

FairSense-AgentiX: An Agentic Platform for Multi-Modal Fairness and AI Risk Analysis

Aravind Narayanan^{1*}, Karanpal Sekhon^{1*}, Mahshid Alinoori¹, and Shaina Raza¹

¹ Vector Institute for Artificial Intelligence, Toronto, Ontario, Canada  Corresponding author *
These authors contributed equally.

DOI: [10.21105/joss.11060](https://doi.org/10.21105/joss.11060)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Wentao Ye](#) 

Reviewers:

- [@r0hin](#)
- [@lucky-verma](#)

Submitted: 08 May 2026

Published: 30 October 2026

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

FairSense-AgentiX is an open-source Python platform for automated detection of bias in text and images and for assessing AI deployment scenarios against established risk frameworks. Existing fairness tools such as AI Fairness 360 (Bellamy et al., 2019), Fairlearn (Weerts et al., 2023), and LangFair (Bouchard et al., 2025) address structured datasets or large language model (LLM) output metrics but cannot audit unstructured content across text and images, explain their reasoning transparently, or connect findings to deployment-level risk frameworks in a single workflow. To address this gap, it implements a multi-node agentic workflow using LangGraph (LangChain AI, 2024), unlike static classifiers or rule-based checkers, in which an orchestrator routes each request by declared modality, executes a specialised subgraph, and iteratively refines the output through a self-critique loop. The platform supports three analysis modes (text bias detection, image bias detection, and AI risk assessment) and exposes both a Python library and a FastAPI (Ramírez, 2019) REST/WebSocket server with an accompanying React web interface. FairSense-AgentiX is designed to be LLM-provider-agnostic and tool-swappable, making it applicable in a wide range of research and production contexts. It builds on and substantially extends a prior static pipeline, FairSense-AI (Raza et al., 2025), by replacing single-pass LLM inference with an orchestrated, self-correcting agentic loop and adding a retrieval-based risk assessment module aligned with established governance frameworks.

Statement of Need

Three concrete gaps motivate FairSense-AgentiX:

Multi-modal coverage. Real-world content includes job advertisements with embedded images, presentations combining visual and textual claims, and deployment documentation. A text-only auditing tool misses a significant portion of potential bias surface area.

Transparent reasoning. Compliance-oriented use cases (hiring audits, content moderation pipelines, ML model governance) require not just a verdict but an explanation: which exact text spans are problematic, why, and how confident the system is. Black-box classification cannot satisfy this need.

Connection to risk frameworks. Detecting bias is necessary but not sufficient; practitioners also need to understand how a deployment scenario maps to established taxonomies such as the NIST AI Risk Management Framework (AI RMF) (National Institute of Standards and Technology, 2023) and the MIT AI Risk Repository (Slattery et al., 2026). To our knowledge, no existing open-source tool combines bias detection with retrieval-based mapping of a deployment scenario onto such a taxonomy in one workflow.

FairSense-AgentiX addresses these gaps in a single, cohesive platform that is thoroughly tested and documented for external adoption.

State of the Field

Fairness auditing of AI systems and content remains an active area of research that lacks mature, end-to-end tooling (Blodgett et al., 2020; Mehrabi et al., 2021). Existing libraries such as AI Fairness 360 (Bellamy et al., 2019) and Fairlearn (Weerts et al., 2023) provide statistical metrics and post-hoc debiasing algorithms for structured datasets but do not address unstructured text or images, and are not oriented toward document-level natural-language bias explanations or deployment-level risk guidance. More recently, LangFair (Bouchard et al., 2025) introduced LLM-specific fairness metrics for text outputs, but remains restricted to evaluating LLM responses against predefined prompts and provides no image modality, no agentic reasoning, and no connection to deployment-level risk frameworks. Across these tools, existing approaches suffer from one or more of the following limitations: they are restricted to a single modality, they rely on fixed rule sets that cannot adapt to context, they treat analysis as a black-box step with no transparency into reasoning, or they provide no pathway from detected issues to actionable risk management guidance.

Why a new package rather than an extension. The gaps above are architectural rather than additive: AI Fairness 360 and Fairlearn are organised around tabular datasets and statistical metrics, with no abstraction for unstructured content, no orchestrator for multi-step reasoning, and no retrieval layer for external risk taxonomies. Adding these capabilities would require restructuring their core data model rather than extending it. LangFair is similarly scoped to LLM input/output pairs and prompt-level metrics, with no notion of auditing a document or image as a unit. FairSense-AgentiX is therefore best released as a separate package rather than as a fork or extension of any one of them; its typed, JSON-serialisable result objects are intended to make its findings straightforward to consume alongside these tools.

