Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

AI-Driven Mobile UI Pattern Recognition and Design Topic Mining on RICO: Semantic Clustering and Screenshot-Based Topic Classification

Jason Kuhn[1], Yushan Chen[1,2], Evelyn Chan[2]

[1]*Data Science, University of Pittsburgh, PA, USA*
[1,2]*Service Design, Savannah College of Art and Design, GA, USA*
[2]*Computer Engineering, Dartmouth College, NH, USA*
yushanchen1029@gmail.com

DOI: 10.69987/JACS.2024.40506

Keywords

mobile UI, design mining, topic modeling, RICO dataset, vision transformer

Abstract

Mobile UI ecosystems contain recurring layout patterns, interaction structures, and visual motifs that collectively form "design topics". This paper presents a data-driven pipeline that mines design topics from the RICO v0.1 semantic-annotation subset and then evaluates screenshot-based topic classification. Using 66,261 RICO screens (PNG screenshots paired with JSON view hierarchies containing semantic fields such as componentLabel, iconClass, text, bounds, and clickable), we extract a compact semantic feature vector per screen and apply MiniBatch K-Means (K=20) to obtain interpretable topic clusters. These clusters serve as pseudo-labels for downstream visual recognition. We compare three lightweight models that predict the mined topics from UI screenshots alone: (i) a small convolutional neural network (CNN), (ii) a compact vision transformer (ViT), and (iii) a lightweight vision–language model (LightVLM) trained with contrastive alignment between screenshots and semantic feature vectors. Experiments use a stratified subset of 4,782 screens (train/val/test = 3,000/594/1,188; 150/30/60 per topic) with deterministic seed 42. On the held-out test set, the ViT achieves the strongest overall performance (Accuracy = 0.345, Macro-F1 = 0.284, Macro-AUC = 0.820), outperforming the CNN (Accuracy = 0.222, Macro-F1 = 0.138, Macro-AUC = 0.764) and LightVLM (Accuracy = 0.243, Macro-F1 = 0.189, Macro-AUC = 0.782). We provide topic distribution analysis, clustering visualizations, confusion matrices, and embedding plots to characterize common failure modes. Finally, a semantic-only prototype baseline (Macro-F1 = 0.605, Macro-AUC = 0.945) clarifies how strongly the mined topics are grounded in view-hierarchy semantics.

1. Introduction

Mobile interfaces are produced on an internet scale: thousands of apps, frequent updates, and rapid shifts in visual and interaction conventions. This scale makes it difficult to manually track "what patterns are common" and "how design is trending" across the ecosystem. Data-driven design mining addresses this problem by automatically discovering recurring interface structures and motifs from large UI corpora, enabling applications such as template recommendation, style transfer, usability benchmarking, accessibility metadata generation, and automated QA. Work on design mining demonstrated that large-scale interface data reveals consistent patterns and powers downstream design tools that help designers reason about existing design spaces [2].

RICO is one of the most widely used public datasets for mobile UI research, combining high-resolution screenshots with view hierarchies collected from Android applications [1]. The view hierarchy captures a structured representation of UI elements and properties (bounds, resource-id, text, clickable), and therefore supports both pixel-based modeling and structure-aware approaches. RICO enables a wide range of tasks: pixel-to-code generation of UI layouts [4], creation of accessibility metadata directly from screenshots [5], learning reusable screen embeddings for retrieval and

clustering [6], and multimodal pretraining that jointly models images and UI structure [7]. In parallel, the Enrico dataset introduced topic modeling for mobile UIs by curating screenshots and topic annotations aimed at discovering higher-level interface topics for analysis and benchmarking [3]. Together, these datasets motivate a general strategy: mine structure from large UI repositories and then build recognition models that generalize across apps.

A key open question is what "topic" should mean for a UI. Topics can be defined by purpose (e.g., login vs. checkout), by interaction pattern (e.g., feed vs. settings), by semantic composition (e.g., image-first vs. text-first), or by fine-grained style (e.g., typography and spacing). Different definitions support different applications. Purpose-oriented topics are useful for flow reconstruction and product analytics; semantic-composition topics are useful for design system auditing and component inventory estimation; style topics are useful for branding and theming analysis. In practice, large-scale datasets typically provide structure and pixels but not purpose labels. This motivates unsupervised topic discovery from available signals, with explicit evaluation of how topics relate to pixels.

From a product and design-engineering perspective, topic mining enables concrete workflows [25-35]. A design system team can estimate how frequently each archetype occurs and prioritize which templates to refine first; a QA pipeline can flag screens whose predicted topic is inconsistent with an app's usual distribution (potential regressions); and competitive analysis can compare topic proportions across app categories. Because topics are derived from structural semantics, they can also be used to estimate component inventory (e.g., how many list-based screens exist) without manual tagging. In addition, topic definitions can serve as scaffolding for human annotation: designers can name and curate a small set of topic prototypes and then propagate labels to large corpora via nearest-prototype assignment.

This paper therefore addresses two concrete questions: (1) Can we mine coherent and interpretable design topics at scale using only semantic annotations from RICO's view hierarchies? (2) Once topics are mined, how accurately can lightweight models predict the topic of a screen from pixels alone, and which architecture provides the best trade-off between accuracy and efficiency?

To answer these questions, we present an end-to-end pipeline (Fig. 1) that first mines topics via unsupervised clustering of semantic annotations and then trains and evaluates models that predict those topics from screenshots. The topic mining stage uses per-screen semantic composition statistics (componentLabel frequencies) augmented with interaction and content meta features (clickable ratio, text volume). We choose MiniBatch K-Means [17] because it is scalable, deterministic under fixed seeds, and produces centroids that are directly interpretable as "average topic signatures." In addition, K-Means clustering aligns well with our evaluation goal: topics are represented by prototype centroids, and downstream visual models can be analyzed against these prototypes.

For screenshot-based recognition, we compare three modeling paradigms. First, we train a compact CNN classifier, which provides local feature extraction and remains a standard baseline for UI vision tasks [8]. Second, we train a small ViT classifier, which represents a UI as patch tokens and uses self-attention to model long-range layout regularities. This inductive bias is well aligned with UI layouts, where global alignment and repeated structural motifs define many screen patterns [9], [10]. Third, we train a lightweight vision–language alignment model inspired by contrastive multimodal learning [11], [12]. Our LightVLM aligns screenshots with semantic feature vectors using InfoNCE contrastive loss and performs CLIP-style prototype classification, enabling a direct bridge between semantic topic definitions and visual inference.

There is an important evaluation nuance: because we mine topics from semantic annotations, the resulting labels do not reflect an independent "ground truth" taxonomy. Instead, they encode a particular perspective on UIs—semantic composition and interaction density. This is deliberate: it yields interpretable and scalable topics. However, it also implies that a purely semantic model can be a strong predictor of the pseudo-labels. We treat this as a feature of the benchmark, and we explicitly measure a semantic-only upper bound in the Results section to quantify the information gap between semantics and pixels.

