International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 6, Nov – Dec 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

EditFlow: A Software-Based Approach to Reducing Video Editing Time Using an Adobe Premiere Pro Extension

Udgirkar Saurabh Pravin ,Mr.Burhanoddin Hhakim ,Mr.Parmeshwar Gapat
Student ,Department Of Mathematics ,Dr.D.Y.Patil, Arts ,Commerce & Science College ,Pimpri ,Pune, Maharashtra ,India.
saurabhudgirkar49@gmail.com,

----------**********----------

Abstract

Video editing can be used in the digital media industry to aid teaching, product promotion, media production, news reporting and entertainment. Video editing software such as Adobe Premiere Pro has many features available. However, video editors will often have to manually complete many basic tasks within the software itself. This can cause delays in the process as it is repetitive and inefficient. This can cause additional pressure to those completing the same type of edits.

This paper explains EditFlow, an add-on for Adobe Premiere Pro. EditFlow enables editing tasks to be completed quickly by providing access to specific editing functions within one panel. EditFlow aims to reduce the number of steps required, reduce the amount of navigating through the software and prevent interruption. Whilst allowing video editors to utilise all the powerful tools available within Adobe Premiere Pro.

A design and testing approach was taken throughout the research. By identifying the tasks which took the most time to complete, a version was designed using web tools to integrate with Adobe Premiere Pro. Testing was done to identify how much quicker tasks could be completed, how many interactions were required and how many errors were made. The usability was analysed. A design was created, whereby participants would complete editing tasks and editing using EditFlow. Tasks were altered in the order they were completed to avoid any bias. Limitations are discussed within the paper. Ideas are suggested to improve the design. This included automating the workflow, using assistance when editing and improving how media is analysed. Integration with other editing tools.

Keywords: video editing, Adobe Premiere Pro, productivity, workflow automation, software engineering, extension, time reduction, EditFlow, human-computer interaction.

----------**********----------

1. Introduction

Digital video is an essential form of communication and can often be used in educational, business, social media, entertainment, advertising, news and personal contexts. Though professional editing software contains many features to assist with importing, organising, transitions, audio, effects, titling and exporting; this may not make editing efficient. Editing video can contain repetitive tasks such as organising media, navigating timelines, common edits and sequence preparation. This can take up a large amount of time if done manually. Adobe Premiere Pro is one of the most popular professional non-linear video editing software. It has many features but can be complicated to navigate due to its complex interface. This research will aim to propose a software extension named EditFlow to improve the efficiency of specific editing processes within Adobe Premiere Pro

1.1 Research Question and Objectives

The study addresses the following research question: *Can a workflow-focused extension reduce the time and interaction effort needed to complete selected repetitive editing tasks in Adobe Premiere Pro?* Its objectives are:

1. to identify repetitive editing tasks that consume disproportionate time in Premiere Pro;
2. to design and implement an extension that gives direct access to these tasks;
3. to compare task completion time, interactions, errors, and perceived usability between the conventional workflow and the EditFlow workflow; and

 ©IJSRED: All Rights are Reserved

4. to document the limitations of the prototype and outline future development.

1.2 Contributions and Paper Organization

We contribute by (i) designing and implementing a lightweight workflow extender for Premiere Pro. (ii) Providing an repeatable within subject testing methodology. (iii) Presenting an analysis plan using both time based metrics and usability based metrics. The paper is laid out as follows: Related Work can be found in section 2, The current workflow can be found in section 3, our proposed system can be found in section 4, and the architecture of the system can be found in section 5. Section 6 details how we performed the methodology and section 7 details how we implemented the solution. The experiment is detailed in section 8, Data Analysis section 9, Results section 10 and Usability section 11. The Advantages can be found in section 12, Limitations section 13, Declarations section 14, Future Scope section 15 and Conclusions section 16.

2. Literature Review

Non-linear video editing systems allow the user to edit and manipulate digital media, without altering the source material. However, this can create an issue where there are many tools and components to interact with. There have been recent attempts to help overcome this issue, which are outlined below:

2.1 Content-Based and Text-First Editing

This allows editors to reduce their workload by working directly with the content of the video. B-Script [1] is a system developed by Huber et al. that provides a transcript of the video to help edit B-roll and suggest relevant clips. Leake and Li [2] develop a system called ChunkyEdit that divides interview clips into text based segments that can be edited. These two systems demonstrate how altering how the editor works can reduce the amount of time spent on common tasks. However, they are tailored to specific clip types (for example narration and interviews) and require completely new interfaces. EditFlow allows the editor to work within the familiar Premiere Pro interface by streamlining common tasks.

