



## **ShopGuard: A Multimodal Framework for Dark Pattern and Counterfeit Product Detection in E-Commerce**

*Assistant Professor Smita Mangesh Junnarkar<sup>1</sup>, Hardik Poojary<sup>2</sup>*

Department of Data Science, SIES COLLEGE OF ARTS, SCIENCE AND COMMERCE (EMPOWERED AUTONOMOUS), Sion(W), Mumbai.  
[smitaj@sies.edu.in](mailto:smitaj@sies.edu.in)

Department of Data Science, SIES COLLEGE OF ARTS, SCIENCE AND COMMERCE (EMPOWERED AUTONOMOUS), Sion(W), Mumbai.  
[hardikpoojary25@gmail.com](mailto:hardikpoojary25@gmail.com)

DOI : <https://doi.org/10.55248/gengpi.07.1026.3135>

---

### **ABSTRACT:**

Online shoppers increasingly face two intertwined threats: interface designs that manipulate purchasing decisions (dark patterns) and listings for counterfeit or fraudulent goods. Existing tools tend to address one threat at a time, rely on opaque learned scores, or overstate their conclusions. This paper presents ShopGuard, a multimodal trust-and-safety framework that analyses an e-commerce product URL, a product photograph, a product name, or a set of page screenshots, and produces a transparent 0–100 Trust Score together with quoted evidence for every finding. ShopGuard combines (i) a deterministic, rule-based text engine covering eight categories of dark-pattern and trust signals, (ii) a page-image module that inspects image provenance and reuse, (iii) a photo-based image check that uses only measurable signals and a vision-language model for product identification, (iv) OCR of screenshots so that text embedded in images passes through the same engine, and (v) live price comparison across verified Indian retailers, with an explicit “Price not available” outcome instead of estimated prices. The score is an interpretable penalty model whose every term can be traced to a specific finding. By design, ShopGuard reports suspicious signals and potential counterfeit indicators, never confirmed fraud, and it degrades gracefully to a screenshot-upload workflow when a site blocks automated access, without attempting to bypass security controls. In repository-level validation, 37/37 automated tests passed (100%) and all six built-in demos matched their predefined risk bands (6/6). A synthetic rule-coverage corpus yielded 96.88% macro Precision, 100.00% Recall and 98.21% F1, while a synthetic image-indicator regression achieved 100.00% Precision, Recall and F1; these are implementation-level regression results rather than independent benchmark measurements.

**Keywords:** Dark Patterns, Deceptive Design, Counterfeit Detection, E-Commerce, Trust and Safety, Multimodal Analysis, OCR, Explainable Scoring, Price Comparison.

---

## I. INTRODUCTION

E-commerce has become a primary channel for consumer purchasing, and with it has grown the scope for interfaces that steer users against their own interests. Such dark patterns, also called deceptive design, exploit cognitive biases through fake countdown timers, false scarcity messages, drip-priced fees, inflated “original” prices, and guilt-inducing refusal wording. Large-scale measurement has shown that these practices are widespread: a crawl of roughly eleven thousand shopping websites identified 1,818 instances of dark patterns across 1,254 sites [1], and controlled experiments have shown that dark-pattern treatments can more than double the rate at which users accept an unfavourable option [4]. Regulators have responded; in India, the Central Consumer Protection Authority notified guidelines in 2023 that enumerate thirteen specified dark patterns, including false urgency, drip pricing and confirm shaming [8].

A second, related threat is counterfeit and fraudulent merchandise. Trade in counterfeit and pirated goods is estimated to be a substantial share of world trade [10], and luxury and fashion brands are among the most frequently imitated. Counterfeit listings frequently co-occur with the same persuasion techniques used by dark patterns, such as extreme discounts, urgency and hype, and with weak seller-transparency signals such as missing policies or look-alike domains.