The pipeline is designed to be reproducible and self-contained: it uses only the information provided in the RICO semantic-annotation subset, avoids external pretraining, and reports explicit splits, hyperparameters, and deterministic seeds. The contributions are:
• A scalable topic mining method for RICO semantic annotations that produces interpretable K=20 design topics and descriptive statistics for each topic.
• A reproducible screenshot-based topic classification benchmark derived from mined topics, including stratified sampling and evaluation with Accuracy, Macro-F1, and Macro-AUC [20].
• A comparative study of CNN, ViT, and lightweight vision–language alignment for UI topic recognition, including confusion matrices and embedding visualizations.
• An ablation that quantifies how much of the mined topic structure is explained by semantics alone versus pixels, clarifying strengths and limitations of pseudo-topic supervision.

The rest of the paper follows the required structure. The Method section details dataset processing, semantic feature extraction [36-42], clustering, model designs, and training procedures. Results and discussion report topic statistics, classification performance, and qualitative analyses. Limitations discuss pseudo-label bias and dataset constraints. Conclusion summarizes findings and practical implications.

Method

A. Dataset and semantic-annotation subset. We use the RICO v0.1 semantic-annotation subset, which contains 66,261 Android screens. Each screen consists of (i) a 1440×2560 PNG screenshot and (ii) a JSON view hierarchy. The semantic-annotation subset enriches nodes with higher-level semantic fields, notably componentLabel (a coarse semantic UI type) and iconClass (icon category). Nodes also include text (visible UI strings), bounds (pixel coordinates), and boolean attributes such as clickable. We treat each screen as an independent sample for both topic mining and classification. Table I summarizes dataset properties and experimental settings.

B. Semantic feature extraction. The view hierarchy provides a natural basis for a "bag-of-components" representation analogous to bag-of-words in document modeling. For each screen, we enumerate all nodes with non-empty componentLabel and count occurrences of each label. The semantic vocabulary contains 25 unique labels in this subset (Table II). Let $c \in \mathbb{R}^{25}$ be the vector of raw counts, where c_j is the number of elements with label j. To remove scale dependence on hierarchy size, we normalize to a frequency vector $f = c / (\sum_j c_j)$. This yields a composition representation where, for example, f_Text is the fraction of labeled nodes that are Text.

The component label distribution is strongly skewed (Table II). The top five labels—Text, Image, Icon, List Item, Text Button—account for 87.65% of all labeled elements across the dataset. This concentration indicates that a small set of primitives dominates Android UIs, while the remaining labels form a long tail of specialized components (e.g., Advertisement, Modal, Drawer). This skew motivates both topic mining and classification: topics are mixtures over common primitives, while rare primitives act as discriminative cues for specialized topics.

In addition to semantic composition, we compute meta features that summarize interaction density and content volume. Specifically:

(1) total_nodes counts all nodes in the hierarchy; larger hierarchies correspond to denser screens (e.g., settings).
(2) clickable_ratio is the fraction of nodes with clickable=true; interaction-heavy screens have higher ratios than passive content views.

(3) text_chars sums character counts across all node text fields; this approximates textual density and is sensitive to forms and lists.

(4) unique_icon_classes counts distinct icon categories; icon-heavy toolbars and navigation bars increase this value.
(5) total_label_elems counts nodes with a defined componentLabel (the same denominator used for f). We apply log1p to count-like features (total_nodes, text_chars, unique_icon_classes, total_label_elems) to reduce heavy-tailed effects and keep clickable_ratio in linear scale. Finally, we standardize all 30 features with z-score normalization prior to clustering. This feature design is intentionally simple and auditable: given a cluster centroid, each dimension is interpretable in terms of a UI property.

C. Topic mining via clustering. We define "design topics" operationally as clusters of screens that share similar semantic composition and meta characteristics. Given standardized feature vectors, we minimize within-cluster squared Euclidean distance with K-Means [17]. We use MiniBatch K-Means for scalability, with batch_size=2048 and deterministic random_state=42. We set K=20 to balance interpretability and diversity and to align with a common granularity used in mobile UI topic analyses [3]. The clustering output assigns each screen a topic ID $t \in \{0,\ldots,19\}$.

We considered alternative topic discovery strategies. Probabilistic topic models such as LDA are natural for bag-of-words data, but they require iterative inference and do not directly incorporate continuous meta features. Hierarchical clustering provides dendrogram structures but scales less efficiently and does not yield simple centroid prototypes. Gaussian mixture models capture elliptical clusters but can be unstable in high dimensions. In contrast, MiniBatch K-Means provides a fast, stable, and interpretable baseline that produces explicit centroids, supports incremental updates as new screens are collected, and fits naturally with prototype-based analyses.

To interpret topics, we compute topic-level summaries: prevalence (number of screens), dominant semantic labels (top componentLabel frequencies), and average meta values (clickable ratio and text volume). We report these in Table III and visualize the topic distribution (Fig. 2) and PCA structure (Fig. 3). Because K-Means produces explicit centroids, each topic is represented by a prototype semantic signature that reflects the average screen in the topic.

D. Screenshot-based topic classification task. After topic mining, we define a supervised recognition

problem: given a screenshot image x only, predict the mined topic ID t. This proxy task evaluates how much of semantic topic structure is visible from pixels. It also reflects practical constraints: in competitive analysis or legacy UI catalogs, view hierarchies are not always accessible. Efficient topic prediction from images therefore enables practical design mining at scale.

E. Data split, balancing, and preprocessing. The full dataset is imbalanced across topics (Fig. 2). To measure per-topic recognition difficulty without prevalence confounds, we build a balanced subset via stratified sampling: 150 screens/topic for training, 30/topic for validation, and 60/topic for testing. This yields 3,000/594/1,188 samples and guarantees that each topic is evaluated with equal support (Table IV). We fix random seed 42 for deterministic splits.

Screenshots are downsampled to 48×48 RGB for computational efficiency and reproducibility on CPU. Although low resolution loses fine typographic details, it retains global layout structure and major visual blocks (e.g., large images, lists, navigation bars). Pixels are normalized to [-1,1] using (p−0.5)/0.5.

F. Models.

F.1 CNN classifier. The CNN baseline uses three convolution blocks with max pooling and global average pooling (GAP), followed by a 64-dimensional MLP projection and a linear classification head. This architecture encodes local patterns and aggregates them through pooling [8]. The CNN's limited depth keeps parameter count small and enables fast CPU inference.

