Jiang, B. et al. (Eds.) (2025). Proceedings of the 33rd International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

# Vid2Log: A Machine Learning Approach to Structured Screen Activity Logging

**Bhavik DODDA[a*], Vishwas BADHE[b] & Ramkumar RAJENDRAN[b]**
[a]*Department of Mathematics, Sardar Vallabhbhai National Institute of Technology, India*
[b]*Centre for Educational Technology, Indian Institute of Technology Bombay, India*
*bhavikdodda22@gmail.com

**Abstract:** Collecting and analyzing learner actions on computer systems during ill-structured tasks presents significant methodological challenges, as these activities often span multiple tools and environments beyond the scope of specialized logging systems. Consequently, researchers frequently rely on screen recordings to capture comprehensive user behavior, necessitating subsequent manual conversion into structured activity logs. This manual annotation process is inherently labor-intensive, error-prone, and severely limits the scalability of behavioral analysis studies. To address these limitations, we present Vid2Log, an open-source tool that automates the conversion of raw screen-recording videos into structured action logs through machine learning techniques. Our approach leverages transfer learning, adapting a pre-trained model to the specific domain of computer screen-recording analysis. This methodology proves highly efficient by significantly reducing both the need for extensive training data and computational resources compared to developing models from scratch. Vid2Log specifically targets the generation of macro-level activity logs from screen-recording videos, thereby reducing annotation overhead while maintaining the granularity necessary for meaningful behavioral analysis. This capability enables large-scale studies of complex workflows across diverse domains, including programming, education, design, and ill-structured problem solving in computer-based systems.

**Keywords:** Video logging, Transfer learning, Programming tasks, Action classification

## 1. Introduction

The analysis of human-computer interaction patterns has become increasingly critical across multiple research domains, particularly in education and software engineering where understanding user behavior can inform both pedagogical strategies and system design improvements (Ge & Land, 2023). Many contemporary research studies depend on screen recordings to capture the full spectrum of learner or developer actions, especially when these activities extend beyond the boundaries of single, monitored environments.

While structured environments often permit the deployment of specialized logging systems that can directly track user actions with precision, a significant portion of real-world tasks are inherently ill-structured. These tasks typically involve complex workflows that span multiple tools, applications, and platforms, creating a distributed interaction landscape that cannot be adequately captured by conventional logging approaches (Jonassen, 2000; Ge et al., 2020). In such scenarios, researchers are compelled to depend on comprehensive screen recordings, which subsequently require manual transformation into analyzable activity logs.
This dependency on manual annotation introduces several critical limitations to behavioral research. The process is not only time-consuming and resource-intensive but also susceptible to human error and subjective interpretation bias (Harper & Sansone, 2018). Moreover, the manual approach fundamentally constrains the scalability of research studies, limiting the scope of investigations that can be feasibly conducted and potentially introducing sampling biases toward smaller, less representative datasets.

To overcome these methodological barriers, we introduce Vid2Log, an open-source tool designed to automate the generation of structured action logs from screen-recorded videos through advanced machine learning techniques. Vid2Log employs transfer learning to efficiently adapt pre-trained models to the specific task of screen activity recognition, producing macro-level logs that capture high-level activity patterns without sacrificing essential behavioral insights.

The distinction between macro-level and micro-level logging is crucial to understanding Vid2Log's approach. While micro-level logs capture granular details such as individual keystrokes, mouse movements, or specific interface interactions, macro-level logs focus on higher-order activities and behavioral patterns. This strategic focus on macro-level logging serves a dual purpose: it significantly reduces annotation overhead while preserving the level of detail most relevant for understanding complex learning and problem-solving processes.

This paper presents a comprehensive examination of Vid2Log structured in two complementary parts. The first part introduces Vid2Log as a broadly applicable tool suitable for diverse domains where ill-structured tasks occur, establishing its theoretical foundation and technical implementation. The second part demonstrates Vid2Log's practical utility through a specific use case, showing how the tool can be applied to log learners' macro actions and extract meaningful insights about learner behavior patterns.

## 2. Related Work

The challenge of analyzing complex human-computer interactions has generated extensive research across multiple domains, each contributing unique perspectives and methodological approaches to understanding user behavior. Traditional approaches to behavior analysis have relied heavily on manual annotation of video recordings or system-generated log data, creating a methodological bottleneck that fundamentally limits research scalability and introduces potential for human error and inconsistency (Romero & Ventura, 2024).

