ScreenSense: Screen Activity Detection in Real-World Environments with Indoor Light Sensors

Tushar Routh, Nurani Saoda, Fateme Nikseresht, Md Fazlay Rabbi Masum Billah, Jiechao Gao, Viswajith Govinda Rajan, Bradford Campbell
University of Virginia, Charlottesville, Virginia, USA
tkr2w,saoda,fn5an,masum@virginia.edu
jg5ycn,gyx4bw,bradjc@virginia.edu

Abstract

Nowadays, light sensors are frequently utilized as wearables for assessing personal light exposure or installed in indoor environments for measuring ambient lighting at areas of interest. Interestingly, light emitted by computer screens records distinct patterns when such sensors are placed nearby. The phenomenon requires analysis for passive sensing and also raises critical privacy concerns. In this paper, we introduce *ScreenSense*, an innovative approach that leverages data from existing framework for detecting screen utilization. For that, we first collect a diversified dataset by placing a light sensor in close proximity to computer screen. We then classify captured dataset into five general categories: Mail, Social, Reading, Video, and No Activity. Our insight is that existing low-power, inexpensive light sensors can be an energy-efficient, low-cost alternative for collecting screen information over extended periods. However, we also observe that for to be effective in real-world, it needs to be robust against several practical factors, including the ambient room lighting where the user device is situated, transitioning between different activities, and examples from unfamiliar arrangements. To overcome these challenges, we propose dataset augmentation including realistic lighting conditions, transition filters, and time series-based augmentation. The system achieves a detection accuracy upto **91.25%** in real world testbed scenarios. ScreenSense also uncovers critical privacy issues inherent in simple IoT based light sensors deployed so commonly in smart buildings.

CCS Concepts

• **Computer systems organization → Embedded systems, Sensor networks**.

Keywords

Screen Activity Detection, Passive Sensing, Indoor Light Sensors

ACM Reference Format:
Tushar Routh, Nurani Saoda, Fateme Nikseresht, Md Fazlay Rabbi Masum Billah, Jiechao Gao, Viswajith Govinda Rajan, Bradford Campbell, University of Virginia, Charlottesville, Virginia, USA, tkr2w,saoda,fn5an,masum@virginia.edu, jg5ycn,gyx4bw,bradjc@virginia.edu. 2024. ScreenSense: Screen Activity De-tection in Real-World Environments with Indoor Light Sensors. In *The 11th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BUILDSYS '24), November 7–8, 2024, Hangzhou, China.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3671127. 3698167

BUILDSYS '24, November 7–8, 2024, Hangzhou, China

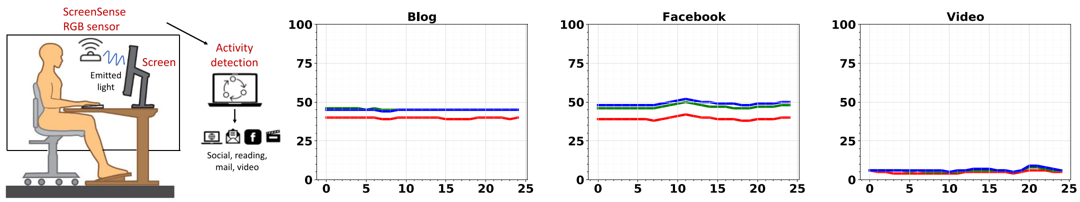
ACM ISBN 979-8-4007-0706-3/24/11
https://doi.org/10.1145/3671127.3698167

1 Introduction

For studying screen time influence, gathering information only regarding how many hours viewers have spent on screen can be insufficient, as the effects of different platforms on people may differ widely [21]. Instead, specific information regarding how much time has been spent on which platforms is useful for many use cases including evaluating employee productivity, observing individuals' digital lives engagement, identifying physical and mental health risks among both children and adults and so on [8, 11].

People today are increasingly adopting indoor light sensors, due to their capacity to prioritize user convenience, energy efficiency, environmental sustainability, and calculating daily exposure hours on different types of lights for human wellness [7, 25]. The popularity and demand for these devices have grown in commercial and residential spaces, and are projected to grow at a compound annual growth rate (CAGR) of 12%, in both wearable and fixed-point formats [29]. Basic indoor light sensors operate by detecting fundamental color parameters, such as RGB values, and the intensity of light in the surrounding environment. Regardless of the purpose, when worn by a user near a monitor or placed close to a workstation, these sensors intriguingly detect unique variations in screen color parameters associated with user engagement on various platforms (Figure 1).

Leveraging color-based data from computer screens for activity classification isn't entirely novel. Experts utilized light sensors as smart eyeglass [34], lux meters from tablets [9] or RGB camera [3] for passive screen sensing. *So why do we need to study the possibility of screen sensing with indoor light sensors?*. The answer is two-fold. In indoor environments, these sensors are often deployed for maintaining consistent lighting near workstations or carried by an individual as a personal smart health gadget while sitting in front of the screen. When intended, professionals can analyze the near-screen recorded data from these sensors to extract screen usage information for further investigation. This method leverages the existing setup, whereas the approaches mentioned above require additional devices or arrangements for sensing. On the flip side, when users operate devices near screens or someone places such sensors intentionally near workstations, they may unwillingly expose their on-screen activities, making them vulnerable to eavesdropping. That is why the experts should have a better understanding of the capabilities of indoor light sensors for passive sensing in indoors and analyze the possible security issues from



(a) (b) (c) (d)

Figure 1: (a) A RGB based indoor sensor was installed near the user's workstation to maintain preferred lighting environment. Due to proximity to a computer screen, it picks up RGB variations over time samples for (b) Blog titled 'Introduction to Attention Mechanism by Kemal Ardem", (c) Facebook with "Dark Mode off" setting, (d) Movie sequence from "Bourne Ultimatum"

the arbitrary usage/deployments. Only then can they take a proactive data-driven approach during indoor deployments remaining vigilant about potential security and privacy threats.

This paper concentrates on addressing two straightforward questions in our contribution: *Is it possible to utilize basic color information from a general light sensor for the classification of screen activities? And with the provided information, how can we effectively navigate real-world factors that might jeopardize the integrity of this approach.*? We propose ***ScreenSense***, an innovative platform for detecting users' activities through passive sensing. The platform leverages existing light sensing framework, which can provide information regarding surrounding lighting environments, as well as unfold users' screen engagement. It not only terminates additional components/arrangements for tracking daily hours on digital social platforms, but also eliminates the need for installing custom software, third-party data sources, and cookie-matching technologies, which can reveal excessive browsing information or capture intricate screen details.

In ScreenSense, we deploy the indoor light sensor in a custom PCB consisting of RGB sensor, a MCU and BLE radio. This board can function as a smart gadget or fixed point device to simulate indoor light sensor usage. The board is optimized **for extremely low power consumption** to enable long-term screen usage data acquisition without frequent power replacements and to support operations in energy harvesting scenarios. As on-board classification can demand additional resources, it transmits the sensed data to a remote platform, such as a nearby cloud computer, for post-processing. After placing the board in controlled indoor environments and near the screen, we collect screen usage information from selected categories in various settings. We find that although the magnitude of the raw data can vary based on the sensor placements, **the patterns are independent of the distances and angular variations.**

To evaluate the effectiveness across different screen activities, ScreenSense design is divided into two crucial phases: dataset generation and dataset augmentation. During the dataset generation phase, we recognize that analysing the true distribution of RGB variations from screen data require an exceptionally extensive dataset, involving numerous sensors deployed across diverse screen settings/indoor environments for an extended duration. ScreenSense first consider a few non-complex and most practical RGB variations over controlled environments, **including different screen settings and operational variations**.