F.2 ViT classifier. The ViT model partitions the 48×48 image into non-overlapping 8×8 patches (36 patch tokens). Each token is embedded into 128 dimensions and processed by a transformer encoder of depth 2 with 4 attention heads. A [CLS] token summarizes the sequence and is mapped to 20 topics by a linear head. Self-attention explicitly models global relationships between distant regions, which is central for layout-dominant patterns [9], [10].

F.3 LightVLM (contrastive vision–semantic alignment). LightVLM combines an image encoder (CNN producing 128D embeddings) and a semantic encoder (MLP mapping 25D semantic vectors to 128D). Training uses InfoNCE contrastive loss on matched (image, semantic-vector) pairs in each batch: $L = \frac{1}{2}[CE(S/\tau, I) + CE(S^T/\tau, I)]$, where S is the cosine similarity matrix between normalized image and semantic embeddings, τ is temperature (0.07), CE is cross-entropy, and I is the identity matching [11], [12]. At inference, we perform prototype-based classification: for each topic k, we compute the mean semantic vector of training screens in topic k, encode it to obtain a topic prototype embedding, and classify an image by softmax over cosine similarity to topic prototypes. This yields an explainable mapping from pixels to semantic prototypes.

G. Training protocol, implementation, and metrics. CNN and ViT models are trained with Adam [14] for 2 epochs using the hyperparameters in Table V. LightVLM is trained for 2 epochs with Adam and τ=0.07. We select checkpoints using validation Macro-F1. We implement clustering with scikit-learn [15] and learning models with PyTorch, and we generate figures with matplotlib [16]. Evaluation on the held-out test set reports Accuracy, Macro-F1, and Macro-AUC computed with one-vs-rest ROC AUC [20]. Macro-F1 is emphasized because the benchmark is balanced and because it penalizes "topic collapse" where a model predicts only a few visually dominant classes.

Figure 1. Overview of design topic mining and UI topic classification pipeline

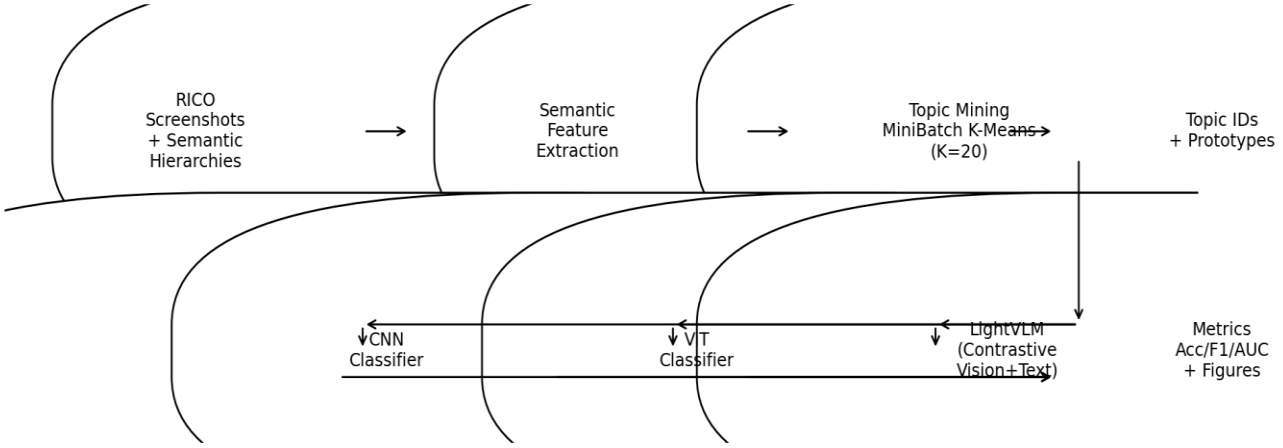

Figure 1. Overview of the proposed pipeline for topic mining and screenshot-based topic classification.

Table I. Dataset and experimental setup summary.

Item	Value
Dataset subset	RICO v0.1 (screenshots + hierarchies with semantic annotations)
Total UI screens	66,261
Screenshot format	PNG (1440×2560)
Hierarchy format	JSON view hierarchy with semantic fields (componentLabel, iconClass, text, bounds, clickable)
Unique componentLabel types	25
Topic mining method	MiniBatch K-Means (K=20) on semantic + meta features
Classification subset (stratified)	4,782 screens
Train / Val / Test	3,000 / 594 / 1,188
Random seed	42

Table II. Global distribution of componentLabel types across 66,261 screens.

ComponentLabel	Count	Percent (%)
Text	455075	34.91
Image	217511	16.69
Icon	179016	13.73
List Item	152165	11.67
Text Button	138834	10.65
Toolbar	34854	2.67
Web View	30788	2.36
Input	21408	1.64
Card	16901	1.30
Advertisement	13338	1.02
Background Image	6785	0.52
Drawer	6642	0.51
Radio Button	5420	0.42
Checkbox	4243	0.33
Multi-Tab	4185	0.32
Pager Indicator	4144	0.32
Modal	3959	0.30

On/Off Switch	2103	0.16
Slider	2016	0.15
Map View	1512	0.12
Button Bar	721	0.06
Video	562	0.04
Bottom Navigation	524	0.04
Number Stepper	427	0.03
Date Picker	291	0.02

Table III. Mined topic signatures (K=20): prevalence and dominant semantic components.

Topic	**Size**	**Top-3 labels**	**Mean clickable_ratio**	**Mean text_chars**
T0	4328	Image, Text, Web View	0.387	67.000
T1	5632	Text, Image, Text Button	0.170	327.700
T10	241	Icon, Image, Bottom Navigation	0.384	9.400
T11	464	Icon, Image, Text	0.427	71.000
T12	3068	Icon, Toolbar, Text	0.452	59.700
T13	1033	Text, Text Button, Modal	0.289	1271.300
T14	2936	Text, Card, Image	0.309	359.800
T15	871	Video, Text, Text Button	0.035	3.400
T16	506	Background Image, Text Button, Image	0.291	75.900
T17	6299	Text Button, Text, Image	0.559	219.800
T18	3322	Text, Toolbar, Icon	0.290	214.500
T19	3887	Icon, Text, Image	0.515	84.700
T2	14581	Text, Image, Icon	0.231	300.900

T3	4390	Text, Input, Text Button	0.524	141.500
T4	3381	Text, Icon, Text Button	0.280	133.900
T5	770	Text, List Item, Checkbox	0.098	272.800
T6	545	Text Button, Text, Button Bar	0.376	666.300
T7	3229	Web View, Icon, Advertisement	0.454	0.300
T8	1455	Text, Pager Indicator, Text Button	0.219	207.400
T9	5323	Text, List Item, Image	0.125	218.300

Table IV. Stratified split counts per topic for screenshot-based classification.

