The Journal of Open Source Software

EchoFlow: End-to-end self-supervised sonar-image pipeline

Erling Devold [1]

1 SINTEF AS, Trondheim, Norway

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Neea Rusch

Reviewers:

- @subhk
- @nkrusch

Submitted: 10 June 2025
Published: unpublished

Summary

EchoFlow is a three-stage, containerised workflow that converts raw Kongsberg EK80 echosounder files into human-readable echograms *and* machine-interpretable attention maps.

1. **Conversion** – raw `.raw` pings are decoded and calibrated to volume-back-scattering strength with *pyEcholab* (Sullivan et al., 2018).

2. **Pre-processing** – echograms are contrast-stretched, down-sampled and tiled as PNGs.

3. **Inference** – a Vision Transformer trained with DINO (Caron et al., 2021) yields per-patch attention heat-maps that highlight fish schools, seabed returns and other salient structures.

Each stage is encapsulated in its own Docker image and orchestrated with Docker Compose; CI ensures that a test file always produces at least one attention map per echogram frequency. The test file can be inspected as artifacts from the CI actions.

Statement of need

Marine-acoustics research collect **terabytes** of multi-frequency sonar per survey but lack an open-source tool-chain that

- converts heterogeneous raw formats,
- scales from a laptop to an HPC cluster, and
- integrates state-of-the-art computer-vision models.

Previous work [Lee et al. (2024)](Sullivan et al., 2018) addresses the first bullet; EchoFlow fills the remaining gap by chaining **conversion → pre-processing → self-supervised inference** in a single, reproducible workflow. This lowers the barrier for fisheries scientists, marine-robotics engineers and citizen scientists who want modern ML without bespoke pipelines. This pipeline also serves as a foundation for new modern framework incorporated into this science.

Implementation and architecture

Each stage lives in its own Docker image and communicates through bind-mounted volumes (`./data/`). A Python watchdog triggers the pipeline when new `.raw` files arrive, and pre-trained DINO weights are cached on first use. Images are multi-platform (`linux/amd64`, `linux/arm64`).

Illustrative example

```
# 0 - fetch a sample EK80 file (NOAA public bucket)
aws s3 cp --no-sign-request \
  "s3://noaa-wcsd-pds/data/raw/Bell_M._Shimada/SH2306/EK80/Hake-D20230811-T165727.raw" \
  data/input

# 1 - run the full pipeline
docker compose up --build raw preprocessing infer

# 2 - open the resulting echogram and attention map
xdg-open data/preprocessing/Hake-D20230811-T165727/38000_debug.jpg
xdg-open data/inference/Hake-D20230811-T165727/70000.png
```

References

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9639–9650. https://doi.org/10.1109/ICCV48922.2021.00951

Lee, W.-J., Setiawan, L., Tuguinay, C., Mayorga, E., & Staneva, V. (2024). Interoperable and scalable echosounder data processing with echopype. *ICES Journal of Marine Science*, *81*(10), 1941–1951. https://doi.org/10.1093/icesjms/fsae133

Sullivan, J., Chu, D., & Lee, W.-J. (2018). PyEcholab: An open-source python-based toolkit to process and visualize echosounder data. *The Journal of the Acoustical Society of America*, *144*(3), 1778. https://doi.org/10.1121/1.5067860