[A23] Deep Learning with python project : ISIC 2024 Challenge, Skin Cancer Classification Using Deep Learning, a Kaggle hosted competition

Team: Aurélie Maugein, Hugo Boileau, Brandt Olson

Introduction

Skin cancer is a significant global health concern, with melanoma being its deadliest form. Early detection is crucial for effective treatment and improved patient outcomes. The International Skin Imaging Collaboration (ISIC) has organized the 2024 challenge to develop advanced deep learning models for binary classification of skin lesion images. This study aims to create a robust and accurate diagnostic tool to assist in triaging potential skin cancer cases, particularly in settings without access to specialized dermatologic care.

1. Short Literature Review

Recent advancements in deep learning have revolutionized medical image analysis, especially in dermatology. Convolutional Neural Networks (CNNs) have shown remarkable performance in skin lesion classification tasks. Notable works include the use of EfficientNet architectures (Tan & Le, 2019) for their efficiency and scalability.

The concept of the "ugly duckling sign" in melanoma diagnosis (Grob & Bonerandi, 1998) suggests that outlier lesions on an individual are more likely to be melanoma. This contextual information has not been fully exploited in previous skin lesion classification algorithms, which often analyze lesions independently.

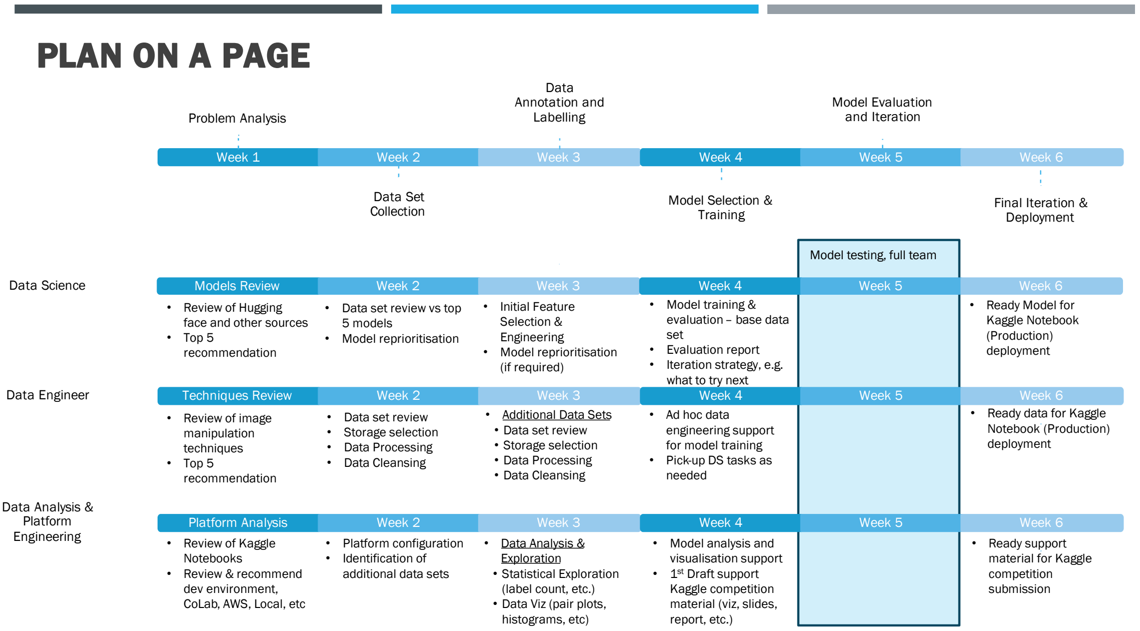
Transfer learning techniques have proven effective in medical imaging tasks (Esteva et al., 2021), allowing models pre-trained on large datasets to be fine-tuned for specific diagnostic purposes. Additionally, addressing class imbalance issues through techniques like oversampling has been shown to enhance model performance in binary classification tasks (Buda et al., 2018).

Variable selection using Random Forests (VSURF) has been demonstrated as an efficient way to identify key features from a large metadata set (Genuer, Poggi, & Tuleau-Malot, 2010).

Hyperparameter optimization utilizing Bald Eagle Search (BES) has been tested as a strategy for tackling the challenge of highly imbalanced data sets found in medical imaging, specifically skin cancer lesion detection (Sayed, Soliman & Hassanien, 2021).

