SIIM-ISIC MELANOMA CLASSIFICATION

Deep Learning Project Report

Identifying Melanoma in Lesion Images using Ensemble Deep Learning

Executive Summary

This project implements a comprehensive deep learning solution for melanoma classification using skin lesion images from the SIIM-ISIC 2020 competition. The final ensemble model achieved a ROC-AUC score of **0.9470** through sophisticated model architectures, extensive data augmentation, and advanced ensembling techniques.

The solution combines 30+ models spanning both PyTorch and TensorFlow implementations, utilizing state-of-the-art architectures including EfficientNet (B0-B6), ResNeSt, and ResNeXt. Key innovations include heavy test-time augmentation (20x), multi-checkpoint averaging, Stochastic Weight Averaging (SWA), and sophisticated preprocessing techniques.

Key Achievements

- **Final ROC-AUC Score: 0.9470**
- 30+ models ensemble (15 PyTorch + 15 TensorFlow)
- Multiple image resolutions (192×192 to 768×768 pixels)
- Heavy TTA with 20× augmentation during inference
- Advanced techniques: Cutmix, Coarse Dropout, SWA
- Multi-year dataset integration (2018, 2019, 2020)

1. Problem Statement

1.1 Objective

The primary objective is to develop an automated system to identify melanoma in dermoscopic images of skin lesions. Melanoma is the deadliest form of skin cancer, and early detection significantly improves survival rates. This binary classification task requires distinguishing between benign and malignant lesions with high accuracy.

1.2 Dataset

The project utilizes multiple datasets from the International Skin Imaging Collaboration (ISIC):

- **SIIM-ISIC 2020 Dataset:** Primary competition dataset
- **ISIC 2019 Dataset:** Additional training data for model generalization
- **ISIC 2018 Dataset:** Historical data for enhanced training
- **Malignant Dataset:** Supplementary malignant cases

The dataset presents significant class imbalance, with benign lesions far outnumbering malignant cases, requiring specialized sampling strategies.

1.3 Evaluation Metric

The competition uses **Area Under the ROC Curve (ROC-AUC)** as the primary evaluation metric. This metric is particularly suitable for imbalanced classification tasks as it evaluates model performance across all classification thresholds.

2. Technical Architecture

2.1 Model Architectures

The solution employs multiple state-of-the-art convolutional neural network architectures:

2.1.1 EfficientNet Family (TensorFlow)

- EfficientNet-B0 through B6 variants
- Utilized compound scaling for optimal depth, width, and resolution
- Multiple image sizes: 192×192, 256×256, 384×384, 512×512, 768×768
- Custom aspect ratios including 384×512 (H×W)
- Pre-trained on ImageNet with custom classification head

2.1.2 ResNeSt (PyTorch)

- ResNeSt-50 architecture with split attention mechanisms
- Primary image size: 384×384 pixels
- Enhanced feature extraction through channel-wise attention

2.1.3 ResNeXt (PyTorch)

- ResNeXt variants with aggregated residual transformations
- Multiple configurations across different image resolutions

2.2 Training Infrastructure

2.2.1 Hardware Acceleration

- **TPU (Tensor Processing Unit):** Used for TensorFlow models with 8 replicas
- **GPU:** Utilized for PyTorch implementations
- **Distributed Training:** Leveraged TPUStrategy and data parallelism

2.2.2 Data Pipeline

- TFRecord format for efficient data loading on TPU
- JPEG compression with varying resolutions
- Balanced sampling with positive ratio of 8%
- Prefetching and caching for optimal throughput

3. Data Augmentation & Preprocessing

3.1 Training-Time Augmentation

Extensive augmentation techniques were applied during training to improve model generalization:

- **Geometric Transformations:**
- • Random rotations (up to 180 degrees)
- • Horizontal and vertical flips (p=0.5)
- • Random sized crops (min 360×360)
- • Shear transformations
- • Zoom (height and width independently)
- • Shift transformations
- **Coarse Dropout (Cutout):**
- • 12 random holes per image
- • Hole size: 24×24 pixels maximum
- • Applied with probability 0.5
- • Helps prevent overfitting to specific image regions
- **Cutmix:**
- • Combines regions from different images
- • Label mixing proportional to area
- • Improves model robustness
- **Microscope Augmentation:**
- • Simulates microscope circle artifacts
- • Random circle masking at image center

