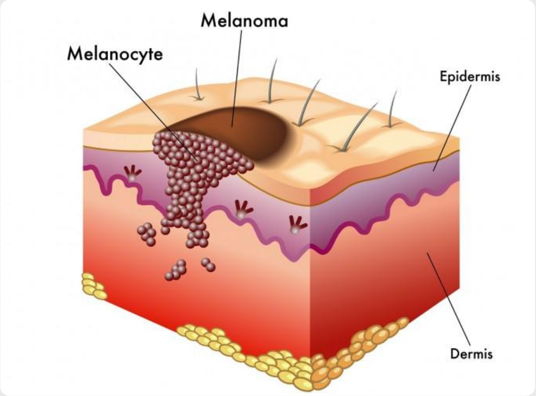
Convolutinal Neural Network for Skin Cancer Detection and Classification

Md Naim Hassan Saykat
ID: 2018511460042

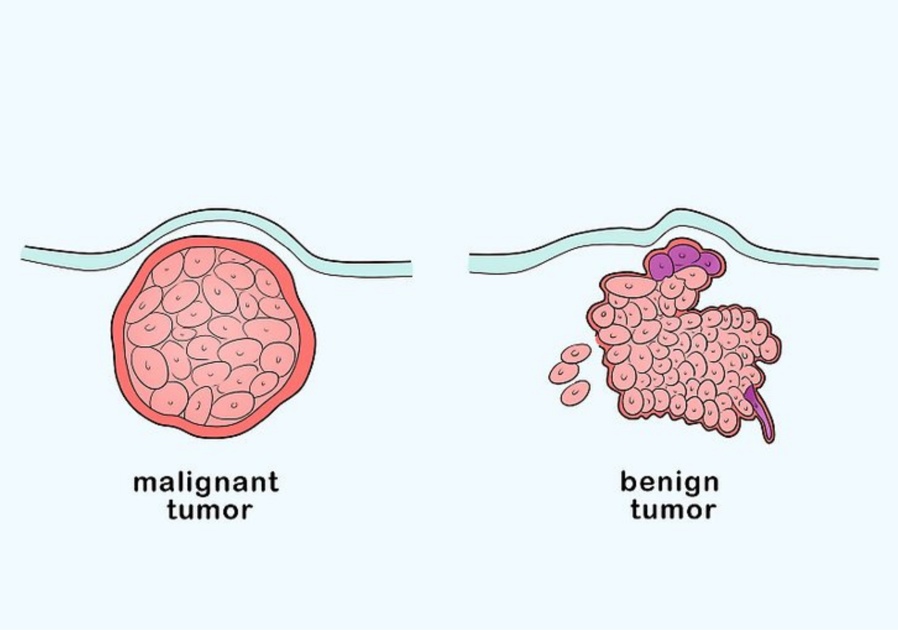
OVERVIEW

SKIN CANCER



Skin Cancer is one of the most deathful of all the cancers. It is bound to spread to different parts of the body on the off chance that it is not analyzed and treated at the beginning time. It is mostly because of the abnormal growth of skin cells, often develops when the body is exposed to sunlight.

MALIGNANT BENIGN



BENIGN

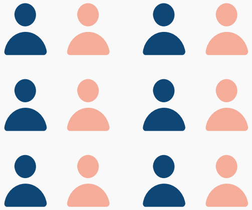
A growth that is not cancer. It does not invade nearby tissue or spread to other parts of the body.

MALIGNANT

A malignant neoplasm is a cancerous tumor, an abnormal growth that can grow uncontrolled and spread to other parts of the body.

The Risk

- **1 in 5 Americans** will develop skin cancer by the age of 70.
- **More than 2 people** die of skin cancer in the U.S. every hour.
- Having **5 or more sunburns** doubles your risk for melanoma.
- When detected early, the **5-year survival rate for melanoma is 99 percent.**



3,000,000

3 million non-melanoma skin cancers and 132,000 melanoma skin cancers occur globally each year.

DATASET

Dataset

MNIST HAM-10000 dataset for Skin Cancer which is available on Kaggle was used in this study. It consists of 10,015 images of skin pigments and 7 features. The dermatoscopic images are divided amongst seven classes. The number of images present in the dataset is enough to be used for different tasks including image retrieval, segmentation, feature extraction, deep learning, and transfer learning, etc. The various features of this dataset are: Lesion id, Image id, Dx, Dx_type, Age, Sex, and Localization.