2. Project delivery

2.1. Plan on a page



PLAN ON A PAGE

Problem Analysis — Week 1

Data Set Collection — Week 2

Data Annotation and Labelling — Week 3

Model Selection & Training — Week 4

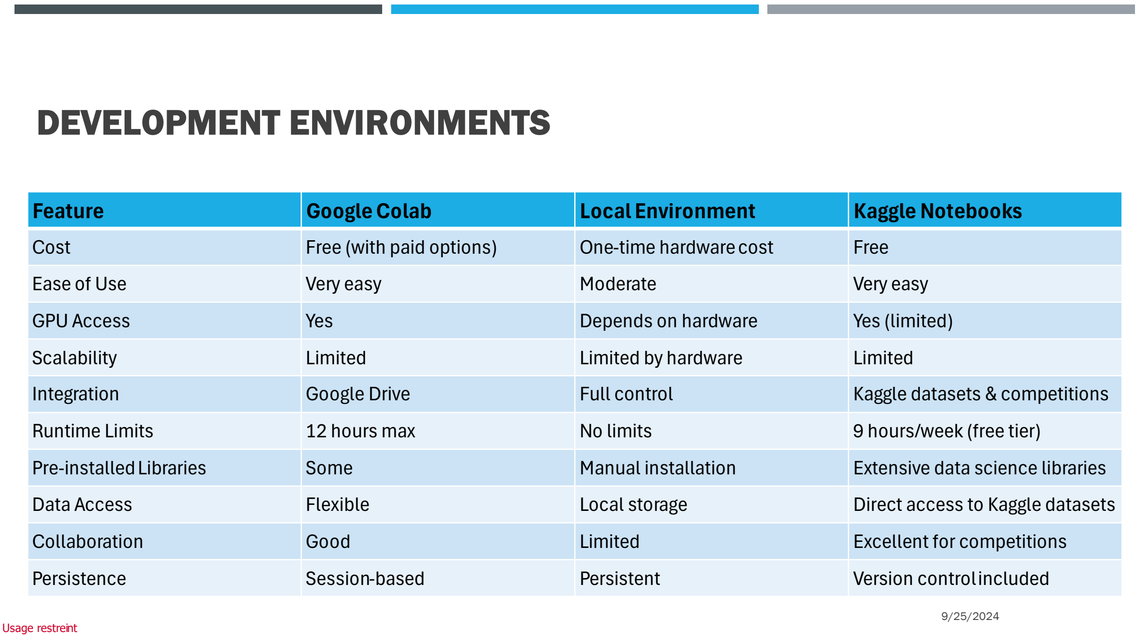
Model Evaluation and Iteration — Week 5

Final Iteration & Deployment — Week 6

Model testing, full team

Data Science	Models Review	Week 2	Week 3	Week 4	Week 5	Week 6
	• Review of Hugging face and other sources • Top 5 recommendation	• Data set review vs top 5 models • Model reprioritisation	• Initial Feature Selection & Engineering • Model reprioritisation (if required)	• Model training & evaluation – base data set • Evaluation report • Iteration strategy, e.g. what to try next		• Ready Model for Kaggle Notebook (Production) deployment
Data Engineer	Techniques Review	Week 2	Week 3	Week 4	Week 5	Week 6
	• Review of image manipulation techniques • Top 5 recommendation	• Data set review • Storage selection • Data Processing • Data Cleansing	• Additional Data Sets • Data set review • Storage selection • Data Processing • Data Cleansing	• Ad hoc data engineering support for model training • Pick-up DS tasks as needed		• Ready data for Kaggle Notebook (Production) deployment
Data Analysis & Platform Engineering	Platform Analysis	Week 2	Week 3	Week 4	Week 5	Week 6
	• Review of Kaggle Notebooks • Review & recommend dev environment, CoLab, AWS, Local, etc	• Platform configuration • Identification of additional data sets	• Data Analysis & Exploration • Statistical Exploration (label count, etc.) • Data Viz (pair plots, histograms, etc)	• Model analysis and visualisation support • 1st Draft support Kaggle competition material (viz, slides, report, etc.)		• Ready support material for Kaggle competition submission

2.2. Problem Analysis (samples, full analysis on Git repo, link below)



DEVELOPMENT ENVIRONMENTS

Feature	Google Colab	Local Environment	Kaggle Notebooks
Cost	Free (with paid options)	One-time hardware cost	Free
Ease of Use	Very easy	Moderate	Very easy
GPU Access	Yes	Depends on hardware	Yes (limited)
Scalability	Limited	Limited by hardware	Limited
Integration	Google Drive	Full control	Kaggle datasets & competitions
Runtime Limits	12 hours max	No limits	9 hours/week (free tier)
Pre-installed Libraries	Some	Manual installation	Extensive data science libraries
Data Access	Flexible	Local storage	Direct access to Kaggle datasets
Collaboration	Good	Limited	Excellent for competitions
Persistence	Session-based	Persistent	Version control included

