Melanoma Detection Using SIIM-ISIC Dataset

Pukar Baral
Chandu Narasimhareddyvari
Eswar Naga Lakshman Chalamalasetty

Background of the problem

- **Skin cancer**:most predominant type of cancer
- The frequency of melanoma doubles every 20 years

Each year (in USA)

- About 100,640 new melanomas will be diagnosed (about 59,170 in men and 41,470 in women).
- About 8,290 people are expected to die of melanoma (about 5,430 men and 2,860 women).
- Melanoma is a deadly form of skin cancer,but survival rates are high if detected and diagnosed early

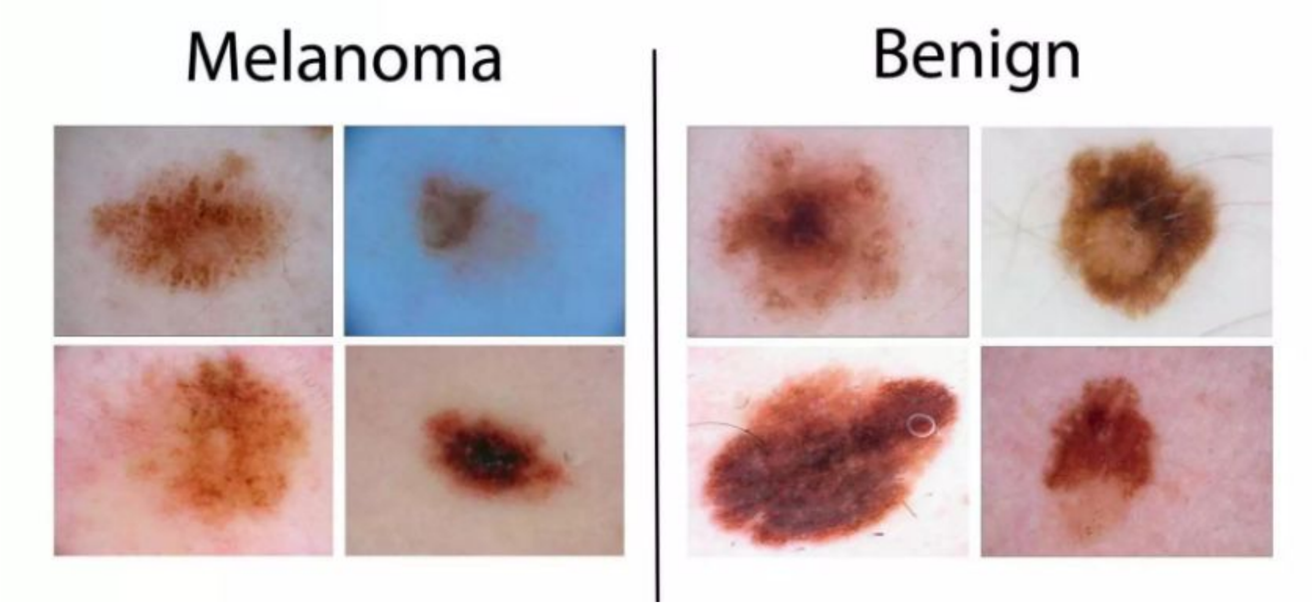
Melanoma detection :rely on handcrafted features

- ABCDE rule (Asymmetry,Border,Color,Dermoscopic structure and Evolving)
- CASH rule (Color,Architecture,Symmetry,and Homogeneity)

Background of the problem

Discriminating between benign and malignant skin lesions is challenging

Without computer based assistance :60-80% accuracy



Objective

Goals :

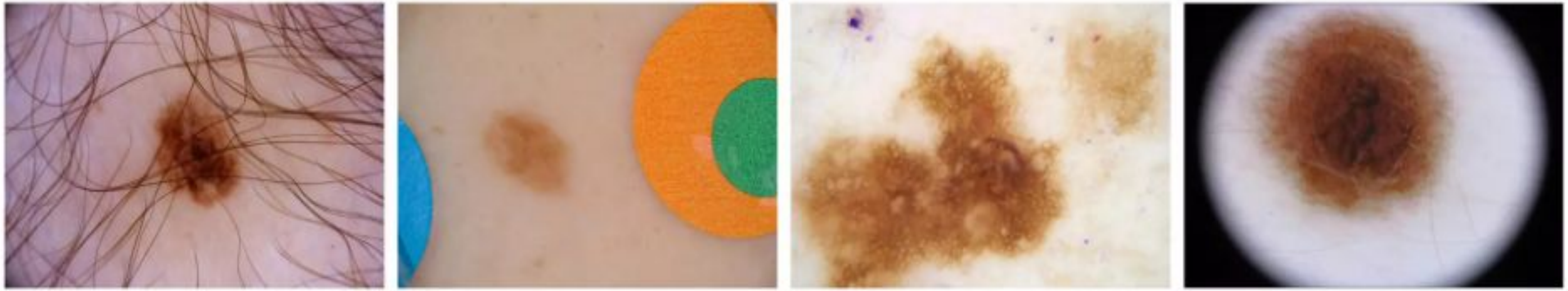
- Develop a deep learning model to classify Melanoma as malignant or benign.