### *2.1 Traditional Logging Approaches and Their Limitations*

Activity tracking tools, including screen-capture utilities and integrated development environment (IDE) logging plugins, have long served as the primary means of capturing user behavior in computational environments. These tools can provide detailed insights into user actions within their specific domains of coverage, offering high-resolution data about interactions within constrained environments (Bond et al., 2024). However, their effectiveness is fundamentally limited when dealing with ill-structured tasks that naturally span multiple tools and platforms. The limitation becomes particularly pronounced in educational contexts where learners frequently navigate between different applications, web browsers, development environments, and external resources as part of their problem-solving process. Traditional logging systems, designed for specific applications or environments, inevitably capture only fragments of the complete behavioral picture, leaving researchers with incomplete data that may not accurately represent the full complexity of user workflows.

### *2.2 Machine Learning Applications in Educational Video Analysis*

Recent advances in machine learning, particularly in computer vision and deep learning, have opened new possibilities for automated video analysis in educational contexts. Convolutional neural networks (CNNs) have demonstrated significant effectiveness in recognizing actions and patterns in constrained video domains, showing particular promise in educational applications where consistent environmental conditions can be maintained (Xu et al., 2024).
Studies in educational data mining have increasingly explored the potential of automated video analysis to supplement traditional data collection methods. The growing field of learning analytics has recognized video data as a rich source of behavioral information that can provide insights into learning processes that are difficult to capture through conventional assessment

methods (Romero & Ventura, 2024). However, the application of these techniques to the specific domain of screen-recording analysis remains relatively underexplored, particularly in the context of transfer learning applications.

### *2.3 Transfer Learning in Video Classification*

Transfer learning has emerged as a particularly powerful approach for video classification tasks, offering significant advantages in scenarios where labeled training data may be limited or expensive to obtain. Recent research has demonstrated the effectiveness of transfer learning techniques in various video analysis contexts, from action recognition to anomaly detection (Scientific Reports, 2024). The fundamental principle of transfer learning leveraging knowledge gained from pre-trained models to solve related but distinct tasks, aligns well with the challenges of screen-recording analysis.
However, despite the proven effectiveness of transfer learning in general video classification tasks, its specific application to screen-recording analysis for educational and behavioral research purposes remains an underexplored area. This gap in the literature represents both a research opportunity and a practical need, as the unique characteristics of screen-recording videos including consistent interface elements, predictable interaction patterns, and domain-specific visual cues may offer particular advantages for transfer learning approaches.

### *2.4 Research Gap and Motivation*

The convergence of these research streams reveals a significant gap in current methodological approaches to behavioral analysis. While manual annotation remains the gold standard for accuracy and interpretability, its limitations in terms of scalability and resource requirements create barriers to large-scale research. Simultaneously, while machine learning techniques have shown promise in related domains, their specific application to screen-recording analysis for behavioral research has not been comprehensively addressed.

This gap motivates the development of Vid2Log as a domain-specific solution that combines the accuracy requirements of behavioral research with the scalability advantages of automated machine learning approaches. By focusing on macro-level activity recognition and leveraging transfer learning techniques, Vid2Log represents an attempt to bridge this methodological gap and provide researchers with a practical tool for large-scale behavioral analysis studies.

## 3. System Workflow

Our process is structured into four main stages: data preparation, model training, frame classification, and log generation. Vid2Log automates the conversion of raw screen-recorded videos into structured action logs through these stages. Each step is described in general terms and illustrated using our lab use case of analyzing programming and learning sessions.

### *3.1 Data Preparation*

The first step involves defining macro-level activity categories and preparing a labeled dataset for training. Researchers must identify activity classes that align with their specific research context. A small but representative dataset of screenshots is sufficient to train the classifier effectively.
Use Case Example: For programming and learning tasks use case, we chose 11 arbitrary action bins (macro-level) to generate an action log: ”browsing”, ”reading documentation”, ”using github”, ”using command prompt”, ”editing docs”, ”coding in VSC”, ”watching YouTube”, ”using file explorer”, ”preview output”, ”other”, ”split screen”. For each category, we manually labeled 10 to 20 screenshots, ensuring coverage of variations

such as dark/light IDE themes and different browser layouts. These categories, illustrated through representative screenshots in Figure 1, provided the foundation for developing the classifier.