To reduce calculation complexity, rapid processing, and avoid unnecessary data storage, **ScreenSense simply utilizes raw variables from sensors for classification.** After analysing recorded data to categorize, we notice that various on-screen activities record **non-linear data for classification**, where classification accuracy can vary based on **the number of advertised packets** considered for classification. Based on this, we aim to optimize ScreenSense's classification performance by minimizing the number of the data packets advertisements required, thereby reducing the overall energy consumption of sensor nodes without compromising system accuracy.

After deploying ScreenSense in realistic scenarios, we observe that the performance of the top-performing classifier can decline significantly. We identify three major drawbacks, *classifying under room light, random switching in between activities and simply identifying examples from unseen environments.* To navigate these performance limiting factors, **ScreenSense proposes several data augmentation techniques.** First, we mix the light data from light sources and screen to mimic light pollution at the screen. Second, we apply filters of variable window size to replace parts of an activity with other activities to capture activity transitioning. Lastly, we adopt time-series GAN to represent the samples from the unseen environments. We discover that substantial improvements in classification accuracy can be achieved by **retraining the classifier with an augmented training dataset** that accounts for realistic variations.

Besides enabling passive activity monitoring, ScreenSense raises a **privacy implication of indoor light sensors** which are mostly installed naively without considering the possibility of leaking privacy-sensitive information. This opens the door to further research regarding how to minimize color data side-channel effects during design, deployment, and data communication process. Below we summarize the contribution of the paper:

- We propose ScreenSense, which utilizes **existing architecture for light sensing** to detect and categorize users' on-screen activities into five distinct classes:***Mail, Social, Reading, Video and No activity***. For activity classification, we leverage **only raw information** advertised through BLE packets after placing the device in front of the screen.
- We consider **several practical factors** including activity data from different users, various types of screens, screen settings, operational settings, and nearby light source effect to design a detailed and realistic screen activity database. To classify captured **non-linear RGB information from screens**, we study several machine learning-based approaches including neural networks,

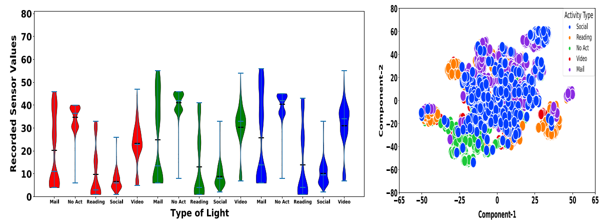


Figure 2: RGB violin plot of activity data (top) (x-axis:source type, y-axis: recorded values). Separate activities generate common readings and cannot be segregated solely based on threshold values. 2D tSNE plot with first two principal components exhibit linear inseparability (bottom)

and time series-based architectures with fine tuning to find the best performing classifier model with **a minimum number of data packets.**

- We implement several **data augmentation techniques** to address real-world performance limiting factors. Our experiment demonstrates a significant performance improvement across all testbed scenarios **up to 7.5%** after performing the augmentation.

2 Related Work and Limitations

In this section, we discuss related approaches to monitor users' screens activity. We categorize major screen monitoring techniques as follows:

2.1 Self-reporting and Software based approaches

Self reporting activities like [15] were largely found to be inaccurate and confusing, which questions the credibility of this approach [12]. Utilization of customized software/platforms such as third-party data sources and cookie-matching technologies have also been adopted because of more precise and accurate tracking [1, 2]. Unfortunately, this requires additional installation for every device a specific user comes across during the day, potentially disclosing superfluous browsing information and even posing the security risk for the classified information within a device if not implemented carefully [24].

2.2 Indirect Sensing Mechanisms

Researchers have also shown that characterising online activity can be accomplished with *specially designed eyeglass [22], head-mounted color light sensor [20] and Keyboard extraction,* [33]. Such indirect methods have multiple disadvantages. For sensing, these devices require additional setups or must be used in a manner that can cause discomfort for continuous, daylong operation. Additionally, they need to be synchronously switched on and off with the monitor's activity status. In addition, the eye level sensor is not quintessential for computer activity usage detection.

2.3 Screen Recording and Snapshots

Screen activity recognition through screen recording and taking snapshots have been attempted in [23, 28]. However, storing high-resolution snapshots of the screen over a prolonged period is memory intensive and privacy compromising. [18, 33]. They also demand complex framework design and post processing, which makes them unsuitable for mass deployment.

2.4 Choice of parameters for classification

Classifying activities based on lux information was attempted in [16]. As lux intensity can vary based on screen size, placement of sensors, and sensor configuration, classification based on it is going to be incorrect. Within the same setup, we discover that different activities share common RGB spectra and magnitudes, which makes it challenging to differentiate solely based on RGB thresholds as shown in Figure 2. t-SNE visualization of RGB values also reveals that dissimilar light sources are linearly inseparable. This necessitates careful classifier architecture selection like Machine Learning, Neural Network or Time Series Classifiers based algorithms, which are efficient in classifying non-linear time-varying signals.

2.5 Memory/Power Inefficiency

Spectra from mini spectrometers can reveal specific activities running on the screen [8]. However, such devices are expensive, power-hungry through unnecessary components like accelerometer, and unsuitable for mass deployment. For screen activity sensing to be practical and ubiquitously adopted, the sensor should be cost-effective, small, and standalone. For IoT devices as sensors, they are expected to optimize available resources to facilitate energy harvesting. For long-term recording and memory friendliness, classification should be based on minimum parameters.

2.6 Real-world Variations

In previous works, screen data has been collected in controlled environments like in a dark room or considering a single screen for the source of emitting color light [20, 22]. In real world, a single computer screen may not be the only RGB source, particularly in workplaces or classroom scenarios. RGB values from a single screen can be polluted with nearby monitors or indoor light source. Moreover, performances in unseen scenarios like with a new monitor, or old monitor but with new setting have not been reported.

2.7 On device Classification

Some approaches used time series-based features for activity classification [34]. However, these methods expect computations to be performed on the sensor node, which is memory-intensive and cause unnecessary delay in classification.

3 ScreenSense for Real-World Applications

Within this section, we illustrate practical, real-world scenarios where ScreenSense can be adopted.

Scenario 1: Consider the scenario where the office authority has gone through renovation and wants to uncover whether that has any impact on the productivity of office staff through analysing the employee's screen duration in a particular platform [26]. In such a case, the authority would like to compare employees on screen behavior pre and post renovation to study the effectiveness of the action. To study this, the authority may desire general screen activity rather than exact details to draw inferences. A simple color sensor, placed near the working area, can generate platform based signature patterns, which then can be analysed to unfold time specific screen usage information.

Scenario 2: Attention Deficit/Hyperactivity Disorders (ADHD) are neurodevelopmental conditions that are prevalent among 2-10% of the total population, where switching screens from one platform to another over long period at abnormal frequencies is considered one of the prime indicators [30]. To detect such disorders, sensing

Systems	Activity-specific detection	Installation requirement	Sensor requirement	Range	Power limitation considered	Parameters for classification
Tiger [22]	Screen View vs Non-view	User-worn (eye-glass)	RGB, IMU, Lidar	80 cm	No	RGB, Hue, Saturation, Intensity
LuxLeak [16]	Ten popular websites	User-worn (eye-glass)	RGB light sensor	60 cm	No	Lux
WISEGlass [34]	Watching Movie/Browsing	User-worn (smartwatch)	Light intensity	30 cm	No	13 statistical features from 23 virtual channels
Head Mount [20]	Document Reading	User-worn (forehead-mounted)	RGB light sensor	cm-scale	No	Same as above
ScreenSense (This paper)	**Activity class**	**User-worn/Fixed deploy**	**RGB light sensor**	**1m**	**Yes**	**Only RGB and Brightness**

Table 1: A comparison of ScreenSense with the most relevant approaches

and studying users' on-screen behavior over a long period is crucial. Nevertheless, the process needs to be completed by acknowledging privacy and significant user attention for device installation. As light sensors are omnipresent nowadays, a simple indoor light sensor placed near the screen can act as a camouflaging device to uncover users' behavioral switching-over patterns.