- Evaluate the performance of the developed model.

Challenges

Dermoscopic images may

- Contain artifacts such as moles, freckles, hair, patches, shading, and noise.
- They may also present low contrast images between the lesion and the background
- Contain multiple skin lesions.

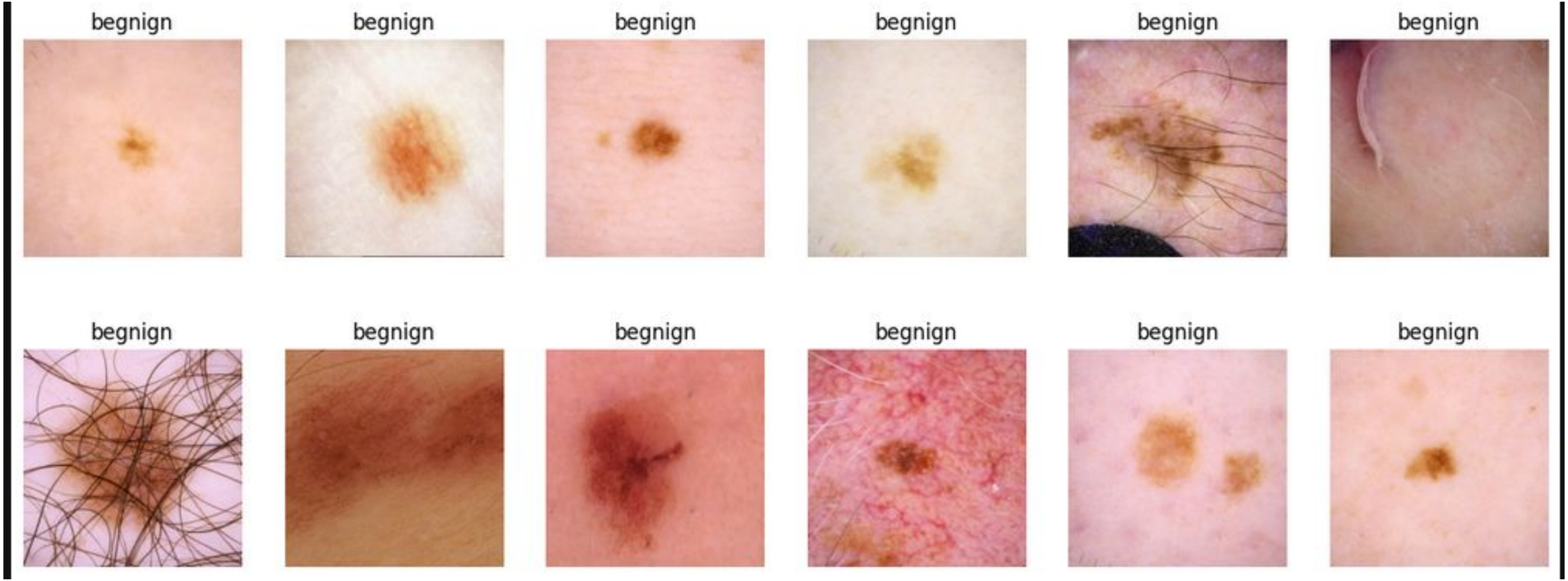


Dataset

- SIIM-ISIC melanoma classification challenge 2020 dataset.
- Skin lesion analysis towards melanoma detection.
- Train set contains 33126 images & features.
- Used this to create train and test set 80-20 split.

	Class		
	Benign	Malignant	Total Images
Training subset	25,571	459	26,030
Test Set	6,385	116	6,501

Dataset


begnign
begnign
begnign
begnign
begnign
begnign
begnign
begnign
begnign
begnign
begnign
begnign