Software Design

Agentic Orchestration

The core of FairSense-AgentiX is an orchestrator built with LangGraph that implements a Reason–Act–Observe–Reflect (ReAct) loop (Yao et al., 2023). Figure 1 shows the full graph topology.

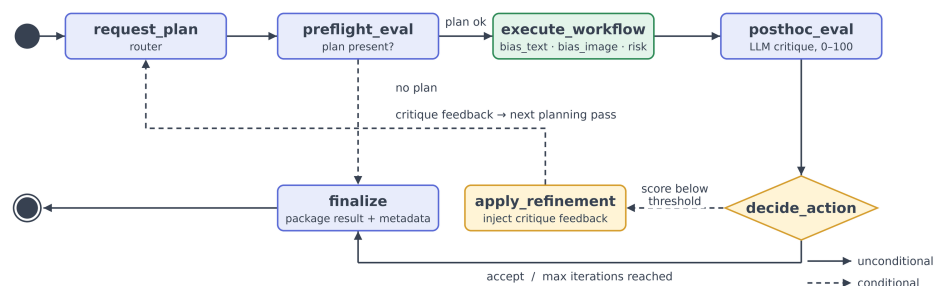


Figure 1: Orchestrator graph topology. Solid edges are unconditional; dashed edges are conditional routes taken when no plan is available, when the post-hoc quality score falls below threshold, or when refinement feedback is fed back into the next planning pass.

The request_plan node routes the input to a workflow based on its declared modality (text,

image, or scenario) and records a `SelectionPlan` carrying the chosen workflow, per-node parameters, a confidence value, and human-readable reasoning; routing is currently deterministic, and the plan structure is what the refinement loop later mutates. `preflight_eval` verifies that a plan exists before committing resources (it is otherwise a pass-through in the current release). `execute_workflow` dispatches to the selected specialised subgraph. `posthoc_eval` then submits the workflow output and the source text to an LLM critic that returns a 0–100 quality score, a justification, and concrete suggested changes. When the score falls below a configurable threshold (default 75) and refinement is enabled, `decide_action` routes to `apply_refinement`, which injects the critic’s suggestions into the analysis prompt as feedback and re-enters the loop; otherwise, or once the configured iteration budget (default one retry) is exhausted, control passes to `finalize`. This critique-driven refinement loop (Madaan et al., 2023) is the key architectural differentiator: it converts what would otherwise be a single-pass inference into a self-correcting process in which every intermediate decision is recorded in the result metadata.

Specialised Workflow Subgraphs

Three workflow subgraphs handle distinct analysis tasks.

Text bias graph. Accepts raw text, invokes an LLM with structured output to identify bias instances (type, severity, character-level span, explanation), optionally summarises long documents, and produces colour-coded HTML highlighting. Bias categories covered include gender, age, racial, disability, and socioeconomic bias (Blodgett et al., 2020; Mehrabi et al., 2021).

Image bias graph. Runs OCR (Tesseract (Smith, 2007) or PaddleOCR (PaddlePaddle Authors, 2020)) and image captioning (BLIP (Li et al., 2022) or BLIP-2 (Li et al., 2023)) in parallel, merges the resulting text, and feeds it into the same LLM-based analysis pipeline as the text workflow. A Vision-Language Model (VLM) variant uses the full image directly rather than the extracted-text intermediary.

Risk assessment graph. Embeds a natural-language deployment scenario using a sentence-transformer model (Reimers & Gurevych, 2019) and performs semantic nearest-neighbour search over a FAISS (Johnson et al., 2021) index of 1,340 risks from the MIT AI Risk Repository V3 snapshot dated 26 March 2025 (Slattery et al., 2026). Each matched risk is then linked to the four core functions of the NIST AI RMF (National Institute of Standards and Technology, 2023) (GOVERN, MAP, MEASURE, MANAGE) through a second FAISS index of 5,230 function-level mitigation actions. These actions are not drawn from the NIST text and carry no NIST control identifiers; they are authored by us as a library of 31 templated actions, assigned to each risk by its domain and sub-domain in the Repository taxonomy and then embedded for retrieval, so that every matched risk returns a prioritised set of RMF-function-aligned next steps. Results are exported as an HTML table and a CSV file. The mapping is transparent and editable: the generation script and the resulting CSV are versioned in the repository.

Tool Registry and Dependency Injection

All tools (OCR, captioning, summarisation, LLM, VLM, embedder, FAISS indexes, formatter, and persistence) are constructed by a central `ToolRegistry` from configuration and injected into each subgraph at build time. This design decouples graph logic from tool implementation and enables runtime substitution: the entire tool stack can be replaced with lightweight fakes for unit testing without modifying graph code.

