduces the computation latency of widely used SIFT descriptors [29]. All these approaches are incorporated in our solution introduced in Section 4, and this new implementation reduces the computation latency by three times over previous approaches such as SafeVchat.

A closer analysis reveals that there is a substantial opportunity to further reduce the latency in the descriptor extraction step. In comparison with descriptor projection and feature classification, we observe that descriptor extraction is usually the most compute-intensive step and involves the highest computation latency. For example, gray DSIFT descriptor extraction takes 0.06 seconds (see Table 1), while the corresponding descriptor projection and classification latency are 0.01 and 0.004 seconds, respectively. Computation latency of classification in general is one magnitude smaller than that of feature extraction. Therefore, the main goal of our work is to reduce the computation latency in the process of descriptor extraction. While Uijlings et al. introduced two approaches to reduce the computation latency for SIFT descriptor extraction [29], these two approaches cannot be used for other types of descriptors (e.g., HOG). Furthermore, these approaches can sacrifice classification precision and recall.

5.2 Fine-Grained Cascaded Classification

Recall our classification approach where the classification process uses 4×21 *regional features* that are generated from 4 descriptors (DSIFT, HOG, color and illumination) in 21 regions defined in a 3-level spatial pyramid (see Section 4.1.3). Note our experiment in Section 6 indicates LBP descriptor is not necessary for our classification. Inspired by the *“image guessing game“* where a player guesses the content of images that are partially blocked, we optimize our classification approach and thus reduce computation latency. Figure 5, for example, shows some snapshot samples where only partial blocks are revealed. Reviewing these partially revealed blocks, one can easily label Figure 5(a) and 5(b) as “normal” with high confidence but may not be able to determine whether obscene content is presented in Figure 5(c). Therefore, more blocks on the third snapshot are revealed in the next-round of image guessing. The guessing game continues until all the images are labeled or all the images are completely revealed. Imagine extending this game in the procedure of misbehaving user classification. Since there are partial snapshot images classified by our classification approach in each round of “guessing”, descriptor computation for these snapshot images is not intensive because descriptors are only extracted from the revealed blocks of these images. Given a snapshot image, the computation for a specific descriptor gradually increases with the rounds of the “guessing” game, because more blocks on the snapshot image are revealed if the image cannot be classified in the previous rounds of “guessing”. According to the characteristics of the guessing game, we name our misbehaving user classification as Fine-Grained Cascaded (FGC) classification.

Our FGC classification has to satisfy two criteria – (1) minimizing the computation latency for descriptor extraction; and (2) maintaining high classification performance in terms of precision and recall. To minimize computation latency, we have to determine which blocks of a snapshot image should be revealed first (i.e., determining which *regional features* should be selected first for misbehaving user classification). This is because different regional features may involve different computation latency (e.g., Table 1 shows that feature extraction for gray DSIFT is faster than HOG) and have different classification capacities. To maintain the same classification precision and recall as the classification approach introduced in Section 4, we have to ensure that the snapshot images that are classified as “normal” in each round of “guessing” have higher classification precision. This is because low recall causes misbehaving users to be mistakenly classified as normal, and adversely affects the overall classification precision and recall for misbehaving user classification. In other words, we want those obscene snapshot images, which are rarer than normal snapshots, to be classified at the last rounds of the “guessing”.

Accordingly, we propose a latency-driven method that can automatically select the appropriate regional feature set (i.e., blocks of snapshot images that need to be revealed) at each round of “guessing”. It is difficult to globally optimize the regional feature sets for each round of “guessing”. Therefore, our latency-driven method selects locally optimal regional features at each round of “guessing”. In other words, we first choose the regional feature set for the first round and then choose the regional feature set for the second round, and so on. The criterion for regional feature selection at each round of “guessing” considers the following three factors: (1) the computation latency of the selected regional features; (2) the proportion of the snapshot images that can be classified by using the selected regional features; and (3) the proportion of the snapshot images that are mistakenly classified as “normal”.