4 Implementation of ScreenSense System Architecture

4.1 System Overview

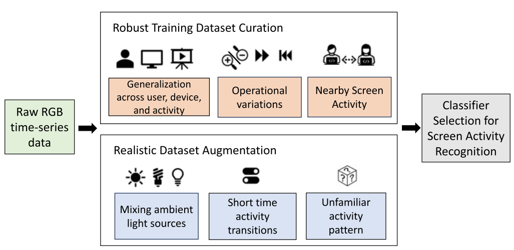


Figure 3: An overview of the ScreenSense framework

ScreenSense design consists of a very low power color sensor deployed near a device user and an activity classification framework that aims to design a high performing real-world activity detection platform. Figure 3 illustrates the basic blocks of operations. Our design consists of two major parts: dataset collection and dataset augmentation. After sensing, ScreenSense is designed to transfer the sensed parameters to a distant platform to minimize on-board resource requirements *(like, cloud)*. Major resource-intensive tasks, like pre-training of the classifier with the diversified dataset, along with identifying recent observations is performed at the distant device. The upper block of Figure 3 depicts the scope of data diversity incorporated within the training set, while the lower section illustrates the approaches used to acquaint that classifier with the real-world variations of the collected diverse dataset. Table 1 compares our proposed method against similar approaches. ScreenSense excels in several areas than similar approaches: it effectively addresses relevant screen activity categories, offers energy efficient passive operation with a wider range and employs significantly fewer parameters for data-efficient classification, thereby minimizing computational complexity and avoiding unnecessary delays.

Data collection process with the sensor in realistic indoor environments have been discussed in the *Experimental Methodology* section. Unless otherwise specified, the plots exhibit **recorded decimal values *(y-axis)* against time samples *(x-axis)***.

4.2 Dataset Overview with Labeling

At first, we try to utilize the existing database of screen sensing. Unfortunately, all the RGB based databases were either purpose specific (e.g., TV on off with distance in [14] or with limited categories/ inappropriate with our purpose (e.g., 800 images from five educational categories in [13, 28]). For that, we decide to collect our dataset and select the top four screen activity classes: **Mail, Video, Social, Reading**, along with inactive hours **class: No Activity** based on screen usage report [17]. We argue that detecting these classes of activities instead of identifying specific applications or websites (Facebook or Gmail) is more insightful for many applications including personal productivity tracking, work-life balance, and attention span of users.

In training dataset generation, we identify inherent variations in the RGB recordings even in dark room environments related to the screen, screen setting, operating nature etc. that can impact any passive screen activity identification approach and need to be addressed for better classification. Including these variations in the dataset aims to acquaint the classifier with realistic scenarios as comprehensively as possible. However, it is not feasible to encompass all these variations in the dataset. Therefore, our approach emphasizes analysing and incorporating RGB variations associated with selected events into the training set to enhance robustness, marking a novel exploration in this area.

Labeling datasets can be confusing for multiple reasons. There is a wide diversity in screen activities, making it challenging to draw clear boundaries (for example, determining whether "watching a video on Facebook" should be categorized as Video or Social?). In addition, multiple activities occur simultaneously. To simplify matters, we assign an activity to its primary platform (like playing a video on Facebook is labeled as "Social") and displays only one activity on one screen at any given time (e.g. [27]).

4.3 Composing Training Dataset

To consider inter and intra-class variability of different on screen activity scenarios, ScreenSense identifies and categorizes several sources of variations that are crucial in practical use-case scenarios. These variations are depicted as recorded RGB decimal value variations over time **(figure 4-12, 14, 18, 20 & 22)**.

4.3.1 ***Generalizing across User, Device, and Activity*** In this section, we consider realistic diversification of screen RGB information that can derive from various screens, screen settings, user operation and screen content.

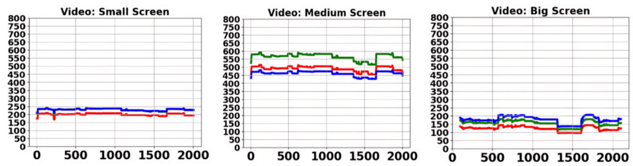


Figure 4: Same video sequence played on different devices have dissimilar RGB patterns. We plot the data collected from Lenovo Ideapad S145 laptop with 15" monitor (left), Dell 32" computer monitor (middle), and LG 55" display (right)

Screen to screen variation: Dimensions and features of different screen can result in variation of sensed values for the same activity. As seen from Figure 4, the same activity sensed at different screens differs from each other. **Although the individual color-magnitude varies, for a particular activity, the fluctuation patterns over timeframes were observed as similar on different screen settings.**

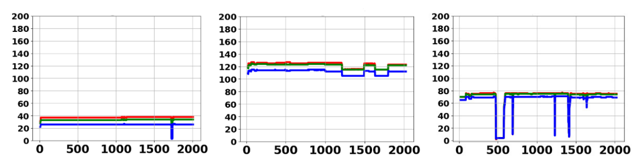


Figure 5: Different users were provided with the same document and were asked to go through it in their usual ways. As seen, RGB patterns generated by *User 1*, *User 2* and *User 3* were different from each other.

User to user variation: We observe that individual persons go through similar content in different ways, based on their style of engagement and preference. This can result in variation of sensed data. To capture them, we ask different individuals to operate their social/email accounts with the same settings and some predefined content *(after IRB approval)*. Figure 5 demonstrates the difference in recorded patterns across different users.

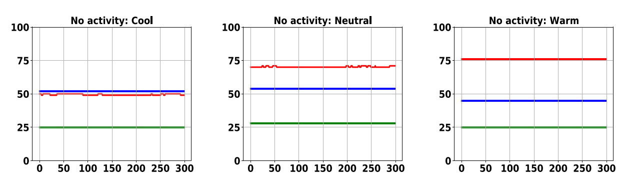
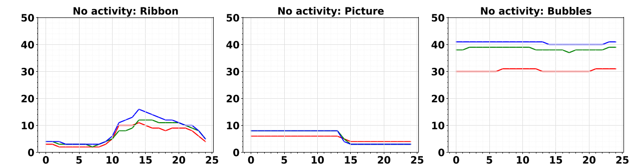


Figure 6: No activity RGB variation of 300 samples, collected from a HP 32" monitor set on four different color modes (left to right): *Cool, Neutral and Warm*

Screen settings variation: People can use different screen settings (*brightness, theme, inverted color* etc.) based on their necessity and preferences, which results in variability of sensed values. To capture this, we collect data by setting the screen with different color themes. We consider common color themes including cool, neutral, warm and custom settings. We also recognize the impact of different display types including LCD and LED. Figure 6 shows how the RGB data differentiates across changing color modes.

Inter class variations: Activities within the same class can derive

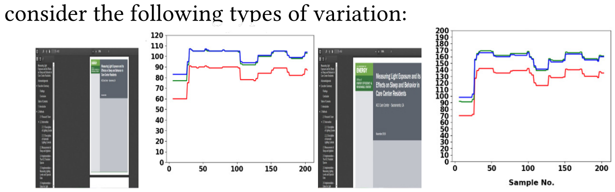


(a) (b) (c)

Figure 7: Analysing RGB variation of different screensaver setup: (a) Ribbons, (b) Pictures (c) Bubbles.

from different sources *(like platforms, websites etc.)*. For example, social interactions can be performed using platforms like *Facebook, LinkedIn* etc., which we cluster as social class. This goes the same for other categories (Mail: *Gmail, Outlook, Yahoo Mail etc.*, Video: *Movie, Gaming, News, Songs etc.*) and so on. Even with screensaver mode, different settings like random flyers, bubbles etc., can generate contrasting patterns. We have captured the variations above in our training set as intra-class variations (Figure 7).

