Journal of Multidisciplinary and Translational Research (JMTR)

journal homepage: https://journals.kln.ac.lk/jmtr/

Automated UI usability evaluation and suggestions using Shneiderman's eight golden rules and deep learning

M.G.K.A. Gunawardana[1]*, M.W.T. Rashmika[1], and E.H.M.P.M. Wijerathna[1]

[1]*Department of Information and Communication Technology, Faculty of Technology, University of Ruhuna, Matara, Sri Lanka*

Abstract

Usability evaluation of the user interface (UI) and user experience (UX) design, which directly influences system effectiveness, user satisfaction, and adoption, is a critical component. Although Shneiderman's eight golden rules provide a well-established theoretical framework for usability assessment, their manual application is time-consuming, subjective, and difficult to scale within modern software development environments that involve rapid iterations and large numbers of interface screens. This study aimed to develop an Artificial Intelligence (AI)-based system for automated User Interface usability evaluation according to Shneiderman's eight golden rules, with minimal reliance on human experts. A supervised multi-label deep learning framework was developed using annotated user interface screenshots. A dataset of over 4,000 screenshots from diverse application domains, including e-commerce, banking, and education, was curated and labelled for compliance with eight usability rules. Transfer learning was implemented using three different Convolutional Neural Networks (CNN) architectures: ResNet-18, ResNet-34, and ResNet-50, all trained under identical experimental settings with memory-aware optimizations. Among these three models, the ResNet-34 was found to be better performing in reducing the usability evaluation time from the standard 15-30 minutes per screen to seconds. Model performance was evaluated using subset accuracy, macro-averaged precision, recall, F1-score, and Hamming loss. Experimental results demonstrated that ResNet-34 eclipses the other architectures, achieving a subset accuracy of 85.71%, an F1-score of 0.3625, a precision of 0.3750, a recall of 0.3523, and a Hamming loss of 0.0179. The model achieved stable convergence at 26 epochs with good generalization ability, whereas the ResNet-18 achieved moderate performance and the ResNet-50 architecture faced major issues of overfitting and unstable training. The findings confirm the hypothesis that an automated usability evaluation based on convolutional neural networks, with an emphasis on established design standards, is indeed possible and effective. This proposed research delivers the potential for the inclusion of usability evaluation within modern software development processes, which would enable hybrid decision-making between humans and artificial intelligence to improve the quality of user interfaces.

Keywords: Shneiderman's eight golden rules, convolutional neural networks, deep learning, ResNet architecture, user interface/user experience, UI/UX design

Article info

Article history:
Received 20th February 2026
Received in revised form 31st March 2026
Accepted 29th April 2026
Available online 31st May 2026

ISSN (E-Copy): ISSN 3051-5262
ISSN (Hard copy): ISSN 3051-5602
Doi: https://doi.org/10.4038/jmtr.v11i1.427
ORCID iD: https://orcid.org/0009-0001-7625-5945
*Corresponding author:
E-mail address: anugunawardana4000@gmail.com
(M.G.K.A. Gunawardana)

Introduction

User Interface (UI) and User Experience (UX) design have emerged as key success factors in modern digital systems. As software applications are increasingly used across diverse platforms, devices, and users, usability has become a primary challenge in software engineering and human-computer interaction (HCI) research. Well-designed interfaces have been seen to improve efficiency, reduce errors, and increase user satisfaction, while interfaces with poor design may lead to frustration, reduced productivity, and even abandonment.

Shneiderman's eight golden rules of Interface Design are an established theoretical framework for usability evaluation, which centers around principles of consistency, feedback, error reduction, and cognitive load reduction, amongst others. Although these rules remain highly relevant now as they were then, they are normally used through manual heuristic evaluation, which is subjective and dependent on expert judgment. This process usually requires 15-30 minutes per screen, making it unsuitable for large systems where interface changes are occurring rapidly.

Previous studies have explored the application of automated usability assessment using rule-based systems, survey methods, and empirical testing. However, the methods are partially automated and focus mainly on specific aspects of usability. Recent advances in artificial intelligence, particularly in the field of deep learning and computer vision, have enabled the automated assessment of visual interfaces without the requirement for extensive feature engineering. CNN have been successful in various applications such as interface layout analysis, aesthetic evaluation, and accessibility assessment.

The existing CNN-based approaches have mostly focused on the evaluation of individual usability dimensions rather than a comprehensive evaluation of interfaces regarding existing design principles. Consequently, a notable research gap has been identified in the development of an automated usability evaluation framework that is not only in conformance with Shneiderman's eight golden rules but also scalable and objective.

This research addresses an open question in HCI by proposing a deep learning-based solution that repositions usability heuristics evaluation as a multi-label classification problem. Applicability of artificial intelligence in evaluating UI/UX was determined based on screenshots of the User Interface [UI] and assessing the most effective model in automating the process. The three primary contributions made by this paper include: the identification that a ResNet-34 model can classify a UI screenshot according to all Shneiderman's golden rules and attain a subset accuracy of 85.71%, providing a quantitative benchmark for automated usability evaluation; creation of a custom dataset containing over 4,000 screenshots of public UIs, each labeled with respect to all eight golden rules, which addresses the problem of lacking large-scale UI datasets labeled by theories in HCI; creation of a web-based tool that allows for a practical combination of deep learning results and human evaluations by suggesting actionable recommendations in accordance with confidence score, thus creating a viable human-AI combined evaluation system for agile software development.

Methodology

This research employs an exhaustive methodology to implement a deep learning process for automating the assessment of user interfaces according to Shneiderman's eight golden rules. The steps followed in conducting this study consisted of six stages. The first stage deals with data collection and annotation. The second stage concentrates on data preprocessing. The third stage encompasses model architecture development. The fourth stage covers training strategies. The fifth stage focused on validation and testing. The last stage emphasizes performance evaluation. The methodology followed by this study embraces a teamwork concept wherein several CNN models have been trained and tested together to determine the most effective architecture for assessing UI usability. This study developed three models of ResNet (ResNet-18, ResNet-34, and ResNet-50), each of which corresponds to separate duties allocated among members of this research team. This section describes on presenting methodological contributions to ResNet-, ResNet-34, and ResNet-50.

