AN AI-POWERED SYSTEM TO INCREASE USER'S PRODUCTIVITY USING COMPUTER VISION AND MACHINE LEARNING

Lida Song

The York School, 1320 Yonge Street, Toronto, Ontario, M4T1X2, Canada

ABSTRACT

DistraXcel represents an innovative AI-powered system designed to enhance productivity by using computer vision and machine learning to mitigate distractions from digital applications and websites [1]. The internet, a "giant hypodermic" filled with distracting "psychoactive drugs," significantly impedes focus, especially for individuals with neurodevelopmental disorders, such as ADHD and autism [2]. DistraXcel addresses this problem by identifying and blocking distracting digital content, leveraging Python and Tkinter for the front-end, Firebase for back-end operations, and Roboflow for building an object detection and classification model. However, challenges such as auto-login hassles, AI's specificity to certain screens leading to false negatives, and cross-platform GUI inconsistencies were encountered [3]. Through methodical experiments, the AI demonstrated strong identification capabilities but also revealed an overfitting issue. A user perception survey highlighted an improvement in perceived productivity post-use, indicating DistraXcel's effectiveness while suggesting room for addressing broader distraction factors. DistraXcel advances beyond current methodologies by offering a more positive feedback mechanism, extensive customization, and prioritizing privacy [4]. It evolves as a user-centric tool aimed at fostering a focused and productive digital environment, providing considerable aid to students and workers alike, especially those with neurodevelopmental disorders. DistraXcel marks a significant step forward in addressing the pervasive issue of digital distractions.

KEYWORDS

Focus Assistance, Computer Vision, Machine Learning, Python

1. INTRODUCTION

It is said that "The internet is a giant hypodermic, and the contents, including social media, are the psychoactive drugs." We've all tried to focus on writing a lab report or an essay while resisting the alluring calls of the internet. My frustration mounted as the distracting apps and websites continued to rob my study time. This led me to ponder existing solutions that could mitigate the disturbance of these troublesome apps and websites. With a quick Google search, I found a BlockSite Chrome extension, which blocks websites that users enter and categories that users select; however, it often fails to detect less popular sites [5].

This sparked an idea to create a program that addresses these problems. I began to consider using artificial intelligence (AI) to detect what app or website the user is currently on, as I discovered that with sufficient data, object recognition AI can detect physical and digital objects [6]. Therefore, I could train the AI model to recognize certain unique features of a specific category of apps or websites, so when the AI detects these elements on the user's screen, it will close the

David C. Wyld et al. (Eds): AdNLP, CDKP, NCO, SAI, MLT, Signal, SOFT, ICAITA, CMC - 2024
pp. 183-192, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141016

application or website. This would allow the app to catch even the most obscure apps and websites.

Python frameworks such as Tkinter will be used for the front-end, as they provide methods that could allow for the display of buttons, search bars, and scrollbars [7]. Firebase, a back-end as a service platform, will be used to construct my back-end with a real-time database for user information storage, including their block sites, their stats, and their settings. Roboflow, a computer vision developer framework, will be used to build an image recognition model by feeding it annotated screenshots of distracting websites and apps.

DistraXcel draws inspiration from three distinct methodologies to address their limitations and enhance personal productivity and user engagement. TimeAware's exploration into positive versus negative feedback on productivity revealed the stress associated with negative framing, prompting DistraXcel to adopt a more positive feedback approach to mitigate stress [8]. WorkAnalytics' focus on software developers' productivity emphasized the importance of experience sampling and data retrospection for increasing awareness of work patterns [9]. DistraXcel builds on this by offering enhanced customization options, allowing users to tailor their experience to their specific productivity needs. Lastly, insights from AI-based suicide risk monitoring in schools highlighted concerns around privacy and data accuracy. DistraXcel addresses these by prioritizing user privacy through local data processing. By integrating these learnings, DistraXcel aims to offer a personalized, stress-free, and ethically considerate tool that enhances productivity without the shortcomings of its predecessors.

2. CHALLENGES

To build the project, a few challenges have been identified:

2.1. The Re-Login Procedure

This program requires users to initially sign-up for an account. Their stats and customizations made in the app, including specific distractions and Pomodoro times, are stored under their account in Firebase. After they make an account, they need to log-in every time the app is reopened. This login process ensures that each user’s preferences and data are maintained securely and consistently across different devices. However, this becomes an unnecessary hassle as if the user uses the app daily, it is needless for them to re-login every day.