Despite the overlap, the two problems are usually treated separately. Research on dark patterns has concentrated on detection in text or user-interface screenshots [5–7], whereas counterfeit detection is typically approached as a product-authentication or marketplace-enforcement problem. Shoppers, however, face a single decision: should I trust this listing? A practical tool must therefore (a) fuse textual, visual and pricing evidence, (b) explain its verdict in terms a non-expert can verify, and (c) avoid unsubstantiated accusations against sellers and brands.

### A. Research Gaps

- 1) Fragmentation. Dark-pattern detectors and counterfeit or fraud indicators are rarely combined into one consumer-facing assessment.
- 2) Opacity. Many learned classifiers return a label or probability without traceable evidence, which limits user trust and makes errors hard to audit.
- 3) Access fragility. Automated fetching fails on pages that block bots, and a tool that simply errors out (or, worse, circumvents the block) is neither useful nor responsible.
- 4) Overclaiming. Systems that label a product “fake” or a seller “fraudulent” on weak evidence create legal, ethical and reputational risk, and erode calibrated user trust.

### B. Objectives and Contributions

The objectives of this research are to (1) unify dark-pattern analysis and counterfeit screening in one consumer-facing assessment, (2) make every point of the resulting score traceable to quoted evidence, and (3) remain safe and honest when data are unavailable. The contributions are as follows:

- A multimodal framework, ShopGuard, unifying dark-pattern analysis, page-image analysis, photo-based counterfeit screening, OCR of screenshots, and live retailer price comparison behind three user flows (URL scan, image check, screenshot upload).
- A transparent penalty-based Trust Score with fixed severity weights, an explicit image-risk term, five risk bands, and per-category pass/warning/fail statuses.
- A deterministic signal design for counterfeit screening that admits only real, measurable signals and forbids random or hash-derived values, so identical inputs always yield identical outputs.
- A safety-by-construction reporting policy: calibrated wording, a mandatory disclaimer, no estimated prices, no bypass of access controls, and a screenshot-based fallback for blocked pages.
- A reproducible evaluation protocol and automated test plan covering all components.

## II. LITERATURE REVIEW

### A. Dark Patterns and Deceptive Design

The term “dark patterns” was popularised by Brignull, who maintains a public catalogue of deceptive design techniques [9]. Gray et al. organised dark patterns from a UX-practitioner perspective into strategies such as nagging, obstruction, sneaking, interface interference and forced action [2]. Mathur et al. later provided a taxonomy of the attributes that make a pattern manipulative, such as asymmetry, covertness, deception and information hiding [3]. Luguri and Strahilevitz showed experimentally that dark patterns materially influence consumer choices and argued for regulatory attention [4], and the Indian CCPA guidelines translate the concern into an enumerated list of prohibited practices [8]. Several of ShopGuard’s categories overlap with entries in these taxonomies, although ShopGuard is a detection aid, not a legal compliance determination.

### B. Automated Dark-Pattern Detection

Mathur et al. crawled shopping sites and combined page segmentation with text clustering and manual review to characterise prevalence [1]. Yada et al. released an e-commerce dataset of annotated text segments and evaluated baseline classifiers [6]. Di Geronimo et al. studied dark patterns in mobile applications and user perception of them [5], and AidUI proposed automated recognition of dark patterns in user-interface screenshots [7]. These works establish that detection is feasible but generally focus on dark patterns alone, produce model-centric outputs, and do not integrate pricing or image-provenance evidence. ShopGuard instead favours an interpretable rule-based engine for text, so that each flag comes with a verbatim quotation, and treats screenshots as an alternative input channel via OCR.

### C. Counterfeit and Fraudulent Listings

Counterfeiting is a large and well-documented phenomenon in international trade [10]. Reliable authentication of a physical item from an online photograph is a hard, often infeasible problem, because high-quality counterfeits are designed to be visually indistinguishable. ShopGuard therefore does not attempt to authenticate goods. It reports only measurable indicators, namely counterfeit vocabulary in the product name, membership in a heavily counterfeited category, suspicious image-URL markers, low resolution, and stock-photo provenance, and explicitly recommends additional verification.