For an online video chat system like Chatroulette, it usually sets up a *maximum tolerable average latency* for classifying a snapshot image. Here we refer to this latency as T_1. When optimizing our classification approach, the average classification latency for a snapshot image should be less than T_1. To satisfy this and minimize the computation latency of descriptor extraction, we select regional features by maximizing the *maximal tolerable latency* at the next round of “guessing”. This is illustrated by the following example. Suppose that an online video chat system needs to classify 100 snapshots and the *maximal tolerable latency of the online video chat system* is 50 milliseconds per snapshot. If the descriptor extraction takes 20 milliseconds per snapshot at the first round of “guessing” and classifies 40 snapshots as “normal”, then the maximal tolerable latency at the second round of “guessing” is $\frac{(50-20)\times 100}{100-40}$ [4]. We further explain our computation for this example as follows. The online video chat system allocates a total of 50×100 milliseconds for the classification process. In the first round, the process of descriptor extraction needs to go through all the snapshot images, which takes 20×100 milliseconds, and so 30×100 milliseconds are remaining for the next rounds of “guessing”.

[4] To simplify our computation, we ignore the latency for descriptor projection and feature classification.

Since 40 snapshot images have been classified in the first round, there are 30×100 milliseconds which can be spent on classifying the remaining 60 snapshot images. Therefore, the average tolerable latency (i.e., the maximal tolerable latency at the second round) left for classifying each snapshot image is $\frac{(50-20)\times 100}{100-40}$ milliseconds.

Let s be the s^{th} round of “guessing“. We formulate the maximal tolerable latency for the $(s+1)^{th}$ round of “guessing” as follows.

$$T_{s+1} = (T_s - f(sel_s) - c_s) \cdot \frac{p_s}{r_s \cdot p_{(s-1)}} \tag{7}$$

where $f(sel_s)$ is the computation latency of descriptor extraction for selected feature set sel_s at the s^{th} round of “guessing”. c_s denotes the computation latency that descriptor projection and feature classification involve at the s^{th} round of “guessing”. p_s and r_s represent the precision and recall for classifying obscene snapshot images at the s^{th} round of “guessing”, respectively. In Equation 7, $\frac{p_s}{r_s \cdot p_{(s-1)}}$ is equal to $\frac{N_{(s-1)}}{N_s}$. Here, $N_{(s-1)}$ and N_s are the number of snapshot images that are classified at the $(s-1)^{th}$ and s^{th} round of “guessing”, respectively. Note that N_0 is the total number of snapshot images that an online video chat system obtains, and p_0 is the proportion of the obscene snapshot images.

Since it is possible to mistakenly classify obscene snapshot images as “normal” in each round of “guessing“, we add a penalty factor $w_s(r_s)$ at the s^{th} round of “guessing”. The penalty factor is a function of r_s – $w_s(r_s) = \alpha^{100\times(1-r_s)}$ where α is a constant and $\alpha < 1$. By combining penalty factor $w_s(r_s)$ with maximal tolerable latency T_s at the s^{th} round of “guessing”, we formulate the objective function of regional feature selection as follows.

$$\max_{sel} \quad w_s(r_s) \cdot T_{s+1}, \tag{8}$$

$$sb.t. \quad sel_s \subseteq X - \bigcup_{m=1}^{s-1} sel_m \tag{9}$$

$$p_s, r_s := boost(\bigcup_{m=1}^{s-1} sel_m, sel_s) \tag{10}$$

$$T_{s+1} = (T_s - f(sel_s) - c_s) \cdot \frac{p_s}{r_s \cdot p_{s-1}} \tag{11}$$

where X is the power set that contains all candidate regional feature sets. sel_m represents the selected regional feature set at the m^{th} round of “guessing”, and $\bigcup_{m=1}^{s-1} sel_m = sel_1 \cup sel_2 \cup \cdots \cup sel_{s-1}$. Equation 10 indicates that precision p_s and recall r_s at the s^{th} round are obtained using a boosting algorithm to combine all the regional features that we used from the first to the $(s-1)^{th}$ round of “guessing” as well as the regional features that we use at the s^{th} round.