3.2 Test-Time Augmentation (TTA)

Heavy TTA with 20-30 augmentations per image was one of the most significant contributors to final performance:

- Each test image processed 20-30 times with different augmentations
- Predictions averaged across all augmented versions
- Significantly improved model stability and confidence
- OOF AUC improved to 0.9142 with TTA

4. Training Strategy

4.1 Cross-Validation

- **5-Fold Stratified Cross-Validation**
- Ensures balanced class distribution across folds
- Patient-level splitting to prevent data leakage
- Different random seeds for diversity

4.2 Optimization

4.2.1 Optimizers

- **AdamW:** Primary optimizer with weight decay
- **Adam:** Alternative optimizer for some models
- Adaptive learning rates for different model components

4.2.2 Learning Rate Schedule

Custom learning rate schedule with three phases:

1. **Warm-up Phase:** Linear increase from 6e-6 to peak (5 epochs)
2. **Sustained Phase:** Maintain peak learning rate
3. **Decay Phase:** Exponential decay with factor 0.85

Peak learning rate scaled by batch size: 1.2e-6 × batch_size

4.3 Loss Functions

- **Binary Cross-Entropy (BCE):** Primary loss function
- **Label Smoothing:** Factor of 0.05-0.09 to prevent overconfidence
- **Focal Loss:** Experimental for addressing class imbalance

4.4 Stochastic Weight Averaging (SWA)

SWA was a critical technique for model stabilization:

- Averages model weights from last 3 epochs (N_SWA=3)
- Decay factor: 0.9
- Produces more generalizable models
- Reduces variance in predictions

4.5 Training Parameters

Parameter	Value
Batch Size	16 (per replica)
Epochs	15-25
Image Sizes	192-768 pixels
TTA Steps	20-30
Positive Class Ratio	8%

5. Advanced Techniques

5.1 Multi-Checkpoint Strategy

A novel approach to model stability:

- Save 5 checkpoints throughout training
- Average predictions from all 5 checkpoints
- Significantly stabilized predictions
- Reduced variance compared to single checkpoint

5.2 Multi-Resolution Training

Different models trained on varying image resolutions:

- 192×192: Fast training, captures coarse features
- 256×256: Balanced speed and detail
- 384×384: Primary resolution for most models
- 512×512: High detail, longer training
- 768×768: Maximum detail, resource intensive
- 384×512 (H×W): Non-square aspect ratio experiments

5.3 Dataset Integration

Incorporation of multiple years of ISIC data:

- **2020 Data:** Competition dataset (33,126 images)
- **2019 Data:** 25,331 additional images
- **2018 Data:** 10,015 historical cases
- **Malignant Dataset:** Focused malignant cases

Total positive cases increased from 581 to 5,670 when all datasets combined

6. Ensembling Strategy

6.1 Model Diversity

The final ensemble combines 30+ models with maximum diversity:

- **Framework Diversity:** 15 PyTorch models + 15 TensorFlow models
- **Architecture Diversity:** EfficientNet, ResNeSt, ResNeXt
- **Resolution Diversity:** Multiple image sizes
- **Data Diversity:** Different dataset combinations
- **Training Diversity:** Different random seeds and folds

6.2 Ensembling Techniques

6.2.1 Weighted Average

Primary ensembling method with model-specific weights:

- Weights determined by validation performance
- Higher-performing models receive greater weight
- Optimized through grid search on validation set

6.2.2 Power Average

Alternative ensembling approach:

- Raises predictions to a power before averaging
- Emphasizes confident predictions
- Helps in cases with high prediction variance

6.2.3 MinMax Ensemble

Experimental technique (did not provide improvement):

- Normalizes predictions to [0,1] range
- Did not yield performance gains in this competition

7. Results and Performance

7.1 Final Competition Score

Metric	Score
Final Competition ROC-AUC	**0.9470**
OOF AUC (with TTA)	**0.9142**

7.2 What Worked

Key factors contributing to the high performance:

4. **Heavy TTA (20-30×):** Single most impactful technique
5. **Multi-checkpoint averaging:** Stabilized predictions significantly
6. **Cutmix augmentation:** Improved model robustness
7. **Coarse Dropout:** Prevented overfitting to specific regions
8. **SWA:** Enhanced generalization
9. **Label Smoothing:** Reduced overconfidence
10. **BCE Loss:** Effective for binary classification
11. **Multiple optimizers:** AdamW and Adam provided complementary benefits
12. **Multi-year datasets:** Increased training data diversity
13. **Non-square image ratios:** Captured different perspectives