Examples of micro-level actions are: “editing in VSC”, “viewing errors in VSC console”, “highlighting text in VSC”, “installing dependencies in VSC”, “installing plugins in VSC” etc.

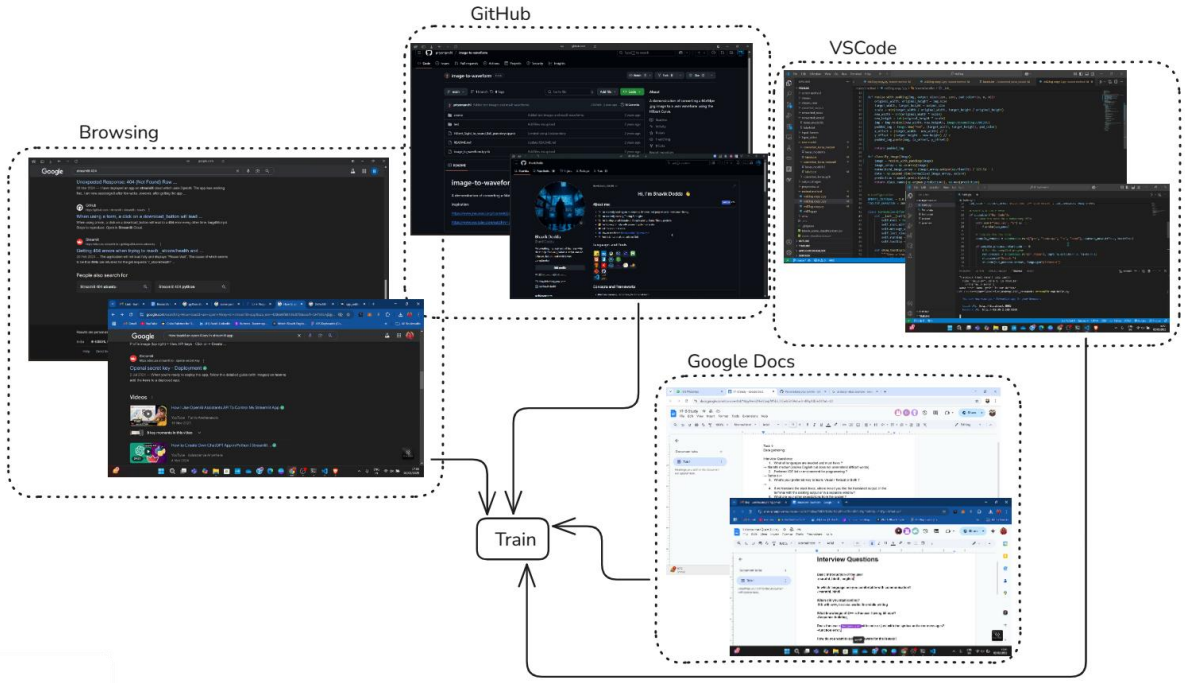



*Figure 1.* Training with a labelled dataset

### *3.2 Model Training*

We employed transfer learning by adapting MobileNet[1], a lightweight convolutional neural network pre-trained on ImageNet. The network was fine-tuned to recognize custom activity categories. Experiments were conducted to address challenges in resizing rectangular 1080p frames to 224x224 while minimizing data loss.

Use Case Example: In the programming dataset, the model successfully distinguished between visually similar activities (e.g., “coding in VSC” vs. “editing docs”) by learning contextual cues such as syntax highlighting, console layout, or text-editing interface.

### *3.3 Frame Classification and Log Generation*

After training, frames from new screen recordings are passed through the classifier to predict activity categories. Each prediction is time-stamped and aggregated into a sequential action log. This log captures macro-level transitions between activities and represents higher-order behavior patterns.

The system also includes an optional OCR feature, which can enhance classification by detecting text-based cues that support or override visual predictions. When detected keywords

[1] MobileNet: It’s a lightweight convolutional neural network architecture designed for efficient image classification tasks on limited-resource devices.

suggest different classifications than visual predictions, the system can override initial classifications (Figure 2).