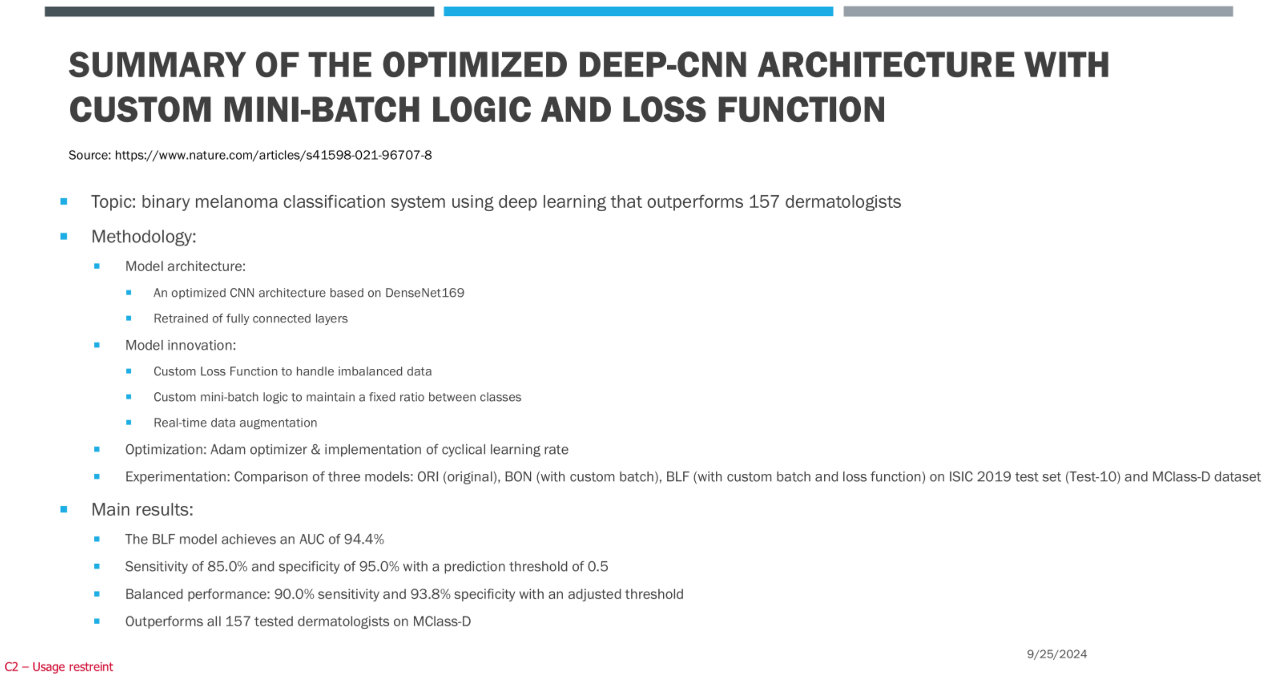
9/25/2024

MOST POWERFUL ALGORITHM COMPARISON FOR IMAGE CLASSIFICATION

	EfficientNet	DenseNet	ResNet	Inception-ResNet	Vision Transformer (ViT)
Definition	CNN architecture using compound scaling to balance network depth, width, and resolution	CNN with dense connectivity pattern between layers	Deep CNN using residual learning	Combines Inception architecture with residual connections	Applies transformer architecture to image patches
Characteristics	• Compound scaling • Mobile Inverted Bottleneck Convolution blocks • Squeeze-and-Excitation blocks	• Dense connectivity • Feature reuse • Compact architecture	• Residual learning • Skip connections • Very deep architecture	• Multi-scale feature extraction • Residual connections • Inception modules	• Patch-based image processing • Self-attention mechanism • No convolutions
Advantages	• Excellent efficiency - accuracy trade-off • Scalable architecture • Strong performance on various tasks	• Efficient parameter usage • Strong feature propagation • Mitigates vanishing gradient	• Enables training of very deep networks • Widely applicable • Well-understood architecture	• High accuracy • Multi-scale feature capture • Benefits of both Inception and ResNet	• Captures global dependencies • Scales well to large datasets • Potential for cross -modal applications
Disadvantages	• Complex architecture • May require large datasets for full benefit	• Memory intensive during training • Complex dense connections	• Very deep variants can be computationally expensive	• Computationally expensive • Complex architecture	• Requires large datasets for training from scratch • May struggle with small objects
Family Models	B0, B1, B2, B3, B4, B5, B6, B7, L2, V2 (S, M, L)	DenseNet-121, 169, 201, 264	ResNet-18, 34, 50, 101, 152, 50V2, 101V2, 152V2	Inception-ResNet-v1, Inception-ResNet-v2	ViT-Base, ViT-Large, ViT-Huge
Comparison of Variants	B0 (smallest) to B7 (largest) Each step increases accuracy and complexity V2 improves training speed and accuracy	Deeper variants (201, 264) offer higher accuracy but more parameters 121 often good balance of efficiency and performance	Deeper variants (101, 152) offer higher accuracy V2 variants improve training stability	v2 generally preferred over v1 for better performance	Larger variants (Large, Huge) offer higher accuracy but require more data and compute