7.3 Model Performance by Architecture

Individual model families contributed as follows:

- **EfficientNet-B6:** Highest single-model performance
- **ResNeSt-50:** Strong PyTorch baseline
- **EfficientNet B0-B5:** Provided diversity in ensemble
- **ResNeXt:** Complementary feature extraction

8. Implementation Details

8.1 TensorFlow Implementation

- **Framework:** TensorFlow 2.x with Keras API
- **EfficientNet:** Using efficientnet.tfkeras package
- **TPU Strategy:** Distributed training across 8 TPU cores
- **Data Format:** TFRecord with JPEG compression
- **Callbacks:** Learning rate scheduler, model checkpointing

8.2 PyTorch Implementation

- **Framework:** PyTorch 1.x
- **ResNeSt:** Using official ResNeSt repository
- **Augmentations:** Albumentations library
- **Data Loading:** Custom Dataset with JPEG files
- **Mixed Precision:** FP16 training for faster computation
- **SWA:** Using torchcontrib.optim.SWA

8.3 Code Organization

- **EfficientNet.ipynb:** TF implementation with full pipeline
- **resnest50-fast.ipynb:** PyTorch ResNeSt implementation

- **Multi-CheckpointPyTorch.ipynb:** Multi-checkpoint strategy
- **IncredibleTPUtf.ipynb:** TPU-optimized TensorFlow training

9. Challenges and Solutions

9.1 Class Imbalance

Challenge: Melanoma cases represent only ~1.8% of the dataset

Solutions:

- Balanced sampling with 8% positive ratio
- Oversampling minority class
- Integration of additional malignant datasets
- ROC-AUC metric (insensitive to class distribution)

9.2 Computational Resources

Challenge: Training 30+ models with heavy TTA

Solutions:

- TPU utilization for TensorFlow models
- Efficient TFRecord data format
- Mixed precision training in PyTorch
- Time budget management (Kaggle 9-hour limit)

9.3 Overfitting Prevention

Challenge: Models tending to memorize training data

Solutions:

- Heavy augmentation during training
- Coarse dropout (Cutout)
- Label smoothing
- Stochastic Weight Averaging
- 5-fold cross-validation
- Early stopping based on validation performance

9.4 Patient-Level Splitting

Challenge: Preventing data leakage from same patient

Solution: Ensured all images from the same patient stayed in the same fold during cross-validation

10. Key Takeaways and Lessons Learned

10.1 Most Impactful Techniques

14. **Test-Time Augmentation:** The 20-30× TTA provided the largest single performance boost
15. **Model Diversity:** Combining PyTorch and TensorFlow models improved robustness
16. **Multi-Checkpoint Averaging:** Reduced variance without additional training
17. **Resolution Variety:** Different image sizes captured complementary features

10.2 What Didn't Work

- **MinMax Ensemble:** Did not provide improvement over weighted average
- **Focal Loss:** BCE with label smoothing performed better
- **Very Small Batch Sizes:** Increased training instability

10.3 Best Practices Identified

- Always use extensive TTA in medical imaging competitions
- Combine multiple frameworks for ensemble diversity
- Save multiple checkpoints and average their predictions
- Experiment with different image resolutions
- Use SWA for improved generalization
- Incorporate external datasets when available
- Patient-level splitting is crucial for medical data

11. Conclusion

This project successfully developed a high-performance melanoma classification system achieving a ROC-AUC score of 0.9470 through comprehensive deep learning techniques. The solution demonstrates that combining multiple proven strategies—extensive test-time augmentation, diverse model architectures, multi-checkpoint averaging, and sophisticated ensembling—can produce state-of-the-art results in medical image classification.

The key to success was not any single technique but rather the systematic integration of multiple complementary approaches. Heavy TTA provided stability, multi-checkpoint averaging reduced variance, diverse architectures captured different aspects of the data, and proper cross-validation ensured robust model selection.

This work contributes to the ongoing effort to leverage artificial intelligence for early melanoma detection, potentially saving lives through more accurate and accessible diagnostic tools. The techniques and insights gained from this project are applicable to broader medical imaging tasks and demonstrate the power of ensemble learning in high-stakes applications.