PROCRASH: A SOLUTION TO PROCRASTINATION BY LIMITING ONLINE DISTRACTIONS USING OPTICAL CHARACTER RECOGNITION

Amanda Zhu[1], Baoyu Yin[2] and Yu Sun[3]

[1]Portola High School, Irvine, CA 92618

[2]Lehigh University, PA 18015

[3]California State Polytechnic University, Pomona, CA 91768

ABSTRACT

For this project, I decided to relieve the tension of procrastination that commonly happens in students and adults. To find a solution to this, I created a program that uses Google Cloud Vision API (Optical Character Recognition) to detect the distracting forms of media such as Twitter, YouTube, and Facebook, and counts the number of times the user visits these websites. After a certain number of visits, the program sends a notification to remind the user to stay focused. If the user ignores the notification message while staying on the unapproved website, the program forces the tab to close. This application was applied to a small user study where a qualitative evaluation of the approach was conducted. After collecting data for two weeks, it concluded that the program was able to effectively reduce and limit the uses of online distractions, allowing the user to manage their time more efficiently by staying off websites they should not visit.

KEYWORDS

Procrastination, Optical detection, Software development

1. INTRODUCTION

Procrastination harms students' schoolwork, grades, and possibly their overall health [1, 2, 3]. According to a survey conducted by Study Mode, an education company, 87% of students say they procrastinate [4]. More than half of high school students (56%) say they procrastinate because they get distracted by other things that attract them. Two of the most popular methods of procrastination are watching TV and browsing social media [8, 9]. As technology becomes more prevalent in society, it will only become easier to procrastinate online. A study in the Psychological Bulletin by Piers Steel , a University of Calgary professor, concluded that about 5% of people were chronic procrastinators in 1978 while about 26% in 2007 [5].Students who procrastinate tend to experience higher levels of stress and anxiety, which lead to side effects such as low self-esteem and depression [10, 11, 12]. Once students start receiving more assignments and larger projects, students who procrastinate have shown that they will receive lower grades than others. So, procrastination will create a cycle of bad grades and low self-confidence that can be difficult for students to overcome.

Due to the COVID-19 pandemic, students throughout the country have been getting educated online through Emergency Distance Learning. While technology can be beneficial and has become part of our daily lives, it can also be distracting. It requires students' discipline to stay on track and pay attention to their homework and lessons [6, 7].

DOI:10.5121/ijcsa.2021.11401

Procrash is an app that aims to assist those who struggle with procrastination by monitoring and blocking potentially distracting websites. The user can customize the list of blocked websites as well as apps. Procrash also utilizes a notification tool to remind the user to stay on task.

Table 1. Comparison of application features

Benefits	Offtime	Zero Willpower	Stay Focusd	Procrash
Preset List of Websites/Apps		✓	✓	✓
Blocks Websites	✓	✓	✓	✓
Blocks Apps	✓		✓	✓
Notification				✓
Optical Character Recognition				✓

When compared to competing apps such as Offtime, Zero Willpower, and Stay Focusd, there are several similarities, but many noticeable differences as well (see Table 1). All of them can block websites, while only a select few use a preset list of websites/apps, and can block apps as well. Procrash is the only program that includes all three attributes, a notification feature, and implements optical character recognition using Google Vision API.

Table 2. Number of visits with and without Procrash

IV: Usage of program	DV: Number of Visits							
	Day of Week							Average
	S	M	T	W	Th	F	S	
Week 1: Without program	20	14	16	17	13	21	23	17.8
Week 2: With program	14	12	9	11	3	10	15	10.6

In a user study, data was collected every day for two consecutive weeks. The first week's data is a control group. The program was not involved. The second week's data is an experimental group. The program was used (seeTable2).All of the data was collected at the same time each day, each for ten minutes, and while the user was doing the same activity (math homework).

As seen in the data table and graph, the number of visits to distracting websites and apps when the use of the program was significantly lower than when the program was not used. Throughout the entire course of this project, the highest number of visits was 23 times, while the lowest number of visits was 3 times (a difference of 20 visits). The average of week one was 17.8 visits and the average of week two was 10.6 visits (a difference of 7.2 visits and a reduction of 40.45%).This shows that the program effectively reduced and limited the use of distracting media, allowing the user to manage their time more efficiently by staying off pages they should not be on.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, followed by presenting the related work in Section 5. Finally, Section6 gives the concluding remarks, as well as pointing out the future work of this project.

2. CHALLENGES

2.1. Challenge 1: Detecting the Text from Screenshot Image with Accuracy

Traditional management of volunteers by event supervisors include email, texting, or online chat services. However, these methods of communication maybe slow and require extra planning. This application aims to provide instant management of events and volunteers. Appropriated for multiple types of users from volunteers to supervisors, this application must facilitate data communication between the supervisors and the volunteers. Volunteers must be able to see event data and join events the supervisors have created, and supervisors must be able to view volunteer profile information, create events, and manage volunteers. When a supervisor creates an event, there must be some way for volunteers to view the event and join the event, and when a volunteer joins an event, the supervisor must be able to view the volunteer and his/her profile. Real-time data updates must be available for users, so no communication errors occur. To solve this challenge, a cloud database will be used, so the application can read and write data at any moment. Any information that users log, including hours, events, or profile changes, will be written to the cloud database. This will allow other instances of the application to read the changes that have been made.