Topic	cluster_size	train	val	test
T0	4328	150	30	60
T1	5632	150	30	60
T2	14581	150	30	60
T3	4390	150	30	60
T4	3381	150	30	60
T5	770	150	30	60
T6	545	150	30	60
T7	3229	150	30	60
T8	1455	150	30	60
T9	5323	150	30	60
T10	241	150	24	48
T11	464	150	30	60
T12	3068	150	30	60
T13	1033	150	30	60
T14	2936	150	30	60
T15	871	150	30	60
T16	506	150	30	60
T17	6299	150	30	60

T18	3322	150	30	60
T19	3887	150	30	60

Table V. Model architectures and training hyperparameters for topic classification.

Model	**Input**	**Backbone**	**Params**	**Optimizer**	**LR**	**Epochs**	**Batch**
CNN	48×48 RGB	3 conv blocks + GAP + 64D MLP	29044	Adam	0.0010	2	128
ViT	48×48 RGB	ViT-Tiny (patch 8, dim 128, depth 2, heads 4)	297364	Adam	0.0002	2	128
LightVLM	48×48 RGB + 25D semantic vector	CNN image encoder + MLP text encoder (InfoNCE, τ=0.07)	41888	Adam	0.0010	2	256

Results and discussion

A. Topic mining outcomes and interpretability. The mined topic distribution (Fig. 2) is imbalanced but still diverse. The normalized Shannon entropy of the topic distribution is 0.872 (1.0 indicates perfectly uniform), and the top-5 topics cover 54.7% of all screens while the top-10 topics cover 82.1%. This indicates that a relatively small set of archetypes accounts for most screens, and a long tail of specialized designs remains.

The component distribution in Table II provides an additional interpretation lens. Because 87.65% of labeled elements belong to Text, Image, Icon, List Item, Text Button, many screens are describable as coarse mixtures over these primitives. Topic mining therefore discovers different "recipes" over a small set of common components, with rarer labels acting as topic-specific identifiers (e.g., Drawer for navigation-heavy patterns, Modal for dialog patterns, Advertisement for ad-dense layouts). This mixture-based view is consistent with how design systems are built: a limited component palette generates a wide variety of screens.

Table III reveals that each topic has a distinctive semantic fingerprint. Several large topics exhibit high Image frequency combined with moderate Text and Icon, consistent with media-centric browsing screens. Other topics have high List Item and Text, consistent with dense lists and settings. Still others have elevated Text Button and Button, consistent with action-heavy dialogs and forms. These signatures provide a practical interpretation layer: the centroid of each cluster is a semantic prototype that indicates which component primitives dominate.

Fig. 3 visualizes a random sample of 6,000 screens in a 2D PCA projection of the semantic feature space. The visualization shows partially separated clusters, demonstrating that the mined topics correspond to consistent semantic compositions. Overlap is expected because screens combine patterns; the clustering therefore yields a usable taxonomy of prototypes rather than a strict partition.

To quantify clustering structure, Table X compares silhouette scores [18] for K=20 on a 2,000-screen sample using two feature sets. Component frequencies alone yield higher silhouette than the full feature set that includes interaction/content meta features. This indicates that semantic composition is the dominant signal for topic separability, while meta features add

nuance but also blur boundaries for mixed screens. We keep meta features in the main pipeline because they improve interpretability and because the downstream recognition models operate on mixed cases regardless.

B. Topic-level case studies. Examining individual topics connects semantic signatures to recognizable UI archetypes. T2 (the largest topic) is dominated by Image and Text, with relatively low clickable ratio; these screens resemble content consumption pages where users scroll through media and captions. T17 is also large but has higher Icon and Image proportions, aligning with navigation-rich screens (toolbars and icon rows) and visually branded layouts. In contrast, a list-centric topic such as T10 shows high List Item and Text and tends to have higher text character counts; these screens match settings pages and feeds with repeated rows. Dialog/form topics exhibit higher Text Button and Button frequencies and increased clickable ratios, reflecting interaction-centric flows. These topic prototypes provide an actionable "component-first" view of the design space, which complements purpose-driven categorizations.

C. Screenshot-based classification performance. Table VI and Fig. 5 summarize test-set performance on balanced topic support (60 screens/topic). The ViT model achieves the strongest overall results (Accuracy = 0.345, Macro-F1 = 0.284, Macro-AUC = 0.820). The CNN baseline is weaker (Accuracy = 0.222, Macro-F1 = 0.138) but still yields a solid AUC (0.764), indicating that it ranks the correct topic relatively high in the probability distribution even when top-1 decisions are incorrect. LightVLM improves over CNN in Macro-F1 (0.189 vs. 0.138) and Macro-AUC (0.782 vs. 0.764), confirming that contrastive alignment transfers semantic structure into the visual embedding space. LightVLM remains below ViT, indicating that explicit global layout modeling is the key factor for topic recognition under low-resolution constraints.

These outcomes reflect the nature of UI topics. Many topics are defined by global layout structure (repeated list rows, full-width media blocks, persistent navigation), which is directly captured by patch-level self-attention in ViT [9], [10]. The CNN infers global structure indirectly through pooling. LightVLM introduces an additional constraint: the image embedding aligns with a semantic embedding, which encourages semantic consistency but regularizes away visually discriminative cues that do not correlate with component frequencies.

D. Confusion analysis, per-topic difficulty, and topic diversity. Fig. 4 presents the normalized confusion matrix for ViT. Confusions concentrate among semantically adjacent topics, especially those dominated by similar primitives such as Text + List Item or Image + Icon. Table VII provides per-topic F1 values and shows large variance. The easiest topics for ViT include T15 and T16 (F1 $\approx$ 0.687 and 0.614), which correspond to topics with distinctive global visual structure. The hardest topics include T19, T0, and T9 (F1 near zero), indicating that these topics encode subtle semantic differences that are not recoverable from downsampled pixels.

Topic size is negatively correlated with ViT per-topic F1 (Pearson r = -0.426). Larger topics contain more diverse screens, which makes them harder to model as a single visual class. This suggests a refinement strategy: if a topic becomes too broad, splitting it (increasing K or using hierarchical clustering) improves both interpretability and recognition. Conversely, if two topics are consistently confused and have highly similar semantic signatures, merging them creates a more reliable taxonomy.

A closer look at the largest off-diagonal confusions highlights this point. On the balanced test set, the strongest confusions include T12→T7 (25/60); T2→T14 (20/60); T10→T16 (20/60); T6→T10 (17/60); T9→T5 (16/60). For example, T12→T7 indicates that screens labeled as T12 are frequently predicted as T7; Table III shows that both topics involve icon-heavy layouts and toolbars, and at 48×48 resolution their global silhouettes are similar. Similarly, T2→T14 reflects confusion between two text+image-dominated topics where differences lie in subtle card/container structure (Card vs. Icon balance) that is hard to recover after aggressive downsampling. Such confusion analysis is actionable: it identifies topic boundaries that are visually brittle and motivates specific interventions (higher resolution, longer training, or topic merging) that directly increase reliability.