2.2. The AI

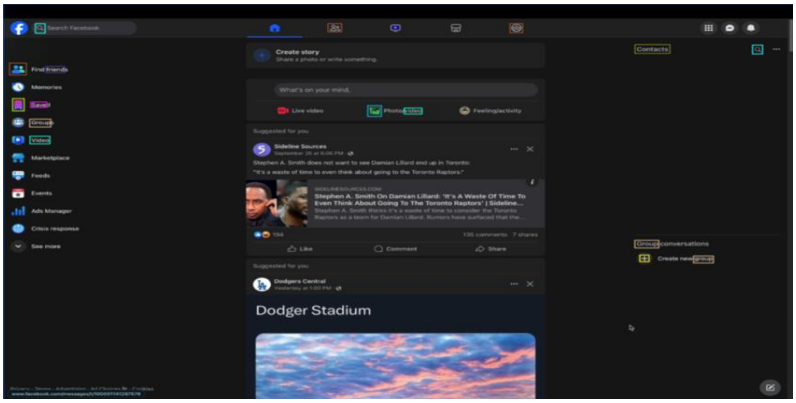


Figure 1: Screenshot of an image annotated in Roboflow

Object detection and classification AI is used to detect if the user is on a distracting app or website. Screenshots of distractions are imported and annotated in Roboflow, which is used to train the AI model, as seen in Figure 1. The model should then be able to detect and classify distracting elements on a user's screen, thereby blocking the distraction. However, after one training session, it was apparent the AI struggled mightily to do so; a glance at the confusion matrix saw an overwhelming amount of false negatives. The reason was that the AI was fully accustomed to the features of my screen, thus it failed to identify features beyond those familiar to my screen.

2.3. The Elements Display

Tkinter is a Python graphical user interface (GUI) library [10]. It enables the creation of frames and grids, which contain widgets such as buttons, labels, search bars, and scroll bars. Tkinter is lightweight, cross-platform, and customizable, making it a great choice for building the front-end of this app. Unfortunately, many problems were encountered when implementing the GUI. Although it is cross-platform, the elements rendered differently on Mac and Windows. Despite having the same resolution, widgets were noticeably bigger and more horizontally stretched on Windows. The most prominent issue was the slider, which was used in the focus screen for the Pomodoro timer. It was glitchy and struggled to sync with the actual value.

3. SOLUTION

DistraXcel is built upon three core pillars to perform the functions it sets out to do. The first component is an AI recognition system built using a combination of OpenCV, PyTorch, and a suite of program management functions. The AI training component serves as the pipeline from which improved variations of the program can be created [11]. This is rounded off by the user account and database back-end where the user's preferences and settings are stored to ensure their settings can transfer so long as they have their account between devices.

The AI recognition system is the first primary component and works by periodically taking a screenshot of the user's screen to then prepare it for analysis. It is fed into a custom model based on YOLOv7 after making the necessary preprocessing changes and those results are then used to carry out a variety of different moderation actions.

Figure 2: Screenshot of the Focus screen

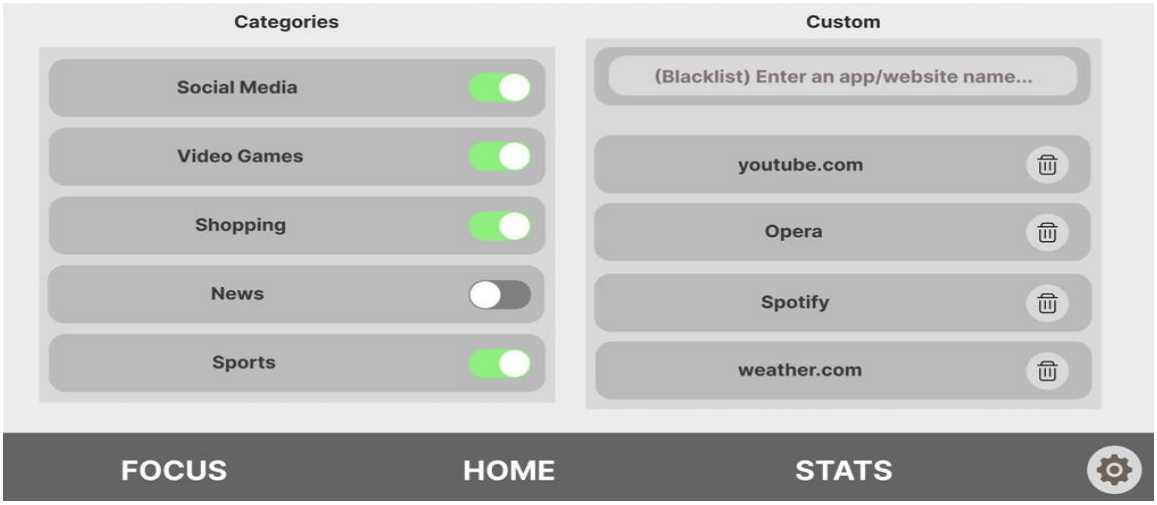
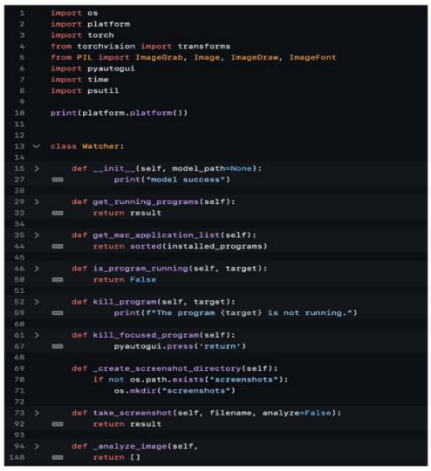