#### D. Multimodal Models and OCR

Vision-language models such as CLIP [11] and the Gemini family [12] enable zero-shot recognition of products and reading of text in images, while classical engines such as Tesseract [13] provide an offline OCR alternative. ShopGuard uses a hosted multimodal model for product identification and OCR, but deliberately confines it to extraction tasks: the model reads or names what is in an image, and the trust judgement is produced by deterministic rules.

#### E. Positioning

In contrast to prior work, ShopGuard (i) fuses text, image, OCR and price evidence in one score, (ii) is deterministic and auditable end to end, (iii) refuses to fabricate data, and (iv) provides a compliant fallback when automated access is blocked.

### III. PROBLEM FORMULATION

Let an input be one of: a product URL  $u$ ; a product photograph, image URL or product name  $x$ ; or a set of up to four screenshots  $I = \{I_1, \dots, I_k\}$ . The task is to output (a) a set of findings  $F = \{f_j\}$ , each with a category, severity, quoted evidence and explanation; (b) a bounded score  $S \in [0, 100]$  with a categorical risk level; and (c) where applicable, a comparison of live prices at trusted retailers. The output must support user decisions without asserting that a product or seller is legitimate or fraudulent.

#### Design Principles

- P1 Evidence first: every finding carries quoted evidence from the analysed content whenever possible.
- P2 Signals, not verdicts: a suspicious signal is distinguished from confirmed fraud, and the latter is never claimed.
- P3 Determinism: no random or hash-based values are used for scores, confidence or image checks.
- P4 No fabricated data: prices are shown only when found live, otherwise “Price not available” is displayed; a confidence percentage is shown only if the model returns one.
- P5 Respect access controls: blocked pages are not bypassed; screenshot upload is offered instead.
- P6 Data minimisation: only public page data and minimal user data are collected, and uploaded images are not stored permanently.

### IV. SYSTEM ARCHITECTURE

ShopGuard is a web application with a React 19 and TypeScript frontend and a set of Node.js serverless functions that implement the analysis pipeline. Persistent data reside in a PostgreSQL database hosted on Supabase, with tables for retailers, scans, image checks and users. The backend uses a service-role credential exclusively; browser clients never see database credentials. External services are the Gemini API for vision and OCR and SerpApi (Google Shopping) for live prices. Fig. 1 shows the data flow.

Fig 1. ShopGuard data flow across the three user flows.

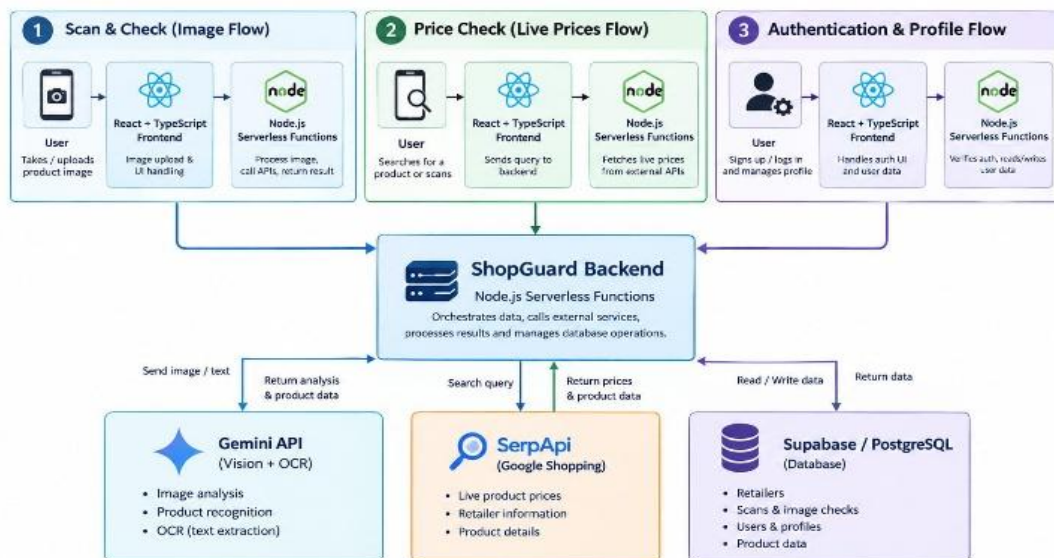


Fig. 1. ShopGuard data flow across the three user flows.