5 SELECTED MODELS FOR SKIN CANCER DETECTION

	EfficientNet-B7	ResNet-152V2	DenseNet-201	Inception-ResNet-V2	Vision Transformer (ViT)
	Compound scaling method to balance network depth, width, and resolution. B7 variant is one of the largest and most powerful	Residual connections, allowing for very deep networks. The 152-layer version with V2 improvements is a powerful model for image classification	Connects each layer to every other layer in a feed-forward fashion, which helps mitigate the vanishing gradient problem and encourage feature reuse	Combines the Inception architecture, which uses multiple filter sizes in each layer, with residual connections from ResNet	Excellent performance on image classification tasks and represents a different paradigm in computer vision (not a traditional CNN)
Characteristics	• Optimized architecture for both accuracy and efficiency • Achieved state-of-the-art performance on ImageNet • Uses compound scaling to systematically scale network dimensions	• Very deep architecture with residual connections • Improved training stability for deep networks • Strong feature extraction capabilities	• Dense connectivity pattern between layers • Requires fewer parameters than traditional CNNs • Encourages feature reuse throughout the network	• Combines Inception modules with residual connections • Very deep and wide network • Efficient use of computational resources	• Treats image patches as tokens and applies self-attention • Inspired by transformer models in NLP • Can capture long-range dependencies in images
Advantages	• Excellent performance-to-parameter ratio • Can capture fine-grained details important in medical imaging • Scalable architecture allows for different model sizes	• Proven architecture in many computer vision tasks • Can capture complex hierarchical features • Good trade-off between depth and computational efficiency	• Efficient parameter usage • Strong gradient flow throughout the network • Can capture fine-grained features effectively	• High accuracy on image classification tasks • Can capture features at multiple scales simultaneously • Benefits from both Inception and ResNet architectures	• State-of-the-art performance on many vision tasks • Can potentially capture more global context than CNNs • Scales well with larger datasets and model sizes



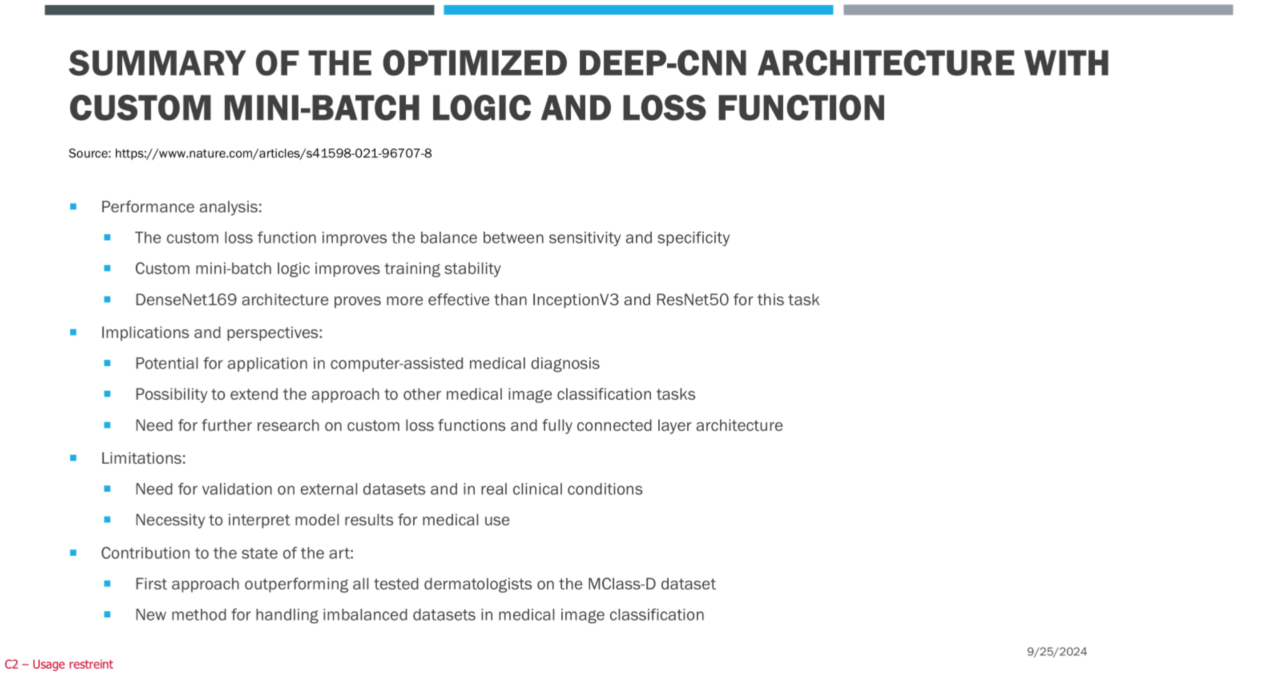
SUMMARY OF THE OPTIMIZED DEEP-CNN ARCHITECTURE WITH CUSTOM MINI-BATCH LOGIC AND LOSS FUNCTION

