

# AN IMAGE CLASSIFICATION-BASED APPROACH TO AUTOMATE VIDEO PLAYING DETECTION AT SYSTEM LEVEL

Eric Liu<sup>1</sup>, Samuel Walcoff<sup>2</sup>, Qi Lu<sup>3</sup> and Yu Sun<sup>4</sup>

<sup>1</sup>Arcadia High School, Arcadia, CA, 92697

<sup>2</sup>Department of Computer Science, University of California, Santa Cruz  
Santa Cruz, CA 95064

<sup>3</sup>Department of Social Science, University of California, Irvine  
Irvine, CA, 92697

<sup>4</sup>Department of Computer Science, California State Polytechnic University,  
Pomona, CA, 91768

## **ABSTRACT**

*Tech distraction has become a critical issue on people's work and study productivity, particularly with the growing amount of digital content from the social media site such as Youtube. Although browser-based plug-ins are available to help block and monitor the sites, they do not work for all scenarios. In this paper, we present a system-level video playing detection engine that captures screenshots and analyze the screenshot image using deep learning, in order to predict whether the image has videos in it or not. A mobile app has also been developed to enable parents to control the video playing detection remotely.*

## **KEYWORDS**

*Machine learning, Tech distraction, Image classification*

## **1. INTRODUCTION**

Raising discussion about Digital responsibility have caught people's attention. Now people get distracted easier than ever because the development of technologies. For example, students usually having difficulty to focus on studying because they are internally preoccupied by new messages on their phone. The process of checking their phones regularly hugely influence their study result thus delays time needed to acquire academic knowledge. People are aware of how technologies distract their normal life and figured out that they need to use technology wisely.

Furthermore, apps installed on devices mostly distract users attention. For example, Youtube is the most commonly used video-sharing website. A lot of users would open Youtube first when they use their phones. Some of them are addicted to Youtube now. Even though they had plans on doing other things, they cannot stop watching the "last" video. People now realized how important it is to control their addiction.

As we all know, there are a lot of blacklist programs in the market, they usually come in the form of Browser Extension [1], and usually do their work perfectly. However, there are quite a few limitations on these traditional blacklist programs, such as the failure against incognito tab, or the inconvenience when the user purposely use another browser to bypass it, which leads to the purpose of this development: a blacklist program that cannot be bypassed with a simple

incognito tab, with many additional features such as remotely customize the list of sites, and blockage based on a certain keyword. Currently, the application “Blacklister” has successfully been uploaded to Google Play [2][3]. Based on the future online reviews, the direction of this development might adjust accordingly, whether it is more type of action instead of brutally pops a warning message or shut down the program, or more accurate ways of detecting such as keyword combination. [4]

## 2. RELATED WORK

One instance of similar program/extension we would like to compare is one of the common blocker extensions in chrome: blacklist. It currently holds a rating of 3.7 out of 5, from some of the comments, we can see that one big problem of this extension is its limitation on having to load the designated page first to be able to block it, which is pretty inconvenient and contradicting.

Super quick and easy! BUT, you have to load the page to block it!?! Some are dangerous, but to block it you need to visit the site... If you could manually add sites, say into some external file, this would be the only "Blacklist" extension you would need!

Was this review helpful?  Yes  No [Reply](#) | [Mark as spam or abuse](#)

Figure 1. a selected review from Chrome Webstore

Aside from that, although this program is working as intended:



Figure 2. showcase of the program

When the user goes into the incognito page of their browser, which most of the commonly used browser has as one of its functions, the designated page can still be freely visit, this entire “unblocking” process can take as little as one tap on the keyboard:

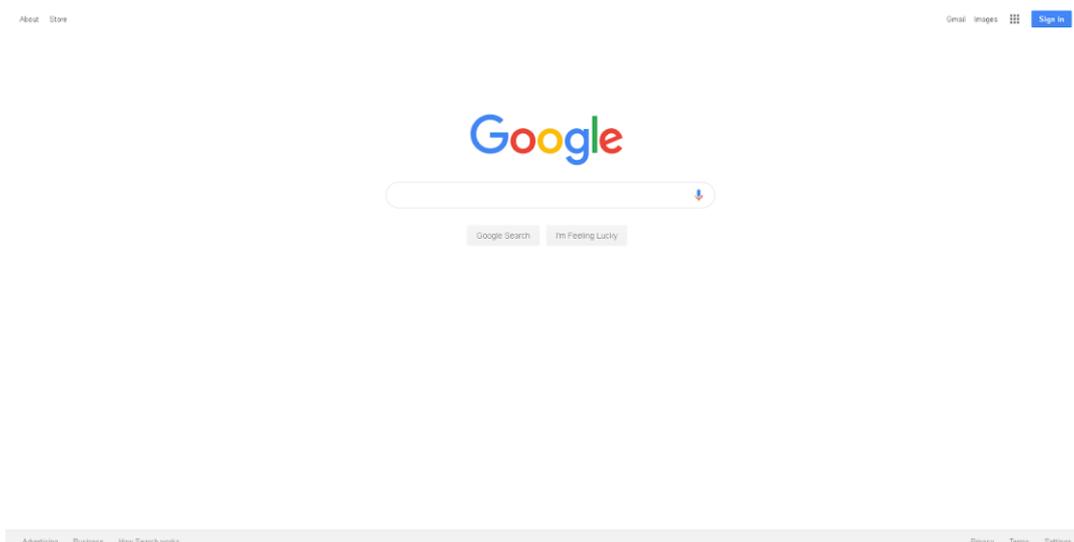


Figure 3. the limitation of extensions against incognito mode[5]

Such limitations exist not only in this specific program, but also in many other similar programs that are frequently used by people, which is what our program is designed to help.

People lack the habit of logging their diet, so it is important to have a simple way for people to track their diet. Controlling diet [6] requires persistence and motivation. While it is difficult to see the subtle changes, if people could monitor the steps and efforts clearly, it can provide another effective way of gaining confidence and motivation.

### 3. DEVELOPMENT OVERVIEW

Development of this program is not a piece of cake, the development went through many phases, switching different concepts and different methods. At first, the main concept of the development is fairly narrow, creating a video monitor program [7] that detects whether the user is currently watching a video on the internet, and potentially detect its origin to make sure the program runs without error, that concept mashed with machine learning sounds very good and applicable, but that idea was quickly put out due to the limitation on the information we can get from a packet. Most of the information inside a packet are encrypted and without a key there is simply no way to decipher it. Occasionally we can catch some DNS packets containing the information we need but they appear randomly and seem to have no pattern whatsoever, which is why that idea was stopped. Unlike the normal top-down approach of development [8], our programming approach is more of a bottom-up [9] one. We did not have a clear target at the beginning, we have a general sense of what this application might look like, but we also change things as we move along. Which is what makes our application unique.

### 4. SOLUTION

Problems we met during the development process include how to make the program recognize that the user is using unauthorized web activity, just like what they do for Chrome extensions. For the problems we have switched between many approaches. As mentioned above the first approach is the packet capturing and try to analyze it with the aid of AI. Although that plan eventually failed due to the limitation on how much information can a computer, using AI as a way to execute our goal is a good concept and worth keeping. Instead of having an AI [10][11] overseeing how many packets are going in and out and break into them, we switched to a

completely different path of on-screen recognition [12], because we can gather a lot more useful information from our program and turn them into something that can actually work. When we go on to a website and try to entertain ourselves, those websites usually have some very significant figures that we can use to determine them. Take Youtube [13], one of the biggest entertaining websites in the world, when we open Youtube, there are many things, especially texts, on the screen, that can be easily used to act as an information giver:

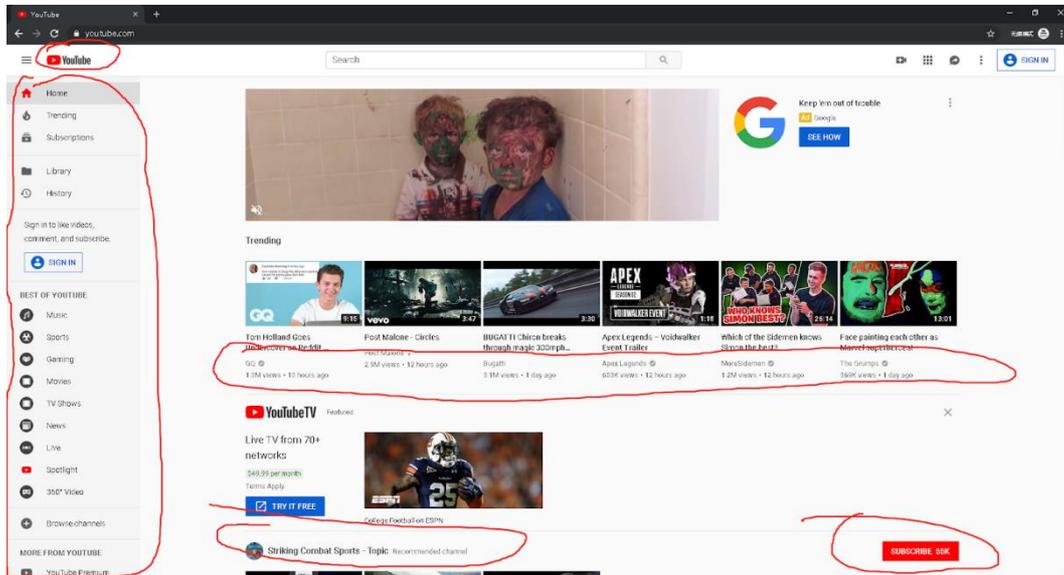


Figure 4. A screenshot of the captured computer screen using Youtube

With the aid of this information, our problem can be easily solved. Simply analyze the screen [14] for preset keywords and the program can easily determine whether it is in its blacklist or not.

```
public static void checker(blacklist bl) throws Exception{
    while(getBoolFire() == false){
        Thread.sleep(1000);
    }
    boolean bool = false;
    syncLists(bl);
    while(bool == false){
        bool = detectText(screencap(), bl);
        Thread.sleep(5000);
    }
    new myGui(bl);
}
```

Figure 5. part of code that checks the screen for a list of keywords.