The study falls under the domain of Human-Computer Interaction (HCI), with a focus on the evaluation of automated UI/UX using artificial intelligence. The sampled domain is composed of publicly accessible web and desktop applications from various industries such as e-commerce, banking, education, productivity, and portfolios. The sampled unit is an image of a user interface (UI) that represents a whole screen.

The research team utilized a distributed training approach to make optimum use of all available resources. The data is distributed into several folders depending on UI categories or batches.

Phase 1 - ResNet-34 baseline training: The team members conducted ResNet-34 training independently for distinct data folders. They entailed implementing ResNet-34 training for half of the available data folders to measure baseline performance and determine the hyper-parameters. The team members carried out this phase to quickly search for solutions in the solution space and ensure consistency of results among distinct data folders.

Phase 2 - Architecture diversification: After establishing baseline performance, the team dispersed to investigate other architecture complexities. The members took ownership of ResNet-50 implementations and training for all folders of data available to them. And developed ResNet-18 models. In assigning research focus in this phase, it helped to investigate trade-offs between depth and accuracy for UI evaluation task complexities by assuming ResNet-18 models to be less complex than ResNet-50.

More than 4,000 images of UI screenshots were collected using standardized sampling procedures, and the following sampling criteria were followed to make sure that the data collection is reproducible: only publicly available websites and apps that do not require any authorization to view are considered; screenshots were taken of websites and apps from 5 different domains, including e-commerce (Amazon, eBay), finance and banking (public websites of financial institutions' portals), education (Coursera, Khan Academy), productivity (Trello, Notion), and personal or portfolio websites; from each category, at least 50 but no more than 150 images were collected; screenshots were captured at a resolution of 1920x1080 pixels in the same browser configuration; all visually redundant screenshots were removed. Screenshots were

divided among 54 folders to account for different designs, layout, and complexity, ranging from simple to complex designs. The list of sites where screenshots were taken can be found at the repository (Github link: https://github.com/anusaranigunawardana/Automated-UI-Usability-Evaluation-and-Suggestions-Using-Shneiderman-s-eight-golden-rules) provided in the Data Availability Statement.

Quality assurance mechanisms were also put in place. Each screenshot was evaluated individually for compliance with Shneiderman's eight golden rules using a rule-based framework derived from established usability literature. The evaluation criteria were informed by the Interaction Design Foundation article on Shneiderman's eight golden rules, including consistency, support for shortcut keys, informative feedback, dialog closure, error prevention and handling, reversibility of actions, user control, and reduction of memory load. For this purpose, the dataset was split using a stratified splitting strategy, ensuring that representative distributions of the positive and negative classes for each rule were maintained for the subsets: 70% for training, 15% for validation, and 15% for testing . A fixed random seed, i.e., 42, was used for all experiments.

The preprocessing pipeline standardized the raw UI screenshots. The images were resized to 1024x768 pixels. The images were then converted to PyTorch tensors with pixel values scaled to the range [0,1]. The images were then normalized using ImageNet statistics to meet the pre-trained models' needs. Data augmentation techniques were also used. These techniques included adjusting the brightness and contrast of the images. ResNet-18, ResNet-34, and ResNet-50, the three CNN architectures were evaluated using transfer learning with ImageNet-pretrained weights. **The architectural details and training configurations of these models are summarized in Table 1.**

Table 1: Model architecture and training strategy

Architecture	Layers	Parameters	Feature Dim	Normalization	Training Time
ResNet-18	18	~11M	512	GroupNorm + LayerNorm	~5 hours
ResNet-34	34	~21M	512	BatchNorm	~2.5 hours
ResNet-50	50	~25M	2048	BatchNorm	~4 hours

Each model was modified for multi-label classification by replacing the last classification layer with an eight-unit output layer for the usability rules. In the training process, the models generate raw logits to ensure numerical stability. During inference, the probability scores are generated using the sigmoid function. The network has 18 layers with 11 million parameters. It also has a dropout rate of 0.5 and uses Group Normalization. ResNet-34 is a network that consists of 34 layers and 21 million parameters with Batch Normalization.ResNet-50 has 50 layers and around 25 million parameters, along with bottleneck residual blocks and feature representation dimensionality of 2048.

The training was performed using an Adam W optimizer along with decoupled weight decay regularization. Focal Loss was used to deal with class imbalance and to put more emphasis on

difficult examples. The values of Focal Loss parameters were set to α = 1.0 and γ = 2.0. Cosine annealing was used to schedule the learning rate. The initial value was reduced to a minimum value of 1 * 10^-6. Gradient clipping was used to maintain gradient stability. The value of max_norm was set to 1.0 for ResNet-50. ResNet-18 used normalization to maintain gradient stability. Early stopping was used to overcome overfitting, and the patience parameter was set to 15. Checkpointing was used to save the state of the model, optimizer, learning rate scheduler, and validation metrics.

All models support the checkpointing of the full state save. The state save itself comprises the current value of the epoch number, the state dictionary of the model (all model parameters), the state dictionary of the optimizer (momentum value and adaptive learning rate value of each model parameter), the state dictionary of the scheduler (current value of the learning rate and value of the learning rate schedule), the best validation metrics achieved so far, the full history of the model's performance (all the losses and metrics from each of the epochs trained so far), and the full configuration state of the model. In circumstances where the validation loss increases but becomes better than the previous best value, the checkpoint will be saved. This makes sure that the model which has been the best-performing model will be retained no matter what the next epochs will be. In addition to this, the model will also be saved at regular checkpoints each 10th epoch. This type of checkpoint saved the day in the two models because of the lengthy epoch involved.