4.3.2 ***Factoring Operational Activity Variations*** Operating style of an activity can influence the recorded RGB patterns. We consider the following types of variation:



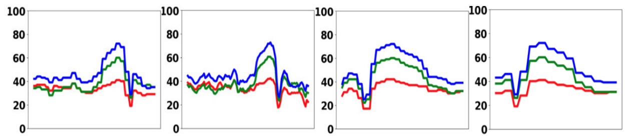
(a) (b) (c) (d)

Figure 8: Analysing effect of zooming with various page setup (a) 50% zoom, (b) RGB readings with 50% zoom, (c) 100% zoom, (d) RGB readings with 100% zoom.

Zooming the screen: Zooming users' screen may record different patterns, as they cause variation in screen contents', their shapes and sizes. However, based on our observation, varying zoom settings while reading have not generated any new pattern (Figure 8).

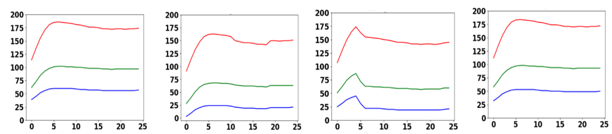
Action speed variation: Different actions can be played at various paces. We vary the playback speed of actions and analyse patterns. As observed, RGB fluctuations within a specific timeframe become faster/slower (*like increasing the playback speed or scrolling a document*), which significantly differs from than original speed version. To capture the effect, we attempt to re-generate alternate patterns by resampling the original speed data. We expect the up-sampled version to have an accurate representation of the actual version played at a faster rate. As shown with a video play, Figure 9 (b) confirms this expectation. However, the reconstructed version have slight deviations from the actual action. We created such variations for different categories and included them in the training set.

Action in reverse: Screen actions may not follow the same sequence of the data acquisition. That is why we observe what happens when the action takes place in reverse order, which is the



(a) (b) (c) (d)

Figure 9: Comparing RGB variations of real vs library generated examples over time (x-axis). (a) Playing a video at a faster speed, (b) Resampling with *scipy*, (c) Playing the same video in reverse, (d) Inverting samples with *Numpy*.



(a) (b) (c) (d)

Figure 10: Analysing activity under light (a) *CFL ("natural daylight", 13 W)*, located 150 cm screen, (b) Mailing (c)Reading, (d) Watching Video under light. As seen, finally recorded values were dominated by the light source, where RGB influence from video activity was the least.

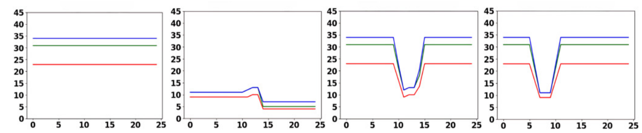
highest level of dissimilarity *(like scrolling a page from bottom to top)*. We replicate those reverse actions using *python* library and include them in the training set. An example has been depicted in Figure 9 (c and d), where we played the same video in the prior example in reverse, where the library-generated version has nearly replicated the real word reverse playback.

4.3.3 ***Nearby Screen Influence*** We place our sensor in multi-display environments, placing displays side by side, and observe RGB variation of the primary screen. In practice, the influence will depend on the positioning of the sensor relative to the secondary screen and its orientation towards that screen. As we point the sensor towards the primary screen, we observe that the primary signal amplitude varies randomly within 2%- 5% in the presence of a background second screen. To model this effect, we add a random RGB noise (5% of components amplitude) to our dataset.

4.4 Realistic Dataset Augmentation

After training our classifier with a diversified database under controlled scenarios, we shift our best-performing classifier in realistic scenarios. Based on our inspection of several testbeds, we monitor that even after training the best model with fine-tuning and wide-ranging examples, performance has deteriorated substantially. Later, we address three major mis-classification incidents. The incidents, along with the approaches we take to encounter them, have been discussed below:

4.4.1 ***Effect of Indoor Light Sources*** Common indoor sensors capture both RGB and the Clear component, which is the unfiltered version of light and represents the brightness information of the source(s) in the surroundings. In the presence of ambient light, there is an interaction in between indoor light and the light coming from the screen. The resultant can differ significantly from the version which is originating solely from the screen, leading to incorrect predictions (Figure 10). [32] did similar work with background lighting, but only with the single resultant at the sensor point with a single type of light. However, in the real world, sources can be of



(a) (b) (c) (d)

Figure 11: Mixing original (a)*no activity* window with interfering (b)*video data* window. (c), and (d) exhibit addition of interfering signal randomly (in a 5 sample point duration) to mimic multi-event window in real world.

multiple types and the resultant at the sensor, however, depends on multiple measurements: (a) the relative distance between two sources, (b) the positioning of the sensor. To enable activity classification under light, we collect samples from 5 different types of major indoor artificial lights from 3 different categories: *LED, Inc and CFL* and natural light at different conditions *(Morning, Noon, Evening, Rainy Day, and Overcast). In all scenarios, color parameters recorded only with light sources were significantly higher in magnitudes than the activity values, to simulate the real-world scenarios.* We mix data from light and screen to mimic scenarios under the light. The highest possible deviations can be for constructive and destructive interference, where the resultant patterns were significantly dissimilar from the darkroom setup. For that, *we have considered both extremes by adding and subtracting signals from Light Source and Screen, which represent the highest possibilities for misclassification.*

4.4.2 ***Switching between Activities*** Within the sampling window, switching from one platform to another or sometimes frequent movement within the same activity generate transitional patterns that are unalike signals captured without switching. These patterns can occur randomly and for an unknown duration and classifiers tend to misidentify when asked for identification. While moving back and forth within a single activity (like *closing an email and opening another*), we want our classifier not to get perplexed. For windows containing multi-activity observations (like *movement from Mail to Video and coming back to Mail*), we want to tag such time frames with the activity that contains the highest number of points within those time frames. For that, we have designed and developed filters for specific variable size window. When applied to regular patterns, these filters are capable of mimicking real-world transitional scenarios. For multi-actions scenarios, we implement these windows in a primary action and replace those points with a secondary action. We keep in mind that the secondary action duration never surpasses the primary one within that particular timeframe and label the frame as the primary category *(like 15 samples of yahoo mail and 10 samples of a video song, will be labeled as "Mail"*) (Figure 11).

4.4.3 ***Unfamiliar Activity Pattern*** With a limited dataset, machine learning models are prone to overfitting, making universal classification problematic. It is also impractical to train models on all possible activity variations a user might encounter. Therefore, we include synthetic examples in our dataset, generated based on the original data distribution, to represent the absent real-world scenarios in the current training set. While there are various methods to generate synthetic data, our time-varying RGB dataset includes

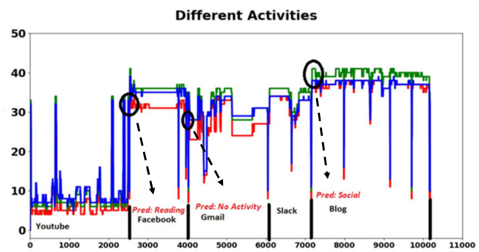


Figure 12: Wrong predictions events for during roaming around multiple platforms.

both static (e.g., constant values during screensaver mode) and dynamic (erratic variations) features, which need to be considered. To tackle the complexity of generating time series data with intricate temporal dependencies, we employ the TimeGAN architecture. This approach captures irregular and random fluctuations, as well as constant attributes, providing more challenging examples for models compared to simpler methods like autoregressive models *(details have been discussed in [35])*. To the best of our knowledge, for color information based real-world passive sensing of on-screen activities,**consideration of constructive and destructive interference from nearby light sources, special filter designing to generate transitional patterns during switching, and fabricating time series synthetic examples for universal classification** have been implemented for the first time for screen activity identification in real world. Now we include all those examples *(under light, switch-over, and synthetic examples)* and retrain our best classifier. Accuracy before and after the introduction of the augmented training set were compared under different real world scenarios.