Source: https://www.nature.com/articles/s41598-021-96707-8

- Topic: binary melanoma classification system using deep learning that outperforms 157 dermatologists
- Methodology:
  - Model architecture:
    - An optimized CNN architecture based on DenseNet169
    - Retrained of fully connected layers
  - Model innovation:
    - Custom Loss Function to handle imbalanced data
    - Custom mini-batch logic to maintain a fixed ratio between classes
    - Real-time data augmentation
  - Optimization: Adam optimizer & implementation of cyclical learning rate
  - Experimentation: Comparison of three models: ORI (original), BON (with custom batch), BLF (with custom batch and loss function) on ISIC 2019 test set (Test-10) and MClass-D dataset
- Main results:
  - The BLF model achieves an AUC of 94.4%
  - Sensitivity of 85.0% and specificity of 95.0% with a prediction threshold of 0.5
  - Balanced performance: 90.0% sensitivity and 93.8% specificity with an adjusted threshold
  - Outperforms all 157 tested dermatologists on MClass-D

9/25/2024

C2 – Usage restreint



SUMMARY OF THE OPTIMIZED DEEP-CNN ARCHITECTURE WITH CUSTOM MINI-BATCH LOGIC AND LOSS FUNCTION

Source: https://www.nature.com/articles/s41598-021-96707-8

- Performance analysis:
  - The custom loss function improves the balance between sensitivity and specificity
  - Custom mini-batch logic improves training stability
  - DenseNet169 architecture proves more effective than InceptionV3 and ResNet50 for this task
- Implications and perspectives:
  - Potential for application in computer-assisted medical diagnosis
  - Possibility to extend the approach to other medical image classification tasks
  - Need for further research on custom loss functions and fully connected layer architecture
- Limitations:
  - Need for validation on external datasets and in real clinical conditions
  - Necessity to interpret model results for medical use
- Contribution to the state of the art:
  - First approach outperforming all tested dermatologists on the MClass-D dataset
  - New method for handling imbalanced datasets in medical image classification

9/25/2024

C2 – Usage restreint

3. Experiments

3.1. Data Distillation

Our experiment utilizes the SLICE-3D dataset provided by ISIC, which consists of standardized cropped lesion-images extracted from 3D Total Body Photography (TBP). These images mimic non-dermoscopic, close-up smartphone photos, making them particularly relevant for telehealth applications. The images are stored in an HDF5 file format, allowing for efficient storage and retrieval of a large number images.

3.2. Methodology

Our approach is based on the EfficientNet architecture, specifically EfficientNet-B0, known for its balanced trade-off between model size and performance. We've extended this architecture to incorporate additional clinical features, creating a hybrid model that leverages both image data and structured clinical information.

Key aspects of our methodology include:

3.2.1. Model Architecture

The model uses a modified EfficientNet architecture, enhanced to incorporate VSURF (Variable Selection Using Random Forests) features.

```
class EfficientNetWithVSURF(nn.Module):
    def __init__(self, model_name='efficientnet_b0', num_classes=1,
vsurf_size=5, dropout_rate=0.5):
        super(EfficientNetWithVSURF, self).__init__()
        self.base_model = timm.create_model(model_name, pretrained=True,
num_classes=num_classes)
        # ... (feature extraction and classifier setup)

    def forward(self, img_inputs, vsurf_features):
        img_features = self.base_model(img_inputs)
        combined_features = torch.cat([img_features, vsurf_features], dim=1)
        output = self.fc(combined_features)
        return output
```