Loading Metadata

- ✓ Initially the metadata which is proved as a .csv file was loaded as pandas dataframe
- ✓ The null values in the data were imputed. Then the given ‘dx’ column which provides the type of cancer was mapped to integer values
- ✓ This column will be the target column of our neural network
- ✓ Then complete image paths were loaded using the Image id column
- ✓ Using these paths images were loaded.


Actinic keratoses (akiec)
Actinic keratoses (akiec)
Basal cell carcinoma (bcc)
Basal cell carcinoma (bcc)
Benign keratosis-like lesions (bkl)
Benign keratosis-like lesions (bkl)
Dermatofibroma (df)
Dermatofibroma (df)
Melanoma (mel)
Melanoma (mel)
Melanocytic nevi (nv)
Melanocytic nevi (nv)
Vascular lesions (vasc)
Vascular lesions (vasc)

Environment And IDE Tools

- Nvidia K80 16GB GPU
- Python

MAIN PROJECT

STEPS For Model Building And Evaluation

- Step 1 : Importing Essential Libraries
- Step 2: Loading Metadata
- Step 3: Loading and Preprocessing Images
- Step 4: Model Architecture
- Step 5: Training the model
- Step 6: Evaluating the model
- Step 7: Training and evaluation of pre-trained models
- Step 8: Performance Comparison

MAIN LIBRARY

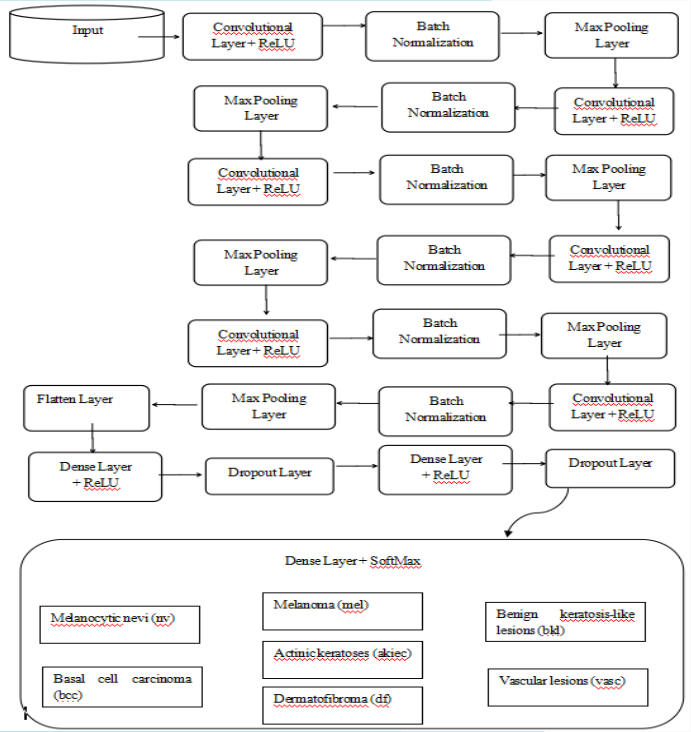
- ✓ Keras
- ✓ sklearn

OTHER SELECTED LIBRARIES

- ✓ numpy
- ✓ pandas
- ✓ os
- ✓ io
- ✓ seaborn
- ✓ PIL
- ✓ glob

Model Architecture

- ✓ The convolutional neural network model proposed in this study includes: Convolution, Rectified Linear Unit (ReLU), Pooling, Flattening, fully connected layers, and Softmax function
- ✓ At the initial step, several convolution filters are applied to the input image to determine features from the images
- ✓ The pooling step aims to simplify the output by performing nonlinear downsampling and decreasing the number of parameters that the network needs to train
- ✓ To develop the training speed, negative values are mapped to zero, and positive values remain unchanged in the ReLU step
- ✓ In the flattening step, all two-dimensional arrays are modified into one single linear vector
- ✓ This process is needed for fully connected layers to be used after convolutional layers
- ✓ Fully connected layers can combine the entire local features of the previous convolutional layers
- ✓ The procedure is completed with the application of the softmax activation function to provide the final classification output.



