DRAFT

PECAN: Programming Encoder Classification Analysis Network

Lightweight Transformer Encoders for Programming Language Identification

Dr. James Ghawaly · Ibrahim Alam · Jackson Descant
AISX Lab · Louisiana State University
Stanford Research Conference 2026

Abstract

PECAN (Programming Encoder Classification Analysis Network) is a research project focused on advancing lightweight, accurate programming language identification using encoder-only neural models. The goal of this work is to design and evaluate efficient transformer-based encoders that classify programming languages from raw code snippets while maintaining strong generalization across diverse and modern codebases. Existing tools such as GuessLang and GitHub Linguist rely on heuristics or limited architectures, which often struggle with ambiguous snippets, mixed-language repositories, and scale. PECAN addresses these limitations by introducing a unified experimental framework that evaluates encoder-only models across a large and diverse corpus of programming languages with respect to accuracy, efficiency, and robustness. The study leverages a dataset of over 42 million code samples spanning more than 300 programming languages and compares multiple encoder model families under consistent evaluation criteria. The results provide a systematic benchmark for lightweight language identification models and offer insights into accuracy–efficiency tradeoffs for real-world deployment. This work contributes to software engineering, code analysis, and the development of scalable AI-assisted programming tools.

Motivation

- Existing systems struggle with ambiguous snippets and mixed-language repositories.
- Large-scale codebases demand scalable, efficient classifiers.
- Lightweight models enable production deployment without sacrificing accuracy.

42M+	319	99.5%
Code Samples	Languages	Accuracy

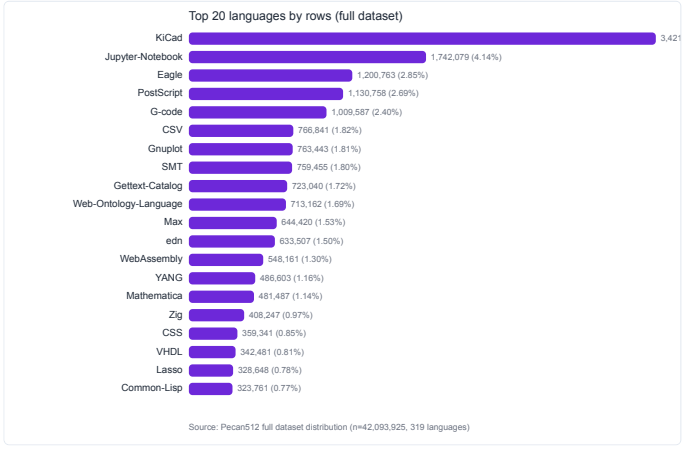
Approach

- Train encoder-only transformers on raw code snippets.
- Benchmark against GuessLang and GitHub Linguist baselines.
- Evaluate accuracy, efficiency, and robustness across model families and baselines.
- Run GuessLang in Docker; run Linguist, highlight.js, and GPT OSS inside the unified pipeline.
- Scale training with distributed multi-GPU infrastructure.

Stack: PyTorch, Hugging Face Transformers, W&B, CUDA.

Dataset

We leverage GuessLang and a larger custom dataset with 42 million+ code samples spanning 319 languages to capture syntax variability and real-world repository diversity.

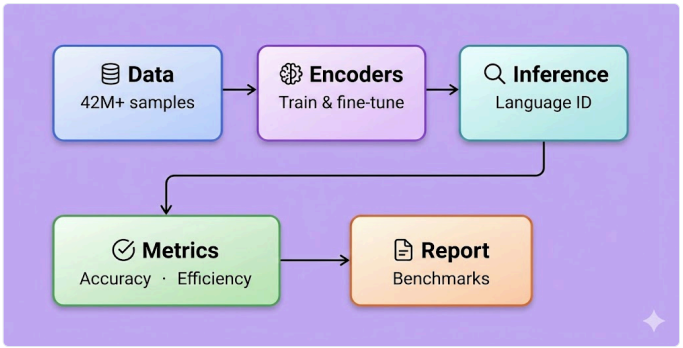


Source: Pecan512 full dataset distribution (provided counts).

Evaluation Pipeline

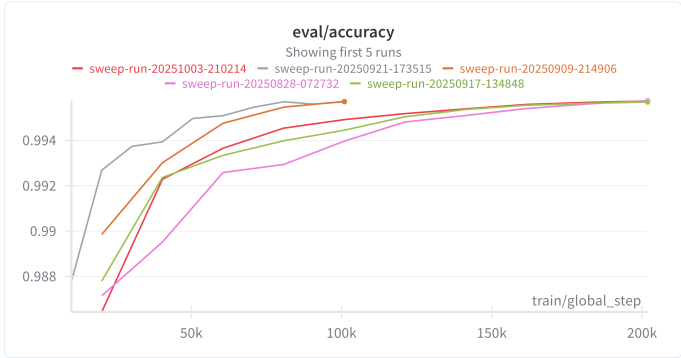
Evaluation routes depend on the model family. Fine-tuned encoder models are evaluated directly in the unified pipeline. GuessLang runs inside its Docker container, while GitHub Linguist is executed in the same pipeline (no Docker). We also evaluate with highlight.js and GPT OSS baselines for broader comparison.

- **Fine-tuned encoders:** Standard pipeline evaluation (accuracy/efficiency/robustness)
- **GuessLang:** Inference via Dockerized runtime
- **GitHub Linguist:** Pipeline integration without Docker
- **highlight.js / GPT OSS:** Additional baseline checks



Results & Contributions

- Established a systematic benchmark for lightweight language ID models.
- Quantified accuracy-efficiency tradeoffs for deployment.
- Produced research artifacts for open-source release and publication.



CodeBERT fine-tuning sweep: eval accuracy over training steps.

Impact & Applications

PECAN enables stronger language detection for IDE tooling, repository analytics, and AI-driven software analysis. The lightweight architecture supports deployment in real-world systems where latency and compute budgets are constrained.

- **IDE/Editor tooling:** Faster, more accurate language ID for mixed files and snippets.
- **Repository intelligence:** Better language composition analysis for large codebases.
- **Security & compliance:** Improved language-aware static analysis pipelines.
- **AI-assisted dev tools:** More reliable code search, tagging, and dataset curation.

Contact

Email: ialam2@lsu.edu

Website: https://ialam04.github.io

Placeholder: QR code to project page.