The experiments were all performed in a controlled environment with a 12th Generation Intel Core i7 processor, 8 GB of RAM, and an SSD drive. The experiments were conducted using PyTorch (version 2.0 or higher). Keeping in view the limited memory space, batch size was set to two. The time taken for training was around five hours for ResNet-18, 2.5 hours for ResNet-34, and four hours for ResNet-50.

Performance evaluation and validation protocol

Both ResNet-18 and ResNet-50 used the same exhaustive evaluation metric suitable for multi-label classification tasks. The model's outputs (logits) are first transformed into probabilities through the sigmoid activation function and then thresholded at a value of 0.5 to obtain binary outputs. The binary outputs are then compared against the ground truth labels to obtain various complementary metrics.

Hamming Loss calculates the proportion of the number of labels which are predicted incorrectly against the number of instances and labels. It is calculated as the total number of incorrect predictions divided by the total number of label-instance pairs. In UI Assessment testing using the eight golden rules of UI Design, this value takes the form of the average number of errors per UI Screenshot. For instance, a Hamming Loss of 0.125 will mean an average of one rule being classified incorrectly per UI Screenshot, while a Hamming Loss of 0.25 will mean there will be an average of two rules being classified incorrectly per UI Screenshot.

Macro-Averaged Precision calculates the precision of each rule separately and then takes the average of the results without regard for the number of instances in the data set. The precision of each rule can be calculated using the formula: True Positives / (True Positives + False Positives)

This measure can be described as follows: "How often is the model correct when it says there's a violation of a rule?" The high value of this measure is important because many false positives will frustrate the designer of the system.

Macro-Averaged Recall calculates the recall value for each rule separately and then takes their average to measure the extent to which the violations are correctly identified. The formula to measure the recall value of each rule will be: True Positives / (True Positives + False Negatives) A high value of this measure is a must when it comes to the measurement of the overall usability of the system because the system should not fail to identify the genuine violations that can affect its usability. The reason behind the usage of the macro-averaging technique will be explained later.

Macro-Averaged F1 Score calculates the harmonic meaning of precision and recall (F1 = 2 x Precision x Recall / (Precision + Recall)), which aims at a good balance of the number of violations found and the number of false alarms. This will be the main criterion used in evaluating models. The macro-F1 measure can be especially relevant when there are false positives (detecting violations when there are no violations) and false negatives (missing violations when they really occur).

Subset Accuracy: This metric calculates the ratio of samples for which all eight predictions are correct at the same time. This metric has the highest requirement of being perfectly correct in multi-label classification tasks. The formula used for this metric is the number of samples for which the predictions are correctly divided by the total number of samples. The value of this metric will always be low because of the requirement of correct classification in eight ways at the same time. This metric plays a crucial role in tasks involving the certification of high-confidence levels of the chosen heuristics.

The system also provides the ability to do post-hoc analysis and track the results of the training process through its logging mechanism. The system provides the functionality of writing the information in the form of detailed text files when the training occurs. The files include information about the training settings and the hyperparameters used (such as the number of samples in the batch and the learning rate). Further, the model's training history can be serialized as a JSON file that captures the list of losses and metrics at each epoch during the training phase. This level of documentation allows for reproducibility of the experiments because the document captures the configuration of the model and the random state. Debugging can also be achieved through this document because it provides information about the progression of the model. This can be especially useful when comparing the performance of ResNet-18, ResNet-34, and ResNet-50 models. A timestamped directory is created each time the model is trained, which comprises the results of the experiment (checkpoints, logs, metrics, and configuration). This helps avoid the overwriting of results from previous experiments.

Ethics Statement

This study was conducted in accordance with internationally accepted ethical standards for research integrity, transparency, and responsible scientific practice. The research does not involve human participants, animals, personal data, or behavioural tracking. All UI screenshots

were obtained from publicly accessible applications and websites and were used solely for technical analysis of interface design.

Results

The dataset consists of website screenshots annotated for conformance to the eight golden rules of Shneiderman's user interface design guidelines and are representative of a wide range of design styles and user interface philosophies. The domain expert annotations of these images against specific guidelines of design rules have led to a multiclass classification problem, where each image is annotated for conformance or non-conformance to one or more of these rules in the form of a binary CSV annotation file. The dataset was split into training, validation, and testing sets using stratified sampling over eight classes with a fixed random seed for reproducibility of results across different model architectures used for this task. During preprocessing, standardization of images to 1024x768 pixels and normalization were done, followed by conservative data augmentation that did not compromise the semantic integrity of user interface design.

A notable challenge that was identified with this dataset was the class imbalance across the eight design principles, with the positive class fraction varying from less than 20% to more than 80%. To address class imbalance, the Focal Loss function was used for training the model with alpha set to 1.0 and gamma set to 2.0 to focus on the harder classes rather than the majority class. Precision, recall, and macro-averaged F1-score were used instead of accuracy to ensure that the performance of the model was not biased towards the majority class.