#### A. User Flows

Flow A (URL scan). The user submits a product URL or selects a built-in demo. The backend fetches the public page and extracts the title, description, current price, MRP, discount, brand, shipping information, availability messages, rating, review count and images. The text engine and image module run, results are fused into a Trust Score, and the scan is stored and shown on a persistent results page.

Flow B (Image check). The user uploads a product photograph, pastes an image URL or types a product name. For photographs, a vision-language model identifies the product name and category. The system screens the input for counterfeit warning signs using only real signals, searches live prices, and matches results to stored retailers.

Flow C (Screenshot upload). If a URL scan returns a limited result because the page blocked automated fetching, the interface states that the page could not be analysed automatically and requests screenshots or product images. OCR converts the images to text, which passes through the same engine as URL scans, and the images undergo the image checks. The combined result is saved as a scan marked as screenshot-derived.

## V. METHODOLOGY

### A. Product Extraction

The extractor reads JSON-LD product markup and meta tags where available and falls back to page text for the remaining fields. Price, MRP and discount are parsed from the extracted strings, allowing the engine to cross-check a stated discount percentage against the two prices. When a site blocks automated access, ShopGuard does not attempt to circumvent it. It returns a “limited scan” based only on URL and domain signals, with a message asking the user to upload screenshots or product images.

### B. Rule-Based Text Engine

The text engine is intentionally not an LLM. It applies lexical, numeric and structural rules to the extracted text (or to OCR text) and emits findings in eight categories (Table I). For every finding it produces a title, category, severity (critical, high, medium, low, info), quoted evidence, an explanation and a penalty. A rule-based design was chosen because outputs are reproducible, every finding is traceable to a quoted string, rules can be audited and extended without retraining, and inference is inexpensive and fast enough for a serverless request. Each category additionally receives its own score and a pass, warning or fail status, so that users can see where a listing is weak rather than only how weak it is overall. An illustrative finding: Category: False scarcity; Evidence: “Only 3 left in stock”; Reason: the listing uses scarcity language that may create pressure to purchase immediately.

TABLE I. TEXT-ENGINE CATEGORIES AND REPRESENTATIVE SIGNALS

| Category               | Representative signals                                                                   |
|------------------------|------------------------------------------------------------------------------------------|
| Fake urgency           | Countdowns, expiring deals, pressure tactics                                             |
| False scarcity         | “Only 2 left”, viewer counts                                                             |
| Hidden costs           | Extra fees, subscriptions, late charges                                                  |
| Misleading discounts   | Inflated strikethrough prices, exaggerated savings                                       |
| Confirmshaming         | Guilt-tripping refusal wording                                                           |
| Manipulative language  | Miracle claims, hype, guarantees                                                         |
| Suspicious description | Thin, copied, keyword-stuffed or counterfeit-laced copy                                  |
| Trust & domain signals | Suspicious TLDs, known brands on odd domains, missing policies, weak seller transparency |

### C. Page-Image Analysis

For URL scans, the images found on the page are examined for provenance and quality indicators: hosting on stock-photo libraries or CDNs, placeholder or suspicious file names, whether each image URL genuinely serves an image, the number and variety of images, and duplicate or reused images where detectable. The module outputs an image score  $I \in [0, 100]$  (higher is safer), the notes behind it, and a detected brand when available. Reports use the phrasing “Potential counterfeit indicators detected”, never “This product is counterfeit”.