Macro-AUC provides an additional lens on difficulty. Even when F1 is low, a model achieves moderate AUC by ranking the correct topic relatively high. This matters for workflows where the top-k topics are presented to analysts or used for retrieval.

E. Representation visualization. Fig. 6 plots PCA projections of ViT [CLS] embeddings on the test set. Topics form visually separable regions, indicating that the model learns a topic-discriminative embedding space even with a short training schedule. Embedding visualization supports taxonomy auditing: if two topics are consistently entangled, they represent closely related archetypes in this embedding projection; merging them yields a coarser taxonomy. Conversely, if a topic forms multiple disjoint clusters, it is broad and benefits from splitting.

F. Efficiency and deployment considerations. Table VIII reports parameter counts and average CPU inference time. The CNN has 29,044 parameters and runs at 3.255 ms/screen, while the ViT has 297,364 parameters and runs at 2.469 ms/screen under batch

inference. The LightVLM image encoder is similarly lightweight (31,904 parameters) and runs at 3.505 ms/screen. These results support deployment in large-scale crawling and design analytics pipelines. In practice, models are selected by operational constraints: ViT offers the best accuracy; CNN offers a minimal baseline; LightVLM provides an alignment-friendly embedding that is useful for retrieval and semantic explanation.

G. Ablation: semantics versus pixels. Table IX highlights a key property of pseudo-topic supervision. Because topics are mined from semantic annotations, a semantic-only prototype classifier achieves strong performance without pixels (Accuracy = 0.608, Macro-F1 = 0.605, Macro-AUC = 0.945). This serves as an upper bound for models that attempt to infer semantic composition from images. In contrast, a random (untrained) LightVLM yields near-chance performance, confirming that the alignment objective is necessary. The contrastive LightVLM improves substantially over random, but still leaves a gap to the semantic upper bound; this gap quantifies the information loss incurred when inferring semantic composition from pixels. In other words, the semantic encoder defines the topic space precisely, while the image encoder approximates it.

H. Practical usage: dashboarding, retrieval, and semi-automatic labeling. Even with moderate top-1 accuracy, topic predictions are useful when aggregated. In a design analytics dashboard, per-screen topic probabilities aggregate into per-app or per-category topic histograms. Such aggregated statistics are robust to occasional misclassifications and allow teams to monitor shifts in design composition. Topic prototypes also enable retrieval: by selecting a topic or a prototype screen, an analyst retrieves visually similar screens in embedding space, similar in spirit to prior UI retrieval systems [6], [7]. In addition, topic mining bootstraps labeling: designers name and curate a small set of topic exemplars, and the model propagates these labels to large corpora to support downstream supervised tasks.

I. Relation to UI representation learning. Our models are trained from scratch on a small balanced subset, but the task itself serves as a pretraining objective for UI representations. Screen embedding work such as Screen2Vec [6] and multimodal UI models such as UIBert [7] show that reusable representations improve multiple downstream tasks. Topic prediction provides a structured supervision signal derived from semantics, and contrastive LightVLM training aligns pixels with structural prototypes. Training on the full 66k screens (using pseudo-topics as weak labels) yields stronger embeddings that transfer to tasks such as accessibility inference [5] and UI code generation [4].

J. Qualitative topic examples and design interpretation. Fig. 7 shows representative screenshots for four topics, each with two examples. Even without reading fine text, the topics show differences in information density, media dominance, and navigation structure (image-first pages vs. list-first pages). Qualitative inspection aligns with Table III and validates that mined topics correspond to recognizable UI archetypes.

K. Explainability and human-in-the-loop topic refinement. A practical topic mining system must support inspection and refinement by designers and analysts. The proposed pipeline provides two complementary explanation channels without additional modeling. First, each topic is associated with an explicit semantic signature (Table III) derived from cluster centroids and per-topic averages. When a screen is assigned to a topic, the signature immediately communicates "what component mixture defines this topic" (e.g., Image-heavy vs. List Item-heavy, or interaction-heavy via clickable ratio). This enables a lightweight, auditable interpretation that is aligned with how design systems are discussed in practice: designers reason about screens through component primitives and interaction affordances. Second, the screenshot classifiers output probability distributions over topics. In a dashboarding setting, analysts can treat the top-k topics as a shortlist rather than a single definitive label, which is especially useful for mixed screens that blend multiple archetypes (e.g., a media feed with a navigation drawer). Because Macro-AUC is relatively high even when Macro-F1 is modest, the probability ranking contains useful information for such shortlist workflows.

LightVLM provides an additional interpretability advantage: classification is performed by similarity to topic prototypes encoded from semantic vectors. This makes it possible to surface "nearest topic prototypes" as explanations and to retrieve representative training screens for the predicted topic. In other words, the model naturally supports example-based explanations ("this screen looks like these prototype screens") alongside semantic explanations ("this topic is defined by these component proportions"). Such example-based explanations are known to be effective for human auditing of machine learning systems because they connect predictions to concrete instances rather than abstract labels.

Human-in-the-loop refinement can then be guided by the quantitative diagnostics already reported. Confusion matrices (Fig. 4) identify topic pairs that are visually difficult to separate; embedding plots (Fig. 6) reveal whether a topic is internally coherent or multi-modal; and Table X indicates whether features beyond component frequencies improve separability. In a deployment cycle, an analyst can (i) merge consistently entangled topics into a coarser but more reliable taxonomy, (ii) split overly broad topics that contain multiple submodes, or (iii) enrich the feature

representation (e.g., include richer text features) for topics that require semantic nuance. This iterative loop is a realistic path to operationalizing topic mining: the initial unsupervised taxonomy provides coverage and interpretability, and subsequent human refinement improves semantic validity for the target application domain.

L. Downstream integration scenarios. Topic mining is most valuable when integrated into downstream analytics and tooling. One direct use is component inventory estimation: by combining predicted topics with Table II component distributions, a team can estimate how frequently specific primitives (e.g., lists, buttons, toolbars) occur in practice and prioritize which components require the most robust theming, accessibility support, and testing. Another use is regression detection: if an app's topic histogram shifts abruptly after an update, screens can be automatically surfaced for review, similar to anomaly detection on design telemetry. Topic predictions also support dataset curation. For example, when training a UI component detector, a balanced training set can be assembled by sampling across topics to avoid over-representation of feed-like screens. Finally, topic-conditioned retrieval can accelerate competitive analysis: analysts can query "show me image-first screens" or "show me dialog-heavy screens" across a large crawl and then drill down into representative examples. These scenarios do not require perfect top-1 accuracy; they primarily require that topic predictions correlate with meaningful archetypes and that errors are understandable and correctable via the refinement loop described above.