1. Base Model: EfficientNet-B0
    - Architecture: It uses mobile inverted bottleneck convolution (MBConv) as its main building block, similar to MobileNetV2.
    - Efficiency: Designed to be computationally efficient while maintaining high accuracy.
    - Compound Scaling: The B0 model serves as the baseline for the EfficientNet family, which uses a compound scaling method to balance network depth, width, and resolution.
    - Performance: Despite its relatively small size, it achieves competitive accuracy on ImageNet classification.
    - Parameters: EfficientNet-B0 has about 5.3 million parameters.
    - Input Size: The standard input size for EfficientNet-B0 is 224x224 pixels.
    - Depth: It has 18 layers in total.
    - Transfer Learning: Due to its efficiency and performance, it's popular for transfer learning in various computer vision tasks, including medical imaging.
2. Feature Extraction:
    - Image features are extracted using the EfficientNet base
    - Variable Selection Using Random Forest (VSURF) features are concatenated with the image features: This allows the model to consider both image features and clinical metadata in its decision-making process, potentially capturing the "ugly duckling" phenomenon within a patient's lesion set.
    - Top 5 Features and their Importance: (only features available in both training and Kaggle model testing set are used)

i. clin_size_long_diam_mm (Importance: 38.60): Maximum diameter of the lesion (mm).+
ii. tbp_lv_H (Importance: 36.38): A inside lesion.+
iii. tbp_lv_deltaLBnorm (Importance: 34.09): Contrast between the lesion and its immediate surrounding skin
iv. tbp_lv_perimeterMM (Importance: 27.89): Perimeter of lesion (mm).+
v. tbp_lv_Hext (Importance: 22.65): Hue outside lesion.+

3. Classifier
    - A custom classifier processes the combined features
    - It consists of fully connected layers with ReLU activation and dropout

3.2.2. Data Augmentation

We employ a comprehensive set of augmentation techniques using the Albumentations library, with the following transformations:

- Random resized cropping (224x224 pixels)
- Horizontal flipping
- Random brightness and contrast adjustments
- Hue, saturation, and value shifts
- Gaussian noise addition
- Coarse dropout (for simulating occlusions).

This diverse augmentation strategy helps in simulating various imaging conditions and lesion presentations, enhancing the model's ability to generalize to the variable quality of telehealth-submitted images.

3.2.3. Optimization

We utilize the Bald Eagle Search (BES) algorithm for hyperparameter optimization. This nature-inspired metaheuristic approach allows us to efficiently search the hyperparameter space for optimal learning rate, batch size, and dropout rate.

3.2.4. Training Process

The model is trained using the Adam optimizer with a learning rate schedule (ReduceLROnPlateau). We implement mixed precision training using PyTorch's autocast and GradScaler for improved computational efficiency, which is crucial given the large dataset size.

3.2.5. Loss Function

Binary Cross-Entropy Loss is used as the optimization criterion, suitable for our binary classification task of differentiating benign from malignant cases.

3.2.6. Evaluation Metrics

As per the competition requirements, we use partial Area Under the ROC Curve (pAUC) above 80% true positive rate (TPR) as our primary evaluation metric. This focuses on the model's performance in the high-sensitivity region, which is crucial for medical screening tasks where false negatives are particularly costly.

3.3. Dataset

The dataset consists of 401060 skin lesion images along with associated metadata, representing every lesion from thousands of patients across nine institutions and three continents. The images are 15x15 mm field-of-view cropped photos from 3D Total Body Photography, captured between 2015 and 2024.

The training dataset is purposely highly imbalanced with only 393 of 401060 images being malignant, and purposely low resolution in effort to represent the balance and quality images captured by a smartphone and emailed/messaged to a trained machine learning model for evaluation. The training set includes both strongly-labelled tiles (histologically confirmed) and weak-labelled tiles (considered benign by a doctor without biopsy)

