Skin Cancer Detection with 3D Total Body Photos

Madelyn Esther Cruz
Maksim Kosmakov

THE ERDŐS INSTITUTE
Helping PhDs get and create jobs they love at every stage of their career.

Project Overview and Objectives

- **Objective:** Develop AI algorithms to **differentiate malignant from benign** skin lesions.
- **Inspiration:** ISIC2024 Kaggle Competition.

Context

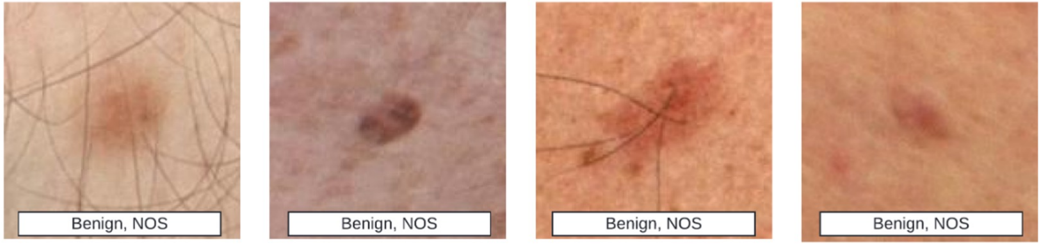
- **Problem: Skin cancer can be fatal if undiagnosed**; many underserved populations lack access to specialized dermatologic care.
- **Solution: AI algorithms to analyze lower-quality images**, similar to smartphone photos used in telehealth
- **Task: Create a binary classifier for skin cancer** using 3D total body photos (TBP) with single-lesion crops.
- **Benefit: Enhances triage and early detection of skin cancer**, especially in settings with limited specialized care.

Melanoma vs Benign Lesion Classification

Highly imbalanced large dataset**,** 401k Images in total, 1042 Patients

Melanoma (393 images, 0.01%)

Benign (400, 666 images, 99.99%)



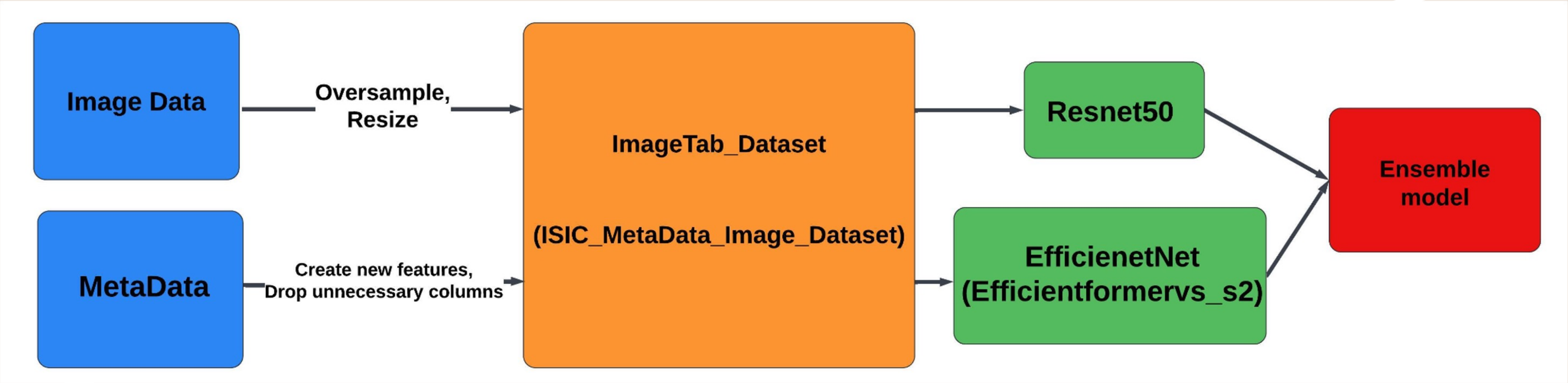
Exploratory Data Analysis (EDA)

- **Images (jpeg and hdf5):** Examined image sizes and reviewed examples of melanoma and benign images.
  - Images vary in size, with a broad range of dimensions:
    top 5: 133x133 - ~21k, 131x131 - ~21k, 129x129 - ~20k, 135x135 - ~20k, 137x137 - ~19k.
    Standard size chosen: 137x137.