To obtain a global optimal for this optimization problem, the straightforward approach is to exhaustively search for optimal regional feature set in power set X. However, exhaustive search is not applicable for the problem where the number of candidate regional feature sets is large. To address the searching problem, sequential forward search [26] is adopted. The pseudo-code is described in Algorithm 1. Sequential forward search is a greedy algorithm, which starts from a null feature set and adds locally optimal feature(s) into the feature set one by one until the expected precision and recall are satisfied. The expected precision and recall are defined based on the maximum number of “guessing” rounds as well as the precision and recall that the non-cascaded solution introduced in Section 4 can achieve. Let P and R be the precision and recall of the non-cascaded solution, respectively. The maximum number of guessing rounds is N. Then, the expected precision ep_s and recall er_s at the s^{th} round are

$$ep_s = P_0 + \frac{s}{N}(P - P_0) \tag{12}$$

$$er_s = R^{1/N} \tag{13}$$

where P_0 is the proportion of obscene snapshot images. The value for the maximum number of guessing rounds can be empirically chosen by selecting the value which achieves the least classification latency.

```
Inputs;
$er$: expected recall at the $s^{th}$ round;
$ep$: expected precision at the $s^{th}$ round;
$p_{s-1}$: precision at the $(s-1)^{th}$ round;
$T_s$: maximal tolerable latency at the $s^{th}$ round;
$avail_s$: feature sets that can be selected at the $s^{th}$ round;
Outputs;
$avail_{s+1}$: feature sets that can be selected at the $(s+1)^{th}$ round;
$sel_s$: selected feature sets at the $s^{th}$ round;
Initialization;
$p = 0, r = 0$;
$sel_s = \emptyset$;
$avail_{s+1} = avail_s$;
while $p < ep \,\&\, r < er \,\&\, avail_{s+1} \neq \emptyset$ do
    for $i = 1 : length(avail_{s+1})$ do
        $p, r := boost(\bigcup_{m=1}^{s} sel_m, avail_s(i))$;
        $t = (T_s - f(avail_s(i)) - c_s) \cdot \frac{p}{r \cdot p_{s-1}}$;
        $score(i) = w_s(r)t$ ;
    end
    $feat := regional\ feature\ with\ maximal\ score$;
    $avail_{s+1} = avail_{s+1} - feat$;
    $sel_s = sel_s \cup feat$;
    $p, r := boost(\bigcup_{m=1}^{s} sel_m, sel)$;
end
return $avail_{s+1}$, $sel_s$;
```

Algorithm 1: Selecting feature set at the s^{th} round.

6. EVALUATION

In this section, using real-world data sets (Section 3) , we evaluate the proposed Fine-Grained Cascaded (FGC) classification solution for misbehaving user detection in online video chat services. Specifically, we validate the effectiveness and efficiency of the new contextual feature descriptors (illumination and color) we have proposed, the 2-stage soft margin SVM algorithm, and our innovative FGC classification scheme.

We have implemented all the descriptors that we introduced in Section 4. The dense SIFT descriptor is implemented in three different color spaces (Gray, HSV and RGB) based on the real time dense SIFT descriptor of the VLFeat library [30]. The HOG descriptor is implemented based on the PHOG descriptor of Vgg MKL [31], and the LBP descriptor is based on the uniform rotation invariant $LBP_{8,1}$ [23]. The parameters used in these descriptors are all empirically tuned to obtain an optimal tradeoff between classification quality and classification efficiency.

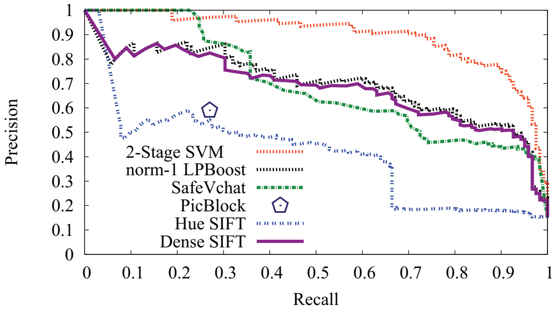


Figure 6: Classification performance comparison.