Figure 3: Screenshot of the Home screen



```
import os
import platform
import torch
from torchvision import transforms
from PIL import ImageGrab, Image, ImageDraw, ImageFont
import pyautogui
import time
import psutil

print(platform.platform())


class Watcher:

    def __init__(self, model_path=None):
        print("model success")

    def get_running_programs(self):
        return result

    def get_mac_application_list(self):
        return sorted(installed_programs)

    def is_program_running(self, target):
        return False

    def kill_program(self, target):
        print(f"The program {target} is not running.")

    def kill_focused_program(self):
        pyautogui.press('return')

    def _create_screenshot_directory(self):
        if not os.path.exists("screenshots"):
            os.mkdir("screenshots")

    def take_screenshot(self, filename, analyze=False):
        return result

    def _analyze_image(self,
        return []
```

Figure 4: Screenshot of the Watcher class

The core function of DistraXcel is its ability to monitor and control various aspects of a user's computer to help them better maintain focus and optimize their time. As shown in Figure 2, on the Focus screen, the user can activate the Pomodoro timer to block all apps and websites besides the ones in the customized whitelist. On the Home screen shown in Figure 3, the user can select which categories (social media, video games, shopping, news, and sports) to block and fill a blacklist to always be blocked. As illustrated in Figure 4, the Watcher class executes the blocking by loading up the custom YOLOv7 model and putting it on standby. It can begin to take action once prompted by this program. It can use a variety of tools given to the Watcher, such as the ability to see what applications and websites are running, kill an application or website by name, and also take screenshots of the user's screen. The Watcher's analysis function takes a screenshot before passing on the image data to the YOLOv7 model, which then outputs any distracting elements it detects on the user's screen [12]. This result is then packaged and returned as a list to be evaluated elsewhere in the application to determine if there is enough confirmation to terminate an application or website.

The second component is the AI training pipeline itself. Data was collected from a variety of sources that were identified as potential sources of distractions and created a dataset with the help

of Roboflow. This dataset was then paired with YOLOv7 to create a fine-tuned checkpoint to accomplish the objectives of DistraXcel.

```
!git clone https://github.com/WongKinYiu/yolov7

%cd yolov7

!pip install -r requirements.txt

!pip install roboflow

from roboflow import Roboflow

rf = Roboflow(api_key="API_KEY")

project = rf.workspace("workspace_name").project("project_name")

dataset = project.version(VERSION_NUMBER).download("yolov7")

!wget
https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7_train
ing.pt

!python train.py --batch 16 --epochs 50 --data DistraXcel-2-1/data.yaml
--weights 'yolov7_training.pt' --device 0

!zip -r export.zip runs/train/exp/weights/best.pt

from google.colab import files

files.download('export.zip')
```

Figure 5: Screenshot of the code ran to prepare a machine learning environment

The code in Figure 5 is designed to create and prepare a machine learning environment using YOLOv7 as the base model for creating an AI meant for real-time image detection. First, the repository is cloned and the necessary configurations are made. Once this is set up, required components and checkpoints are then installed within the directory for functionality and efficiency's sake. Then, model training is initiated with a GPU-capable computer, before packaging the best candidate for use in future testing and eventually DistraXcel itself.