4.5 Classifier Selection

After collecting original and real world variation data sets, we study several *ML, NN, and Time Series* based classifiers. For ML based classification, we select Decision Tree (DT), Random Forest (RF), K-Nearest Neighbour (KNN) and Support Vector Machines with radial (SVM-Rad) and polynomial (SVM-poly) kernels. However, with dataset having a small set of examples with high variation, classifiers can behave as weak learners and may tend to overfit. To improve prediction in general, we introduce ensemble-based boosting algorithms and select Adaboost (Ada) and Extreme Gradient Boosting (Xboost) for classification. To extract diverse features from our time-series observations, we use Random Convolutional Kernel Transform (Rocket) and its faster variant Minimally Random Convolutional Kernel Transform (MiniRocket). As signals from BLE devices are multi-channel time series data, we have included both Neural Networks (Feedforward Multilayer Perceptron Models (MLP), Convolutional Neural Networks for classification. In activity data, we expect some underlying relationship among samples. For that, we have also included Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) for classification.

As our goal was to minimize parameters for classification, we scale, normalize and divide only the RGB dataset into training, test, and validation sets (80%, 10% and 10% respectively). For better evaluation of the built model, we have implied stratified 10-fold cross validation by tuning to their best hyper parameters using *Gridsearch*. After reporting mean accuracy with standard deviation, we analyze the mis-classified examples for betterment. After discovering certain patterns of wrong predictions, we decide to include brightness in our dataset and re-record accuracy in controlled environments.

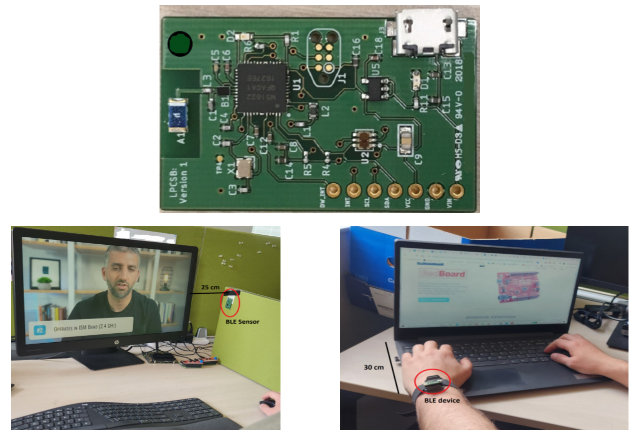


Figure 13: ScreenSense color sensor PCB (top), Deploying the board in a personal workspace as a fixed-point installation (bottom-left) and as a handheld device (bottom-right) during data collection

5 Implementation

5.1 ScreenSense RGB Sensor Design

To collect the screen activity data, we implement a small, custom-made prototype PCB accommodating a low power color sensing chip. An ultra-low power MCU interfaces with the color sensor to periodically sense, process, and transmit the color data over BLE. The board adopts *TCS34725* color sensor with high sensitivity and wide dynamic range. The active current consumption while sampling the internal ADC channel is 235 μA, which can be reduced to 2.5μA during the sleep mode. The PCB is powered through a micro-USB B-type connector and a 3.3V linear regulator. The MCU periodically samples the red, green, blue, and clear channels of the sensor, calculates the color temperature and lux value, and transmits these values with raw color data as BLE advertisements. The dimension of the PCB is $24mm \times 39.5mm$, allowing it to be ubiquitously deployed in indoor environments. With an advertisement interval of 12 ms (5 adv/second), the sensor can run upto 45 days on a 3.3V lithium coin battery. The overall cost per board ranges between \$10-\$15 which can be mass-produced on a reasonable budget (Figure 13) [31].

5.2 Experimental Methodology

To collect the BLE advertisements, we use a BLE receiver [6] connected to a nearby computer. This computer also processes sensed parameters and runs the classifier. The sensor transmits BLE data packets containing an ID and raw data including three different color filters (red, green, and blue) and no filter clear component. Recorded decimal numbers are proportional to the intensity of each component. Before collecting the data, we configure the advertising rate of *BLE packets* for ScreenSense. Considering statistics on average people's stopovers on a website [19], the persistence of

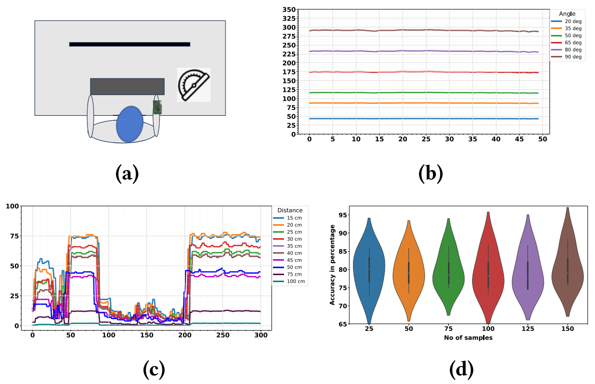


Figure 14: (a) Analysing angular and distance variation in wearable fashion, Recorded *Blue components* while playing the same video and placing the sensor at different angles (b) and distances (c). Pointing the hand horizontally and placing the hand at a distance of *20 cm* from screen records the largest values. As observed, both the angular and distance variation record similar RGB variations with different amplitudes. With 1X gain, sensor records RGB variations upto 1m . Accuracy with variable length window is shown with violin plots (d). Higher mean accuracy was recorded with 25 and 150 samples.

human vision and recording adequate fluctuations of RGB values for activity identification, we configure the device to transmit **5 advertisements per second.** We place the sensor near the screen in a dark room to get the true patterns first. We collect **2000 samples for each activity.**

Regarding sensor placement, our goal was to verify multiple issues: the operational range (distances/angles) of typical indoor light sensors, whether placement variation causes dissimilar patterns or not, and finally, for best output, in what fashion the sensors should be installed/used. Based on [5, 34], **we placed the sensor at a distance ranging from 10 cm to 100 cm and at an angular variation from 0-90 degrees relative to the screen** ***(both screen tilted and wrist tilted)***. As seen,sensor below 15 cm distance and 20 degree angle records too low magnitude to analyze with 1X gain setting to analyse [Figure 13].

To determine the number of samples required for classification, We start by randomly picking up a fixed number of observations from the training set. Next, we divide the original time series observation into multiple equal-length sub-sequences by choosing six different length slicing windows (consisting of *25,50,75,100,125,150* RGB samples). With overlapping and non-overlapping samples, all those windows contain an equal number of examples. To the best of our knowledge, **no previous research has explored variations of classification accuracy based on the number of sensed observations and optimizing the number for energy efficient classification.**

6 Evaluation

To evaluate the proposed system, we evaluate the system in a controlled environment by turning off the light sources and in real testbeds without changing the lights. Specifically, We compare the accuracy of the classifiers and determine the number of samples required without compromising classification accuracy. We discuss the efficacy of the timeGAN-based dataset augmentation. We also highlight the importance of incorporating clear filter data for some activity classes.