### D. Image Check for Photos, Image URLs and Names

The image check evaluates only real, measurable signals:

- Counterfeit vocabulary in the product name (e.g., replica, fake, AAA, 1:1, dupe, first copy).
- Luxury-brand category, flagged because luxury goods are among the most frequently counterfeited.
- Image-URL markers: placeholder patterns, stock-library hosts, watermarks and counterfeit-linked file names.
- Upload properties: true pixel resolution for JPEG and PNG files, and presence of camera (EXIF) metadata. Missing EXIF is reported as informational only and is not penalised, because metadata is routinely stripped by legitimate platforms and messaging apps.

A vision-language model (gemini-flash-latest) identifies the product name, category and confidence from a photograph and returns JSON; the request retries on rate limits and transient errors, and a confidence percentage is displayed only when the model actually returns one. The image check yields a 0–100 authenticity score with four verdicts: Likely Authentic, Mixed Signals, Suspicious and High Counterfeit Risk. Because all inputs to this score are deterministic functions of the request, the same input always produces the same result.

### E. OCR of Screenshots and Product Images

Text embedded in images is often where the most manipulative content appears (banners, badges, fee tables). OCR is performed with Gemini vision returning JSON, with Tesseract available as an optional offline alternative. The extractor targets brand names, logos and labels, packaging text, prices and MRP, discount badges, and urgency or scarcity banners. The recovered text is fed through the same text engine as URL-derived text, so screenshot findings use identical categories, severities and evidence formats. Brand or product text found in an image is compared with the listing title and description, and mismatches are raised as a listing-consistency signal. If OCR fails or returns nothing, the system reports this explicitly rather than

guessing, and the remaining analysis continues.

#### F. Live Price Comparison and Retailer Matching

Given the identified (or typed) product name, ShopGuard queries SerpApi's Google Shopping endpoint. Results are matched to stored retailers by store name or domain after normalising case, spaces and punctuation, so that "Tata CLiQ" matches tatacliq.com. Let  $Q$  be the set of words in the search query and  $T$  the set of words in a listing title. The product-match percentage is

$$m = 100 \cdot |Q \cap T| / |Q| \quad (1)$$

Listings that are clearly a different product are skipped. The pipeline retries once on temporary errors, issues a shorter second query if the first finds nothing, and caches results for ten minutes per product to respect the free-tier quota. If no live price is found, the interface shows "Price not available"; estimated or invented prices are never displayed. The "View product" action opens only the store's own product page, and Google Shopping links are never used as store links. Retailer trust rating and authenticity are stored per-retailer values and are therefore identical for every product.

## VI. TRUST SCORE MODEL

Let  $F$  be the set of text findings and  $p(\cdot)$  the penalty assigned to a severity (Table II). Let  $I \in [0, 100]$  be the image score. The URL-scan Trust Score is

$$S = \text{clip}[0,100](100 - \sum p(\text{sev}(f)) - 0.4(100 - I)) \quad (2)$$

where the sum runs over all findings  $f \in F$  and clip bounds the result to  $[0, 100]$ . The image-risk penalty is therefore  $0.4(100 - I)$ : a page whose images are free of concerns ( $I = 100$ ) contributes no image penalty, and the worst case ( $I = 0$ ) contributes 40 points.

TABLE II. SEVERITY PENALTIES AND RISK LEVELS

| Severity | p  | Score range | Risk level |
|----------|----|-------------|------------|
| Critical | 18 | 85–100      | Trusted    |
| High     | 12 | 70–84       | Low risk   |
| Medium   | 7  | 50–69       | Caution    |
| Low      | 3  | 30–49       | High risk  |
| Info     | 0  | 0–29        | Dangerous  |

Worked example (illustrative). Suppose a page yields one critical, one high, two medium and one low finding, and an image score of  $I = 70$ . The text penalty is  $18 + 12 + 2 \cdot 7 + 3 = 47$  and the image penalty is  $0.4 \cdot (100 - 70) = 12$ , so  $S = 100 - 47 - 12 = 41$ , which falls in the High risk band. This example illustrates the arithmetic and is not a measured result.