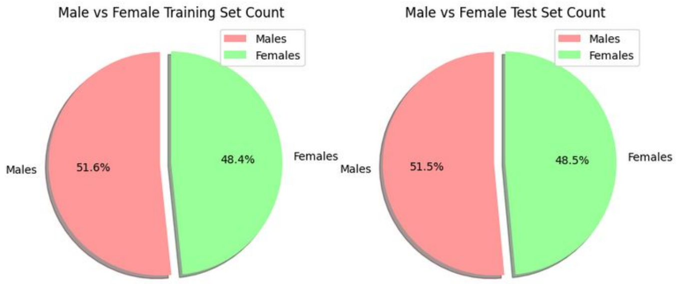
Dataset

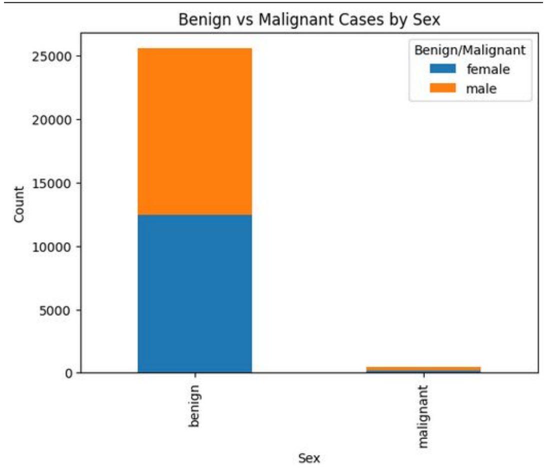
- Data also contains statistical features
- Sex, Age, Location of lesion.

	image_name	patient_id	sex	age_approx	anatom_site_general_challenge	diagnosis	benign_malignant	target
0	ISIC_8104064	IP_7207550	male	55.0	torso	unknown	benign	0
1	ISIC_6917587	IP_0894335	female	40.0	head/neck	unknown	benign	0
2	ISIC_3391651	IP_2842809	female	55.0	torso	unknown	benign	0
3	ISIC_4547675	IP_7279968	male	45.0	upper extremity	unknown	benign	0
4	ISIC_7289411	IP_5439716	male	70.0	torso	nevus	benign	0

Data Distribution

Visualizing Data Distributions





Preprocessing

Handling Missing Values

- Identified columns with missing values
- Dropped rows with missing values (since the dataset was large and missing values were less for malignant type)

Dropped Redundant columns

- *diagnosis* and *benign_malignant*
- *diagnosis* contains major unknown values
- *Benign_malignant* identical to *target column*

Encoding Variables

- Used label encoding to convert categorical variables to numerical labels

Normalizing Features

- Normalized numerical features to all features had the same scale

Data Augmentation

Train	Val	Test
RandomResizeCrop()		RandomResizeCrop()
ScaleShiftRotate()		ScaleShiftRotate()
HorizontalFlip()		HorizontalFlip()
VerticalFlip()		VerticalFlip()
Normalize()	Normalize()	Normalize()
ToTensor()	ToTensor()	ToTensor()

Model Overview

- The Classifier is a PyTorch implementation of a neural network model for melanoma image classification.
- The CNN feature extractor is a EfficientNet B2 model initialized with ImageNet weights, and the FNN processes the non-image features.
- The final classification is made by concatenating the features from the CNN and FNN, and passing them through a classifier.

Model Development

Efficient-Net B2

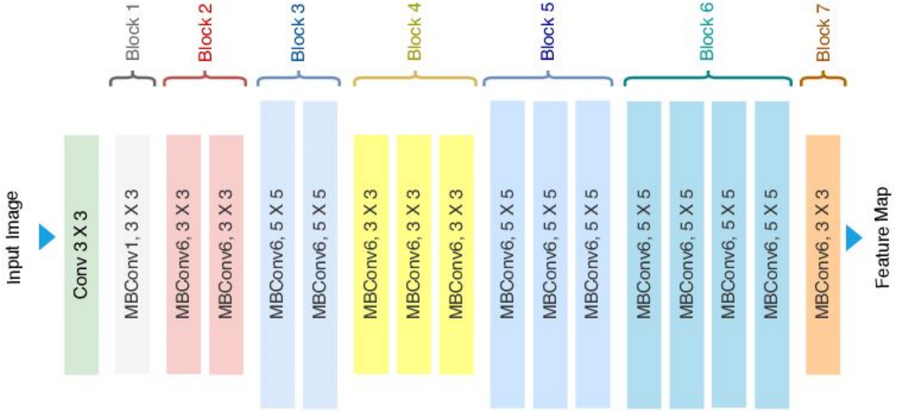
- Based on convolutional neural network architecture designed for image classification tasks.
- Achieves high accuracy while being efficient in terms of computational resources.
- Extracts image features.