6.1 Classification Precision and Recall

To evaluate the feature descriptors and the improved combination algorithm, we conduct several experiments and compare our combination algorithm with the state-of-the-art pornographic content detection techniques. Table 1 shows the latency of feature extraction using our implementation. In addition, the Pearson correlations between features and the label of snapshot images (i.e., misbehaving or normal users) are also calculated. As shown in the table, the Pearson correlations are either moderate or medium.

We can observe that the features we use for the classification of misbehaving users are meaningful and the correlations between the features and the snapshot labels (i.e., flasher or not) are not a chance occurrence. Before combining these features, a multicollinearity diagnosis however is necessary, because strong correlations among these features may impose multicollinearity threats. Our Pearson correlation calculation among all the features identifies a few highly-correlated features. Specifically, gray DSIFT has high Pearson correlation values with rgb and hsv DSIFT ($corr = 0.854$ and $corr = 0.746$), and the Pearson correlation value between rgb and hsv DSIFT is also fairly high ($corr = 0.851$). To avoid the multicollinearity threat, the most straightforward approach is to drop any two of DSIFT features. As shown in Table 1, gray DSIFT has higher Pearson correlation than DSIFT feature in rgb and hsv color spaces, which indicates better classification precision and recall. Further, the feature extraction latency for hsv and rgb DSIFT is three times the extraction latency of gray DSIFT. Therefore, we drop rgb and hsv DSIFT features and do not consider them in our combination algorithm.

Combining the features (i.e., gray DSIFT, illumination context, color context, HOG and LBP) using 2-stage SVM, as shown in Figure 6, significantly outperforms the state-of-the-art skin color based detection technique (PicBlock [1] [5]), because the skin colors in snapshot images captured from online video chat systems are diverse and thus the statistical skin-color model used in PicBlock cannot provide effective discriminative characteristics for misbehaving user classification.

Descriptor	Latency (sec.)	Correlation
Gray DSIFT	.06	$.552^{**}$
RGB DSIFT	.18	$.460^{**}$
HSV DSIFT	.18	$.354^{**}$
HOG	.22	$.202^{**}$
LBP	.08	$.251^{**}$
Color Context	.02	$.284^{**}$
Illumination Context	.01	$.271^{**}$

** Correlation is significant at the 0.01 level (2-tailed).

Table 1: Latency of feature extraction and Pearson Correlation.

In addition to the skin color based detection techniques, we also compare our solution with the state-of-the-art Hue SIFT based pornographic content detection technique [19]. Figure 6 shows that the Hue SIFT based technique provides poor classification performance in terms of precision and recall. The reason behind this poor classification performance is that the SIFT descriptor is a sparse feature representation which may cause the loss of some discriminative characteristics. To address this issue, the most straightforward solution is to simply replace the SIFT descriptor with the Dense SIFT descriptor. We repeated the experiment that we conducted for the SIFT descriptor using the Dense SIFT descriptor. We observe that the classification performance in terms of precision and recall is significantly improved and approximately the same as SafeVchat (see Figure 6). However, the classification performance of this straightforward improvement is still lower than our combined solution – combining five features including gray DSIFT, illumination, color, HOG and LBP using 2-stage SVM, because snapshot images have smaller inter-class distance between normal and misbehaving users.

We also compare the performance between the actively deployed SafeVchat [21] and our new solution introduced in Section 4. Figure 6 shows a significant performance improvement in terms of precision and recall. Since SafeVchat uses facial features to discriminate normal users from misbehaving ones, normal users might be mistakenly classified as misbehaving ones when facial features are not presented in the normal users' snapshots. This results in the lower classification performance of SafeVchat. On the other hand, the new solution which combines five features using 2-stage SVM uses other discriminative features to classify misbehaving users. Though the individual features among the five features do not show strong classification capacity (see Table 1), the combination of the features achieves higher classification performance in terms of precision and recall.