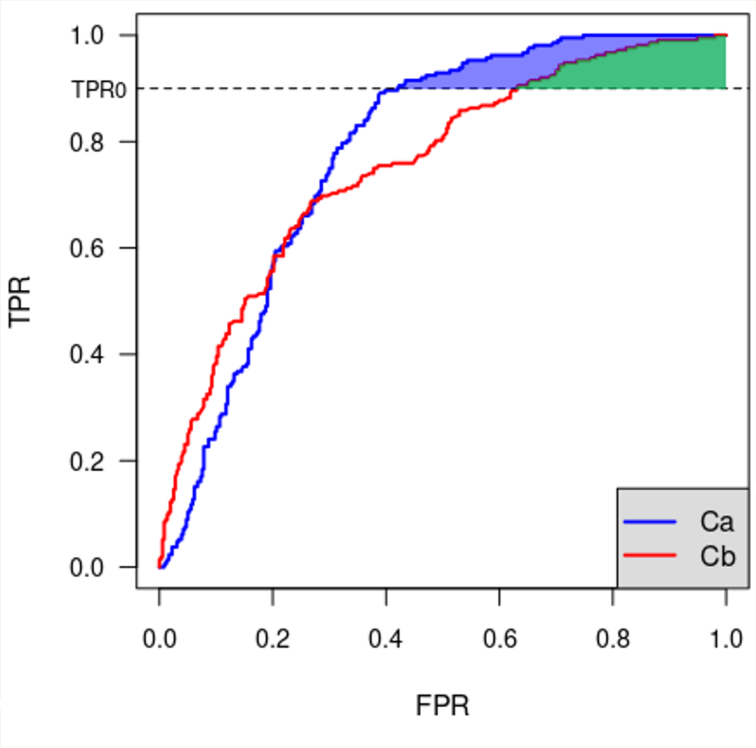
- **Meta Data (csv)**
  - 401059 Rows, 55 columns
  - **Missing Values:** 3k for age, 12k for sex, and 6k for anatomical site general.
    NAN values are replaced with the mode of the respective feature.
  - **Feature Types:** Includes both categorical and continuous variables.
- **Train-Test Split:** Stratified split ensures a balanced representation of both target classes in each set.
- **Feature Importance:** A Random Forest model computes feature importance scores.

Ensemble Model

Fastai | **PyTorch**



Evaluation



- Evaluated on partial area under ROC curve (pAUC) above 80% TPR.
- Other surrogate loss functions: Focal Loss, MSE Loss, Binary Cross Entropy Loss
- Results on 28% of hidden test set, which contains approximately 500k images

Results

Oversampling	Image model	Epochs	Loss function	Valid score*	pAUC
100k:10K	Resnet50 +Efficient_v2	0+5 0+5	CrossEntropy	0.0025	0.140
No	Resnet	0+1	CrossEntropy	0.0064	0.133
10k:1k	Resnet50	5+2	CrossEntropy	0.108	0.127
100k:10k	Resnet50	7+3	pAUC	0.962	0.018
100k:10k	Resnet50 +Efficient_v2	5+2 5+2	CrossEntropy	0.088	0.090
No	Pytorch ResNet50	22	FocalLoss	0.0043	0.126
No	Pytorch ResNet50 + EfficientNet (Ensemble)	22+11	FocalLoss	-	0.126

Conclusion

- The FastAI ImageTab model delivered the best performance.
- **Challenges:** High RAM usage for DataLoader (slow training and requires lots of memory), Possible overfitting
- **Advantages of our work:** We explored multiple approaches and leveraged various pretrained models, which saved time and yielded good results.
- **Future work:**
    - We plan to explore other ensemble models and techniques to improve image quality, such as hair removal.
    - Why does more training lead to a lower score, despite overfitting penalties?
    - Find more efficient ways to overcome the RAM requirements in the pytorch code.