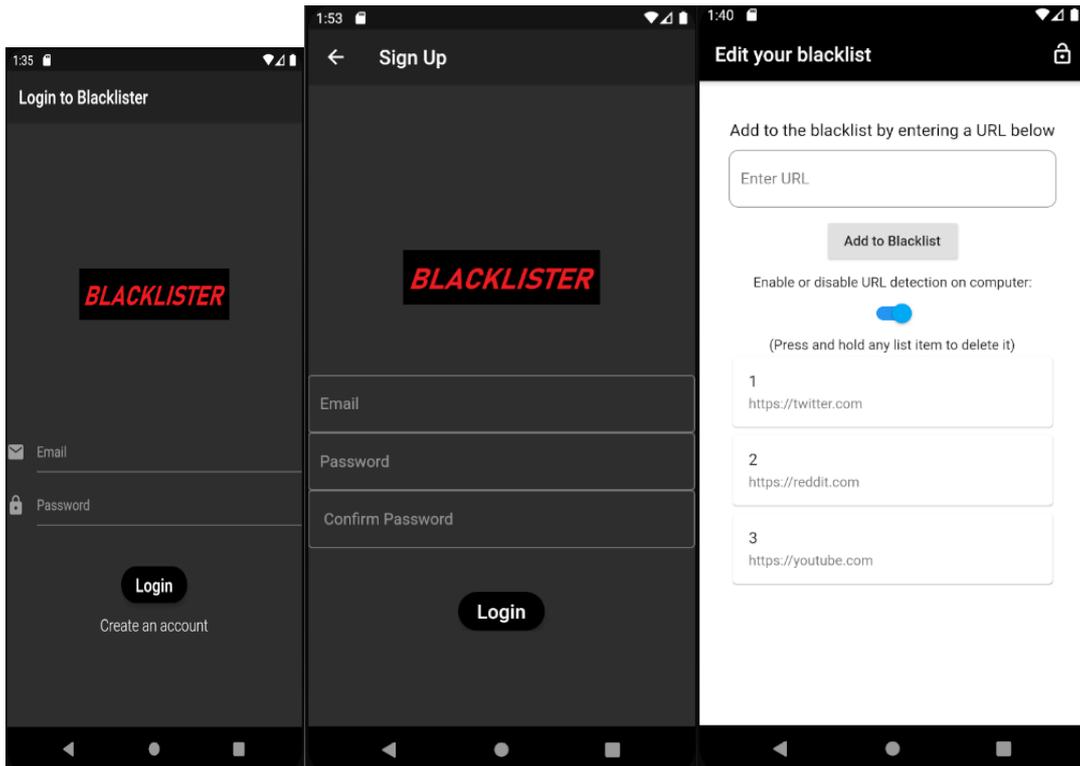


Figure 6. Screenshots of the app

The login page is displayed first and requires email and password. Once both are entered, the user must press the “Login” button to navigate to the third page containing their blacklist editor. If the user does not yet have an account, they may hit the “Create an account” button at the bottom of the first page to be directed to the signup page.

In the signup page the user must enter their email and then set and confirm their password. Once that is complete, they can press the “Login” button which will direct them to the blacklist editor. The user is also free to navigate back to the Login page using the arrow at the top left corner of the page.

On the blacklist editor, the user is free to add, delete, disable, and enable the blacklist. To add, they must type the URL into the text box and press “Add to Blacklist”. To delete, they can long press any list item and it will disappear. In the top right of the page, they can press the unlock button to logout and navigate back to the login page.

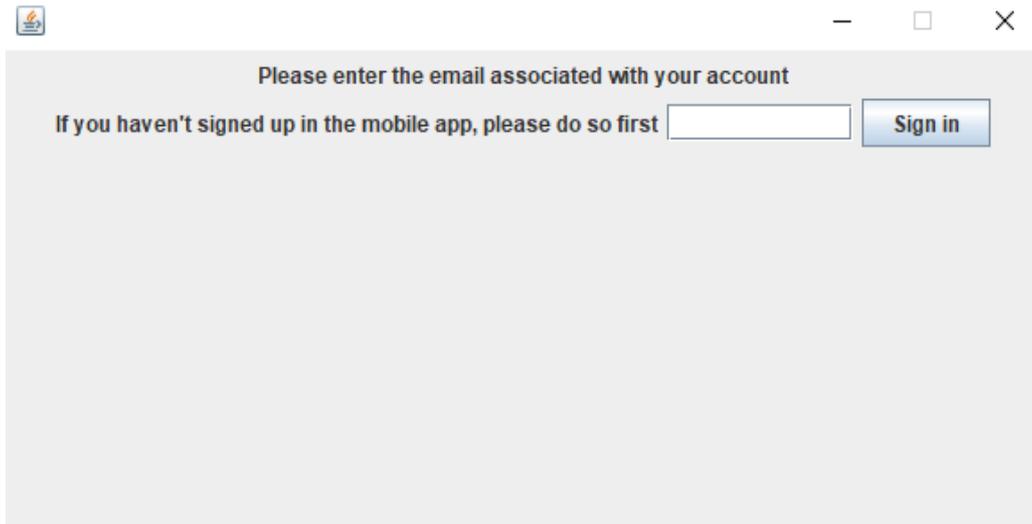


Figure 7. Screenshots of the desktop admin app

This window appears when the user opens the app. The user may enter their email in the text box and press “Sign in”. After doing so, the window will disappear and the program will begin running in the background.