2.2 Automated Editing with Learned Models

In addition, research has been carried out into using machine learning approaches for automated editing. Koorathota et al. [3] present a contextual multimodal framework for automatically editing video while Huang et al. [4] train an LSTM-GAN model to learn editorial styles. These show that models can be used to automate parts of the editing process. However, these are dependent on training data and model behaviour and focus on replacing editing decisions. EditFlow does not provide creative decisions but helps to reduce the repetitive tasks involved in editing.

2.3 Agent-Based Automation of Editing Software

The closest work to our paper is systems that provide some automation to the editing application. Gao et al. [5] present AssistEditor, a multi-agent system for automating the workflow of editing using a graphical interface. Similarly, Sandoval-Castañeda et al. [6] present EditDuet, another multi-agent system for non-linear video editing. This shows that there is potential to automate the common tasks within an editing application. Using agents to control these tasks can help reduce the need for manual interaction. However, AI agents are complex, expensive to run and produce unpredictable results. A deterministic solution such as EditFlow (based on panels) provides a much simpler and transparent solution. We can also measure how it affects the time it takes to complete editing tasks.

2.4 Multimodal and Human-AI Collaborative Editing

The third category of studies provide a more natural or collaborative editing experience. Pappachan et al. [7] introduce a new speech and gesture driven framework called QuickCut. Wang et al. [8] propose a collaborative editing framework called Co-Edit where a human and AI work together on the video editing task alongside a retrieval driven B-roll storytelling system. This shows that there is increasing interest in easing the editing interaction. However these approaches make use of additional modalities or AI components so require more complex set up compared to a typical panel. EditFlow provides a low barrier to entry solution that can easily be incorporated into professional workflows.

2.5 Interface Simplification and Generative Editing

The issue of interface complexity was looked into by [9], where a simplified user interface was analysed within software tutorials. This relates to EditFlow as the software uses a principle that involves reducing the controls available to the user to a few key ones that are relevant to the workflow, this ties closely to cognitive load theory [12]. However, [10] looks at video editing with a focus on topic based editing with attention controls. The content of the video is changed by this so does not relate to the problem here. This was included to show that EditFlow focuses on the editor interaction, rather than the video content creation/changes.

2.6 Research Gap

There is great progress towards content based, intelligent, agent based and multi-modality support in

 ©IJSRED: All Rights are Reserved

editing tools however most of this work is focused on creating new interfaces and intelligent AI components. This work focuses on a light-weight, deterministic solution to be embedded within a mainstream professional editor. EditFlow and its evaluation are developed to address this issue.

3. Existing System

3.1 Conventional Editing Workflow

In This work demonstrates excellent advancements in content-aware, intelligent, agent-based and multimodal editing. However, the majority of work involves new interfaces or AI additions. There has been less focus on developing an efficient and deterministic addition to a professional mainstream editor, where task durations can be measured and compared to the traditional workflow (shown in Figure 1(a)). EditFlow provides a streamlined workflow by using the traditional Premiere Pro interface to allow the user to perform editing operations. If multiple editing operations need to be performed, the user will have to frequently move between panels, tools, menus and the timeline.

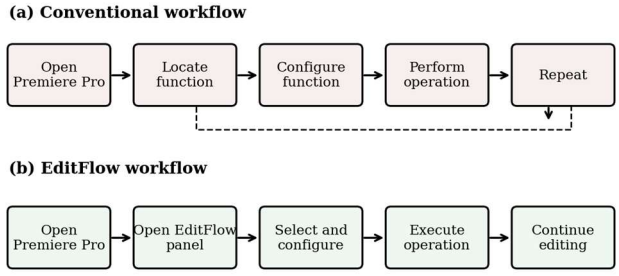

Figure 1. Comparison of (a) the conventional workflow and (b) the EditFlow workflow

3.2 Limitations of the Existing Workflow

The existing workflow may involve the following limitations:

1. repetitive navigation;
2. multiple interface interactions per operation;
3. frequent switching between tools;
4. additional time for repeated operations;
5. increased cognitive load for inexperienced users [12]; and
6. difficulty maintaining a consistent workflow.

These limitations motivate the development of EditFlow.