2.2. Challenge 2: Larger User Study

Another challenge is finding a way to conduct a larger user study. It is difficult to share the Python program with other people since not everyone has Python installed on their computer. Also, some of the code is operating system-specific. Because of this, testing was limited to one user, which does not give the most accurate data. The experiment takes two weeks of consistent testing, and not everyone may have the time, leading to a limited pool of testers. It is also a challenge to find people to test the program.

2.3. Challenge 3: Force Closing the Browser Application/Tab

Selecting the most effective method of closing the browser can also be a challenge. There are multiple ways of closing the browser, including force closing the window, force closing the tab, or redirecting the user to another application. Force closing the entire window risks losing all the other tabs in the same window and possibly work.

3. SOLUTION

This program monitors and tracks the different media being use outside of the premises of school or work. By implementing Google Cloud Vision API, the program is able to identify distracting websites and apps and how much time the user spends on them. The program takes a screenshot of the user's computer screen every ten seconds, and the image is uploaded to Cloud Storage with libraries of text. Then, a Cloud Function is triggered, which uses the Vision API to extract the text. The program utilizes an internal counter that increases by one each time a distracting form of media is detected in the screenshot, for example, Instagram, Twitter, Reddit, Facebook, and so on. Once the counter reaches ten, the program will prompt the device to send a notification that reminds the user to stay on task. After this, the program continues to take screenshots and add them to the counter, and once the counter reaches fifteen, the window with the distraction will automatically be closed. After the window closes, the counter restarts at zero. However, if the screenshot shows that the user is on task, for example, Google or Khan Academy, the counter decreases by one every two minutes. The goal for this project is to test if the software helps to reduce wasted time on computers, by collecting data on the time spent on/off task with and without the program.

The screenshot feature is implemented through the PyAutoGUI library. When a screenshot is taken, an image object is returned, and it is saved to the user's computer.

```
pyautogui.screenshot('./temp/temp.png')
```

Every screenshot replaces the previous one, because the above line is executed in a loop and the image name does not change. Then, the program uses Google Cloud Vision API to extract text from the screenshot.

```
detect_text('./temp/temp.png', finalresult, calcScore)
```

The image is uploaded to Cloud Storage with libraries of text. Then, a Cloud Function is triggered, which uses the Vision API to extract the text. The extracted text is saved into a local variable in detect _text and passed through the callback function calcScore, which modifies a global variable score.

```
if url in text.description: saveData('experiment2_' +
        str(sys.argv[3])) score = score + 1
```

Score is incremented by one if the URL matches a URL from the blacklist. Firebase Realtime Database is used to store data in real time. The first step to utilizing the database is to connect it to a variable.

```
firebase =  firebase.FirebaseApplication("https://procrash-
            d8b5f.firebaseio.com/", None)
```

Whenever the black list is updated by the user in the mobile app, the list is automatically refreshed to contain the most recent information. The program receives a list of websites from the database and saves it as result. Because the statement is in a loop, result is frequently updated. Result is used with Google Cloud Vision API to determine if the URL is on the blacklist.

```
result = firebase.get("Blacklist", "").get('-
    M1rexaILOfsIJ1oM_Gl').get('Sites')
```

In addition to the Python program, I also used Swift to create a supplementary iOS app. This app allows the user to add and delete from the list of websites being monitored and blocked. The app comes with a preset list of websites, but the user can customize the list according to their preference. Firebase was imported to Swift so that when the user adds or removes a website, the new data would be sent to the database. After this, the list automatically refreshes so it always has the most current information. They can also start or stop monitoring their activity with a simple touch of a button. When the user clicks start, Fire base is triggered to start the program on the user's computer, and the button changes to stop so the user can stop the program as well.

4. EXPERIMENT

To evaluate the accuracy of Procrash, three experiments were conducted: detecting text from the screenshot with accuracy, a user study, and a survey.

The first experiment tests the accuracy of Procrash's ability to detect text from the screenshot. Adversarial analysis was use d in an attempt to fool the program by supplying deceptive input. False positives that were supplied include white listed sites that reference a blacklisted site and google results for a black-listed site. False negatives that were supplied include windows that overlap the browser, the mouse cursor overlapping the website URL, and hiding the URL in full screen. By implementing fuzzy string matching, a type of search that finds matches even when users misspell words or enter only partial words for the search, the probability of the program identifying false positives and false negatives decreases.

The second experiment tests the overall effectiveness of Procrash through a user study. During the first week, the program ran in the background of the user's computer and collected the frequency of visiting sites on the blacklist without interference (ground truth). During the second week, the was also used, but with interference (experiment). The data from both weeks was saved to Firebase, and then the data was downloaded to produce the graph below. The third experiment is a survey completed by twenty people. The data was used for feedback regarding the design and effectiveness of the program.