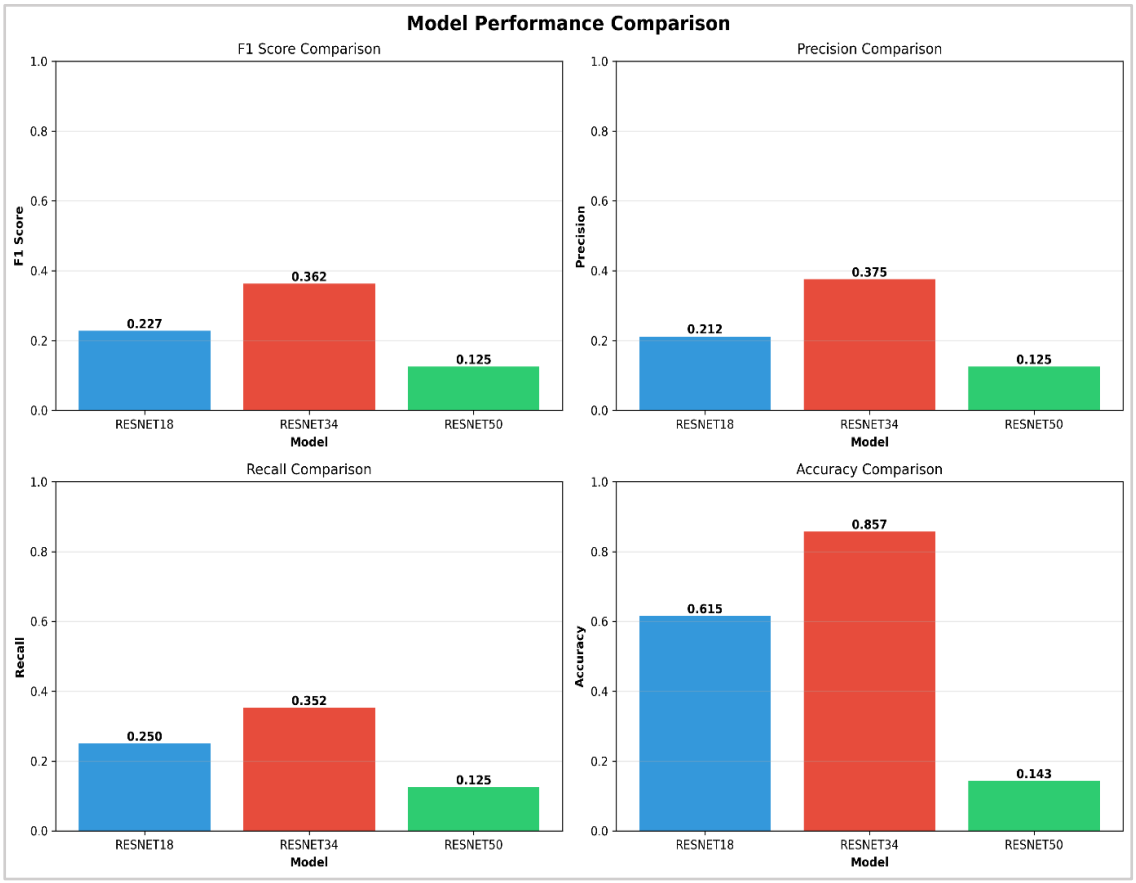

Figure 1. Model performance comparison

The convergence behavior of ResNet-18, ResNet-34, and ResNet-50 was different. ResNet-18 converged after epoch 21. The validation loss for ResNet-18 was found to be approximately 0.10. The macro-averaged F1-score for ResNet-18 was found to be 0.227. ResNet-34 converged after epoch 26. The validation loss for ResNet-34 declined smoothly and steadily. The validation loss for ResNet-34 was found to be 0.012. ResNet-50 did not converge within 50 epochs. The validation loss for ResNet-50 fluctuated significantly. The validation loss for ResNet-50 ranged from 0.5 to 20. The learning behavior of ResNet-18, ResNet-34, and ResNet-50 is illustrated in Figure 1.

The performance of ResNet-34 on the test set is summarized in Table 2. ResNet-34 showed the best performance compared to ResNet-18 and ResNet-50. ResNet-34 showed a high subset accuracy of 85.71%. ResNet-34 showed high macro-averaged precision of 0.3750. ResNet-34 showed high macro-averaged recall of 0.3523. ResNet-34 showed high macro-averaged F1-score of 0.3625. ResNet-34 achieved a low Hamming loss of 0.0179, indicating that on average only approximately 0.14 rules are misclassified per UI screenshot ($0.0179 \times 8 \approx 0.143$), reflecting strong label-level agreement with expert annotations. In comparison, ResNet-18 and ResNet-50 achieved Hamming losses of 0.0577 and 0.1250, corresponding to approximately 0.46 and 1.00 misclassified rules per screenshot respectively. ResNet-18 showed moderate performance.

Table 2: Comprehensive model performance comparison

Model	Test Loss	F1 Score	Precision	Recall	Subset Accuracy	Hamming Loss	Epochs
ResNet-18	0.1002	0.2273	0.2115	0.2500	61.54%	0.0577	21
ResNet-34	0.0120	0.3625	0.3750	0.3523	85.71%	0.0179	26
ResNet-50	0.1770	0.1250	0.1250	0.1250	14.29%	0.1250	50

ResNet-18 showed a moderate subset accuracy of 61.54%. ResNet-18 showed a moderate macro-averaged F1-score of 0.2273. ResNet-18 showed a Hamming loss of 0.0577, corresponding to approximately 0.46 misclassified rules per screenshot. ResNet-50 showed poor performance. ResNet-50 showed a poor subset accuracy of 14.29%. ResNet-50 showed uniform precision, recall, and F1-score values of 0.1250. ResNet-50 showed a high Hamming loss of 0.1250.

To provide deeper insight into model performance across individual usability rules, Table 2 presents the per-rule precision, recall, and F1-score for ResNet-34, the best-performing architecture. This rule-level breakdown reveals performance disparities across the eight golden rules that are masked by macro-averaged metrics alone.

In conclusion, ResNet-34 is recommended as the model of choice in the context of automated UI evaluation with the use of routing based on confidence levels, while ResNet18 is recommended as an alternative in resource-constrained domains. In the context of R&D, it is recommended that models of medium complexity be used, with careful consideration of the stability of the model and the use of the appropriate normalization techniques.