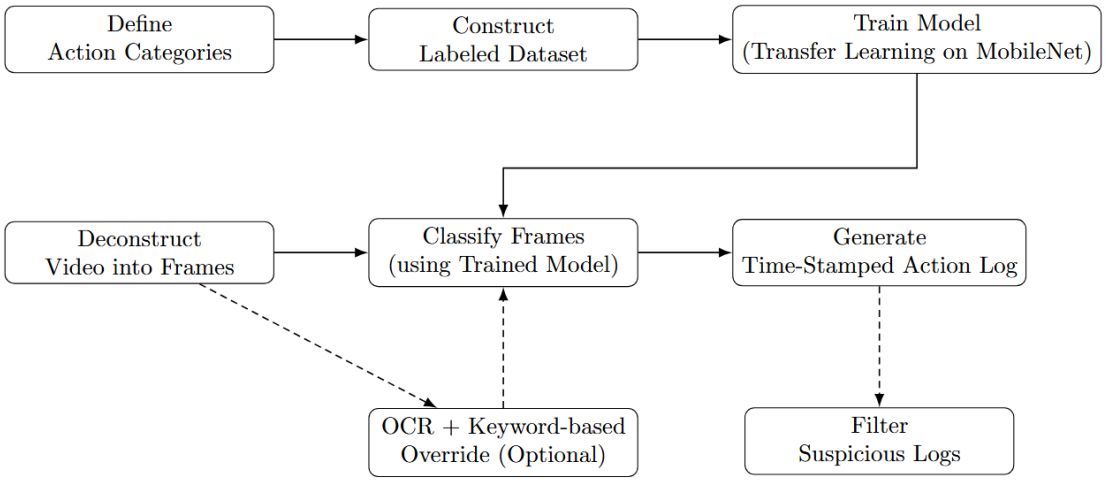



*Figure 2.* Methodology for macro log generation

*Use Case Example:* In a 5-hour programming session, a representative structured log generated by the system shows transitions such as a learner moving from coding to browsing documentation or consulting GitHub (Table 1).

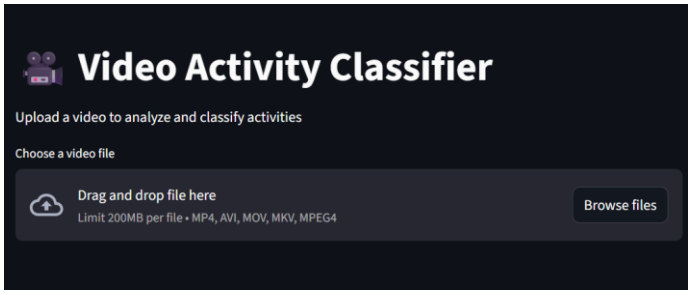



*Figure 3.* Tool Frontend

↓

| Start_time | End_time | Duration | Index | Class Name |
|---|---|---|---|---|
| 0:57:49 | 0:57:59 | 0:00:10 | 5 | coding in VSC |
| 0:57:59 | 0:58:00 | 0:00:01 | 7 | using file explorer |
| 0:58:00 | 0:58:05 | 0:00:05 | 2 | using github |
| 0:58:05 | 0:58:09 | 0:00:04 | 7 | using file explorer |
| 0:58:09 | 0:58:11 | 0:00:02 | 5 | coding in VSC |
| 0:58:11 | 0:58:18 | 0:00:06 | 4 | editing docs |
| 0:58:18 | 0:59:01 | 0:00:43 | 5 | coding in VSC |
| 0:59:01 | 0:59:03 | 0:00:02 | 6 | watching YT |
| 0:59:03 | 0:59:05 | 0:00:02 | 2 | using github |
| 0:59:05 | 0:59:07 | 0:00:01 | 1 | reading documentation |
| 0:59:07 | 0:59:08 | 0:00:01 | 4 | editing docs |
| 0:59:08 | 1:02:29 | 0:03:21 | 5 | coding in VSC |
| 1:02:29 | 1:02:30 | 0:00:00 | 6 | watching YT |
| 1:02:30 | 1:02:30 | 0:00:00 | 1 | reading documentation |
| 1:02:30 | 1:04:24 | 0:01:53 | 4 | editing docs |

*Table 1.* Sample log from a 5:22 hour video.

Note: Each row shows class-index (here ranging from 0-10) corresponding to the activity name.