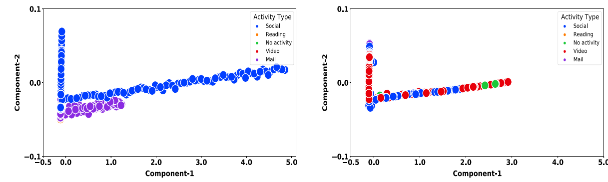


Figure 15: Comparing first two principal components of Captured (left) and TimeGAN generated examples (right). As observed, maximum variation occurred in *social platforms*, as it contains videos, images and texts

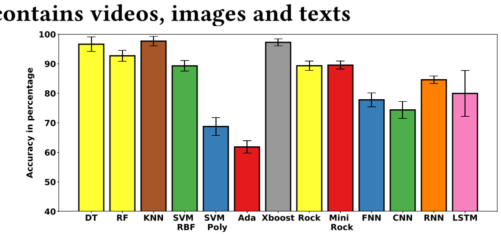


Figure 16: Variation of *Mean Accuracy* along with *Standard Deviations* for tuned classifiers.

6.1 Sensor Setup and Classification Window Selection

After setting our sensor at various distances *(range: 15cm-100cm)* and angle *(range:20-90 degrees)*, we observe that variation in sensor placement generates **similar patterns** and the recorded values are not **inversely proportional to distance but proportional to the angle (Figure 14)**. Highest amplitudes were recorded with *the sensor facing towards the screen at a distance 20cm.* The best performance was observed with the window of 150 samples. However, accuracy with the window of 25 samples was the second best, **which offers classification with significantly less time and power requirements.** We fix the sample size to 25 for the next experiments.

6.2 Analysing Synthetic Examples

For generating synthetic examples, we first gather examples from controlled environments and their transitional and under light variations and use them to generate an equal number of synthetic examples utilizing TimeGAN architecture. Figure 15 demonstrates the distribution of the first two principal components of real and synthetic examples generated using TimeGAN. To measure the utility of the data, we build a dataset using synthetic and original data (70% synthetic, 30% original) and train KNN with previously determined best parameters. The accuracy with 10% random testset, after training with rest of the 90%, was 92%.

6.3 Performance in Controlled Environments

As observed (Figure 16), the overall classifying performance of ML algorithms are better than NNs in the controlled scenario. The first and second best results are achieved with *KNN* (**mean accuracy: *97.67%, F1 score= 0.89***) and *Xboost* (**mean accuracy: *97.25%, F1 score= 0.88***).

6.4 Performance Boost with Screen Brightness

We analyze some of the misidentifying examples from a random test set and with the highest performing classifiers: *KNN* and

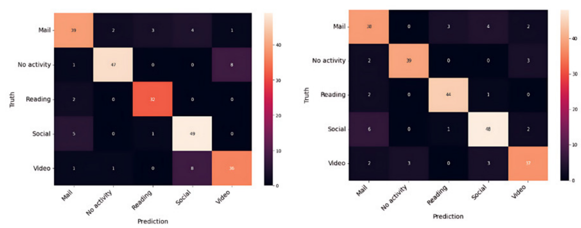


Figure 17: Confusion matrices for random test examples for *KNN* (left) and *Xboost* (right) with only RGB Data

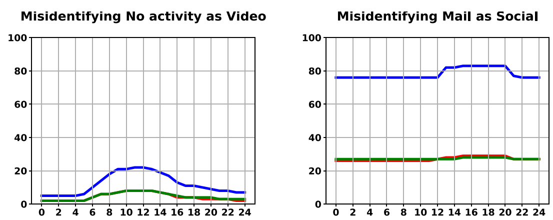


Figure 18: Certain patterns of misidentifying when KNN was trained with only RGB Data

Xboost. As discovered from confusion matrices of those classifiers (Figure 17), some of the wrong predictions display certain patterns. For instance, during screensaver *Ribbon* mode, the classifier would sometimes misclassify the event as a video due to the similarities in their RGB variations, leading to confusion (Figure 18). In spite of having similarities in RGB patterns, screen's brightness differs between screensaver mode and playing a video. For that, we have brightness information *(represented by unfiltered clear channel in the sensor)* in training set and re-train our classifier from scratch. As seen from Figure 19, still *KNN* ***(accuracy: 98.875%, F1 score= 0.93)*** and *Xboost* ***(accuracy: 98.91%, F1 score= 0.94)*** were the best performers with elevated accuracy/F1 score. We proceed with our analysis with *KNN* trained with 4-channel information *(R,G,B and Clear components)*, considering ease of deployment on resource-constrained edge devices than *Xboost*.

6.5 Performances at Realistic Testbeds

We conduct three experiments at three different realistic testbeds (Figure 21). They include samples taken in a dark room with unseen user, not included in the training set (*Testbed-1 (top)*), under unknown room light single display with a completely unseen screen settings (*Testbed-2 (middle)*), and unknown room light multi-display scenario (*Testbed-3 (bottom)*). All testbeds contain five activity types for classification, with the first having seen the content and the others having unseen content with practical variations *(zooming, varying playback speed)*.

Following the segmentation of the observations into 25 RGB-C sample sub-sequences, we employ our top-performing *KNN* classifier, which has been trained using RGB and Clear components in controlled scenarios, to determine the activity type represented by these sub-sequences. We observe that the classification accuracy has been degraded significantly in all the testbeds (*as low as **72.5%***). In the dark room scenario, the classifier failed to identify timeframes where the user was switching from one activity to another. At other testbeds, the classifier simply gets confused with patterns under light and practical variations of activities.

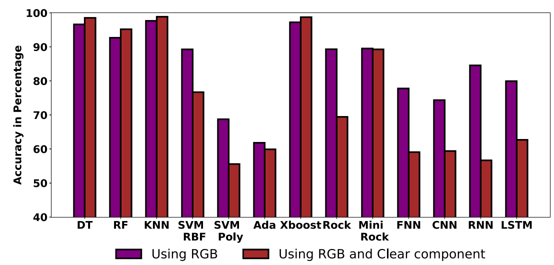


Figure 19: Recalculated Accuracy after addition of brightness information.

6.6 Performance after augmentation-retraining

We then add augmented data in the present training set and retrain our *KNN* classifier with this augmentation. Accuracy of *KNN* classifier pre and post-implementation of augmented data were recorded. As seen from Figure 22, in all three scenarios, performances were elevated (***upto 7.5%***) with the extraneous training set, and the with highest accuracy recorded was for *Testbed-1* ***(accuracy: 91.25%, F1 score= 0.81)***, compared to test set accuracy of ***79.3%*** with Random forest and ***70.1%*** with Naive Bias reported in [20]. Figure 23 (top) depicts a few examples where the introduction of extraneous training sets aid in accurate predictions of activity class, which were previously misidentified with the limited training set. However, there were a few timeframes that were still misidentified (bottom). After analysis, no common patterns of inaccurate predictions were observed.

6.7 Sensor Energy Consumption

To understand the energy requirement of ScreenSense sensor, we computer the average current consumption of the device. Figure 20 shows the current trace during one compute and transmission cycle For each screen event transmission, the device consumes an average current of only around 0.22mA. With an advertisement interval of 12 ms (5 adv/second), the sensor can run upto 45 days on a 3.3V lithium coin battery [31].

7 Discussion

Data acquisition setup: For ScreenSense, we set *TCS3475* sensor with 1X ADC gain with an integration time of 700 ms, which allowed us to read color values up to value 65535. For low light conditions or activity detection beyond 1 m, tuning ADC gain settings achieve this, though with the trade-off of higher energy consumption. The average current consumption can be reduced by reducing the sampling and transmission frequency. However, the frequency should be high enough to capture enough variations and enable real-time data acquisition.