To demonstrate that the 2-stage SVM can achieve significant performance improvement, we use the traditional LPBoost algorithm to combine the same features that 2-stage SVM combines. As shown in Figure 6, the classification precision and recall for 2-stage SVM are significantly improved in comparison with the traditional LPBoost algorithm (norm-1 LPBoost). As shown in Table 1, the classification capacity of the features that the combination algorithms use are fairly diverse. For example, Gray DSIFT has strong classification capacity ($corr = 0.552$) while the classification capacity for HOG is weak ($corr = 0.202$). The classification results for traditional LPBoost can be easily biased by the feature with strong classification capacity (i.e., gray DSIFT), while our two stage SVM can balance well the weights of the features and thus achieves higher precision and recall.

Finally, we evaluate the contribution of each feature for classification performance improvement when using 2-stage SVM. Figure 7 shows the incremental classification performance improvement by adding features one by one in the following order – gray DSIFT, illumination, color context,

[5] PicBlock is a commercial software and we are not able to obtain the whole precision-recall curve.

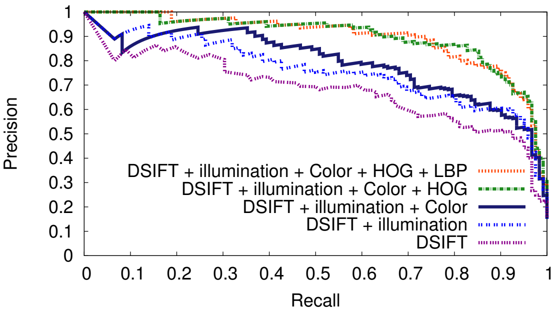


Figure 7: Classification performance variation of 2-stage SVM algorithm using different feature sets.

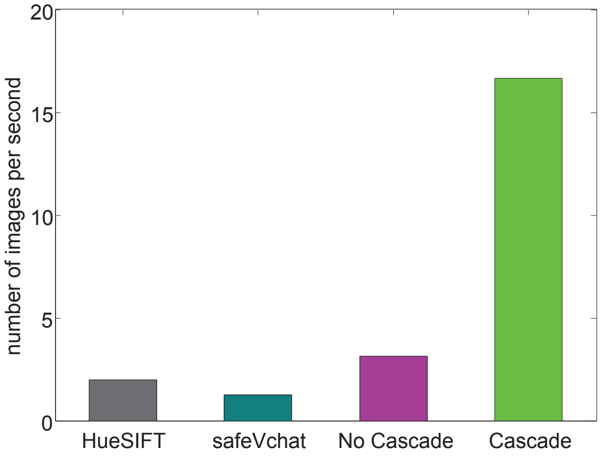


Figure 8: Comparison of classification efficiency. (“Cascade” = classifier with FGC scheme)

HOG, and LBP. As shown in the figure, sequentially adding the features - illumination, color context, and HOG - on gray DSIFT feature significantly boosts the classification performance in terms of precision and recall. However, the classification performance improvement, as shown in Figure 7, is not significant when combining the LBP feature with the other four features. To balance the need for accuracy on one hand and the need for efficiency on the other hand, we therefore do not consider the LBP feature in the classification of misbehaving users.

6.2 Classification Latency

To evaluate the classification latency of our FGC scheme, the *Maximal tolerable latency of the online video chat system* T_1 is set based on the requirements of Chatroulette.

Figure 8 compares of classification efficiency of different schemes, i.e., number of snapshot images that can be classified per second. We observe that FGC classification (cascaded classification) enables one Chatroulette computation infrastructure to classify approximately 17 snapshot images per second. Compared with the SafeVchat deployment [33] and the state-of-the-art pornographic content detection technique (Hue SIFT) [19], FGC classification achieves 92% and 89% reduction in computation latency, respectively. The reason behind this significant improvement in terms of classification efficiency is quite straightforward – both SafeVchat and Hue SIFT involve compute-intensive feature extraction procedures.

