

# Improving Human-computer Interaction through Innovative Adaptation, a Case Study in End-user Development for Digitization

Jeremy D. Moore, University of Tennessee, Knoxville; Andrew Coggins, University of Tennessee, Knoxville

## Abstract

*The Digital Production Lab in John C. Hodges Library at the University of Tennessee, like many university imaging studios, has long relied on a seasonal and temporary labor force drawn from our student body. Improving human-computer interaction in our production workflows should reduce training time and errors while increasing throughput and worker confidence.*

*In this paper, we present our on-going efforts adapting Elgato's Stream Deck XL hardware to control a computer running Apple's macOS operating system and Phase One's Capture One Pro software using Python and AppleScript code. The paper outlines our custom code linking these parts together, describes how this relatively inexpensive input device streamlines our digitization process, and includes ideas for future application.*

## Introduction

The Digital Production Lab (DPL) at the University of Tennessee (UT) is responsible for digitizing materials for UT Libraries' Digital Collections [1]. In 2019, two major events affected the DPL: the hiring of a new Technical Manager and a plumbing leak requiring the lab to be physically relocated. This new manager was tasked with designing and building a modern imaging studio for large-scale digitization projects because the DPL was inadequately equipped for high-quality, high-efficiency digitization work and lacked well-established workflows and documentation [2]. The authors' viewpoints are informed by their roles as the new, but experienced Technical Manager, and a student Digital Imaging Technician collaborating to create, test, and implement digitization workflows in the updated DPL.

## Motivation

Training student workers to be digital imaging technicians is a complicated and time-consuming process. Before any significant imaging training can begin, new hires must learn how to safely handle archival materials and use a computer with networked storage in a managed, enterprise environment in addition to administrative tasks that may be new to student workers seeking employment for the first time. This is exacerbated by the use of complicated software, such as Capture One Pro, that often has important settings which may be difficult to find and easy to unknowingly change. During the physical relocation of the DPL, new capture stations were built using cameras (see Fig. 1) that capture images much more quickly than the flatbed and planetary scanners previously in use. With this increased speed, if unusable images are captured due to a maladjusted setting, user error, or system design, then it is likely that many unusable images will be

made and subsequently require quality control, handling, and rescanning. Additionally, while the DPL's manager is experienced in building and using camera-based capture stations [3], the student digital technicians were not and had to be trained to use the new hardware and software.



**Figure 1.** Andrew Coggins prepares a fragile map for digitization using a custom, camera-based capture station in the relocated Digital Production Lab at UT Libraries in late 2019. To his left is an Apple iMac running Phase One's Capture One Pro. This computer is tethered to and controls the camera. Part of Andrew's right hand is visible on the LCD-screen on the back of the camera in the upper-right hand corner of the image.

## Problem

The initial workflow for our first and on-going project, digitizing historical athletics programs and media guides with a camera-based capture station, was quite onerous. Any difficulties navigating the process were compounded for anyone unfamiliar

with the software we use to tether our cameras to the computer for rapid capture, Phase One's Capture One Pro (see Fig. 2).



**Figure 2.** Screenshot of Phase One's Capture One Pro 20.

The following is a simplified list of ten steps necessary to capture and process one item, e.g. a single program, media guide, or other print document, assuming the capture station has already been powered on, Capture One Pro started, and the camera successfully tethered to the computer:

### **Steps to Capture and Process a Book-like Item**

#### **Capture**

1. Create new folder
2. Set new folder as Capture Directory
3. Capture each page of the item
4. Select all images and batch rename
5. Check images shot for completeness e.g. correct number of pages, total number of pages is even

#### **Process**

6. Select representative image and run AutoCrop script
7. If necessary, adjust crop box location or size
8. Select images and run AutoCrop script that enforces a crop box the same size as that on the primary variant
9. Check each image and, if necessary, adjust its automated crop box or, if the AutoCrop was unsuccessful, apply one manually
10. Repeat steps 6-9 if there are pages of different sizes that need differently sized crop boxes, e.g. covers versus interior pages, fold-outs

Clearly, this simplified list of steps requires extra tasks and navigation through Capture One Pro. There are currently no built-in methods to quickly and simply adjust an image's crop box without clicking and dragging its location or boundaries with a mouse nor are there methods to match crop box sizes across multiple images, e.g. pages from a book, without also synchronizing the box's location. Hardware malfunctions also sometimes occur, requiring troubleshooting that adds extra steps. Due to the various ages, conditions, formats, and styles of the athletics programs and media guides, the item being digitized may

also require exceptional handling and more complicated processing.