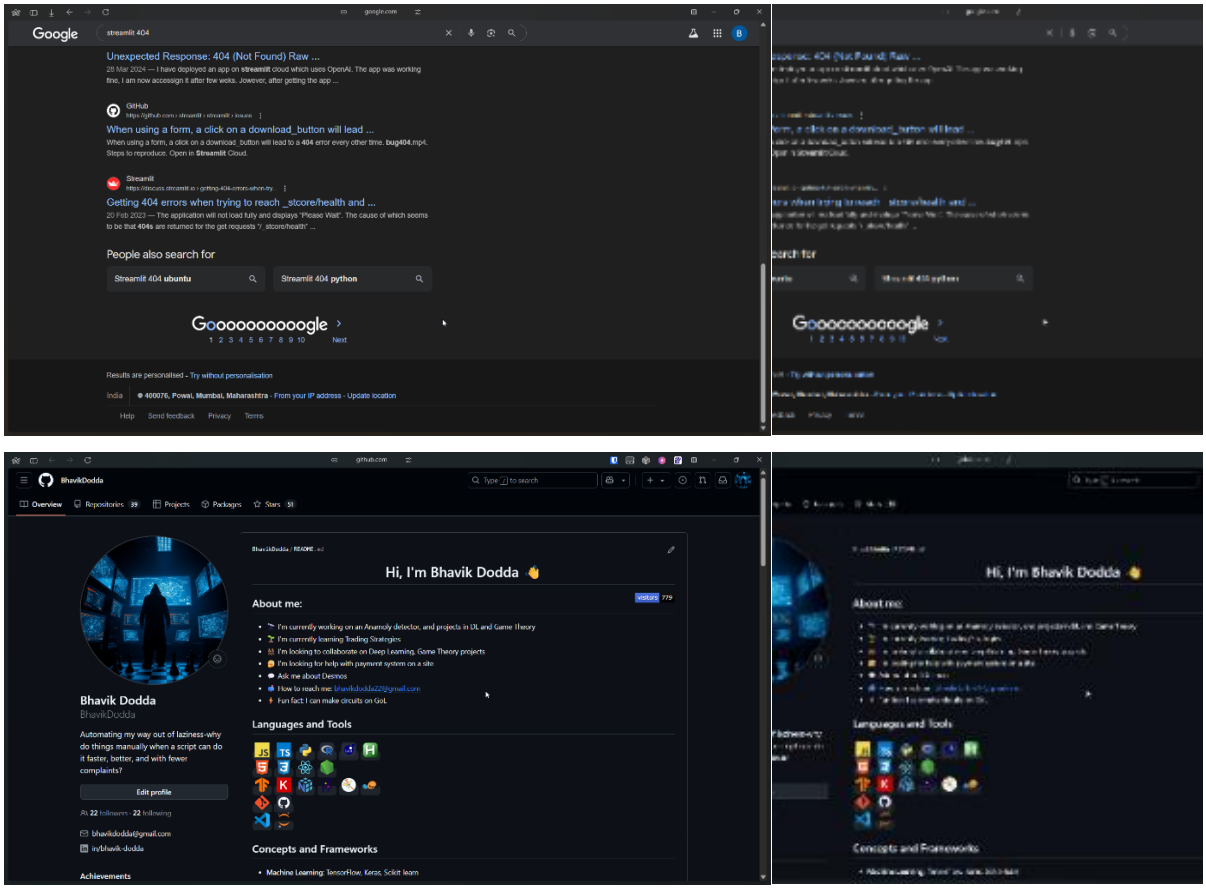
## 4. Challenges

### *4.1 Resizing*

Recordings varied in resolution, aspect ratio, and clarity, and we had to resize them to 224x244 for MobileNet input,