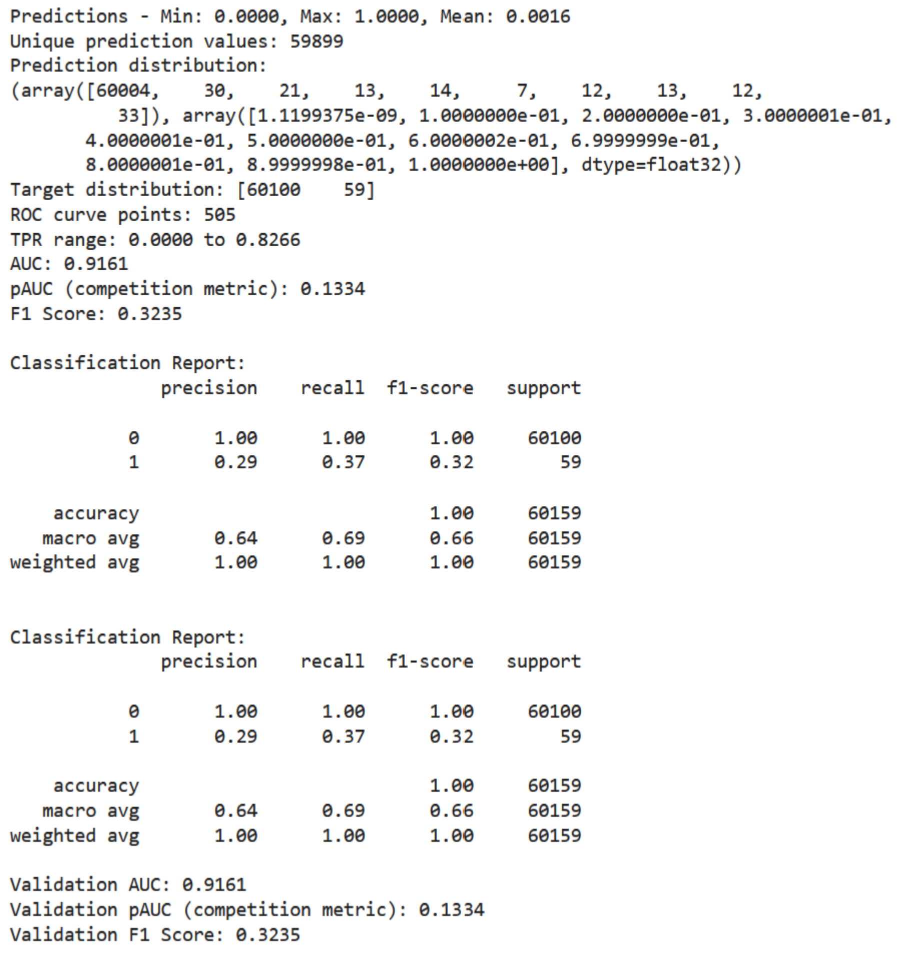
To train our model the data is split into training, test and validation sets, using balanced randomizer with oversampling to ensure malignant images are present throughout.

We utilized the provided metadata, which contains 55 potential input variables such as age, sex, and lesion size and location, with the most important features selected utilizing VSURF.

3.4. Reproducibility

To ensure reproducibility, we've implemented several measures including fixed random seeds, deterministic CUDA operations, checkpointing mechanisms, and comprehensive logging of hyperparameters, training progress, and evaluation metrics.

3.5. Model Analysis



```
Predictions - Min: 0.0000, Max: 1.0000, Mean: 0.0016
Unique prediction values: 59899
Prediction distribution:
(array([60004,    30,    21,    13,    14,     7,    12,    13,    12,
          33]), array([1.1199375e-09, 1.0000000e-01, 2.0000000e-01, 3.0000001e-01,
       4.0000001e-01, 5.0000000e-01, 6.0000002e-01, 6.9999999e-01,
       8.0000001e-01, 8.9999998e-01, 1.0000000e+00], dtype=float32))
Target distribution: [60100    59]
ROC curve points: 505
TPR range: 0.0000 to 0.8266
AUC: 0.9161
pAUC (competition metric): 0.1334
F1 Score: 0.3235

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     60100
           1       0.29      0.37      0.32        59

    accuracy                           1.00     60159
   macro avg       0.64      0.69      0.66     60159
weighted avg       1.00      1.00      1.00     60159


Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     60100
           1       0.29      0.37      0.32        59

    accuracy                           1.00     60159
   macro avg       0.64      0.69      0.66     60159
weighted avg       1.00      1.00      1.00     60159

Validation AUC: 0.9161
Validation pAUC (competition metric): 0.1334
Validation F1 Score: 0.3235
```

The model excels at identifying benign cases, which is expected given the class imbalance.

For malignant cases, the model struggled, correctly identifying 37% of malignant cases (recall). When it predicts malignant, it's correct 29% of the time (precision).

The high AUC (0.9161) suggests that the model ranks predictions well, but the default threshold may not be optimal, however, it was not adjusted as it was eventually used in a Kaggle competition which sets a default threshold.

The overall accuracy of 1.00 is misleading due to the extreme class imbalance.

The forecasted pAUC of 0.1334 score for the Kaggle competition, was encouraging, but ultimately disappointing with a score of .10695 achieved when tested against the hidden holdout set. The maximum competition score was .2, with .17264 achieved by the winner of the competition.

3.6. Alternative approaches

To address the very high imbalance a variety of options were explored such as using Synthetic Minority Oversampling Technique (SMOTE) which proved futile due to the very high similarity between benign and malignant skin lesions. Incorporating additional images was considered, however, competition discussions noted indicated that this did little to address the class imbalance.