Performance limiting factors: With ScreenSense, we have not

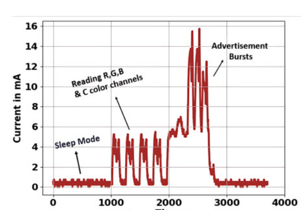


Figure 20: Current draw of the ScreenSense color sensor board

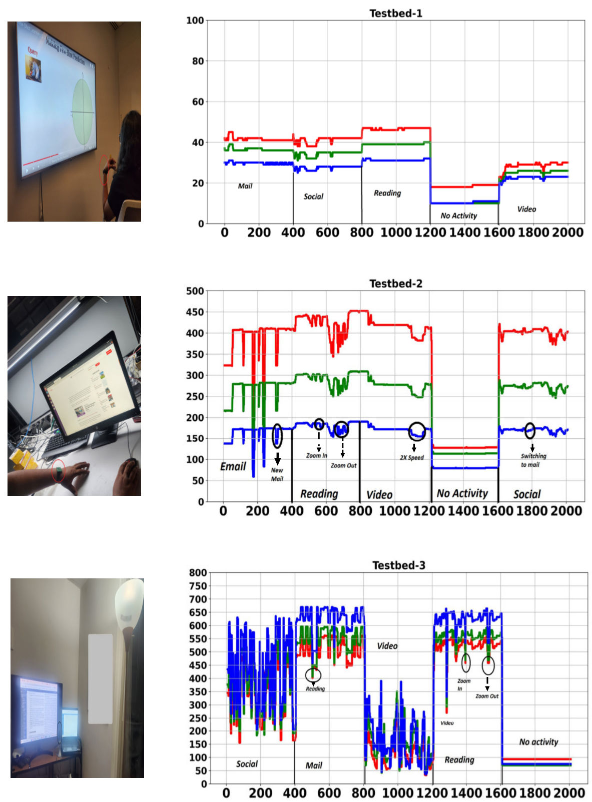


Figure 21: RGB signals at different test beds from screen activities. Each testbed comprises of 2000 samples including five activities with realistic variations. Collecting signal: (top) as wearable at dark conference room at *1m* distance from *LG 55" screen with custom setting*, (middle) as wearable in a lab with *HP 32" monitor* under *LED: 21W, 3000K* (middle), as fixed point device with *LG 34" & Ideapad S145 monitors* under *LED: 65W, 5000K* lamp (bottom)

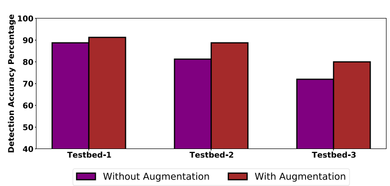


Figure 22: Comparing Testbed accuracy with training with and without augmented dataset

considered what happens when the sensor is positioned at mid-point and equal distance from two displays. We assume that while carrying as a smart device, user hand was not in motion, which can impact the final recordings. With low RGB amplitudes from screen, too bright room light in many scenarios may entirely overshadow on-screen information. ScreenSense can be arranged to be auto-synchronous with the active screen. However, additional arrangements may be called for to ensure users' actual involvement on a particular screen, like gaze control or face detection arrangements [4, 10], as the user may easily switch in between monitors or looking places other than monitor in realistic scenarios. Our training set includes a limited type of screen setting variations. For

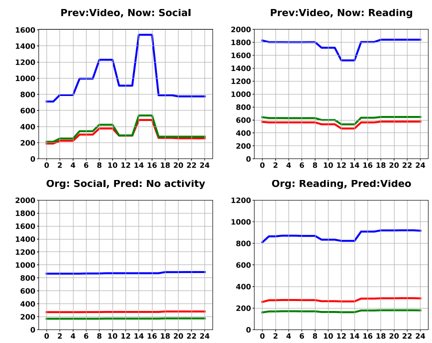


Figure 23: Correct Predictions with extraneous training set (top), Miscategorized examples recorded after retraining with extraneous set (bottom)

customized/specific settings, the performance of the classifier can be elevated by introducing few examples in the training set with that setting. With limited examples and features, ML algorithms have outperformed neural networks, but as the dataset and features expand, neural networks may surpass traditional ML algorithms. Finally, in real world, screen usage is highly dynamic, which demands augmentation/modification of the number of categories.

8 Conclusion and Future work

ScreenSense platform elicits the fact that, whether self-deployed or an open source RGB dataset from an indoor light sensor positioned near a screen, **can be revealing** of the on-screen activities of a digital user. ScreenSense will help experts understand **the risks associated** with the expected widespread indoor deployment of smart color sensor systems and establish guidelines for their placement in smart buildings. It also **alerts users** to permit their daylong exposure for open analysis. On the flip side, ScreenSense offers **enhanced privacy** compared to software-based identification or screen recording/snapshot methods. It acknowledges several drawbacks of at present indoor light sensors for passive screen sensing, including *energy and memory inefficiency for long-term data collection, accumulating needless derivatives or statistical features, lack of knowledge regarding the impact of sensor positioning/orientation on classifying performance, choice of best performing classifier with appropriate parameters for such dataset and finally, addressing performance depreciation in realistic scenarios.* ScreenSense explores the **multi-modality of indoor light sensors, where the existing framework can provide information regarding daylong light-source exposure [31].** In considering the privacy implications of ScreenSense, we assume smart indoor environments where continuous light-sensitive data is collected in close proximity to screens, with the potential for this information to be accessed by individuals other than the user, such as facility managers or cybercriminals. The applicability of the system is constrained in scenarios where light sensors are positioned at a greater distance from the screens or where data collection occurs at extended intervals.

Given that *KNN* demonstrated strong performance, ScreenSense offers insights like **similar on-screen activities within basic color-brightness feature spaces are likely to exhibit significant localities.** For exhibiting superior performance in unfamiliar atmospheres, ScreenSense is anticipated to classify screen content **independent of particular sensing device**, as the classification is based on **the patterns rather than absolute values** of variables. Considering one device per machine, ScreenSense provides distinct advantages **for mass deployment** due to its low cost, flexibility, and energy efficiency compared to the majority of sensors. Additionally, it provides flexibility in deployment within a meter with minimum gain settings, **allowing it to be worn, placed at the desk, or positioned near monitors.** It supports **off-board storage and classification for long-term analysis.** As a result, ScreenSense will be helpful for the physiological and psychological study related to on-screen behavior, associating social website hours on learning/social interactions/sleep duration, prediction of human personalities and so on.

In the future, we plan to analyse the maximum ratio of indoor intensity light to screen light for successful classification. As ScreenSense runs the classifier to a distant device, we plan to calculate off-board parameters, such as processing time, daylong memory requirement, latency, and power consumption. On-board classification approach is also planned for places like commercial buildings and offices, where resource availability may not be an issue and activity detection may be intended only during business hours.

9 Acknowledgement

The authors gratefully acknowledge the support provided by the National Science Foundation under Awards NSF #2326406, NSF #2217071, and NSF #2144940 and the Commonwealth Cyber Initiative (CCI). We appreciate the guidance and resources that have been instrumental in advancing our work.