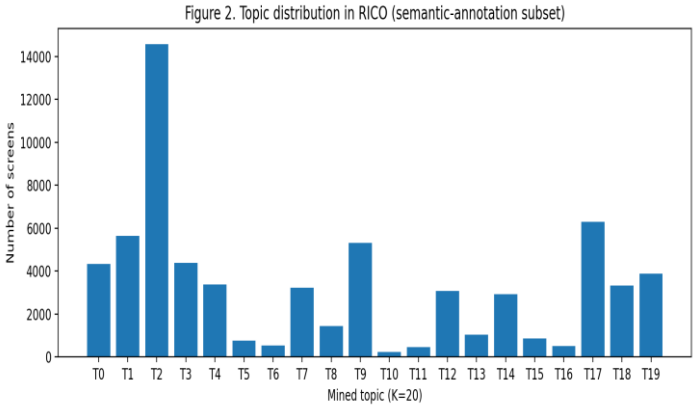

Figure 2. Topic distribution across 66,261 RICO screens (K=20).

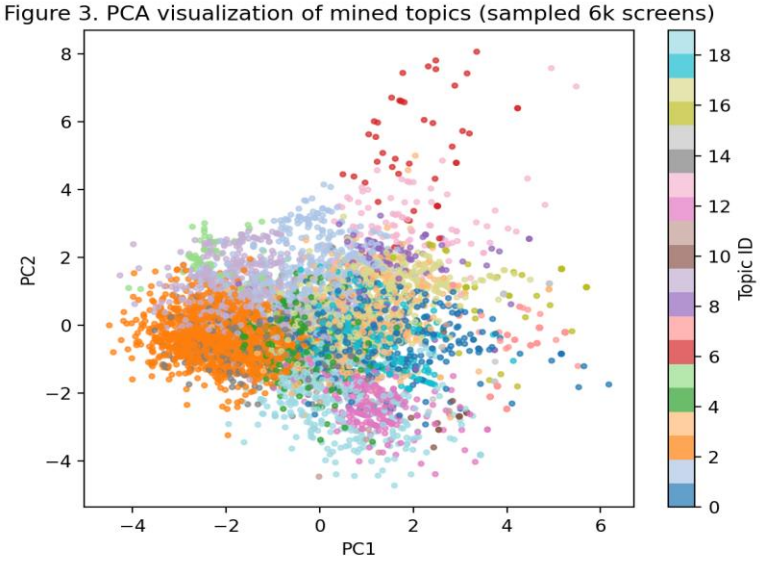

Figure 3. PCA visualization of semantic feature vectors colored by mined topic (6k sampled screens).

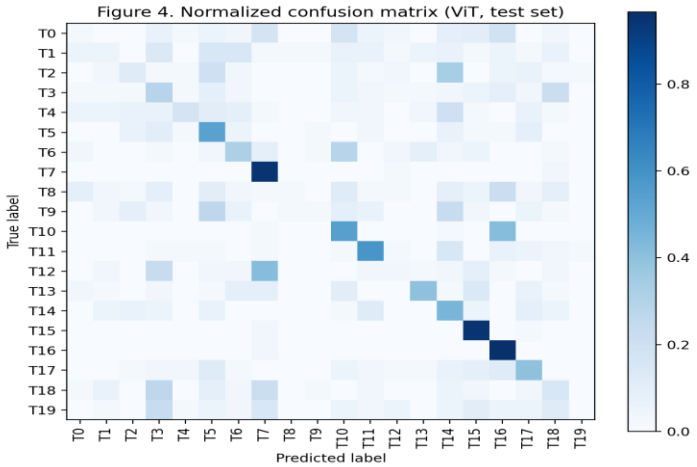

Figure 4. Normalized confusion matrix of ViT topic predictions on the test set.

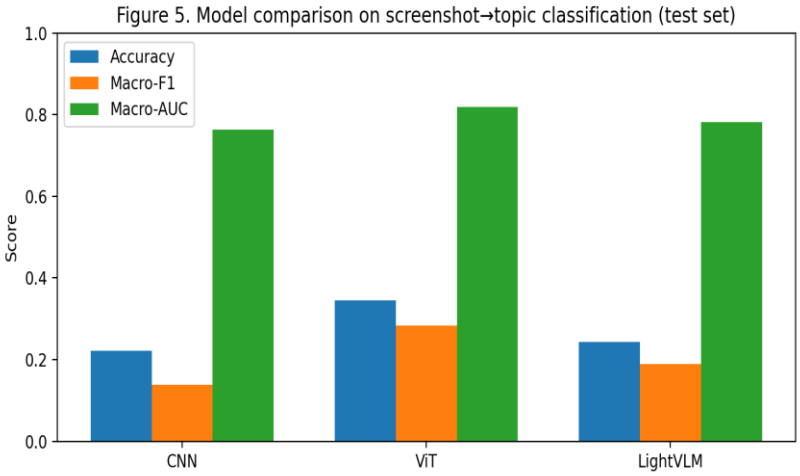

Figure 5. Overall quantitative comparison of CNN, ViT, and LightVLM on the test set.

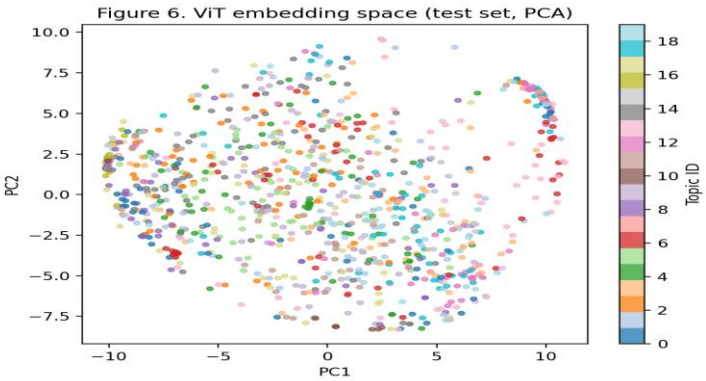

Figure 6. PCA visualization of ViT [CLS] embeddings for test screens.

Figure 7. Representative screenshots for four mined topics (two examples per topic).

Table VI. Overall test-set results for screenshot-based topic classification.

Model	**Accuracy**	**Macro-F1**	**Macro-AUC**
CNN	0.2222	0.1377	0.7636
ViT	0.3451	0.2840	0.8196
LightVLM	0.2433	0.1891	0.7817

Table VII. Per-topic F1 scores on the test set (higher indicates better topic recognition).