FNN

- Takes meta data as input
- Outputs 250 features

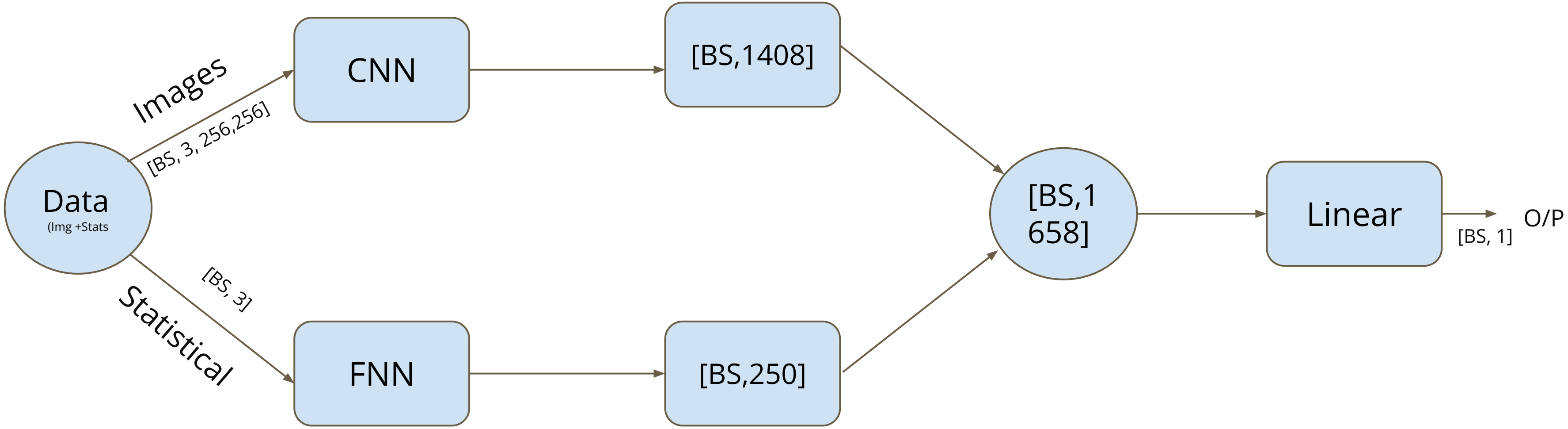
Classification head

- Combine above two features (1408 + 250)
- Feed into classification head
- Linear Layer (in =1408+250, out=1)



Architecture of Efficient-Net (Source)

Architecture Overview



BS: Batch Size
O/P:Malignant\Benign (probs)

Model Training

- Uses GroupKFold to train and evaluate the model with data grouped based on patientID.
- For each fold:
  - Get train and validation data.
  - Initialize model, optimizer, schedular and loss function.
  - Train and evaluate the model, get out of fold predictions.
  - Use early stopping to prevent overfitting.
  - Use the model with best ROC from fold on test data to calculate evaluation metrics.
  - Used test time augmentation in the test set.

Model Training

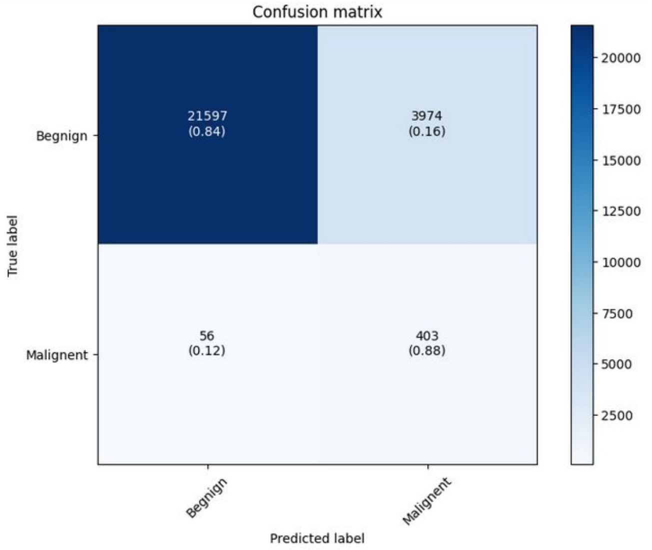
- Epochs: 30
- K: 6
- Learning Rate: 0.0001
- Batch Size: 32
- Optimizer: Adam
- Scheduler: ReduceLROnPlateau()
- Loss function: BCEWithLogitsLoss()

Performance Metrics

- The performance metrics (accuracy, sensitivity, and specificity) for each fold are stored in the results list.
- The out of fold predictions are used to evaluate model along with test data.
- Mean accuracy of around 0.7536 for the test data.

Fold	Accuracy	True Positive	True Negative	False Positive	False Negative	Sensitivity	Specificity
1	0.722350	75	4621	1764	41	0.646552	0.723727
2	0.644362	90	4099	2286	26	0.775862	0.641973
3	0.776188	76	4970	1415	40	0.655172	0.778387
4	0.800954	71	5136	1249	45	0.612069	0.804385
5	0.794801	67	5100	1285	49	0.577586	0.798747
6	0.783264	79	5013	1372	37	0.681034	0.785121

Results



Confusion Matrix for Out of fold prediction

	precision	recall	f1-score	support
0	1.00	0.84	0.91	25571
1	0.09	0.88	0.17	459
accuracy			0.85	26030
macro avg	0.54	0.86	0.54	26030
weighted avg	0.98	0.85	0.90	26030

ROC: 0.933

THANK YOU