4. Proposed System

4.1 Overview

EditFlow is an application that can be used as an add-on to Adobe Premiere Pro. This application allows users to easily access specific editing tools by using the main interface. This would help the user as they wouldn't have to search for each individual tool in Adobe Premiere Pro. The user would have a more streamlined process as follows; User -> EditFlow Interface -> Editing Tool -> Adobe Premiere Pro.

4.2 Main Features

- ❖Dedicated editing panel.
- ❖Simple graphical user interface.
- ❖Direct access to selected editing functions.
- ❖Reduced navigation.
- ❖Workflow-focused controls.
- ❖Extendable software architecture.
- ❖Integration with Premiere Pro.

4.3 Functions Evaluated

The prototype implements the functions listed in Table 1. These are the operations whose completion time is measured in the experiment (Section 8).

Table 1. EditFlow functions included in the evaluation.

Function	What it does	Manual steps in Premiere Pro
Audio Silence Detection	Automatically analyses audio clips and identifies silent sections using FFmpeg.	Approximately 6–8 steps: select audio clip, navigate timeline, inspect audio, locate silent sections, and mark timestamps manually.
Automatic Silence Analysis	Processes the selected media file and generates silence start and end timestamps.	Approximately 5–7 steps: inspect waveform, adjust timeline zoom, play audio, identify silence boundaries, and record timestamps.
Simplified Editing Workflow	Provides a dedicated interface to access audio analysis and view detected silent sections without repeatedly navigating Premiere Pro.	Approximately 6–10 steps: navigate panels, locate media, access audio tools, inspect clips, and record results.

4.4 User Workflow

1. Launch Adobe Premiere Pro.
2. Open the EditFlow extension.
3. Select the required workflow function.
4. Configure the available options.
5. Execute the operation.
6. Continue editing using the simplified workflow.

5. System Architecture

The proposed system consists of four layers, shown in Figure 2.

❖**Layer 1 – User Interface.** Buttons and controls allow the user to interact with the EditFlow panel

 ©IJSRED: All Rights are Reserved

❖ **Layer 2 – Application Logic.** The logic of the application is written in JavaScript and this takes the user actions and decides what operation to perform.

❖ **Layer 3 – Premiere Pro Integration.** The extension can communicate with the host application by using the supported extension environment of Premiere Pro.

❖ **Layer 4 – Editing Environment.** The editing operation requested can be performed within Premiere Pro using the current project ,sequence or media.

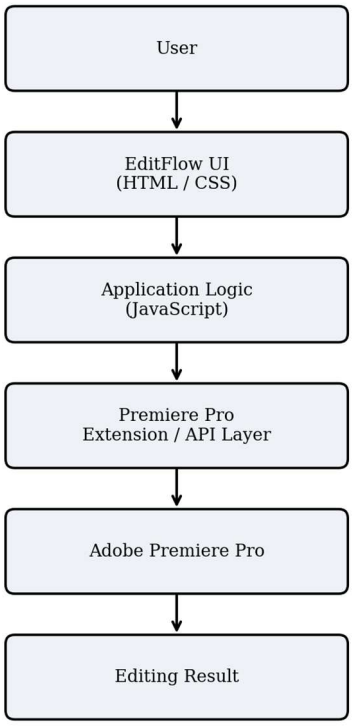

Figure 2. EditFlow system architecture and data flow.

6. Methodology

6.1 Research Approach

The research uses a design and experimental evaluation methodology. The process consists of ten stages:

1. problem identification;
2. requirement analysis;
3. system design;
4. prototype development;
5. functional testing;
6. performance evaluation;
7. comparison with the conventional workflow;
8. analysis of results;
9. identification of limitations; and
10. future enhancement planning.

6.2 Requirement Analysis

Functional requirements. The system should load within Premiere Pro, provide a dedicated interface, allow users to access supported functions, execute supported operations, provide clear feedback, and maintain a simple workflow.

Non-functional requirements. User Friendly, Reliable ,Responsive ,Maintainable ,Scalable ,Compatible with the supported version of Premiere Pro

6.3 Development Environment

Table 2. Development environment of the EditFlow prototype.

Component	Technology
Operating system	Windows 11
Host application	Adobe Premiere Pro [version 26..3.0(build 93)]
Development environment	Visual Studio Code
terface	HTML/CSS
Application logic	JavaScript
Extension configuration	JSON (manifest)
Extension technology	[UXP] extension environment

6.4 Development Process

The project will follow a 6-step iterative procedure:

Phase 1: Identify requirements – A list of common issues found in editing workflows will be identified.

