Deep Learning for Automated Skin Cancer Detection

Adilet Akimshe,
Rakhim Baimurzin,
Lazzat Zhengisova,
Alisher-Polat Kurmanayev,
Yelzhas Omarov,
Aidana Orazbay,
Arailym Mukhametkali

Problem & Impact

We aim to build a system that maps dermoscopic and clinical skin images (plus optional metadata) to a diagnosis of malignant vs. benign lesion. Hypothesis: a convolutional neural network (CNN) trained on diverse, well-curated datasets can achieve dermatologist-level accuracy. This matters because early melanoma detection saves lives, but access to dermatologists is limited worldwide.

Related Work

TODO

Data

Primary source: ISIC Archive/HAM10000 (around 10k labeled dermoscopic images, CC-BY-NC). Optional additional clinical photos from other open repositories. Labels: benign/malignant and subtype. Planned split: 70/15/15 by patient ID. Preprocessing: resizing, color normalization; augmentation: flips, rotations, lighting variation.

Method

Baseline: pretrained ResNet-50 fine-tuned on ISIC. Loss: weighted cross-entropy (to counter class imbalance). Plan: hyperparameter search (learning rate, weight decay, augmentation strength) via grid/random search. Explore multimodal fusion of metadata and image features.

Evaluation

Metrics: ROC-AUC, PR-AUC, confusion matrix, and sensitivity at fixed specificity (90–95%). Baselines: random, majority-class, pretrained ResNet. Target: ROC-AUC > 0.90 on held-out test.

Resources & Reproducibility

Compute: student laptops or in the best case university GPU cluster (1–4 GPUs). Libraries: PyTorch, scikit-learn. Reproducibility: Git repo with fixed seeds, Docker/Conda env, dataset versioning on HuggingFace Datasets.

Risks & Ethics

Risks: (i) Class imbalance (mitigate with reweighting/augmentation); (ii) Patient-level leakage (mitigate by splitting by ID); (iii) Overfitting (early stopping, regularization). Ethics: avoid misuse as a diagnostic tool; dataset bias (under-representation of darker skin)

Timeline & Deliverables

Weeks 5–6: finalize data sources, preprocessing pipeline. Weeks 7–8: implement baseline CNN, establish repo. interim demo (baseline results). Weeks 9-10: advanced models + metadata fusion. Week 11: evaluation and error analysis. Week 12: final code, report, and poster submission.

References

https://github.com/proxy-pylon/medical-deep-learning-project