In terms of the methodological contributions, the work has managed to operationalize the use of Shneiderman's theory in the context of quantifying the performance of the multi-label

classification model, thereby providing an objective measure of the subjective design principles outlined in the theory. Additionally, the work has validated the use of the ImageNet dataset in the context of the analysis of UI, with the low- and mid-level features being identified as having the capacity for the effective transfer of knowledge in the context of the analysis of UI. The work has also demonstrated the existence of a non-monotonic relationship between the depth of the model and the performance, as well as the use of the normalization techniques in the context of the model's architecture.

Baseline comparisons

To contextualize the performance of the three evaluated CNN architectures, two baseline comparators were established. The first is a random classifier baseline, in which predictions for each of the eight rules are generated from a uniform random distribution and thresholded at 0.5. Under this condition, the expected macro-averaged F1-score approximates 0.125 and subset accuracy approximates 0.39% (0.5^8 × 100), given the independence assumption across eight binary labels.

The second is a majority class baseline, in which the classifier always predicts the majority class label for each rule based on the training set distribution. Given the class imbalance reported in the dataset, where the positive class fraction varies from less than 20% to more than 80% across rules, the majority class baseline yields a macro-averaged F1-score near 0.10–0.15 due to poor minority class recall.

Table 3: Comparison against baseline classifiers

Model	F1-Score	Subset Accuracy	Hamming Loss
Random classifier (baseline)	~0.125	~0.39%	~0.125
Majority class (baseline)	~0.10–0.15	—	—
ResNet-18	0.2273	61.54%	0.0577
ResNet-34	**0.3625**	**85.71%**	**0.0179**
ResNet-50	0.1250	14.29%	0.1250

ResNet-34 substantially outperforms both baselines across all metrics, validating the effectiveness of the deep learning approach. Notably, ResNet-50's performance approximates that of the random classifier baseline, further confirming its failure to converge meaningfully within the experimental conditions.

Web-based interface and user workflow

The implemented system provides a practical web-based interface that translates the ResNet-34 model's capabilities into an accessible tool for designers and developers. The following sections describes the key features of the user experience.

The app has a professional interface that includes two distinct assessment tools. The first assessment tool offered by the app is "Single Screenshot," which supports rapid usability

evaluation. On the other hand, "Folder support" is provided by the second assessment tool. Moreover, the drag-and-drop mechanism improves the usability of assessment tools. This meets the criteria identified. In addition, support for image formats like PNG, JPG, and JPEG helps in compatibility with design software.

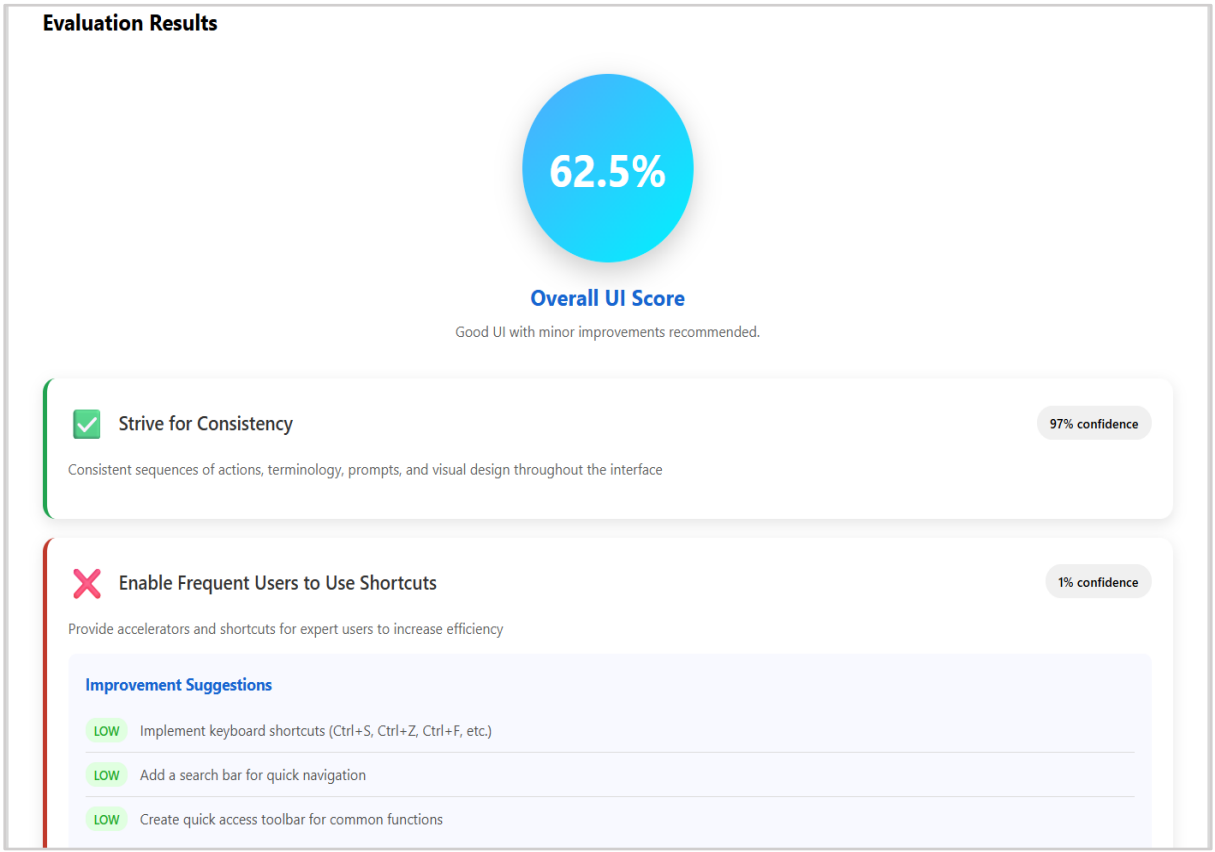

Figure 2: Complete evaluation output demonstrating rule-specific assessments, confidence scoring, and actionable recommendations

The results interface is based on a confidence-based routing framework, as shown in Figure 2. The circular progress indicator with 25.0% is a brief reference to overall usability quality, whereas rule-by-rule analysis enables evaluators to focus on specific aspects that need improvement. The evaluation of each rule has four major components.