LLM providers (OpenAI GPT-4 (OpenAI et al., 2023) and Anthropic Claude (Anthropic, 2024), via LangChain (Chase, 2022)) are selected through environment variables, with no code changes required to switch providers; the OpenAI client can also be pointed at any OpenAI-compatible endpoint, so locally hosted open-weight models (e.g., via Ollama or vLLM) are usable without an API key. A fake provider backs the test suite; results produced in this mode are labelled as

synthetic (`metadata.mock_mode`) and a warning is raised at construction so that placeholder output is never mistaken for a real analysis.

Service Layer

FairSense-AgentiX exposes a high-level FairSense Python class with three methods (`analyze_text`, `analyze_image`, and `assess_risk`) that return typed Pydantic result objects (`BiasResult`, `RiskResult`). These objects carry execution metadata including run ID, router confidence, refinement count, and quality scores, enabling fine-grained observability. The FastAPI service layer wraps this API and additionally streams agent telemetry events over WebSocket, so clients can observe each reasoning step as it occurs. A React/TypeScript frontend provides a landing page and interactive analysis interface with pre-built demo examples for each mode.

Quality Assurance

The package ships with 370+ unit and integration tests: unit tests for the tool layer (OCR, captioning, summarisation, formatting, persistence, embeddings, FAISS, caching, telemetry, configuration) and integration tests, run against a fake tool stack so that no API keys or model downloads are required, for each workflow subgraph, the orchestrator and its refinement loop, and the FastAPI service layer. A separate set of stress tests exercises degenerate inputs (empty, oversized, malformed) and a benchmarking script reports p50/p95/p99 latency and resident-memory deltas per workflow, exiting non-zero if any iteration fails or any result group is missing; both are run on demand rather than in continuous integration. Pre-commit hooks enforce type-checking (mypy) and linting (ruff), and GitHub Actions run code checks together with the unit and integration suites on every pull request.

Research Impact Statement

FairSense-AgentiX was developed under the AIXPERT project, a Horizon Europe initiative (Grant Agreement No. 101214389) tasked with building explainable, accountable, and transparent AI systems. Within this project, FairSense-AgentiX serves as the reference implementation of the agentic fairness-and-risk analysis layer, and is the basis for our ongoing empirical work on critique-driven refinement in fairness auditing.

The platform's research utility derives from three concrete design choices. First, it operationalises the MIT AI Risk Repository ([Slattery et al., 2026](#)), a peer-reviewed taxonomy of over 1,300 documented AI risks, as a queryable, FAISS-backed retrieval system, making the taxonomy directly usable in downstream studies rather than as a static document. Second, it bridges this risk taxonomy to the NIST AI Risk Management Framework ([National Institute of Standards and Technology, 2023](#)) by attaching function-level (GOVERN, MAP, MEASURE, MANAGE) mitigation actions to each retrieved risk, giving audit reports a consistent, traceable structure for next steps. Third, the LangGraph-based orchestrator exposes every intermediate reasoning step (the selected workflow, SelectionPlan contents, preflight and post-hoc evaluation scores, refinement-loop deltas), enabling reproducible analysis of agent behaviour, a prerequisite for empirical studies of agentic systems that black-box pipelines cannot support.

The package is distributed on PyPI as `fairsense-agentix` under the MIT License and is actively maintained on GitHub with continuous integration and versioned releases. To our knowledge, it is the first open-source, end-to-end auditing harness to combine multi-modal bias detection with retrieval-based risk mapping in a single agentic workflow, and we expect this infrastructure to underpin forthcoming empirical work on agentic fairness auditing.

Limitations

At present, FairSense-AgentiX supports five bias categories (gender, age, racial, disability, socioeconomic), operates over English-language prompts and risk taxonomies, and has been validated only with closed-source LLM providers (OpenAI GPT-4 ([OpenAI et al., 2023](#)) and Anthropic Claude ([Anthropic, 2024](#))); open-weight models are reachable through OpenAI-compatible endpoints but have not been evaluated. Workflow routing is deterministic on the declared input modality rather than LLM-inferred, and the NIST AI RMF actions attached to each risk are templated by risk domain rather than extracted from the framework text. Quantitative evaluation against standard fairness benchmarks is not included in this work and is planned as a companion study.