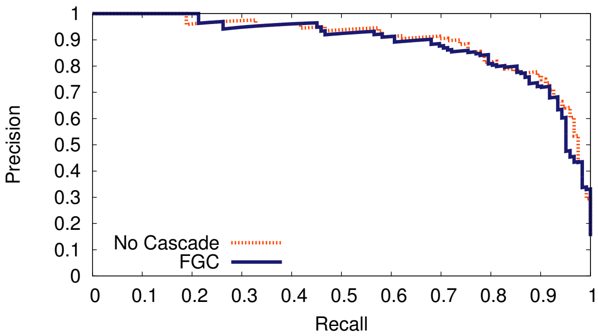


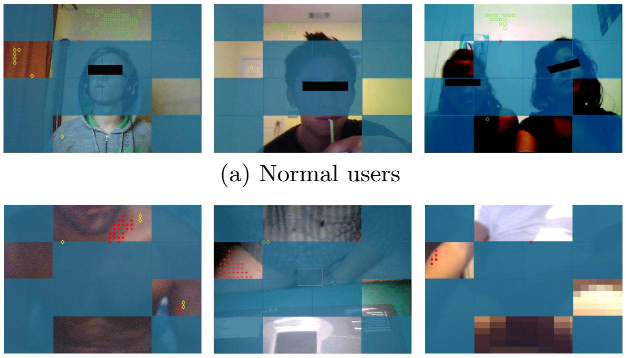
Figure 9: Classification precision and recall (FGC classification vs. No-cascaded classification.

We also observe that our FGC classification achieves significant efficiency improvement in comparison with the non-cascaded classification (82% reduction in computation latency). To understand the reason behind this efficiency improvement, we show the selected regional feature set, partial classified snapshots and partial non-classified snapshots in the first round (see Figure 10). The selected regional feature set by FGC classification in the first round contains the DSIFT feature in six regional blocks (i.e., the areas without blue screen in Figure 10). The first interesting observation here is that, among all the trained DSIFT features, some features capture the ceiling-like line structure and are more abundant in the snapshots from normal users (shown as green “□” in Figure 10(a)). Since ceiling information can infer the upward orientation of webcams and thus identify the snapshots as “normal”, the snapshots shown in Figure 10(a) can be classified as “normal” and thus reduce computation latency for these snapshots. While non-cascaded classification can also obtain the same classification results based on the ceiling information, it needs to process the whole snapshot image (all blocks) rather than partial (revealed) blocks in the snapshot image. Likewise, some DSIFT features capture the texture of hairy skin (shown as red “+” in Figure 10(b)). Since the snapshots with hairy skin are more likely to contain obscene content, those snapshot images are passed to the subsequent rounds of “guessing”. For those snapshot images that cannot be classified by using a few revealed blocks, the computation latency involved in FGC classification remains the same as that in non-cascaded classification. Therefore, the overall efficiency improvement is contributed by those snapshot images that can be classified using only a few of blocks of the images.

Finally, we compare classification precision and recall for our FGC classification with non-cascaded classification (See Figure 9). We can see that the precision and recall in FGC classification are similar to those in non-cascaded classification. Our feature descriptors are designed based on discriminative characteristics and the presence of the discriminative characteristics at certain regions of an snapshot image dominates the classification result. As a result, using certain regions of snapshot images to classify obscene and normal snapshot images can obtain approximately the same classification performance in terms of precision and recall.

7. CONCLUSION

This paper presents a novel solution to identify misbehaving users efficiently and accurately in online video chat sys-



(a) Normal users

(b) Misbehaving users

Figure 10: FGC classification results at the first round of the “guessing”.

tems. Two new contextual features, illumination and color contextual features are introduced. Combining these two new features along with a few existing features using a 2-stage SVM algorithm, the proposed solution can discriminate misbehaving users from normal ones with high classification performance in terms of precision and recall. To achieve higher classification efficiency, the paper further proposes a new fine-grained cascaded (FGC) classification approach, which orders the compute-intensive feature extraction process into multiple rounds and allows normal users to be classified in fewer rounds. Experimental results using real-world data demonstrate that our solution can significantly improve the classification precision and recall as well as reduce classification latency.

