Deep Learning for Malignant Melanoma Detection on the ISIC 2020 Dataset

Anurag Yadav,6th of November,2025

Abstract
This project presents a deep learning solution for the classification of skin lesions as benign or malignant, utilizing the highly imbalanced ISIC 2020 Challenge dataset. Skin cancer, particularly melanoma, is one of the most dangerous forms of cancer, but early detection dramatically increases survival rates. The primary challenge of this dataset is its extreme class imbalance, with malignant cases representing only 1.8% of the training data. This makes standard accuracy a misleading metric. Our approach uses a PyTorch-based EfficientNetB0 model with a weighted loss function to prioritize the detection of the rare, malignant class. Our final trained model achieved a strong Validation AUC (Area Under the Curve) of 0.8523 and, most critically, a Malignant class Recall of 91%, demonstrating its effectiveness as a high-sensitivity screening tool.

1. Introduction

Skin cancer is the most common form of cancer, with malignant melanoma being the most lethal. The 5-year survival rate for melanoma, when detected early, is over 99%. However, if it metastasizes, the survival rate plummets to around 30%. This stark difference highlights the critical importance of early and accurate diagnosis.

While dermatologists use dermoscopy, visual inspection can be subjective. Artificial Intelligence, specifically Convolutional Neural Networks (CNNs), has shown immense promise in analyzing medical images to provide rapid, objective, and accurate classifications.

The goal of this project was to develop a robust deep learning model to classify dermoscopic images from the ISIC 2020 dataset, with a specific focus on overcoming the severe class imbalance to build a clinically useful tool.

2. Dataset and Preprocessing

We utilized the **ISIC 2020 Challenge Dataset**, a large public archive of 33,126 high-resolution dermoscopic images for training. Each image is associated with metadata, including a binary target: `0` (Benign) or `1` (Malignant).

2.2. The Challenge: Extreme Class Imbalance

The dataset's primary challenge is its severe imbalance:

- **Benign (0):** 32,542 images (98.2%)
- **Malignant (1):** 584 images (1.8%)

This imbalance makes "accuracy" a dangerous and useless metric. A model that simply predicts "Benign" for every image would achieve 98.2% accuracy while being 100% ineffective at its primary goal of detecting cancer.

2.3. Data Splitting

The 33,126 training images were split into a training set (80%) and a validation set (20%). This split was **stratified** to ensure that both sets retained the same 98%/2% class distribution, allowing for a realistic evaluation of the model's performance.

3. Methodology

To tackle the defined problem, we implemented a multi-stage pipeline within the PyTorch framework.

3.1. Model Architecture: Transfer Learning

We employed **Transfer Learning** using an **EfficientNetB0** model, pre-trained on the ImageNet dataset. This model was chosen for its high efficiency and top-tier performance. We used the `timm` library to load the model, froze the weights of all base convolutional layers, and replaced the final classifier with our own custom head:

1. A `Dropout` layer (p=0.4) to prevent overfitting.
2. A `Linear` layer (128 units, ReLU activation).
3. A final `Linear` layer (1 unit) to output a raw logit for classification.

Only this new custom head was trained, making the process fast and data-efficient.

3.2. Handling Class Imbalance: Weighted Loss

This was the most critical decision of the project. Instead of naive oversampling, we directly addressed the imbalance at the loss-function level. We used PyTorch's `BCEWithLogitsLoss` and provided it with a `pos_weight` argument.

This weight was calculated as `(Number of Negatives / Number of Positives)`, which was **~55.7**. This tells the model that a single misclassification of a Malignant case is **55.7 times more costly** than a misclassification of a Benign case. This forces the model to prioritize **Recall (Sensitivity)**, which is the correct strategy for a medical screening test.

3.3. Data Augmentation

To prevent overfitting and teach the model rotational/spatial invariance, we applied on-the-fly data augmentation to the training images:

- `RandomHorizontalFlip()`
- `RandomVerticalFlip()`
- `RandomRotation(degrees=20)`

3.4. Training & Optimization

The model was trained for 10 epochs. To accelerate this long process, we utilized **Automatic Mixed Precision (AMP)** (`torch.amp.autocast`), which leverages the T4 GPU's tensor cores for faster, lower-precision (float16) calculations.

- **Optimizer:** `Adam` (Learning Rate = 1e-3)
- **Scheduler:** `ReduceLROnPlateau` (to reduce LR if AUC stalled)
- **Stopping:** `EarlyStopping` (Patience = 5) and saving the best model based on `Val AUC`.

4. Results and Analysis

The model training was successful, completing 10 epochs. The best model was saved from **Epoch 8**, which achieved the peak **Validation AUC of 0.8523**.

4.1. Primary Metric: ROC Curve (AUC)

The **Receiver Operating Characteristic (ROC) Curve** is the best metric for binary classification on imbalanced data. It plots the True Positive Rate vs. the False Positive Rate. Our model's **AUC of 0.8523** signifies a strong ability to distinguish between the two classes—far better than the 0.5 baseline of random guessing.

4.2. Diagnostic Performance: Confusion Matrix & Recall

The **Classification Report** and **Confusion Matrix** provide the most critical insights into the model's diagnostic behavior.

Key Insights:

1. **High Recall (The Goal):** The model achieved a **Recall of 0.91** for the **Malignant (1)** class. This is the project's biggest success. It means that of all *actual* cancer cases in the validation set, **our model successfully identified 91% of them** (107 out of 117).

2. **Low Precision (The Trade-off):** This high recall came at the cost of a **Precision of 0.04**. This means that when the model predicted "Malignant," it was correct only 4% of the time (2,496 false alarms).
3. **Clinical Justification:** This trade-off is not a failure; it is the **correct clinical strategy**. In a real-world screening scenario, it is **infinitely preferable to have a false positive** (sending a healthy patient for a follow-up) than to have a **false negative** (missing a real cancer case). Our model is successfully optimized as a "high-sensitivity" tool.

4.3. Threshold Analysis

The "Precision & Recall vs. Threshold" plot confirms this strategy. It shows that achieving high precision is nearly impossible without sacrificing almost all recall. Our chosen threshold of 0.5 correctly prioritized high recall.

---

5. Conclusion

This project successfully demonstrates a complete pipeline for training a robust, clinically-useful skin cancer classification model on a highly imbalanced dataset. By prioritizing the **AUC** and **Recall** metrics over simple accuracy and implementing a **weighted loss function**, we developed a model that achieves a strong **0.8523 AUC** and a **91% recall** for malignant lesions.

The final model serves as an excellent "screening assistant," effectively flagging potential risks for further review by a dermatologist, which aligns perfectly with the goal of early detection to save lives.

6. References

1. **(The Dataset)** Rotemberg, V., Kurtansky, N., Betz-Stablein, B. *et al*. A patient-centric dataset of 10,000 clinical images for skin cancer screening and education. *Sci Data* 8, 39 (2021). https://doi.org/10.1038/s41597-021-00815-z
2. **(The Medical Statistics)** American Cancer Society. (2024). *Survival Rates for Melanoma Skin Cancer*. Retrieved from https://www.cancer.org/cancer/types/melanoma-skin-cancer/detection-diagnosis-staging/survival-rates-for-melanoma-skin-cancer-by-stage.html
3. **(The Model Architecture - Optional but good)** Tan, M., & Le, Q. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. Proceedings of the 36th International Conference on Machine Learning, PMLR 97:6105-6114.
4. **(The Software - Optional but good)** Paszke, A., *et al*. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Advances in Neural Information Processing Systems 32.