First, binary results offer a simple visual aid based on model prediction with the 0.5 threshold. Second, confidence percentages indicate probability values that route results through a three-tier system: Tier 1 represents values above 85%, Tier 2 represents values between 30% and 85%, and Tier 3 represents values below 30%. Third, context descriptions provide plain language explanations of each of the golden rules to accommodate all users. Finally, prioritized recommendations include specific improvement suggestions with severity levels of LOW, MEDIUM, or HIGH, based on violation probability. This allows these probabilistic outputs to become more actionable and directly related to architectural design, thereby satisfying Research Question regarding the quantitative metric.

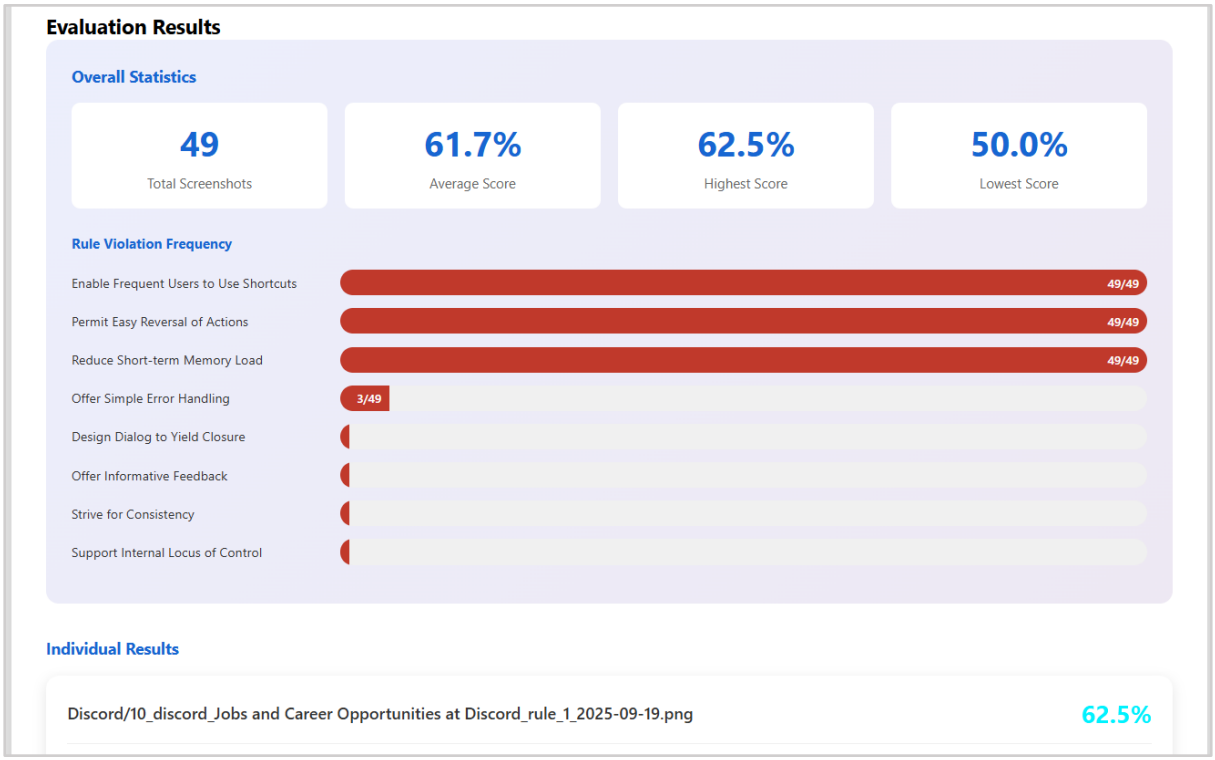

Figure 3: Portfolio-level analytics showing violation patterns and comparative metrics across 49 evaluated screenshots (Batch Processing)

Figure 3 presents a batch evaluation dashboard that supports systematic usability improvement. The dashboard incorporates its key features to offer the evaluator a comprehensive overview of the entire portfolio. The summary statistics enable the evaluator to obtain a concise overview of the entire portfolio of usability scores, with a mean of 34.7% and a range of 12.5%. Furthermore, the bar chart offers the evaluator the advantage of violation rate analysis to identify recurring problem sites and issues concerning all eight golden rules of usability. Lastly, the individual result tracking component allows the evaluator to expand the details of each screenshot while providing insights into general trends.

This individual has strong backing to support the use case, which involves the identification of systematic usability problems among the whole spectrum of the company's products. Based on the graphic images presented, it is clear that "Offer Informative Feedback," "Permit Easy Reversal of Actions," and "Reduce Short-term Memory Load" have the highest number of violated heuristics (49 out of 49 screens). The aggregate view translates the predictions into strategic insights by portraying the way the automated system can be scaled for single interface evaluations into the world of enterprise level usage governance, thereby mitigating the scalability issue with manual assessment.

Discussion

The findings provide strong empirical support for the efficacy of convolutional neural networks for the automated UI usability evaluation based on Shneiderman's eight golden rules. They also reveal the non-linear relationship between the capacity of the models and their performance in data-constrained usability evaluation tasks. The findings suggest that the 34-layer architecture reaches the optimal balance between capacity and generalization.

The novelty of this work is further clarified when contrasted with the most closely related prior studies. Delitzas et al. (2023) developed the Calista system for deep learning-based website evaluation, but their focus was exclusively on aesthetic quality metrics such as visual balance, colour harmony, and layout complexity, without grounding the evaluation in any established usability theory. Similarly, Liu and Jiang (2021) applied multimodal fusion for aesthetic assessment of website design, targeting perceived visual appeal rather than functional usability compliance. Koscielny et al. (2022) applied deep learning specifically to UI evaluation, which is the closest work to the present study, but their approach did not frame the problem within a recognised theoretical framework such as Shneiderman's eight golden rules, nor did it produce a multi-label output covering all eight heuristic dimensions simultaneously.