Acknowledgment

We thank Andrey Ternovskiy (Founder of Chatroulette.com) and Kirill Gura (CTO of Chatroulette.com) for providing us with these otherwise unobtainable internal data traces and computation resources. This work is supported in part by the unrestricted gift funds from Chatroulette.

8. REFERENCES

[1] Picblock web site. http://www.cinchworks.com/.

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

[3] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, page 22, 2004.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886 –893 vol. 1, june 2005.

[5] T. Deselaers, L. Pimenidis, and H. Ney. Bag-of-visual-words models for adult image classification and filtering. In *ICPR 2008.*, pages 1 –4, dec. 2008.

[6] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, volume 2, pages 524 – 531 vol. 2, june 2005.

[7] M. Fleck, D. Forsyth, and C. Bregler. Finding naked people. In *ECCV' 96*, pages 593–602. Springer Berlin / Heidelberg, 1996.

[8] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, pages 221–228. IEEE, 2009.

[9] M. Hammami, Y. Chahir, and L. Chen. Webguard: A web filtering engine combining textual, structural, and visual content-based analysis. *IEEE Trans. on Knowl. and Data Eng.*, 18:272–284, February 2006.

[10] W. Hu, O. Wu, Z. Chen, Z. Fu, and S. Maybank. Recognition of pornographic web pages by classifying texts and images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1019–1034, 2007.

[11] C. Jansohn, A. Ulges, and T. M. Breuel. Detecting Pornographic Video Content by Combining Image Features with Motion Information. In *Proceeding of the seventeen ACM international conference on Multimedia*, 2009.

[12] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. In *International Journal of Computer Vision*, pages 274–280, 1999.

[13] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *Pattern Recognition*, 40(3):1106–1122, 2007.

[14] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K. R. Müller, and A. Zien. Efficient and accurate lp-norm multiple kernel learning. *NIPS*, 22(22):997–1005, 2009.

[15] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, volume 2, pages 2169 – 2178, 2006.

[16] J.-S. Lee, Y.-M. Kuo, P.-C. Chung, and E.-L. Chen. Naked image detection based on adaptive and extensible skin color model. *Pattern Recognition*, 40(8):2261 – 2270, 2007.

[17] P. Lee, S. Hui, and A. Fong. An intelligent categorization engine for bilingual web content filtering. *IEEE Transactions on Multimedia*, 7(6):1183 – 1190, dec. 2005.

[18] A. Lopes, S. de Avila, A. Peixoto, R. Oliveira, and A. Araújo. A bag-of-features approach based on hue-sift descriptor for nude detection. In *Proceedings of the 17th European Signal Processing Conference, Glasgow, Scotland*, 2009.

[19] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, volume 2, pages 1150 –1157 vol.2, 1999.

[20] Flasher detection algorithm aims to clean up video chat. January 19, 2011. http://www.technologyreview.com/blog/arxiv/26281/.

[21] Myyearbook live web site. http://live.myyearbook.com/.

[22] T. Ojala, M. PietikÃd’inen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, pages 51–59, 1996.

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

[24] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *ICCV*, pages 1–8, 2007.

[25] P. Pudil, J. Novovicová, and J. Kittler. Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125, 1994.

[26] G. Ratsch. Robust boosting via convex optimization: Theory and applications. PhD thesis, University of Potsdam, Potsdam, Germany, 2001.

[27] Tinychat web site. http://tinychat.com/.

[28] J. Uijlings, A. Smeulders, and R. Scha. Real-time bag of words, approximately. In *Proceeding of the ACM International Conference on Image and Video Retrieval*, page 6. ACM, 2009.

[29] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2011.

[30] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.

[31] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *CVPR*, pages 3539–3546. IEEE, 2010.

[32] X. Xing, Y.-l. Liang, H. Cheng, J. Dang, R. Han, X. Liu, Q. Lv, and S. Mishra. Safevchat: Detecting obscene content and misbehaving users in online video chat services. In *WWW*, 2011.

[33] J. Ze Wang, J. Li, G. Wiederhold, and O. Firschein. System for screening objectionable images. *Computer Communications*, 21(15):1355–1360, 1998.