The third component is a user account and database system which enables the user to save their preferences and settings in the cloud. Firebase authentication and Firestore are used as the back-end, making a point to ensure no sensitive information about a given user is stored.

```
import re
import threading
import time


class UserData: # TODO: the timer currently has a major acceleration issue that needs adjustment
    _instance = None

    valid_program_names = {
    }

    valid_category_names = {
    }

    def __init__(self):
        self.is_working = True
        self.timer_active = False
        self.timer_should_run = False
        self.timer_seconds_remaining = 600
        self.timer_callback = None
        self.timer_thread = None
        # self.timer_lock = threading.Lock()

        if UserData._instance is not None:
            raise Exception("This class is a singleton!")
        else:
            UserData._instance = self
            self.user_data = {
                "blacklist": {},
                "whitelist": {},
                "categories": [False] * 5,
                "settings": {
                    "pomodoro_work_slider_val": 50,
                    "pomodoro_rest_slider_val": 50,
                    "dark_mode": False,
                    "consecutive_cycles": 1
                },
                "stats": {
                    "block_count":
                    dict(),  # number of times a given program was blocked
                    "category_block_count": dict(
                    ),  # number of times something from a category was blocked
                    "focus_time":
                    0,  # total amount of time (in minutes) spent on focus mode during pomodoro session
                    "rest_time":
                    0,  # total amount of time (in minutes) spent on break mode during pomodoro sessions
                    "current_focus_streak":
                    0,  # current daily consecutive uses of pomodoro
                    "longest_focus_streak":
                    0,  # longest daily consecutive uses of pomodoro
                }
            }
```

Figure 6. Screenshot of the UserData Class

Given the large reach a program like DistraXcel needs to properly function, it is extremely important to make sure that the data of the user remains private. To that end, the only user data that is stored in the back-end is whatever whitelist or blacklist they may have alongside some general settings and statistics on application usage statistics, as seen in Figure 6. There is no data

about the actual contents of the user's screen nor any other sensitive information as any operations for DistraXcel that depend on it (e.g., capturing the screen and terminating applications) are done entirely locally without the need for the internet to function.

4. EXPERIMENT

4.1. Experiment 1

One of the most important aspects that needs to be addressed is the intrinsic accuracy and strength of the AI running within the program itself.

To conduct this particular experiment, the current model will be evaluated by running it against the following tests and settings:

- Configure the AI to visually annotate what it sees along with its confidence in identification.
- Track the performance of the AI in identifying one or a few specific elements across different apps and websites. Given that the goal of the program is to shut down distractions, the correct identifications relative to any false positives that may occur will also be logged. In addition, the confidence score will be recorded to determine the degree of strength.
- Elements to be focused on:
  - Like icon
  - Emojis
  - Notification icon
  - Reply icon
- Websites to be tested:
  - Reddit
  - Instagram
  - Facebook

Programs to be tested:

- League of Legends
- Genshin Impact
- Minecraft

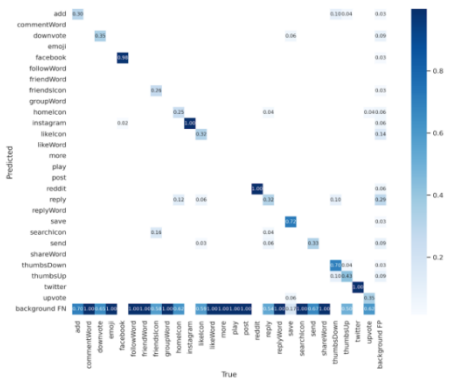


Figure 7. Confusion Matrix of Experiment 1

As shown in Figure 7, the confusion matrix created during this testing process reveals that the AI has a generally significant capability of identifying targets, but also exposes certain vulnerabilities that arise from certain characteristics regarding the dataset the AI was derived from. The overfit nature of the AI towards certain classes like Reddit, Facebook, and Instagram exposes the overrepresentation of those members at the expense of other less significant members.

4.2. Experiment 2

The other element for DistraXcel that needs to be addressed is the extrinsic element, that is to say, what the general user thinks of the product and its perceived effectiveness.

To properly gauge the reactions of the user, a survey will be conducted both before and after the user is exposed to the product to get a better sense of both their initial biases and thoughts as well as any changes that may be caused by DistraXcel.

The pre-survey is meant to capture the general disposition, inclinations, and perspectives of the people before their exposure to DistraXcel. The demographic and dynamics of the user will also be collected to understand the approach of different groups and to distinguish the difference between a user using DistraXcel independently and a user using it under a parent or educator's supervision.