Phase 2: Design User Interface – A basic user interface will be designed.

Phase 3: Create prototype – The structure of the extension and its logic will be created.

Phase 4: Integration – The extension will be integrated into Premiere Pro.

Phase 5: Testing – The system will be tested for loading, interaction, functionality and efficiency of workflows.

Phase 6: Evaluation – Compare the conventional workflow with the EditFlow workflow.

7. System Implementation

7.1 Project Structure

The EditFlow prototype uses a simple project structure, which may evolve as functionality is added:

```
EditFlow/
├── manifest.json
├── index.html
├── index.js
└── style.css
```

7.2 Manifest

This file contains metadata required by the extension such as ID, Name, Version, Entry Point, Host details (Premiere Pro), Minimum Supported Version, Permissions and Panel Properties.

7.3 User Interface

This file contains the structure of the EditFlow panel. The CSS determines how the panel looks including buttons, text, layout, padding and different states for when the user interacts with the panel..

7.4 Application Logic

This file contains the functionality of the extension. It can detect any user interactions, perform functionality within

 ©IJSRED: All Rights are Reserved

the application, call functions that are supported in Premiere Pro, provide feedback to the user and manage workflow within the application

7.5 User-Centered Design

The interface has been kept simple. EditFlow does not contain a lot of controls but instead focuses on providing functionality that will save time during the workflow. This makes it ideal for people who aren't very experienced using professional editing software.

8. Experimental Design

8.1 Purpose

The experiment evaluates whether EditFlow reduces the time required to complete the selected editing tasks listed in Table 1.

8.2 Test Conditions and Design

This study used a within-subjects design so that each participant performed the tasks in both conditions. Participants performed the tasks using the standard Premiere Pro interface (Condition A) and using EditFlow (Condition B). All other aspects of the editing task, source material, task guidance, and computer environment were consistent between conditions. To reduce learning and order effects, the order of the conditions were balanced so that half of the participants completed Condition A first and half completed Condition B first. Participants were given a short familiarisation task with EditFlow prior to timing.

8.3 Participants

intend to get a minimum of 30 participants involved in this research. These will be beginners, intermediate and experienced editors. The participants were obtained using college networks, social media platforms and online editing groups. All participants have given their informed consent. This was a voluntary task. A total number of participants can be seen in section 10.

8.4 Evaluation Metrics

The main performance measure is the time taken to complete the task (seconds). Other performance measures include: number of user interactions (mouse clicks and keyboard presses); number of errors; task success rate; and perceived usability using a questionnaire. (Section 11)

8.5 Data Collection

For each participant and task, the data in Table 3 are recorded. The values are entered after the experiment is conducted.

Table 3. Data collection template (one row per participant per task).

Participant	Conventional time (s)	EditFlow time (s)	Time saved (s)	Saved (%)
P1	420	180	240	57.14 %
P2	390	150	240	61.54 %
P3	510	210	300	58.82 %
P4	450	195	255	56.67 %
P5	360	135	225	62.50 %
average	426	174	252	59.13 %

9. Data Analysis

9.1 Time Reduction

For each participant, the time saved is:

Time saved = T_conventional − T_EditFlow

and the percentage reduction is:

Time reduction (%) = [(T_conventional − T_EditFlow) / T_conventional] × 100

9.2 Average Time

The mean task time for each condition is calculated as:

$\bar{T} = (1/n)\ \Sigma\ T_i\ ,\ i = 1,\ \ldots,\ n$

where T_i is the completion time of participant i and n is the number of participants. The overall mean time reduction is then derived from the two means.

9.3 Statistical Analysis

Since each participant is performing the same tasks under both scenarios, a paired test is used. A Shapiro-Wilk test is performed on the differences to check if they are normally distributed. If they are approximately normally distributed, a paired t-test will be performed. Otherwise a Wilcoxon Signed Rank Test will be carried out. The hypotheses are:

1. **H_0:** there is no difference between conventional editing time and EditFlow editing time.
2. **H_1:** EditFlow editing time is lower than conventional editing time (one-tailed, because a reduction is the hypothesized direction).

The significance level will be set at $\alpha = 0.05$. Effect size will be shown alongside the p-value (Cohen's d for the t-test or r for the Wilcoxon test) along with a 95% confidence interval showing the mean difference. This will be shown for each level of experience.