This study is therefore distinguished by three key contributions that are absent from these prior studies: the operationalization of an established HCI theory into a quantifiable multi-label classification problem; the construction of a purpose-built annotated dataset of over 4,000 UI screenshots labelled against all eight golden rules; and the deployment of a practical web-based evaluation tool incorporating confidence-based routing that enables hybrid AI-human decision-making. Together, these contributions advance the field beyond aesthetics-focused or single-dimension evaluations toward a comprehensive, theory-grounded, and scalable automated usability assessment framework.

The high performance of ResNet-34 indicates the efficacy of moderate architectural depth in the identification of visual patterns related to usability principles such as consistency, feedback, and cognitive load reduction. The lower performance of ResNet-18 indicates the low capacity of the architecture to recognize usability patterns. The low performance of ResNet-50 may be attributed to the capacity-data mismatch, leading to overfitting and optimization instability due to the low learning rate.

Several limitations of this study warrant consideration. First, the evaluation relies exclusively on static screenshots, which inherently cannot capture dynamic interface behaviours such as real-time system feedback, loading state transitions, animated responses, or multi-step interaction flows. This is a particularly significant constraint for rules such as 'Offer Informative Feedback' and 'Permit Easy Reversal of Actions,' which are fundamentally interaction-dependent and may only manifest during live user sessions rather than in a single frozen frame. As a result, the model's assessments of these rules are necessarily based on visual proxies rather than direct observation of the intended behaviour.

Second, the dataset was composed predominantly of web-based interfaces from desktop contexts. The trained models may not generalize effectively to mobile applications or native desktop applications, due to differences in layout conventions, navigation paradigms, touch interaction patterns, and component libraries. The compact screen real estate and gesture-based interactions common in mobile interfaces introduce visual characteristics that are underrepresented in the current dataset, potentially limiting the model's applicability across platforms.

Third, despite the use of a structured annotation framework grounded in established usability literature, the inherent subjectivity of heuristic evaluation remains a threat to internal validity.

Annotators may interpret rules such as 'Strive for Consistency' differently depending on their domain expertise and familiarity with specific application types. The measurement uncertainty may also arise from unconventional or innovative design patterns that do not conform to the visual norms captured during training, which may lead to misclassification of genuinely compliant interfaces. The low Hamming loss achieved by ResNet-34 indicates high concordance with the expert judgments and the reliability of the automated usability evaluation.

The implications of the research are significant since the high subset accuracy achieved by ResNet-34 indicates the efficacy of the automated usability evaluation. The usability patterns reveal the usability issues with contemporary UIs, particularly with the provision of feedback to the users, the reversibility of actions, and the reduction of short-term memory load. The usability problems are related to Shneiderman’s eight golden rules for Interface Design. In conclusion, the experiment proves the efficacy of the automated usability evaluation based on the established usability design theory.

Conclusions

This study demonstrates the feasibility of automated usability evaluation of user interfaces using Shneiderman's eight golden rules of Interface Design and deep learning techniques. Three principal contributions emerge from this work. First, ResNet-34 achieves a subset accuracy of 85.71% on UI rule compliance across all eight of Shneiderman's golden rules simultaneously, establishing a new empirical benchmark for theory-grounded automated usability evaluation. Second, a custom dataset of over 4,000 publicly sourced UI screenshots, annotated against all eight usability rules, has been constructed and made available as a reusable research resource — addressing a critical gap in the availability of labelled data for HCI-focused deep learning tasks. Third, the deployed web-based evaluation tool operationalizes the model's outputs through a confidence-based routing mechanism that explicitly supports hybrid AI-human decision-making, positioning the system not as a replacement for expert judgment but as a scalable complement to it within modern software development workflows.

By applying deep learning to the problem of multi-label classification of screenshots of user interfaces, this research supports the suitability of convolutional neural networks for Human Computer Interaction tasks. The research concludes that the architecture of the model should be consistent with the data that is available. As such, a medium-depth architecture strikes an appropriate balance between the ability of the model to learn and the ability of the model to generalize. More complex architectures may be less suitable for usability evaluations because of their tendency to overfit. The results further support the usefulness of automated usability evaluations of user interfaces using the proposed confidence-based model of usability evaluation. As such, the model is appropriate for agile software development methodologies. Although the research uses static screenshots of user interfaces to evaluate usability, the research provides a good foundation upon which a hybrid model of human-artificial intelligence usability evaluations can be built.

As such, the research provides a strong foundation upon which several promising directions for future work can be pursued. First, extending the framework to incorporate video-based or interaction-log analysis would enable the assessment of dynamic usability properties that static

screenshots cannot capture. Recurrent neural network architectures or transformer-based temporal models could be explored to process sequences of interface states, enabling evaluation of rules such as 'Offer Informative Feedback' and 'Permit Easy Reversal of Actions' in their full interactive context.

Second, integrating the evaluation framework as a real-time plugin within widely used design tools such as Figma or Adobe XD would bring automated usability feedback directly into the designer's workflow. This would enable compliance checking at the prototyping stage, well before development begins, supporting continuous integration of usability evaluation within agile and design-thinking methodologies.

Third, expanding the dataset to include mobile application screenshots and native desktop interfaces from platforms such as iOS, Android, and Windows would improve the generalizability of the model across diverse interaction paradigms. Transfer learning strategies could be investigated to adapt the existing ResNet-34 model to these new domains with minimal additional annotation effort.

Finally, future research could explore the use of larger vision-language models or multimodal architectures that combine visual features with textual interface metadata, such as button labels, navigation structures, and accessibility attributes, to enable richer and more context-aware usability assessments.