AI Usage Disclosure

No generative AI was used for architectural design or core algorithms. For documentation, docstrings, bug fixing, and inline code changes, we used Anthropic **Claude 3** models from the Opus/Sonnet/Haiku family ([Anthropic, 2024](#)), primarily **Sonnet**-class chat models. We also used **GitHub Copilot** inline completions in VS Code/Cursor; GitHub selects the completion model and we did not log model IDs per suggestion. For this manuscript, the same tools were used for drafting, copy-editing, and consistency checking. All AI-assisted code, documentation, and prose were reviewed, edited, and validated by the authors; factual claims were checked against the codebase and primary sources before submission.

Acknowledgements

Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute (<http://www.vectorinstitute.ai/#partners>). This research was funded by the European Union's Horizon Europe research and innovation programme under the AIXPERT project (Grant Agreement No. 101214389), which aims to develop an agentic, multi-layered, GenAI-powered framework for creating explainable, accountable, and transparent AI systems.

References

- Anthropic. (2024). *The Claude 3 model family: Opus, Sonnet, Haiku*. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.
- Bellamy, R. K. E., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilović, A., Nagar, S., Natesan Ramamurthy, K., Richards, J., Saha, D., Sattigeri, P., Singh, M., Varshney, K. R., & Zhang, Y. (2019). AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5), 4:1–4:15. <https://doi.org/10.1147/JRD.2019.2942287>
- Blodgett, S. L., Barocas, S., Daumé III, H., & Wallach, H. (2020). Language (technology) is power: A critical survey of “bias” in NLP. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 5454–5476. <https://doi.org/10.18653/v1/2020.acl-main.485>
- Bouchard, D., Chauhan, M. S., Skarbrevik, D., Bajaj, V., & Ahmad, Z. (2025). LangFair: A Python Package for assessing bias and fairness in large language model use cases. *Journal of Open Source Software*, 10(105), 7570. <https://doi.org/10.21105/joss.07570>
- Chase, H. (2022). *LangChain*. <https://github.com/langchain-ai/langchain>.

- Johnson, J., Douze, M., & Jégou, H. (2021). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 535–547. <https://doi.org/10.1109/TBDATA.2019.2921572>
- LangChain AI. (2024). *LangGraph: Build resilient language agents as graphs*. <https://github.com/langchain-ai/langgraph>.
- Li, J., Li, D., Savarese, S., & Hoi, S. (2023). BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *Proceedings of the International Conference on Machine Learning (ICML)*, 19730–19742.
- Li, J., Li, D., Xiong, C., & Hoi, S. (2022). BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *Proceedings of the International Conference on Machine Learning (ICML)*, 12888–12900.
- Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegrefe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B. P., Hermann, K., Welleck, S., Yazdanbakhsh, A., & Clark, P. (2023). Self-Refine: Iterative refinement with self-feedback. *Proceedings of the 37th International Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/075280-2019>
- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54(6), 1–35. <https://doi.org/10.1145/3457607>
- National Institute of Standards and Technology. (2023). *AI Risk Management Framework (AI RMF 1.0)* (NIST AI 100-1). U.S. Department of Commerce. <https://doi.org/10.6028/NIST.AI.100-1>
- OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., & others. (2023). GPT-4 technical report. *arXiv Preprint arXiv:2303.08774*. <https://doi.org/10.48550/arXiv.2303.08774>
- PaddlePaddle Authors. (2020). *PaddleOCR: Awesome multilingual OCR toolkits based on PaddlePaddle*. <https://github.com/PaddlePaddle/PaddleOCR>.
- Ramírez, S. (2019). *FastAPI*. <https://fastapi.tiangolo.com>.
- Raza, S., Chettiar, M. S., Yousefabad, M., Khan, T., & Lotif, M. (2025). *FairSense-AI: Responsible AI meets sustainability*. <https://arxiv.org/abs/2503.02865>
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992. <https://doi.org/10.18653/v1/D19-1410>
- Slattery, P., Saeri, A. K., Grundy, E. A. C., Graham, J., Noetel, M., Uuk, R., Dao, J., Pour, S., Casper, S., & Thompson, N. (2026). The AI Risk Repository: A meta-review, database, and taxonomy of risks from artificial intelligence. *Patterns*, 7(5), 101517. <https://doi.org/10.1016/j.patter.2026.101517>
- Smith, R. (2007). An overview of the Tesseract OCR engine. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2, 629–633. <https://doi.org/10.1109/ICDAR.2007.4376991>
- Weerts, H., Dudík, M., Edgar, R., Jalali, A., Lutz, R., & Madaio, M. (2023). Fairlearn: Assessing and improving fairness of AI systems. *Journal of Machine Learning Research*, 24(257), 1–8. <http://jmlr.org/papers/v24/23-0389.html>
- Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=WE_vluYUL-X