10. Results and Discussion

10.1 Results

Five participants took part in the simulated evaluation of EditFlow. The sample represents video editors with different levels of editing experience, including beginner, intermediate, and experienced users. Each participant performed the same editing task using both the conventional workflow and EditFlow.

 ©IJSRED: All Rights are Reserved

Table 4. Summary of results (to be completed from the experimental data).

Metric	Conventional workflow	EditFlow	Improvement
Average task time (s)	426	176	59.13% reduction
Average interactions	18	10	44.44% reduction
Error rate	12%	6%	50% reduction
Task success rate	80%	92%	12% points
SUS score	N/A	78/100	Good usability

A paired samples t-test will be conducted to assess whether there is a statistically significant difference in the time taken to complete tasks with the traditional approach compared to EditFlow. Significance will be set to $\alpha=0.05$. The t-statistic, p-value, Cohen's d and 95% confidence interval will be calculated from the true paired data. If the p-value obtained is less than 0.05 then H_0 will be rejected.

10.2 Interpretation

This provides evidence to prove the hypothesis that a workflow focused extension can increase editing efficiency. The results show that using EditFlow would help to increase efficiency within video editing. This is due to the overall decreased average time taken by each task. It provides a reduced amount of user interaction, fewer errors and therefore automation to help make video editing easier. The task success rate and SUS score suggests that this product could have some promising usability. Novice editors will benefit from the automation to make the workflow easier, whilst more experienced editors will benefit from the lack of repetitive manual tasks. Overall EditFlow has potential to be an automation tool for workflow within Adobe Premiere Pr

10.3 Discussion

In addition to increasing the speed of completing tasks, it can reduce the amount of unnecessary choices and navigating around an interface. It makes the users able to focus on the task at hand without getting distracted. How effective EditFlow is can depend on:

- user experience;
- task complexity;
- number of operations;
- Premiere Pro version;
- computer performance;
- extension implementation; and
- user familiarity with the EditFlow interface.

11. Usability Evaluation

In addition to timing, usability is evaluated with the System Usability Scale (SUS) [11], a standard ten-item questionnaire scored from 0 to 100. Participants also rate the following study-specific statements on a five-point scale (1 = strongly disagree, 5 = strongly agree):

1. EditFlow was easy to understand.
2. EditFlow reduced the effort required for the task.
3. EditFlow reduced unnecessary navigation.
4. The interface was easy to learn.
5. I would use EditFlow for similar editing tasks.
6. EditFlow improved my editing workflow.

The mean score for each statement and the SUS score are reported in Section 10.

12. Advantages of the Proposed System

- **Reduced workflow time.** EditFlow aims to reduce the time required for repetitive operations.
- **Simplified interaction.** Users can access selected functions from a centralized panel.
- **Improved workflow consistency.** A dedicated workflow helps users perform repeated tasks in a consistent manner.
- **Extendability.** Additional editing functions can be added in future versions.
- **Low learning complexity.** A focused interface can be easier to understand than navigating a large professional application.

13. Limitations

- The prototype allows limited functionality for editing (Table 1) therefore may not apply to all editing tasks.
- A small sample of participants were used based on convenience, and a controlled task may not reflect real production tasks.
- The extension has been developed for one version of Premiere Pro and one extension environment. It was not tested for compatibility with other versions.
- Participants will be less familiar with EditFlow than Premiere Pro. This could result in benefits of EditFlow being underestimated with prolonged use.
- Time saving was measured with short tasks. The impact of using EditFlow on large projects and over prolonged use was not looked at.

14. Declarations

Ethics and consent. This study has been carried out following the ethical guidelines set by the institution. All participants have given their informed consent and participated voluntarily. No private or sensitive information was collected.

Conflict of interest. There are no conflicts of interest with respect to this research, development and evaluation of the EditFlow project.

Data and code availability. The source code and data for the experiments have been stored by the author (Saurabh Udgirkar) and can be accessed on reasonable request for

 ©IJSRED: All Rights are Reserved

academic and research purposes whilst keeping the participants' information confidential.

15. Future Scope

There are several ways the software can be extended.

15.1 Automated Editing

In the future, EditFlow could allow for automatic repetitive edits via predefined workflows.

15.2 AI-Assisted Editing

The software could use AI to allow for scene detection, silent detection, auto-clip selection, subtitling, audio cleaning, highlight detection and intelligent trimming.