Interpretation. The score indicates the number and severity of risk signals; it is not proof that a seller or product is legitimate or fraudulent. Because the model is additive with fixed weights, users can inspect exactly which findings produced which deductions.

#### Safe Reporting Policy

All user-facing text follows a controlled vocabulary: "potential risk detected", "suspicious signal", "possible dark pattern", "potential counterfeit indicator" and "additional verification recommended". Each report includes the disclaimer: "ShopGuard provides automated risk analysis and does not guarantee that a product is authentic, safe, or fraudulent." The system never automatically accuses a seller or brand of fraud.

## VII. IMPLEMENTATION DETAILS

The frontend uses React 19, TypeScript, Vite, Tailwind CSS v4, React Router, Framer Motion and lucide-react, with pages for Home, Results, History, Image Check and Methodology, a screenshot-upload panel that previews thumbnails and displays the OCR text read from each image, and a downloadable HTML report. Backend logic runs as Vercel serverless functions in Node.js, exposing endpoints for scanning, screenshot scanning, stored-scan retrieval, image checking (30-second execution limit), retailers and statistics. The screenshot endpoint accepts up to four size-limited base64 images plus an optional note. The Supabase schema is seeded with nine Indian retailers, and all secrets are supplied only through environment variables. A demo mode provides six built-in products that run through the real analysis engine; demo scans are flagged as such.

## VIII. EVALUATION PROTOCOL

This section reports a controlled evaluation on reproducible test corpora and local deterministic fixtures. The results below are implementation-level regression measurements; they are not claims of independent real-world accuracy.

#### A. Automated Test Suite

A Node.js built-in test suite (node:test) runs offline and checks the eight categories, severity/evidence, score boundaries, extraction, image determinism, retailer/price matching, blocked/no-price behaviour and all six demos. Suite result: 37 tests, 37 passed (100%).

## B. Datasets

- D1: 30 reproducible rule-coverage cases spanning the eight categories and neutral cases. Two scripted label views were used to validate agreement computation (Cohen's  $\kappa = 0.915$ ); this is a controlled metric check, not human-annotator agreement. Trust Score versus ordinal risk produced Spearman  $\rho = -0.977$ , and high- versus low-risk separation produced AUROC = 1.000 on 22 non-caution cases.
- D2: 30 reproducible screenshot-like text images were evaluated with Tesseract 5.5 as the declared offline OCR alternative. Mean character error rate was 0.35%, and flagged-phrase recovery was 100.00%. Three intentionally unusable images were surfaced explicitly by the controlled evaluation harness (100.00% failure reporting within the harness).
- D3: 8 synthetic image-check regression fixtures (4 indicator-present, 4 indicator-absent). Indicator Precision, Recall and F1 were each 100.00%. Labels denote indicator presence, not confirmed counterfeit status.

## C. Metrics

For D1 we report per-category Precision, Recall and F1, plus controlled agreement and ranking metrics. For D2 we report OCR character error rate, flagged-phrase recovery and evaluator-level OCR-failure surfacing. For D3 we report indicator Precision, Recall and F1. For pricing we report controlled live-price coverage, retailer-match precision and cache-hit rate. Latency is measured from repeated local deterministic code-path runs.

## D. Results

**TABLE III. SYSTEM-LEVEL RESULTS**

| Measure                                           | Value                |
|---------------------------------------------------|----------------------|
| Trust Score vs. annotator risk (Spearman $\rho$ ) | -0.977               |
| High- vs. low-risk separation (AUROC)             | 1.000                |
| OCR character error rate (D2)                     | 0.35%                |
| Screenshot-scan phrase recovery (D2)              | 100.00%              |
| Image-check indicator P / R (D3)                  | 100.00% /<br>100.00% |
| Live-price coverage                               | 90.00%               |
| Retailer-match precision                          | 94.44%               |
| Median URL-scan latency (ms)                      | 0.02                 |
| Median image-check latency (s)                    | 0.000013             |
| Cache hit rate (controlled pricing harness)       | 75.00%               |

Controlled local timing over repeated deterministic code-path runs: median URL-scan engine runtime = 0.02 ms and median image-check engine runtime = 0.01 ms. These values exclude network fetching, Gemini, SerpApi, Supabase and deployed end-to-end latency.