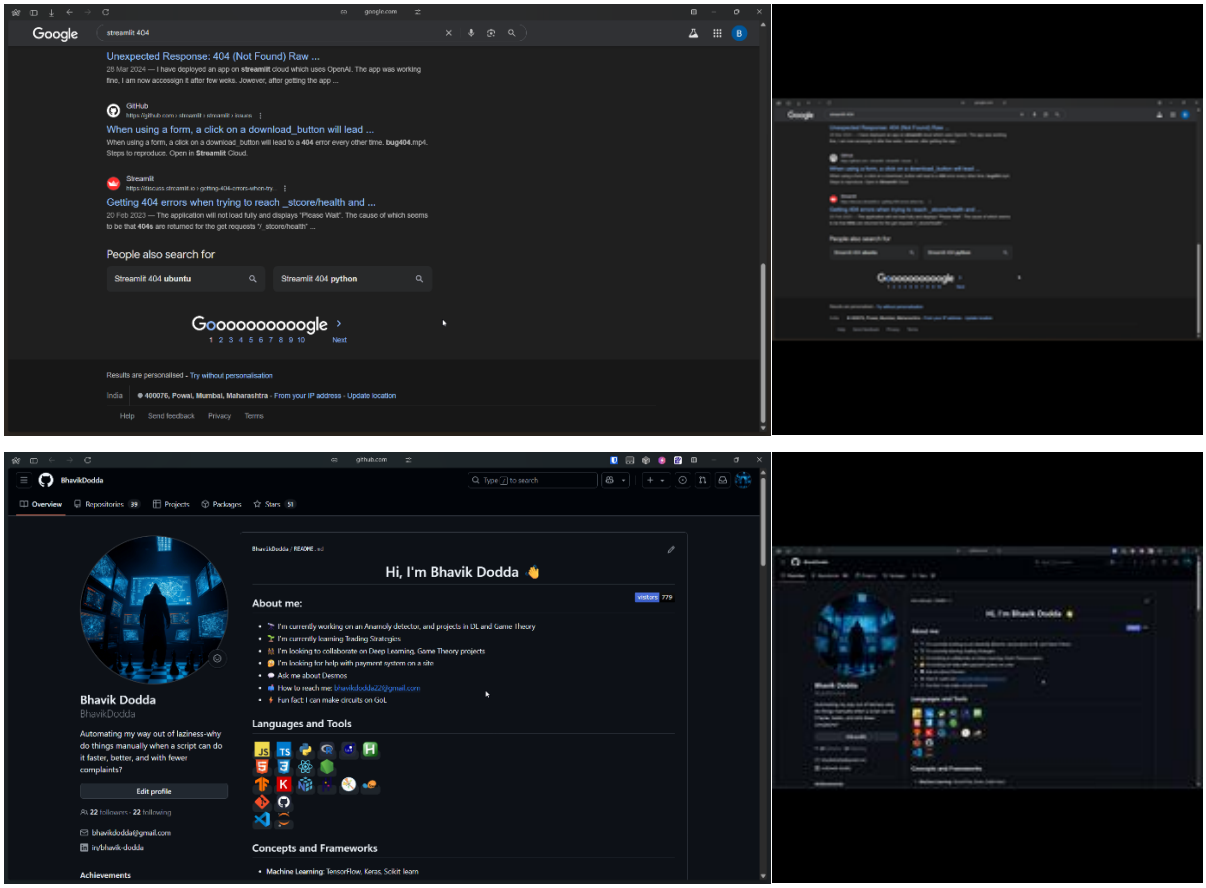
#### *4.1.1 Resize Method 1: Cropped Region with Higher Resolution*

In this method, only the central portion of the screen is captured. As shown in the browsing and GitHub examples, this method preserves more detail and clarity, which can be useful for distinguishing fine-grained features such as code structure or UI elements. However, the trade-off is that some contextual information near the edges of the screen is lost.




#### *4.1.2 Resize Method 2: Full Screen with Padding*

This method addresses the limitations of the previous one by preserving the entire screen layout. It scales the full screen to the target size and introduces padding at the top and bottom. This ensures that no contextual elements are cropped out, which is valuable when interpreting activities that rely on peripheral details. The trade-off is reduced effective resolution, making smaller elements less sharp while maintaining screen context. The following samples show how it preserves screen context.




During model training we also encountered challenges related to performance.

*Overfitting*

The primary challenge was preventing overfitting when adapting MobileNet to our small, manually labeled dataset. Limited training data required careful tuning to ensure generalization across unseen screen recordings.

Solution: Limit image quantities while ensuring comprehensive coverage of all variations within each category to maintain representativeness without redundancy.

*Accuracy*

Maintaining balanced accuracy across all 11 categories proved challenging due to potential class imbalance in real-world screen recording data.

Solution: Maintain equal sample numbers across categories to prevent underrepresentation of specific activity types.

## 5. Results

Vid2Log achieved consistent performance in classifying macro-level actions. In our programming use case, we observed an accuracy exceeding 85% across the 11 categories. To estimate this performance, we manually validated approximately 50 classified rows for each of at least 20 videos, each ranging from 3 to 6 hours in length. For every selected row, the predicted class label was compared with the actual class observed in the screen recording.

This manual cross-checking provided a reliable measure of classification accuracy at the macro-action level.