### **Approach**

We expected that simplifying the human-computer interface to limit the number and complexity of decisions that could be made should result in novice technicians making fewer accidental inputs while increasing confidence that their choice was the correct one. We were already using a foot pedal to speed up each capture and were calling a combination of custom Python and AppleScript code to automatically crop our captured images from Capture One Pro's "Scripts" drop-down in the menu bar when we happened upon the Elgato's Stream Deck XL hardware [4] in an advertisement online. This inexpensive human interface device (HID) retails for \$249.99 and consists of an LCD screen covered by 4 rows of 8 clear, physical buttons (see Fig. 3). The authors are unaware of any similar devices on the market, which combine physical buttons with screens in such a user-friendly way. The hardware has a USB-C port and came with a USB-C to USB-A cord to connect it to a computer. It is heavy enough to not slide on the desk when a button is pressed, but small enough that it can be easily held in one's lap where buttons can be quickly pressed with the thumbs, like a video game controller. The Stream Deck XL is marketed to Internet personalities live streaming video to automate various actions and engage with their viewers, such as sending a Tweet with a screenshot from the video stream or loading and playing a soundbite.



**Figure 3.** Elgato Stream Deck XL, image courtesy Elgato.com.

The Stream Deck XL's buttons can be extensively customized in both appearance and action. Each button is a 144-pixel square image that can be created using Elgato's Key Creator website [5] chosen from a set of included icons, created in an imaging program like Adobe Photoshop, or automatically generated. Elgato provides software and an SDK to build your own plug-ins and actions. A profile is first created including a set of button images and their associated actions. Profiles can be imported, exported, and set to load whenever a particular application is currently active. A profile of buttons can also include folders with sub-screens of buttons that automatically include navigation back up to the parent folder.

Creating profiles and changing between them is very streamlined allowing one to easily switch between multiple

profiles on the fly by clicking the icon for the Stream Deck software in the macOS menu bar and selecting another by name (see Fig. 4). Multiple Stream Decks (there are currently 3 different sizes) can be connected to a single computer and can affect each other, such as using a 6-button Stream Deck Mini to set the active profile on the larger, 32-button Stream Deck XL. We recommend the largest Stream Deck, the XL, as buttons can be left blank if unneeded or to simplify an interface (see Fig. 5), but adding extra buttons later would require buying another device. Once technicians are taught how to modify profiles they can alter the user interface in ways that makes the most sense to them, for example, some may naturally orient and think of an image having a “top” and “bottom”, while others prefer buttons that refer to the directions “up” and “down” (see Fig. 4). We have focused our efforts on using the Stream Deck XL to assist in cropping our images consistently due to high labor costs of doing so manually.

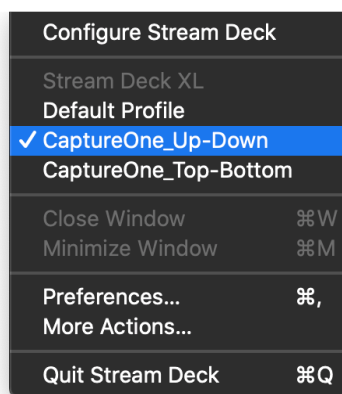


Figure 4. macOS menu allowing one to select different Stream Deck profiles.

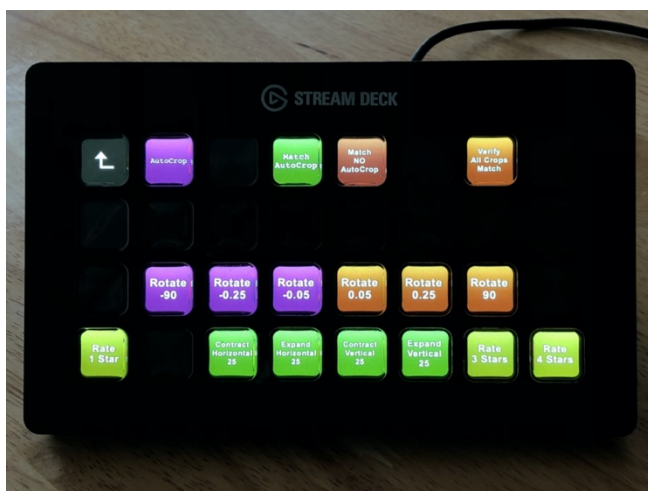


Figure 5. Stream Deck XL showing active buttons for AutoCropping and adjusting, matching, and verifying crop box sizes. Note the navigational arrow on the upper-left button to move back up one level of Stream Deck XL screens.