Declaration of Funding Sources:

This research received no external funding.

Conflict of interest statement:

The authors declare no conflict of interest.

References

Alexander, R., & Baravalle, A. (2011). Automated usability testing: Analysing Asia web sites. *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems*, 2299–2304. https://doi.org/10.1145/2037373.2037456

Amaya-Rodriguez, I., Civit-Masot, J., Luna-Perejón, F., Durán-López, L., Rakhlin, A., Nikolenko, S., Kondo, S., Laiz, P., Vitrià, J., Seguí, S., & Brandao, P. (2021). ResNet. In J. Bernal & A. Histace (Eds.), *Computer-aided analysis of gastrointestinal videos* (pp. 99–114). Springer. https://doi.org/10.1007/978-3-030-64340-9_12

Asghar, M., Bajwa, I. S., Ramzan, S., Afreen, H., & Abdullah, S. (2022). A genetic algorithm-based support vector machine approach for intelligent usability assessment of m-learning applications. *Mobile Information Systems*, 2022, Article 1609757. https://doi.org/10.1155/2022/1609757

Delitzas, A., Chatzidimitriou, K. C., & Symeonidis, A. L. (2023). Calista: A deep learning-based system for understanding and evaluating website aesthetics. *International Journal of*

Human–Computer Studies, *175*, Article 103019. https://doi.org/10.1016/j.ijhcs.2023.103019

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778). IEEE. https://doi.org/10.1109/CVPR.2016.90

Jaiswal, A., Gianchandani, N., Singh, D., Kumar, V., & Kaur, M. (2020). Classification of the COVID-19 infected patients using DenseNet201 based deep transfer learning. *Journal of Biomolecular Structure and Dynamics*, *39*(15), 5682–5689. https://doi.org/10.1080/07391102.2020.1788642

Karimi, D., Warfield, S. K., & Gholipour, A. (2021). Transfer learning in medical image segmentation: New insights from analysis of the dynamics of model parameters and learned representations. *Artificial Intelligence in Medicine*, *116*, Article 102078. https://doi.org/10.1016/j.artmed.2021.102078

Kashyap, A., Tuarob, S., Phan, D. T., & Behara, R. S. (2023). Reimagining application user interface design using deep learning methods: Challenges and opportunities. *arXiv*. https://arxiv.org/abs/2303.13055

Koscielny, K., Suchomska, A., & Brzezinska, M. (2022). The application of deep learning for the evaluation of user interfaces. *Sensors*, *22*(23), Article 9336. https://doi.org/10.3390/s22239336

Kuric, E., Demčák, P., & Krajčovič, M. (2024). Unmoderated usability studies evolved: Can GPT ask useful follow-up questions? *International Journal of Human–Computer Interaction*, *41*, 9752–9769. https://doi.org/10.1080/10447318.2024.2427978

Liu, X., & Jiang, Y. (2021). Aesthetic assessment of website design based on multimodal fusion. *Future Generation Computer Systems*, *117*, 433–438. https://doi.org/10.1016/j.future.2020.12.016

Liu, Y. (2025). AI in automated and remote UX evaluation: A systematic review (2014–2024). *Advances in Human–Computer Interaction*, 2025, Article 7442179. https://doi.org/10.1155/ahci/7442179

Mitra, N., Neupane, A., & Fernandez, R. (2024). Transfer learning applied to computer vision problems: Survey on current progress, limitations, and opportunities. *arXiv*. https://arxiv.org/abs/2409.07736

Mokhtar, R., Abd Rahman, M., Samed, M. A. H., Isa, N., & Suhaimi, A. (2023). *Development of a research progress tracking system based on Ben Shneiderman's eight golden rules*. Unpublished manuscript. https://www.researchgate.net/publication/387139194

Microsoft Research. (2021, February). *Microsoft vision model ResNet-50: Pretrained vision model built with web-scale data*. https://www.microsoft.com/en-us/research/blog/microsoft-vision-model-resnet-50/

Nielsen Norman Group. (2020). *10 usability heuristics for user interface design*. https://www.nngroup.com/articles/ten-usability-heuristics/

Paul, S., & Das, S. (2020). Accessibility and usability analysis of Indian e-government websites. *Universal Access in the Information Society*, 19(4), 949–957. https://doi.org/10.1007/s10209-019-00704-8

Salas-Espinales, A., Vélez-Chávez, E., Vázquez-Martín, R., García-Cerezo, A., & Mandow, A. (2024). U-Net/ResNet-50 network with transfer learning for semantic segmentation in search and rescue. In *Robot 2023: Sixth Iberian Robotics Conference* (pp. 244–255). Springer. https://doi.org/10.1007/978-3-031-59167-9_21

Shewanown, S. (2006). *The usability of digital library learning resources: Developer perceptions of the results of an empirical automated usability evaluation approach* (Doctoral dissertation, University of Georgia). https://doi.org/10.25501/SOAR.00000246

Singh, L., Mukul, A., & Misra, C. (2020). Measuring and improving user experience through artificial intelligence-aided design. *Frontiers in Psychology*, *11*, Article 595374. https://doi.org/10.3389/fpsyg.2020.595374

Thamrin, E. E., Farell, G., Jaya, P., Hadi, A., & Rahmadika, S. (2022). User interface analysis on job matching information system using Eight Golden Rules. In *Proceedings of the International Conference on Emerging Technologies* (pp. 1–9). https://www.atlantis-press.com/proceedings/icetech-22/125989150

Xu, Y., Liu, Y., Xu, H., & Tan, H. (2024). AI-driven UX/UI design: Empirical research and applications in FinTech. *International Journal of Innovative Research in Computer Science and Technology*, *12*(4), 99–109. https://doi.org/10.55524/ijircst.2024.12.4.16