According to the results shown above, 75% of the participants are willing to try software that may help manage their procrastination.

5. RELATEDWORK

The research of Richard W. Patterson presents three software tools: a commitment tool, a reminder tool, and a focusing tool [13]. The commitment tool allows users to pre-commit time limits on certain websites. The reminder tool sends a reminder to the user after they spend thirty minutes on a distracting website. The focusing tool blocks certain websites for a given amount of time. Procrash also offers similar reminder and focusing tools. The program sends a notification to remind the user to stay on task if it detects that they are distracted for a prolonged amount of time. If the user continues to be distracted, instead of blocking the website, Procrash closes the distracting tab.

Matthias Laschke, et al. present a tangible calendar-like object in order to overcome procrastination. This model is designed to help people reflect and help overcome their procrastination by allowing users to set personal goals. Procrash also aims to help users overcome their procrastination but in the form of a computer program instead of an object. Procrash helps divert the user's attention by redirecting their attention from distracting websites, but not specifically to a calendar or to-do list.

Christian Aljoscha Lukas, et al. address the issue of procrastination by using a smart phone-based intervention system. Although both of our solutions are technology-based and mainly for computers, this one is implemented exclusively on smart phones. In addition, this paper only includes one experiment, which is a user study.

6. CONCLUSION AND FUTUREWORK

The use of Procrash limits the use of online distractions and therefore contributes to more efficient time management. The experimental data supports this, indicating a direct correlation between the usage of the program and the number of times the user is off task. The average number of visits to distracting websites and apps in ten minutes was 17.8 times without the program, and 10.6 times with the program. When the program was in use, the time the user spent with their attention diverted from their work was significantly lower when compared to the data from not using the program (a reduction of 40.45%).

Currently, this application uses optical character recognition, which provides 98 to 99 percent accuracy. In most cases, this is an acceptable accuracy level. The user can remotely control and configure the status/blacklist at real-time. In addition, the application currently closes the entire window with disregard of other active tabs.

A reliable level of accuracy can be maintained with high image quality. Making sure that the source image is visible enough can lead to better optical character recognition results. An additional future plan is to conduct a larger user study to collect a wider range of data on how the program performs.

REFERENCES

[1] Day, Victor, David Mensink, and Michael O'Sullivan. "Patterns of academic procrastination." Journal of College Reading and Learning 30.2 (2000): 120-134.

[2] Hussain, Irshad, and Sarwat Sultan. "Analysis of procrastination among university students." Procedia-Social and Behavioral Sciences 5 (2010): 1897-1904.

[3] Tice, Dianne M., and Roy F. Baumeister. "Longitudinal study of procrastination, performance, stress, and health: The costs and benefits of dawdling." Psychological science 8.6 (1997): 454-458.

[4] StudyMode. "Eighty-Seven Percent of High School and College Students are Self-Proclaimed Procrastinators." PR Newswire: Press Release Distribution, Targeting, Monitoring and Marketing, 30 June 2018.

[5] Steel, Piers. "The nature of procrastination: a meta-analytic and theoretical review of quintessential self-regulatory failure." Psychological bulletin 133.1 (2007): 65.

[6] Kovbasnyuk, Nadiya, and Iryna Styfanyshyn. "Distance learning." Матеріали конференцій Молодіжної наукової ліги (2020): 38-39.

[7] Joy, Kelvin. "THE EFFECTS OF INTRODUCING ONLINE EDUCATION VERSUS CLASSROOM LEARNING." Academy of Educational Leadership Journal 25.5 (2021): 1-2.

[8] Gillick, Mitchell, and Christie Magoulias. "Competing Against Outside Distractions in Online Classrooms for Grade-School Students." (2020).

[9] Rathi, Udbhava, Daniel John KJ, and Joel Mathew Jose. "Exploring the relationship between social media usage and academic performances." (2020).

[10] Özer, Bilge Uzun, and Mesut Saçkes. "Effects of academic procrastination on college students' life satisfaction." Procedia-Social and Behavioral Sciences 12 (2011): 512-519.

[11] Roster, Catherine A., and Joseph R. Ferrari. "Time Is on My Side—or Is It? Assessing How Perceived Control of Time and Procrastination Influence Emotional Exhaustion on the Job." Behavioral Sciences 10.6 (2020): 98.

[12] Duru, Erdinç, and Murat Balkıs. "Procrastination, self- esteem, academic performance, and well- being:A moderated mediation model." (2017).

[13] Richard W. Patterson. "Can behavioural tools improve online student outcomes? Experimental evidence from a massive open online course." Journal of Economic Behaviour & Organization, 153 (2018): 293-321.

[14] Laschke, Matthias, and Marc Hassenzahl, Jan Brechmann, Eva Lenz, and Marion Digel. Overcoming Procrastination with ReMind. New York, Association for Computing Machinery, 2013.

[15] Aljoscha Lukas, Christian, and Matthias Berking. Reducing procrastination using a smartphone- based treatment program: A randomized controlled pilot study. Internet Interventions, 12 (2018): 83-90.