The survey data revealed the following insights:

- The mean effectiveness rating before using DistraXcel was 5.91, which improved to 6.57 after using the program.
- The median values show a shift from 6 (pre-use) to 7 (post-use), indicating a positive shift towards higher effectiveness ratings.
- The lowest pre-use effectiveness rating was 1, with the post-use rating improving to a minimum of 3, suggesting the program effectively eliminated the lowest effectiveness perceptions.
- The highest rating, both pre- and post-use, was 10, indicating that some users found the program extremely effective even before its use, possibly due to anticipation or general expectations towards such solutions.
- The data shows an overall positive shift in perceived effectiveness after using DistraXcel. Surprisingly, the improvement, while positive, is moderate (mean increase of 0.66). This could be due to the varying levels of distractions users face, and while DistraXcel was effective in managing distractions for most, there might be external factors influencing productivity that DistraXcel does not address.

The significant shift in median values suggests a tangible improvement in user experience, highlighting DistraXcel's potential to enhance productivity by mitigating digital distractions. This data underscores the importance of addressing digital distractions through tailored AI-driven interventions, reinforcing the potential of such technologies in improving focus and productivity in study environments.

In Experiment 1, the focus was on evaluating the AI's intrinsic accuracy in identifying specific elements, including emojis, as well as like, notification, and reply icons, across various apps and websites, crucial for minimizing distractions [13]. The AI was tested against platforms such as Reddit, Instagram, and Facebook, as well as games including League of Legends, Genshin

Impact, and Minecraft. The findings highlighted a generally high identification capability, but also pointed out the AI's overfitting towards more common platforms, leading to vulnerabilities in recognizing elements from less represented sources. This overfitting suggests the AI's training data may overly represent certain sites, impacting its versatility.

Experiment 2 delved into the extrinsic evaluation, assessing user perceptions of DistraXcel's effectiveness through surveys conducted before and after the program's use. This aimed to capture changes in perceived effectiveness, with results indicating a positive shift in users ' ratings post-use. The moderate improvement suggests that while DistraXcel effectively reduces digital distractions, other productivity-affecting factors remain unaddressed by the program. The experiments collectively reveal DistraXcel's potential in enhancing productivity by reducing digital distractions, but also underscore the need for comprehensive solutions addressing a broader range of distractions and enhancing the AI's adaptability to varied digital environments.

5. RELATED WORK

Young Ho Kim et. al. presented a study on a self-monitoring system called TimeAware, designed to improve personal productivity by tracking and reflecting on computer usage behaviours [14]. The system uses an ambient widget and an information dashboard for visual feedback and incorporates two versions with different framing settings: one highlighting productive activity (positive framing) and the other emphasizing distracting activities (negative framing). An eight-week study with 24 participants showed that only the negative framing condition led to a significant improvement in productivity, suggesting the effectiveness of emphasizing distractions in enhancing personal productivity. However, this improvement was not sustained after the feedback was removed, indicating limitations in lasting behavioural change. Additionally, participants in the negative framing condition reported feeling stressed by the feedback. This points out challenges such as the stress associated with negative feedback which DistraXcel seeks to remedy through a more positive feedback loop.

Meyer et al. explored the impact and design considerations of self-monitoring tools aimed at improving productivity for knowledge workers, with a focus on software developers [15]. Through an iterative development process involving user feedback, a survey of 413 developers, and a field study with 43 participants, the research identifies key design elements for effective workplace self-monitoring. The developed tool, WorkAnalytics, incorporates these design elements and was tested to assess its influence on self-awareness and productivity. Findings suggest that experience sampling and data retrospection significantly enhance user engagement and awareness of work patterns and productivity. Participants valued the ability to personalize the tool to their individual needs, including selecting which metrics to track and how they are presented. DistraXcel seeks to expand on this idea through the ability for users to customize their preferences and requirements to match their productivity needs.

Lynsay Ayer et al. investigated the usage of AI-based tools for monitoring suicide risk among K–12 students [16]. As schools increasingly turn to educational technology, including AI, to support student mental health, this study provides a critical examination of AI-based suicide risk monitoring programs' implementation, stakeholder perceptions, and the potential benefits and risks associated with these tools. Key findings from interviews with school staff, EdTech representatives, healthcare professionals, and advocacy groups indicate that AI-based tools can identify students at risk for suicide and offer reassurance to school staff and parents. However, concerns arise regarding student privacy, the perpetuation of inequalities, and the accuracy of AI-based suicide risk prediction algorithms. DistraXcel addresses these concerns by focussing more on the aspect of productivity and prioritizes the privacy of its users through measures such as running the AI model locally rather than server side.