CNN MODEL BUILDING

- ✓ The CNN model takes 224x224x3 images as input and passes them through six sets of Convolution2D
- ✓ The second important layer in Normalization and MaxPooling2D layers
- ✓ In this process the images get converted into different dimensions and after the sixth MaxPooling layer the dimension of the image becomes 3x3x512
- ✓ A Flatten layer is then used to convert the three-dimensional image to one dimensional vector
- ✓ Then this vector goes through two Dense-Dropout layer blocks and finally gets classified at the last Dense layer
- ✓ The total number of parameters generated in this process is 16,717,063 where 16,714,375 parameters are trainable and 2,688 parameters are non-trainable.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 64)	1792
batch_normalization (BatchNo	(None, 224, 224, 64)	256
max_pooling2d (MaxPooling2D)	(None, 112, 112, 64)	0
conv2d_1 (Conv2D)	(None, 112, 112, 128)	73856
batch_normalization_1 (Batch	(None, 112, 112, 128)	512
max_pooling2d_1 (MaxPooling2	(None, 56, 56, 128)	0
conv2d_2 (Conv2D)	(None, 56, 56, 128)	147584
batch_normalization_2 (Batch	(None, 56, 56, 128)	512
max_pooling2d_2 (MaxPooling2	(None, 28, 28, 128)	0
conv2d_3 (Conv2D)	(None, 28, 28, 256)	1605888
batch_normalization_3 (Batch	(None, 28, 28, 256)	1024
max_pooling2d_3 (MaxPooling2	(None, 14, 14, 256)	0
conv2d_4 (Conv2D)	(None, 14, 14, 256)	3211520
batch_normalization_4 (Batch	(None, 14, 14, 256)	1024
max_pooling2d_4 (MaxPooling2	(None, 7, 7, 256)	0
conv2d_5 (Conv2D)	(None, 7, 7, 512)	6423040
batch_normalization_5 (Batch	(None, 7, 7, 512)	2048
max_pooling2d_5 (MaxPooling2	(None, 3, 3, 512)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 1024)	4719616
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 512)	524800
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 7)	3591

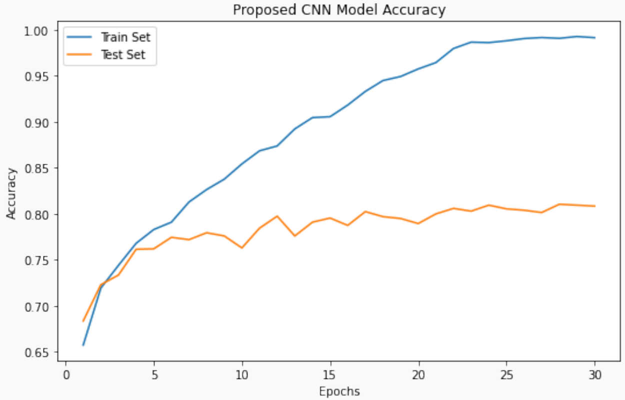
Total params: 16,717,063
Trainable params: 16,714,375
Non-trainable params: 2,688

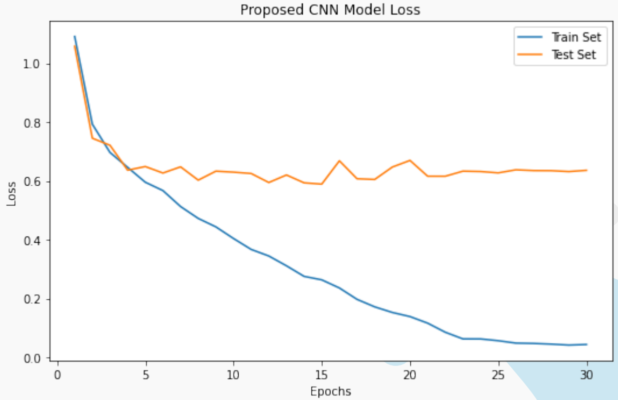
CNN MODEL TRAINING

- ✓ The model was trained in 30 epochs.
- ✓ Categorical Cross-entropy was used to calculate the loss. Stochastic Gradient Descent (SGD) with a learning rate of 0.001 was used to optimize the loss
- ✓ A batch size of 32, steps per epoch of 5712//32 ≈ 178 and validation steps of 1311//32 ≈ 41 was used
- ✓ The model took nearly 1 hour to be trained with a Nvidia K80 16GB GPU
- ✓ The number of epochs of 30 has been standardized for this model because the learning rate degrades if the model is fed with too many iterations. Callback functions
- ✓ Early Stopping and Reducing Learning Rate on Plateau were also used to reduce overfitting
- ✓ Model checkpoints was also used to save the best model while training
- ✓ A gradual dropping of the validation accuracy of the model during the 17th and 24th epochs has been noticed, which is not at all a good sign
- ✓ the model loses its effectiveness because of the over-fitting of the data