Topic	**F1_CNN**	**F1_ViT**	**F1_LightVLM**
T0	0.000	0.026	0.183
T1	0.161	0.071	0.182
T2	0.202	0.161	0.109
T3	0.220	0.212	0.086
T4	0.000	0.256	0.000

T5	0.000	0.368	0.055
T6	0.000	0.317	0.106
T7	0.589	0.591	0.606
T8	0.196	0.032	0.000
T9	0.000	0.031	0.075
T10	0.000	0.374	0.500
T11	0.000	0.543	0.000
T12	0.000	0.052	0.000
T13	0.367	0.495	0.193
T14	0.088	0.303	0.029
T15	0.299	0.687	0.628
T16	0.633	0.614	0.626
T17	0.000	0.390	0.374
T18	0.000	0.158	0.000
T19	0.000	0.000	0.030

Table VIII. Model efficiency: parameter count and average CPU inference time.

Model	**Params**	**Inference time (ms/screen)**
CNN	29044	3.2547
ViT	297364	2.4693
LightVLM (image encoder)	31904	3.5051

Table IX. Ablation and baselines: random alignment vs. trained alignment vs. semantic-only upper bound.

Method	**Accuracy**	**Macro-F1**	**Macro-AUC**
Random LightVLM (no training, seed=42)	0.0623	0.0130	0.4718
LightVLM (contrastive trained)	0.2433	0.1891	0.7817
Text-only prototypes (semantic freq cosine)	0.6077	0.6054	0.9453

Table X. Clustering quality comparison on a 2,000-screen sample (K=20).

Feature set	**Inertia**	**Silhouette(sample500)**

Component frequencies only (25D)	29273.100	0.153
Frequencies + meta (30D)	31175.200	0.109

Limitations

First, the topics used in this paper are pseudo-labels obtained from clustering semantic annotations rather than human-defined design categories. As Table IX shows, semantic features alone predict these pseudo-topics with high accuracy, which implies that the task is "semantic composition recognition" rather than subjective "design intent" recognition. Semantic composition topics quantify component inventory and design-system coverage, but they do not directly capture user goals such as "shopping checkout" or "profile editing." Future work can incorporate human-centered topic definitions or combine RICO with datasets that contain explicit topic labels such as Enrico [3].

Second, our screenshot-based models are trained and evaluated on a balanced subset with low-resolution images and short schedules to ensure CPU reproducibility. This design choice limits the amount of typography, iconography, and fine-grained styling that the models exploit. Higher resolutions and longer schedules increase the amount of visual detail available during optimization and directly address the resolution bottleneck. Stronger architectures and external pretraining increase representational capacity. Table VI therefore represents a resource-constrained operating point rather than an accuracy ceiling.

Third, semantic annotations are derived from automated extraction and contain noise. WebView containers and custom views reduce the quality of componentLabel assignments and cause semantically distinct screens to appear similar. The semantic vocabulary is also limited (25 labels), so subtle distinctions are not represented. Enriching semantics with richer text tokenization and structure-aware encoders is a direct extension.

Fourth, "trend mining" is limited by the dataset's lack of temporal metadata. Our analysis discovers prevalent topics and their signatures, supporting comparative design-space analysis. Trend analysis in the strict temporal sense requires longitudinal snapshots and release timelines.

Finally, because topics are mined from semantics, screenshot classification mixes two sources of difficulty: visual distinguishability and clustering tightness. Hard topics reflect visually subtle boundaries or noisy clustering. For deployment, merging entangled topics into a coarser taxonomy (as indicated by confusion matrices and embedding overlap) yields higher reliability and higher F1.

Conclusion

This paper presented an AI-driven pipeline for mobile UI pattern recognition and design topic mining on the RICO semantic-annotation subset. Using semantic composition features extracted from 66,261 view hierarchies, we mined 20 interpretable design topics via MiniBatch K-Means clustering and characterized each topic by prevalence and dominant componentLabel types. We then defined a screenshot-based topic classification benchmark and compared three lightweight modeling approaches: a CNN baseline, a compact ViT, and a contrastive LightVLM that aligns screenshots with semantic features.

On a balanced stratified evaluation set (3,000/594/1,188 train/val/test), the ViT achieved the best overall topic recognition (Accuracy = 0.345, Macro-F1 = 0.284, Macro-AUC = 0.820), confirming that attention-based global layout modeling is effective for UI topics. LightVLM improved over CNN in Macro-F1 and Macro-AUC, and ablation results clarified that mined topics are strongly grounded in semantic annotations. The combined topic mining and visual recognition pipeline supports practical design analytics workflows: semantic topic mining provides interpretable summaries of UI ecosystems, and lightweight models enable topic prediction from pixels when view hierarchies are unavailable.

References

[1] B. Deka, Z. Huang, C. Franzen, J. Hibschman, D. Afergan, Y. Li, J. Nichols, and R. Kumar, "Rico: A Mobile App Dataset for Building Data-Driven Design Applications," in Proc. 30th ACM Symp. on User Interface Software and Technology (UIST), 2017, pp. 845–854, doi: 10.1145/3126594.3126651.

[2] R. Kumar, A. Satyanarayan, C. Torres, M. Lim, S. Ahmad, S. R. Klemmer, and J. O. Talton, "Webzeitgeist: Design Mining the Web," in Proc. SIGCHI Conf. on Human Factors in Computing Systems (CHI), 2013, pp. 3083–3092, doi: 10.1145/2470654.2466420.

[3] L. A. Leiva, A. Hota, and A. Oulasvirta, "Enrico: A Dataset for Topic Modeling of Mobile UI Designs," in Proc. 22nd Int. Conf. on Human-Computer Interaction

with Mobile Devices and Services (MobileHCI), 2020, Article 9, doi: 10.1145/3406324.3410710.

[4] T. Beltramelli, "Pix2Code: Generating Code from a Graphical User Interface Screenshot," arXiv:1705.07962, 2017.

[5] X. Zhang, A. Swearngin, S. White, K. Murray, L. Yu, Q. Shan, J. Nichols, J. Wu, C. Fleizach, A. Everitt, and J. P. Bigham, "Screen Recognition: Creating Accessibility Metadata for Mobile Applications from Pixels," in Proc. CHI Conf. on Human Factors in Computing Systems (CHI), 2021, Article 275, doi: 10.1145/3411764.3445186.

[6] T. J. J. Li, A. Azaria, S. J. Pan, and J. Nichols, "Screen2Vec: Semantic Embedding of GUI Screens and GUI Components," in Proc. CHI Conf. on Human Factors in Computing Systems (CHI), 2021.

[7] C. Bai, X. Zhang, J. Zhang, J. Yu, Y. Xie, and X. Zhou, "UIBert: Learning Generic Multimodal Representations for UI Understanding," in Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI), 2021.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[9] A. Dosovitskiy et al., "An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale," in Int. Conf. on Learning Representations (ICLR), 2021.