References

[1] 2023. Activtrak. https://www.activtrak.com/.
[2] 2023. Teramind. https://www.teramind.co/.
[3] Amit Adate, Soroush Shahi, Rawan Alharbi, Sougata Sen, Yang Gao, Aggelos K Katsaggelos, and Nabil Alshurafa. 2022. Detecting screen presence with activity-oriented RGB camera in egocentric videos. In *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 403–408.
[4] Daniel Avrahami, Eveline van Everdingen, and Jennifer Marlow. 2016. Supporting Multitasking in Video Conferencing using Gaze Tracking and On-Screen Activity Detection. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*. 130–134.
[5] Tanja Blascheck, Anastasia Bezerianos, Lonni Besançon, Bongshin Lee, and Petra Isenberg. 2018. Preparing for perceptual studies: Position and orientation of wrist-worn smartwatches for reading tasks. In *Proceedings of the Workshop on Data Visualization on Mobile Devices held at ACM CHI*.
[6] Bluegiga BLED112 Bluetooth Smart Module. 2016. Silicon Labs. https://www.silabs.com/documents/public/data-sheets/BLED112-DataSheet.pdf.
[7] Timothy M Brown, George C Brainard, Christian Cajochen, Charles A Czeisler, John P Hanifin, Steven W Lockley, Robert J Lucas, Mirjam Münch, John B O'Hagan, Stuart N Peirson, et al. 2022. Recommendations for daytime, evening, and nighttime indoor light exposure to best support physiology, sleep, and wakefulness in healthy adults. *PLoS biology* 20, 3 (2022), e3001571.
[8] Sean W Cain, Elise M McGlashan, Parisa Vidafar, Jona Mustafovska, Simon PN Curran, Xirun Wang, Anas Mohamed, Vineetha Kalavally, and Andrew JK Phillips. 2020. Evening home lighting adversely impacts the circadian system and sleep. *Scientific reports* 10, 1 (2020), 1–10.
[9] Supriyo Chakraborty, Wentao Ouyang, and Mani Srivastava. 2017. LightSpy: Optical eavesdropping on displays using light sensors on mobile devices. In *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2980–2989.
[10] Giuseppe De Palma, Emma Sala, Sofia Rubino, Simone Dalola, Marco Ferrari, Daniele Marioli, Pietro Apostoli, Cesare Tomasi, Francesca Righetti, Flavia Mattioli, et al. 2023. Objective Evaluation of Active Interactions between the Operator and Display Screen Equipment Using an Innovative Acquisition System. *Bioengineering* 10, 6 (2023), 686.
[11] Katherine L Downing, Xanne Janssen, and John J Reilly. 2019. Feasibility of wearable cameras to assess screen time and time spent restrained in children aged 3 to 5 years: a study protocol. *BMJ open* 9, 5 (2019), e028265.
[12] Sindhu Kiranmai Ernala, Moira Burke, Alex Leavitt, and Nicole B Ellison. 2020. How well do people report time spent on Facebook? An evaluation of established survey questions with recommendations. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–14.
[13] Bilkis Jamal Ferdosi, Mohaomd Sadi, Niamul Hasan, and Md Abdur Rahman. 2023. Tracking Digital Device Utilization from Screenshot Analysis Using Deep Learning. In *Proceedings of International Conference on Data Science and Applications: ICDSA 2022, Volume 1*. Springer, 661–670.
[14] Richard Ribón Fletcher, Daniel Chamberlain, D Richman, Nicolas Oreskovic, and Elsie Taveras. 2016. Wearable sensor and algorithm for automated measurement of screen time. In *2016 IEEE Wireless Health (WH)*. IEEE, 1–8.
[15] Garrett Hisler, Jean M Twenge, and Zlatan Krizan. 2020. Associations between screen time and short sleep duration among adolescents varies by media type: evidence from a cohort study. *Sleep medicine* 66 (2020), 92–102.
[16] Ashton Holmes, Sunny Desai, and Ani Nahapetian. 2016. Luxleak: capturing computing activity using smart device ambient light sensors. In *Proceedings of the 2nd Workshop on Experiences in the Design and Implementation of Smart Objects*. 47–52.
[17] Josh Howarth. 2024. *Alarming Average Screen Time Statistics*. Technical Report. DataReportal, Reuters.
[18] Linda K. Kaye, Amy Orben, David A. Ellis, Simon C. Hunter, and Stephen Houghton. 2020. The conceptual and methodological mayhem of "screen time". *International Journal of Environmental Research and Public Health* 17, 10 (2020), 3661.
[19] Gitte Lindgaard, Gary Fernandes, Cathy Dudek, and Judith Brown. 2006. Attention web designers: You have 50 milliseconds to make a good first impression! *Behaviour & information technology* 25, 2 (2006), 115–126.
[20] Tommaso Martire, Payam Nazemzadeh, Alberto Sanna, and Diana Trojaniello. 2018. Digital screen detection enabled by wearable sensors: application in ADL settings. In *2018 International Conference on Intelligent Systems (IS)*. IEEE, 584–588.
[21] Dean McDonnell, Evan Hunt, Laura Griffin, and Ryoko Sasamoto. 2019. Evaluating the relationship between multimedia viewing, sedentary behavior, and executive function: A systematic review. *Journal of Technology in Behavioral Science* 4, 4 (2019), 318–331.
[22] Chulhong Min, Euihyeok Lee, Souneil Park, and Seungwoo Kang. 2019. Tiger: Wearable glasses for the 20-20-20 rule to alleviate computer vision syndrome. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*. 1–11.
[23] Durjoy Mistry, MF Mridha, Mejdl Safran, Sultan Alfarhood, Aloke Kumar Saha, and Dunren Che. 2023. Privacy-Preserving On-Screen Activity Tracking and Classification in E-Learning Using Federated Learning. *IEEE Access* (2023).
[24] Anthony D Miyazaki. 2008. Online privacy and the disclosure of cookie use: Effects on consumer trust and anticipated patronage. *Journal of Public Policy & Marketing* 27, 1 (2008), 19–33.
[25] Ashish Pandharipande, David Caicedo, and Xiangyu Wang. 2014. Sensor-driven wireless lighting control: System solutions and services for intelligent buildings. *IEEE Sensors Journal* 14, 12 (2014), 4207–4215.
[26] Ruth K Raanaas, Katinka Horgen Evensen, Debra Rich, Gunn Sjøstrøm, and Grete Patil. 2011. Benefits of indoor plants on attention capacity in an office setting. *Journal of Environmental Psychology* 31, 1 (2011), 99–105.
[27] Nilam Ram, Xiao Yang, Mu-Jung Cho, Miriam Brinberg, Fiona Muirhead, Byron Reeves, and Thomas N Robinson. 2020. Screenomics: A new approach for observing and studying individuals' digital lives. *Journal of adolescent research* 35, 1 (2020), 16–50.
[28] Byron Reeves, Thomas Robinson, and Nilam Ram. 2020. Time for the human screenome project.
[29] Grand View Research. 2022. *Smart Lighting Market Size, Share & Trends Analysis Report By Component (Hardware, Software, Services), By Connectivity (Wired, Wireless), By Application (Indoor, Outdoor), By Region, And Segment Forecasts, 2023 - 2030*. Technical Report. DataReportal, Reuters.
[30] Sara Rosenblum, Elad Yom-Tov, et al. 2017. Seeking web-based information about attention deficit hyperactivity disorder: where, what, and when. *Journal of medical Internet research* 19, 4 (2017), e6579.
[31] Tushar Routh, Nurani Saoda, Md Fazlay Rabbi Masum Billah, and Brad Campbell. 2022. Low Cost Light Source Identification in Real World Settings. In *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 226–234.
[32] Lorenz Schwittmann, Viktor Matkovic, Torben Weis, et al. 2016. Video recognition using ambient light sensors. In *2016 IEEE international conference on pervasive computing and communications (PerCom)*. IEEE, 1–9.
[33] Tung Vuong, Giulio Jacucci, and Tuukka Ruotsalo. 2017. Watching inside the screen: Digital activity monitoring for task recognition and proactive information retrieval. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–23.
[34] Florian Wahl, Jakob Kasbauer, and Oliver Amft. 2017. Computer screen use detection using smart eyeglasses. *Frontiers in ICT* 4 (2017), 8.
[35] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. 2019. Time-series generative adversarial networks. *Advances in neural information processing systems* 32 (2019).