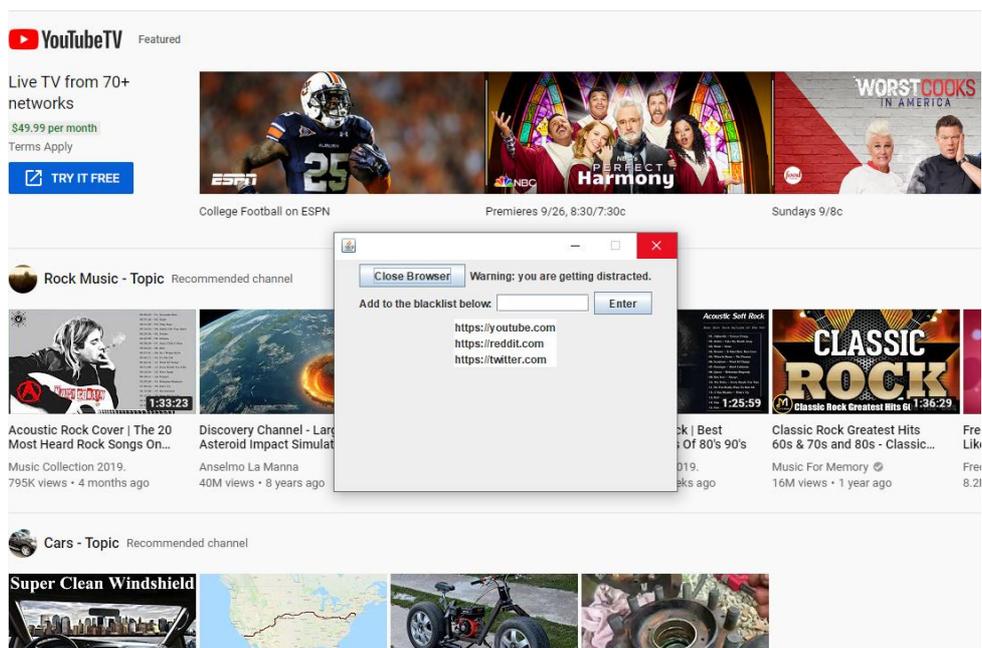


Figure 8. Screenshots of the blocking feature

Upon browsing a blacklisted website, this window will pop up giving the user a warning. The user is then only allowed or leave the website. They are also not allowed to close the window until some action is taken.

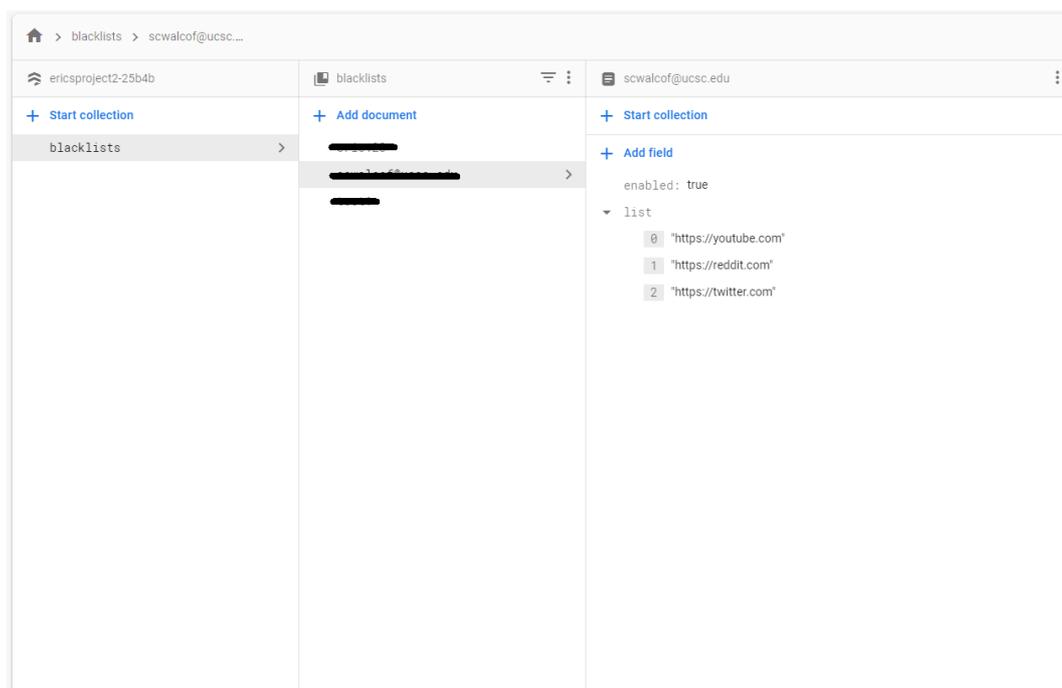


Figure 9. Screenshots of the Firebase database structure

Here is the database where each user's blacklist is stored. The blacked out document titles are where the emails are stored. The collections contain the list of blocked URLs as well as the "enabled" variable that controls the computer application URL monitor.