[10] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.

[11] A. Radford et al., "Learning Transferable Visual Models from Natural Language Supervision," in Proc. Int. Conf. on Machine Learning (ICML), 2021, pp. 8748–8763.

[12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in Proc. Int. Conf. on Machine Learning (ICML), 2020, pp. 1597–1607.

[13] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," arXiv:1807.03748, 2018.

[14] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in Int. Conf. on Learning Representations (ICLR), 2015.

[15] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[16] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," Comput. Sci. Eng., vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.

[17] S. Lloyd, "Least Squares Quantization in PCM," IEEE Trans. Inf. Theory, vol. 28, no. 2, pp. 129–137, 1982, doi: 10.1109/TIT.1982.1056489.

[18] P. J. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," J. Comput. Appl. Math., vol. 20, pp. 53–65, 1987, doi: 10.1016/0377-0427(87)90125-7.

[19] I. T. Jolliffe, Principal Component Analysis, 2nd ed. Springer, 2002.

[20] T. Fawcett, "An Introduction to ROC Analysis," Pattern Recognit. Lett., vol. 27, no. 8, pp. 861–874, 2006, doi: 10.1016/j.patrec.2005.10.010.

[21] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in Int. Conf. on Learning Representations (ICLR), 2015.

[22] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proc. Int. Conf. on Machine Learning (ICML), 2019, pp. 6105–6114.

[23] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861, 2017.

[24] A. P. Bradley, "The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms," Pattern Recognit., vol. 30, no. 7, pp. 1145–1159, 1997, doi: 10.1016/S0031-3203(96)00142-2.

[25] Xinzhuo Sun, Yifei Lu, and Jing Chen, "Controllable Long-Term User Memory for Multi-Session Dialogue: Confidence-Gated Writing, Time-Aware Retrieval-Augmented Generation, and Update/Forgetting", JACS, vol. 3, no. 8, pp. 9–24, Aug. 2023, doi: 10.69987/JACS.2023.30802.

[26] Hanqi Zhang, "DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR Models", JACS, vol. 3, no. 7, pp. 24–40, Jul. 2023, doi: 10.69987/JACS.2023.30703.

[27] Meng-Ju Kuo, Boning Zhang, and Haozhe Wang, "Tokenized Flow-Statistics Encrypted Traffic Analysis: Comparative Evaluation of 1D-CNN, BiLSTM, and Transformer on ISCX VPN-nonVPN 2016 (A1+A2, 60 s)", JACS, vol. 3, no. 8, pp. 39–53, Aug. 2023, doi: 10.69987/JACS.2023.30804.

[28] Xinzhuo Sun, Jing Chen, Binghua Zhou, and Meng-Ju Kuo, "ConRAG: Contradiction-Aware Retrieval-Augmented Generation under Multi-Source

Conflicting Evidence", JACS, vol. 4, no. 7, pp. 50–64, Jul. 2024, doi: 10.69987/JACS.2024.40705.

[29] Hanqi Zhang, "Risk-Aware Budget-Constrained Auto-Bidding under First-Price RTB: A Distributional Constrained Deep Reinforcement Learning Framework", JACS, vol. 4, no. 6, pp. 30–47, Jun. 2024, doi: 10.69987/JACS.2024.40603.

[30] T. Shirakawa, Y. Li, Y. Wu, S. Qiu, Y. Li, M. Zhao, H. Iso, and M. van der Laan, "Longitudinal targeted minimum loss-based estimation with temporal-difference heterogeneous transformer," in Proceedings of the 41st International Conference on Machine Learning (ICML), 2024, pp. 45097–45113, Art. no. 1836.

[31] K. Xu, H. Zhou, H. Zheng, M. Zhu, and Q. Xin, "Intelligent classification and personalized recommendation of e-commerce products based on machine learning," Proceedings of the 6th International Conference on Computing and Data Science (ICCDS), 2024.

[32] Q. Xin, Z. Xu, L. Guo, F. Zhao, and B. Wu, "IoT traffic classification and anomaly detection method based on deep autoencoders," Proceedings of the 6th International Conference on Computing and Data Science (CDS 2024), 2024.

[33] B. Wang, Y. He, Z. Shui, Q. Xin, and H. Lei, "Predictive optimization of DDoS attack mitigation in distributed systems using machine learning," Proceedings of the 6th International Conference on Computing and Data Science (CDS 2024), 2024, pp. 89–94.

[34] Z. Ling, Q. Xin, Y. Lin, G. Su, and Z. Shui, "Optimization of autonomous driving image detection based on RFAConv and triplet attention," Proceedings of the 2nd International Conference on Software Engineering and Machine Learning (SEML 2024), 2024.

[35] J. Chen, J. Xiong, Y. Wang, Q. Xin, and H. Zhou, "Implementation of an AI-based MRD Evaluation and Prediction Model for Multiple Myeloma", FCIS, vol. 6, no. 3, pp. 127–131, Jan. 2024, doi: 10.54097/zJ4MnbWW.

[36] K. Yu, D. Yuan, and S. Min, "Enhancing credit decision transparency for small business owners: An explainable AI approach to mitigate algorithmic bias in micro-lending," Journal of Computing Innovations and Applications, vol. 2, no. 2, pp. 66–77, 2024.

[37] H. Weng, S. Zhang, and S. Min, "Multi-constraint optimization for real-time bidding: A reinforcement learning approach," Artificial Intelligence and Machine Learning Review, vol. 5, no. 1, pp. 93–104, 2024.

[38] A. Kang, S. Min, and D. Yuan, "Comparative analysis of foreign exchange market shock transmission and recovery resilience among major economies under geopolitical conflicts," Journal of Computing Innovations and Applications, vol. 2, no. 1, pp. 46–61, 2024.

[39] S. Min and C. Wei, "Comparative analysis of filter-based feature selection methods for high-dimensional data in classification tasks," Journal of Advanced Computing Systems, vol. 3, no. 8, pp. 25–38, 2023.

[40] S. Min, L. Guo, and G. Weng, "Alert fatigue mitigation in anomaly detection systems: A comparative study of threshold optimization and alert aggregation strategies," Journal of Computing Innovations and Applications, vol. 1, no. 2, pp. 59–73, 2023.

[41] W. Chen, S. Min, A. T. Chala, Y. Zhang, and X. Liu, "Assessing compaction of existing railway subgrades using dynamic cone penetration testing," Proceedings of the Institution of Civil Engineers – Geotechnical Engineering, 2022.

[42] S. Wang, S. Min, J. Yu, H. Cheng, Z. Tse, and W. Song, "Contact-less home activity tracking system with floor seismic sensor network," in Proc. 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), 2021, pp. 13–18.