Figure 6. Stream Deck XL showing active buttons to adjust the size, location, and rotation of the crop box of the currently active image in Capture One Pro. Depending upon reproduction, it may be difficult to note that each type of action, such as “Up 25” and “Up 50”, share the same color buttons. Note buttons in the upper-left and right-hand corners to rate an image as “1 star” or “4 stars”.

## Results

We have successfully begun using the Stream Deck XL in production to improve student workers’ confidence in their interactions with the complicated software we use. It should be noted, though, that many of the Stream Deck’s advantages could be replicated with any touch-based interface or an alternative control device, such as a Microsoft Xbox or Sony PlayStation controller, but Elgato’s hardware is the only system the authors are aware of to combine a screen with physical buttons. After introducing the Stream Deck XL into our workflow, a recently hired digital imaging technician was taught how to confidently process a book-like item in less than 15 minutes. This is unusually fast, but by limiting input options to the buttons of the Stream Deck XL many possible errors and edge cases did not have to be mentioned or explained, such as “don’t unknowingly change the cursor tool or a hidden option by pressing one or more of a multitude of keys on the keyboard”. The new digital technician is over three times faster using the Stream Deck XL to adjust and validate crop boxes than doing so with a keyboard and mouse in informal tests.

Tasks are easily separated and organized through the simple visual and tactile interface placing technicians’ focus on doing tasks rather than how to do them. Likewise, separated tasks allow for the steps to be organized, scheduled, and assigned to different technicians as needed. For example, images needing the location and size of a crop box validated are rated 3 stars in Capture One Pro and then the software is set to filter by rating and only show 3-star images. A technician will select the appropriate Stream Deck XL profile and button-set, visually validate the location of each crop box and adjust as necessary. If, after any adjustments are made, the image passes quality control it is rated 4 stars (if not, it is rated 1 star) using a dedicated button (see Fig. 6) that simply enters the number “4” as if one had typed it on a keyboard, it will leave view and the next 3-star image will be automatically selected and

can be processed via button presses, though not all of the buttons are so simple.

Currently, for anything more complicated than a keyboard shortcut, we use a Stream Deck XL with a suite of custom Python and AppleScript scripts to control a computer running Apple's macOS operating system and Capture One Pro version 20. AppleScript is first used to create a script that the Stream Deck software can open when a button is pressed [6]. This AppleScript is mostly boilerplate only serving to call a Python script with hard-coded arguments, such as the direction and number of pixels a crop box should be shifted. Once one of these AppleScript scripts is written (see Fig. 7), it is a much simpler task for novice programmers, i.e. the more advanced student technicians in the DPL, to tweak this code to create the different combinations they may need for a particular project. For example, the students decided that buttons to rotate images 0.05 and 0.75 degrees clockwise and counter-clockwise were useful, but they replaced the initial buttons created to rotate 1.5 degree with 0.25 degree rotation buttons that gave them more granular control (see Fig. 6).

The Python script then performs whatever action it was designed to do, which may involve calling out to other Python scripts as needed. While Capture One Pro does not have a Python-accessible API, Phase One does provide an AppleScript Dictionary allowing end-users access to a limited set of controls in that language. Due to the authors' preference for Python over AppleScript, two Python scripts were initially written to programmatically generate AppleScript to control Capture One Pro, `applescript.py` and `capture_one_pro_12.py`, but in early 2020, the DPL upgraded from Capture One Pro version 12 to Capture One Pro version 20 and created an updated script, `capture_one_pro_20.py`, that is still undergoing testing [7].

```
set direction to "left"
set pixels to "25"

set local_python to "/path/to/Python"
set python_script to
"/path/to/modify_crop_box.py"

set script_arguments to "--direction " &
(quoted form of direction) & " --pixels " &
(quoted form of pixels)

set run_local_python_script to (quoted form
of local_python) & " " & (quoted form of
python_script) & " " & script_arguments

do shell script run_local_python_script
```

**Figure 7.** Sample AppleScript to move a crop box 25 pixels, making the border on the left side of an image 25 pixels larger and the right side 25 pixels smaller.

The software allowing us to write Python and create AppleScript on the fly is a built-in macOS command-line application named `osascript` that was designed to execute a supplied script or standard input in AppleScript [8]. This command-line application is called by `applescript.py` using the `run` command from Python's `subprocess` module in the standard library (see Fig. 8). If the AppleScript command produces any output the

information can be captured by Python as a string from standard output then converted to a list of values by splitting the data on commas and stripping out the new lines and leading and trailing whitespace (see Fig. 9). These values can then subsequently be used as arguments for further actions; an example of this process starts with manually selecting all interior pages of a book in Capture One Pro before starting a combination of scripts that get the crop box size of the currently active image and apply that size box to the rest of the selected images.

```
from subprocess import run

def process(command):

    run_command = ['osascript']

    if isinstance(command, list):
        for line in command:
            run_command.append('-e')
            run_command.append(line)
    else:
        run_command.append('-e')
        run_command.append(command)

    result = run(run_command,
                 encoding='utf-8',
                 stdout=PIPE,
                 stderr=PIPE)

    return result
```