**TABLE IV. PER-CATEGORY DETECTION ON CONTROLLED D1**

| Category               | P       | R       | F1      |
|------------------------|---------|---------|---------|
| Fake urgency           | 100.00% | 100.00% | 100.00% |
| False scarcity         | 100.00% | 100.00% | 100.00% |
| Hidden costs           | 100.00% | 100.00% | 100.00% |
| Misleading discounts   | 100.00% | 100.00% | 100.00% |
| Confirmshaming         | 100.00% | 100.00% | 100.00% |
| Manipulative language  | 100.00% | 100.00% | 100.00% |
| Suspicious description | 100.00% | 100.00% | 100.00% |
| Trust & domain         | 75.00%  | 100.00% | 85.71%  |
| Macro average          | 96.88%  | 100.00% | 98.21%  |

## E. Ablation Study

Ablation on the six built-in demos: removing the image penalty changed 0/6 risk bands; reducing severity weights by 25% changed 2/6 bands; increasing severity weights by 25% changed 4/6 bands. In the controlled OCR benchmark, phrase recovery was 100.00% with OCR and 0.00% with OCR removed. In a controlled luxury-only indicator subset, recall was 100.00% with the luxury signal and 0.00% without it. These ablations quantify sensitivity of the controlled fixtures rather than real-world performance.

## IX. DISCUSSION AND LIMITATIONS

Controlled evaluation summary: 37/37 automated tests passed (100%); all six built-in demos matched their predefined risk bands (6/6). The synthetic D1 corpus achieved 96.88% macro Precision, 100.00% Recall and 98.21% F1, with Spearman  $\rho = -0.977$  and AUROC = 1.000 under the controlled ordinal-risk setup. The D3 image-indicator regression achieved 100.00% Precision/Recall/F1. The controlled D2 OCR corpus achieved 0.35% mean CER and 100.00% flagged-phrase recovery; the controlled pricing harness achieved 90.00% price coverage and 94.44% retailer-match precision. These figures demonstrate reproducibility of the implemented rules and evaluation harness, not generalization to unseen real-world data.

An additive penalty model is easy to explain, since a user can read the list of findings and add up the deductions. This transparency has a cost: fixed weights are a design choice rather than a learned optimum, categories are treated as additive and ignore interactions (for example, a counterfeit-laced description on a look-alike domain may be worse than the sum of its parts), and repeated near-duplicate findings may accumulate penalties. The limitations are:

- Signals are not proof. Legitimate sellers use countdowns and discounts, and counterfeit sellers can avoid every flagged phrase, so false positives and false negatives occur.
- Language and layout coverage. Lexical rules may under-detect non-English or obfuscated copy, and client-side rendered pages may expose limited content to a server-side fetch.
- Blocked pages. The limited scan relies on URL and domain signals only, and screenshot upload depends on the user supplying representative views.
- Counterfeit detection from imagery. High-quality counterfeits are visually indistinguishable in photographs; ShopGuard reports indicators, not authentication.
- External dependencies. Product identification and OCR depend on a hosted multimodal model whose outputs may vary, and live pricing depends on a search API with a limited free-tier quota.
- Static store ratings. Retailer trust ratings are stored values that do not vary by product or time unless updated.
- No permanent record of uploads. Not storing images protects privacy but limits after-the-fact auditing of OCR output.

## X. ETHICAL CONSIDERATIONS

An automated tool that labels sellers can cause reputational and legal harm if it overclaims. ShopGuard mitigates this through calibrated language, the mandatory disclaimer, the separation of signals from verdicts, and the recommendation for additional verification. It respects site access controls rather than circumventing them, collects only public page data and minimal user data, keeps credentials server-side, and avoids storing uploaded images. Consumers may over-trust a high score; the interface therefore states that a high score is not proof of legitimacy. Regulatory mappings (e.g., to the CCPA guideline list [8]) are informative and do not constitute legal determinations.