## 5. CONCLUSIONS AND FUTURE WORK

Thus, a sketchy build has been created with dedication, although it is nothing tricky, it offers a solution to many problems and limitations of current blacklist application, and in the future we have the confidence to make it one step further, perfecting this application such that it can be widely used in areas such as education and workplace productivity.

## REFERENCES

- [1] Leonard, Huw, and Gary Farmaner. "Method and system for administering a customer loyalty reward program using a browser extension." U.S. Patent Application 09/908,615, filed April 18, 2002.
- [2] Viennot, Nicolas, Edward Garcia, and Jason Nieh. "A measurement study of google play." In ACM SIGMETRICS Performance Evaluation Review, vol. 42, no. 1, pp. 221-233. ACM, 2014.
- [3] Liu, Charles Zhechao, Yoris A. Au, and Hoon Seok Choi. "Effects of freemium strategy in the mobile app market: An empirical study of google play." Journal of Management Information Systems 31, no. 3 (2014): 326-354.
- [4] Reddington, Thomas B. "Keyword search automatic limiting method." U.S. Patent 4,554,631, issued November 19, 1985.
- [5] Lerner, Benjamin S., Liam Elberty, Neal Poole, and Shriram Krishnamurthi. "Verifying web browser extensions' compliance with private-browsing mode." In European Symposium on Research in Computer Security, pp. 57-74. Springer, Berlin, Heidelberg, 2013.

- [6] Young, Simon N. "The use of diet and dietary components in the study of factors controlling affect in humans: a review." *Journal of Psychiatry and Neuroscience* 18, no.5 (1993): 235.
- [7] Buxton, J., M. White, and D. Osoba. "Patients' experiences using a computerized program with a touch-sensitive video monitor for the assessment of health-related quality of life." *Quality of Life Research* 7, no. 6 (1998): 513-519.
- [8] Craddock, Deborah, Cath O'Halloran, Kathryn Mcpherson, Sarah Hean, and Marilyn Hammick. "A top-down approach impedes the use of theory? Interprofessional educational leaders' approaches to curriculum development and the use of learning theory." *Journal of Interprofessional Care* 27, no. 1 (2013): 65-72.
- [9] Chamaret, Aurélie, Martin O'Connor, and Gilles Récoché. "Top-down/bottom-up approach for developing sustainable development indicators for mining: application to the Arlit uranium mines (Niger)." (2007).
- [10] Neches, Robert, Richard E. Fikes, Tim Finin, Thomas Gruber, Ramesh Patil, Ted Senator, and William R. Swartout. "Enabling technology for knowledge sharing." *AI magazine* 12, no. 3 (1991): 36-36.
- [11] Polit, Stephen. "R1 and beyond: Ai technology transfer at digital equipment corporation." *AI Magazine* 5, no. 4 (1984): 76-76.
- [12] Lee, Dar-Shyang, Lee-Feng Chien, Aries Hsieh, Pin Ting, and Kin Wong. "On-screen guideline-based selective text recognition." U.S. Patent 8,515,185, issued August 20, 2013.
- [13] Alcock, Shane, and Richard Nelson. "Application flow control in YouTube video streams." *ACM SIGCOMM Computer Communication Review* 41, no. 2 (2011): 24-30.
- [14] Sheiner, Lilach, Jessica L. Demerly, Nicole Poulsen, Wandy L. Beatty, Olivier Lucas, Michael S. Behnke, Michael W. White, and Boris Striepen. "A systematic screen to discover and analyze apicoplast proteins identifies a conserved and essential protein import factor." *PLoS pathogens* 7, no. 12 (2011): e1002392.