**Figure 8.** Sample Python function to run an AppleScript command using macOS's `osascript` via Python's `subprocess.run()` function and return the result.

```
def result_to_list(result):

    return [x.strip('\n').strip(' ') for x
            in result.stdout.split(', ')]
```

**Figure 9.** Sample Python function to parse the standard output of an AppleScript command captured with the `process()` function (see Fig. 8) into a list of values.

## Conclusion & Future Work

Our adaptation of the Stream Deck XL as an input device for our camera-based capture stations using sets of custom buttons sorted by the different stages of digitization has been successful from the viewpoint of a student Digital Imaging Technician by simplifying or eliminating repetitive tasks and providing a flexible interface that can be customized by each employee according to their needs and preferences. And from a Technical Manager's perspective by increasing throughput, lowering training time, and increasing worker confidence.

In streamlining our system's human-computer interface by adapting Elgato's hardware, we have developed a flexible system that can be customized as needed to alleviate confusion and avoid errors and inconsistencies. Onboarding new hires will take less time and more of their training can be overseen by a senior imaging technician, thus freeing up the DPL's manager to continue

researching and developing new methods to improve our workflows. By using boilerplate code and custom Python modules to generate AppleScript to control macOS and Capture One Pro, new actions can be quickly prototyped and tested based on the suggestions of technicians or the needs of a particular project.

Future research includes adapting the device for sheet-feed scanning and metadata creation and logging data during its use to track the work of individual technicians and our progress on projects. We are also investigating methods to use the Stream Deck XL as a dashboard to retrieve and report information on the buttons themselves, such as the number of images in the current capture directory.

We believe our adaptation of Elgato's hardware using custom software can serve as a model and inspiration to others labs and organizations looking to innovate and improve upon their current practices and processes.

## References

- [1] UT Libraries, "Digital Collections | Libraries at University of Tennessee, Knoxville," [Online]. Available: <https://digital.lib.utk.edu>. [Accessed 12 April 2020].
- [2] J. D. Moore and L. Trott, "[Re]building the digitisation lab at the University of Tennessee Libraries: A framework for success," *Journal of Digital Media Management*, 2020 (forthcoming).
- [3] J. D. Moore and T. Peters, "A camera-based approach for digitising a large-scale photographic negative collection," *Journal of Digital Media Management*, vol. 7, no. 2, pp. 163-173, 2019.
- [4] Elgato, "Stream Deck XL | elgato.com," [Online]. Available: <https://www.elgato.com/en/gaming/stream-deck-xl>. [Accessed 12 April

2020].

- [5] Elgato, "Stream Deck Key Creator | elgato.com," [Online]. Available: <https://www.elgato.com/en/gaming/keycreator>. [Accessed 12 April 2020].
- [6] J. D. Moore, "automating\_capture\_one\_pro/Python Scripts at master · jeremydmoore/automating\_capture\_one\_pro," [Online]. Available: [https://github.com/jeremydmoore/automating\\_capture\\_one\\_pro/tree/master/Capture%20One%20Scripts](https://github.com/jeremydmoore/automating_capture_one_pro/tree/master/Capture%20One%20Scripts). [Accessed 12 April 2020].
- [7] J. D. Moore, "automating\_capture\_one\_pro/Python Scripts at master · jeremydmoore/automating\_capture\_one\_pro," [Online]. Available: [https://github.com/photosbyjeremy/automating\\_capture\\_one\\_pro/tree/master/Python%20Scripts](https://github.com/photosbyjeremy/automating_capture_one_pro/tree/master/Python%20Scripts). [Accessed 12 April 2020].
- [8] S. Sheppard, "osascript Man Page - macOS - SS64.com," [Online]. Available: <https://ss64.com/osx/osascript.html>. [Accessed 12 April 2020].

## Author Biographies

*Jeremy Moore has over 20 years' experience in photography and 13 years managing digitization labs and grant projects. He holds an MFA in Photography from the University of North Texas and has been the Technical Manager of the Digital Production Lab in Hodges Library at the University of Tennessee since 2019.*

*Andrew Coggins is a senior undergraduate student in the College of Communications and Information at the University of Tennessee, and expects to graduate in 2021 with a Bachelor of Arts in Advertising. He has been a Digital Imaging Technician in the Digital Production Lab at Hodges Library since 2018.*