6. CONCLUSION

With the use of object detection AI, DistraXcel marks a significant advancement in the realm of productivity tools — it demonstrates the capability of AI to detect and block digital distractions, a noteworthy upgrade from previous apps that relied purely on hardcoding app and website names, which proved to be inadequate in various situations. With the use of the YOLOv7 model, Tkinter for the graphic user interface, and Firebase for cloud storage, DistraXcel serves as an effective solution for the pervasive issue of digital distractions. From the user feedback received, it is apparent that DistraXcel is successful in increasing the efficiency of users, although it has some weaknesses. The most apparent limitation of the project is the overall performance and capability of the AI. This is in large part due to the general quality of the dataset itself, which has an overrepresentation of certain classes like Reddit and Facebook [17]. This can be remedied in the future by making more efforts to create a stronger and more balanced dataset to train on, or by using a newer version of the YOLO model. A general rewards system can also be developed to inject a gamified component, incentivizing users to better pursue more efficient workflows. Another direction that could be explored is the creation of a companion mobile app to allow administrators and parents to have easier access to the information gathered and acted upon by the productivity tools built into DistraXcel.

DistraXcel will continue to evolve following the initial release through the introduction of a newer, better-trained detection model, along with new incentives and additional user experience features to make it an even more valuable tool for users who are looking to boost their study or work productivity.

REFERENCES

[1] Leung, Kimberly A. The effect of distractions on task performance and enjoyment as moderated by regulatory fit. San José State University, 2015.

[2] Schmidt, Shelly J. "Distracted learning: Big problem and golden opportunity." Journal of Food Science Education 19.4 (2020): 278-291.

[3] Wirzberger, Maria, et al. "Optimal feedback improves behavioral focus during self-regulated computer-based work." Scientific Reports 14.1 (2024): 3124.

[4] Conaway, Emily Mae. "Student distractibility with school-issued computers." (2019).

[5] Rosen, Larry, and Alexandra Samuel. "Conquering digital distraction." Harvard business review 93.6 (2015): 110-113.

[6] Almoallim, S., and C. Sas. "Functionalities review of digital wellbeing apps: towards research-informed design implications for interventions limiting smartphone use." JMIR Form Res 6 (2022): e31730.

[7] Han, GuHyun, et al. "Human–Computer Interaction Lab (HCIL) in Seoul National University." Visual Informatics 4.4 (2020): 35-39.

[8] Carvalho, Luiz Roberto, et al. "Evaluation of the effectiveness of the FOCUS ADHD App in monitoring adults with attention-deficit/hyperactivity disorder." European Psychiatry 66.1 (2023): e53.

[9] Józsa, Krisztián, et al. "Design, Development and Application of Focus App for Assessment of School Readiness." INTED2022 Proceedings. IATED, 2022.

[10] Ghosh, Arup Kumar, et al. "Safety vs. surveillance: what children have to say about mobile apps for parental control." Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. 2018.

[11] Badillo-Urquiola, Karla, et al. "Beyond parental control: Designing adolescent online safety apps using value sensitive design." Journal of adolescent research 35.1 (2020): 147-175.

[12] Biwer, Felicitas, et al. "Understanding effort regulation: Comparing 'Pomodoro'breaks and self-regulated breaks." British Journal of Educational Psychology 93 (2023): 353-367.

[13] Ghosh, Arup Kumar, et al. "A matter of control or safety? Examining parental use of technical monitoring apps on teens' mobile devices." Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. 2018.

[14] Cranefield, Jocelyn, et al. "Partnering with AI: The case of digital productivity assistants." Journal of the Royal Society of New Zealand 53.1 (2023): 95-118.

[15] Mark, Gloria, Shamsi Iqbal, and Mary Czerwinski. "How blocking distractions affects workplace focus and productivity." Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers. 2017.

[16] Meyer, Andre N., et al. "Design recommendations for self-monitoring in the workplace: Studies in software development." Proceedings of the ACM on Human-Computer Interaction 1.CSCW (2017): 1-24.

[17] Ayer, Lynsay, et al. "Artificial Intelligence–Based Student Activity Monitoring for Suicide Risk: Considerations for K–12 Schools, Caregivers, Government, and Technology Developers." Rand Health Quarterly 11.2 (2024).