The system demonstrated strong generalization to unseen videos despite wide variations in resolution, color themes, and tool configurations. Rather than relying on superficial cues, the model learned structural features such as code editor layouts, terminal appearances, and browser window organization, enabling stable predictions across diverse contexts. Notably, the classifier maintained accuracy even when users employed different IDE themes, programming languages, or operating systems, demonstrating robust feature extraction capabilities. The temporal consistency of predictions was particularly impressive, with minimal flickering between activity classifications during extended periods of consistent user behavior.

These results suggest that transfer learning with MobileNet is a practical and effective approach for small, domain-specific datasets. MobileNet's pretrained weights provided a strong starting point, reducing the amount of labeled data needed. Its lightweight design also made the pipeline efficient enough for large-scale video processing. Together, these factors establish Vid2Log as a scalable tool for programming education research.

The resulting action logs provide structured insights into learner behavior, enabling large-scale studies that were previously infeasible due to manual annotation requirements.

## 6. Discussion and Future Work

Manual annotation of screen recordings for ill-structured tasks is labor-intensive, prone to errors, and difficult to scale. Converting raw video into structured logs alleviates this reliance on manual work and makes large-scale analysis of learner workflows feasible. To address this challenge, we developed Vid2Log, a tool that automates the generation of structured action logs from screen recordings.

For researchers, Vid2Log provides a means of observing learner behavior on the computer screen over extended periods of time. This perspective is important because learning involves both cognitive processes and the sequence of actions that students carry out, such as searching, coding, reading, or shifting between tools. Without access to this screen-level activity, researchers may find it difficult to establish links between learners' thought processes and their observable actions. Structured logs help bridge this gap by aligning verbal or written evidence with actual computer-based activity, producing a more complete account of learning behavior.

The tool is also useful for teachers in their classrooms. It provides a means of examining student activity beyond what can be observed during live instruction. Students frequently engage in diverse computer-based tasks such as coding, browsing for information, or working with documents, and much of this behavior is difficult to monitor in real time. By producing macro-level logs, Vid2Log enables post-hoc analysis of learner workflows allowing teachers to identify when students encounter challenges, switch contexts, or engage in unproductive detours. These activity traces provide evidence of where learners struggled and help design more targeted scaffolds, feedback mechanisms, and instructional interventions. This reduces the burden of manual observation while also strengthening the feedback loop between classroom practice and learner needs.

Future work will focus on improving classification granularity, increasing the size and diversity of the training dataset, letting users define custom classes and create their own workflow. We can also explore micro-level logging expanding the dataset with more sub-classes, and training models within each class to further classify into a sub-class. The open source release also invites researchers to contribute to the datasets and implementation.

## References


Ge, X., & Land, S. M. (2023). A conceptual framework for panning as a screen recording technique in software education: Implications for instructional design in educational videos. Educational Technology Research and Development, 71(4), 1245-1268.

Ge, X., Law, V., & Huang, K. (2020). Understanding learners' challenges and scaffolding their ill-structured problem solving in a technology-supported self-regulated learning environment. In Self-Regulation in Technology Enhanced Learning Environments (pp. 285-302). Springer. https://link.springer.com/chapter/10.1007/978-3-030-36119-8_14

Harper, B., & Sansone, C. (2018). Using computer screen recordings and think aloud protocols to study students' cognitive strategies while working online. SAGE Research Methods Cases.

Jonassen, D. H. (2000). Toward a design theory of problem solving. Educational Technology Research and Development, 48(4), 63-85.

Bond, M., Khosravi, H., De Laat, M., Bergdahl, N., Negrea, V., Oxley, E., Pham, P., Chong, S. W., & Siemens, G. (2024). A meta systematic review of artificial intelligence in higher education: A call for increased ethics, collaboration, and rigour. Educational Data Mining Conference Proceedings.

Romero, C., & Ventura, S. (2024). Educational data mining and learning analytics: An updated survey. arXiv preprint arXiv:2402.07956. https://arxiv.org/abs/2402.07956

Scientific Reports. (2024). Transfer learning model for anomalous event recognition in big video data. Nature Scientific Reports. https://www.nature.com/articles/s41598-024-78414-2

Xu, L., Chen, Y., & Wang, J. (2024). From recorded to AI-generated instructional videos: A comparison of learning performance and experience. British Journal of Educational Technology.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv. https://arxiv.org/abs/1704.04861