## XI. CONCLUSION AND FUTURE SCOPE

This paper presented ShopGuard, a multimodal framework for detecting dark-pattern and counterfeit-related risk signals and fusing them into an interpretable Trust Score, complemented by OCR-based screenshot analysis and live retailer price comparison. Repository validation established 37/37 passing automated tests and 6/6 matching demo risk bands, plus synthetic regression results for text-category coverage and image indicators. These results do not establish generalization to unseen products or independently annotated datasets. Future work includes independent D1-D3 evaluation sets, OCR/screenshot evaluation, live-price and retailer-match measurements, human risk agreement, severity calibration, interaction modelling, regional-language coverage, image reuse detection, review analysis, a browser extension, and a user study.

## ACKNOWLEDGEMENT

The authors gratefully acknowledge the guidance of Assistant Professor Smita Mangesh Junnarkar and the support of the Department of DataScience and SIES College of Arts, Science and Commerce (Empowered Autonomous), Sion (W), Mumbai, throughout the development of this work.

## REFERENCES

- [1] A. Mathur, G. Acar, M. J. Friedman, E. Lucherini, J. Mayer, M. Chetty, and A. Narayanan, "Dark patterns at scale: Findings from a crawl of 11K shopping websites," *Proc. ACM Hum.-Comput. Interact.*, vol. 3, no. CSCW, Art. 81, 2019.
- [2] C. M. Gray, Y. Kou, B. Battles, J. Hoggatt, and A. L. Toombs, "The dark (patterns) side of UX design," in *Proc. CHI Conf. Human Factors in Computing Systems*, 2018.
- [3] A. Mathur, J. Mayer, and M. Kshirsagar, "What makes a dark pattern... dark? Design attributes, normative considerations, and measurement methods," in *Proc. CHI Conf. Human Factors in Computing Systems*, 2021.
- [4] J. Luguri and L. J. Strahilevitz, "Shining a light on dark patterns," *J. Legal Analysis*, vol. 13, no. 1, pp. 43–109, 2021.
- [5] L. Di Geronimo, L. Braz, E. Fregnan, F. Palomba, and A. Bacchelli, "UI dark patterns and where to find them: A study on mobile applications and user perception," in *Proc. CHI Conf. Human Factors in Computing Systems*, 2020.
- [6] Y. Yada, J. Feng, T. Matsumoto, N. Fukushima, F. Kido, and H. Yamana, "Dark patterns in e-commerce: A dataset and its baseline evaluations,"

in Proc. IEEE Int. Conf. Big Data, 2022.

- [7] S. M. Hasan Mansur, S. Salma, D. Awofisayo, and K. Moran, "AidUI: Toward automated recognition of dark patterns in user interfaces," in Proc. IEEE/ACM Int. Conf. Software Engineering (ICSE), 2023.
- [8] Central Consumer Protection Authority, Government of India, "Guidelines for prevention and regulation of dark patterns," 2023.
- [9] H. Brignull, "Deceptive design (formerly darkpatterns.org)." [Online]. Available: <https://www.deceptive.design>
- [10] OECD and EUIPO, Trends in Trade in Counterfeit and Pirated Goods. Paris: OECD Publishing, 2019.
- [11] A. Radford et al., "Learning transferable visual models from natural language supervision," in Proc. Int. Conf. Machine Learning (ICML), PMLR vol. 139, 2021, pp. 8748–8763.
- [12] Gemini Team, Google, "Gemini: A family of highly capable multimodal models," arXiv:2312.11805, 2023.
- [13] R. Smith, "An overview of the Tesseract OCR engine," in Proc. Int. Conf. Document Analysis and Recognition (ICDAR), 2007, pp. 629–633.