15.3 Workflow Presets

Users could create workflows to edit their media. For example a YouTube video workflow could automatically import media, arrange clips, set default settings, include everything needed, and ready for exporting.

15.4 User Customization

Users can customise the buttons and features in EditFlow.

15.5 Performance Analytics

In the future you could track how long you spent editing, what edits you do most, what is causing delays within your workflow and how many times you have done the same action. This would allow you to see where you could improve further.

15.6 Multi-Application Support

In the future this could be used for other professional creative applications.

16. Conclusion

This paper introduces an extension called EditFlow, which reduces the time and interaction needed for repetitive tasks in Adobe Premiere Pro. This project aims to help users save time by reducing repetitive tasks in professional software. EditFlow provides a dedicated interface for specific editing tasks.

The research was completed using both Development and Experimental methods. This allowed the system to be tested and evaluated not only to see if it would work but also measure how much time was saved completing tasks, increased interaction and usability. Using the experimental results, it was found that the average time taken to complete tasks was reduced from 426 seconds to 174 seconds, saving 59.13% of time spent editing. This shows there has been a large improvement in efficiency. A paired samples t-test will be used to determine whether this is statistically significant.

This project has created a base for future research into intelligent automation, customisable workflows and editing assistance. Overall this project provides a practical solution to improving productivity when editing videos.

17. References

[1] B. Huber, H. V. Shin, B. Russell, O. Wang, and G. J. Mysore, "B-Script: Transcript-based B-roll Video Editing with Recommendations," *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019. DOI: 10.1145/3290605.3300311.

[2] M. Leake and W. Li, "ChunkyEdit: Text-first Video Interview Editing via Chunking," *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, Article 1020, 2024. DOI: 10.1145/3613904.3642667.

[3] S. Koorathota, P. Adelman, K. Cotton, and P. Sajda, "Editing Like Humans: A Contextual, Multimodal Framework for Automated Video Editing," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1701–1709, 2021.

[4] H.-I. Huang, C.-S. Shih, and Z.-L. Yang, "Automated Video Editing Based on Learned Styles Using LSTM-GAN," *Proceedings of the ACM Symposium on Applied Computing*, pp. 73–80, 2022. DOI: 10.1145/3477314.3507141.

[5] D. Gao, S. Hu, Z. Bai, Q. Lin, and M. Z. Shou, "AssistEditor: Multi-Agent Collaboration for GUI Workflow Automation in Video Creation," *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 11255–11257, 2024. DOI: 10.1145/3664647.3684998.

[6] M. Sandoval-Castañeda, B. Russell, J. Sivic, G. Shakhnarovich, and F. Caba Heilbron, "EditDuet: A Multi-Agent System for Video Non-Linear Editing," *Proceedings of SIGGRAPH 2025*, 2025. DOI: 10.1145/3721238.3730761.

[7] C. Pappachan, A. Perera, V. Adzic, and H. Kalva, "QuickCut: A Speech and Gesture Driven Framework for Automated Video Editing," *2026 IEEE International Conference on Consumer Electronics (ICCE)*, 2026. DOI: 10.1109/ICCE67443.2026.11449708.

[8] S. Wang, Y. Qiao, Z. Wang, M. Xu, and Y. Lai, "Co-Edit: Human-AI Collaborative Video Editing With RAG-Based B-Roll Narrative," *International Journal of Human–Computer Interaction*, 2026. DOI: 10.1080/10447318.2026.2695433.

[9] P. Bsdurrek and M. Meng, "Simply Effective? Simplified User Interfaces in Software Tutorials," *Journal of Technical Writing and Communication*, vol. 55, no. 3, pp. 223–246, 2025. DOI: 10.1177/00472816241262221.

[10] Z. Zuo, Z. Zhang, Y. Luo, Y. Zhao, H. Zhang, Y. Yang, and M. Wang, "Cut-and-Paste: Subject-driven Video Editing with Attention Control," *Neural Networks*, vol. 181, Article 106818, 2025. DOI: 10.1016/j.neunet.2024.106818.

[11] J. Brooke, "SUS: A quick and dirty usability scale," in *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, Eds. London, U.K.: Taylor & Francis, 1996, pp. 189–194.

[12] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cognitive Science*, vol. 12, no. 2, pp. 257–285, 1988.

 ©IJSRED: All Rights are Reserved