A generative adversarial network (GAN), specifically a Cycle GAN, was also explored as an option for generating synthetic malignant images (Zunair & Hamza, 2020), though, ultimately time did not allow for this promising option to fully explored. The competition winner was revealed to have used Stable Diffusion 1.5 to generate synthetic images.

Other architectures such as ViT along with alternate EfficientNet models, such as B7, were also explored without success.

Reducing the number of hyperparameters to be tunned by BES did show some promise, with a SqueezeNet model achieving a late submission score of .12203. The model had been trained for 10 hours and, with more time available, it would be interesting to see how much further it could been improved.

3.7. Conclusion

Tackling such a highly imbalanced data set, 333 malignant images out of 401060, provide to be a very testing, but highly rewarding, challenge as the research required to address the challenge uncovered a range of techniques, such as BES, and models such as SqueezeNET, which we would have not been exposed to otherwise. Why we are disappointed that we did not score higher in the Kaggle competition, we do feel fully rewarded with the knowledge and techniques that we have learned as a result of entering the competition as part of [A23] Deep Learning with python project.

3.8. Links to competition and code

3.8.1. Kaggle, ISIC 2024- Skin Cancer Detection with 3D-TBP

https://www.kaggle.com/competitions/isic-2024-challenge/overview

3.8.2. Kaggle Notebooks

Competition Submission, https://www.kaggle.com/code/brandtolson/isic-sub-en0-bes-2?scriptVersionId=195630541

Late Submission, https://www.kaggle.com/code/brandtolson/isic-sub-sn-v1?scriptVersionId=196350145

3.8.3. Git Repository

https://github.com/Brandt-DSTI/Computer_Vision_ISIC_2024

3.8.4. Colab Notebooks

Model for competition submission, https://colab.research.google.com/github/Brandt-DSTI/Computer_Vision_ISIC_2024/blob/main/Copy_of_EN_B0_BES_v_03.ipynb

Model for late submission, https://colab.research.google.com/github/Brandt-DSTI/Computer_Vision_ISIC_2024/blob/main/Copy_of_SN_BES_v1.ipynb

3.9. References

1. J J Grob, J J Bonerandi, The 'ugly duckling' sign: identification of the common characteristics of nevi in an individual as a basis for melanoma screening, https://doi.org/10.1001/archderm.134.1.103-a
2. Mingxing Tan, Quoc V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, https://arxiv.org/abs/1905.11946v5
3. Andre Esteva, Katherine Chou, Serena Yeung, Nikhil Naik, Ali Madani, Ali Mottaghi, Yun Liu, Eric Topol, Jeff Dean, Richard Socher, Deep learning-enabled medical computer vision, https://doi.org/10.1038/s41746-020-00376-2
4. Robin Genuer, Jean-Michel Poggi, Christine Tuleau-Malot. Variable selection using Random Forests. Pattern Recognition Letters, 2010, 31 (14), pp.2225-2236. hal-00755489
5. Gehad Ismail Sayed, Mona M. Soliman, Aboul Ella Hassanien, A novel melanoma prediction model for imbalanced data using optimized SqueezeNet by bald eagle search optimization, https://doi.org/10.1016/j.compbiomed.2021.104712
6. Aarushi Shah, Manan Shah, Aum Pandya, Rajat Sushra, Ratnam Sushra, Manya Mehta, Keyur Patel, Kaushal Patel, A comprehensive study on skin cancer detection using artificial neural network (ANN) and convolutional neural network (CNN), https://doi.org/10.1016/j.ceh.2023.08.002
7. Tri-Cong Pham, Chi-Mai Luong, Van-Dung Hoang, Antoine Doucet, AI outperformed every dermatologist in dermoscopic melanoma diagnosis, using an optimized deep-CNN architecture with custom mini-batch logic and loss function, https://doi.org/10.1038/s41598-021-96707-8
8. Kazi Tanvir, Sharia Arfin Tanim, Md Sadi Al Huda, Akinul Islam Jony, Enhancing Early-Stage Detection of Melanoma using a Hybrid BiTDense, http://dx.doi.org/10.5281/zenodo.10049652#160

9. Kanchana, Kavitha, Anoop, Chinthamani B3, Enhancing Skin Cancer Classification Using Efficient Net, B0-B7 through Convolutional Neural Networks and Transfer, Learning with Patient-Specific Data, DOI:10.31557/APJCP.2024.25.5.1795
10. Hasib Zunair, A. Ben Hamza, Melanoma Detection using Adversarial Training and Deep Transfer Learning, https://arxiv.org/abs/2004.06824v2