CNN MODEL EVALUATION

- ✓ The best performing model achieved an accuracy of 99% on training set and 81% on test set
- ✓ The categorical loss for training set was 0.0435 and for test set it was 0.6314





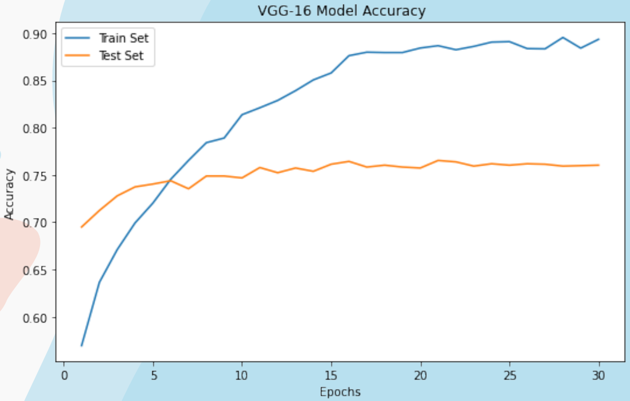
VGG-16 training and evaluation

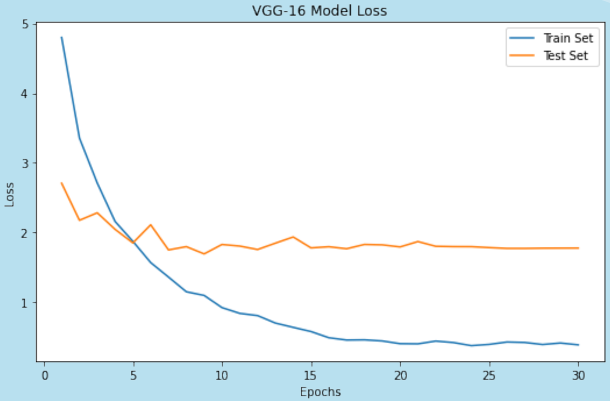
- ✓ The VGG-16 model was implemented using Keras's API
- ✓ VGG-16 model weights were loaded and the VGG16 base model was extended by flatten, dropout and the final fully connected layer
- ✓ This model was also trained for 30 epochs
- ✓ The total number of parameters for this model was 14,890,311 of which 175,623 parameters were trainable and the remaining were pretrained
- ✓ While training this model the learning rate reduced three times by a factor of 0.2.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Para
m #
=================================================================
vgg16 (Functional)     |     (None, 7, 7, 512)         1471
4688
_________________________________________________________________
flatten_1 (Flatten)          (None, 25088)             0
_________________________________________________________________
dropout_2 (Dropout)          (None, 25088)             0
_________________________________________________________________
dense_3 (Dense)              (None, 7)                 1756
23
=================================================================
Total params: 14,890,311
Trainable params: 175,623
Non-trainable params: 14,714,688
_________________________________________________________________
```

OUTPUT

✓ The highest training and testing accuracies of this model were respectively 89% and 76%





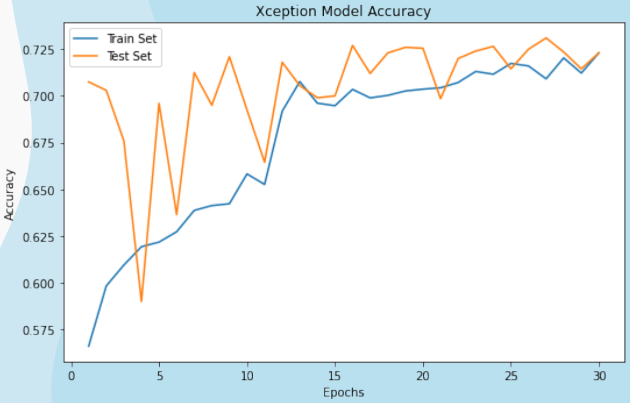
Xception training and evaluation

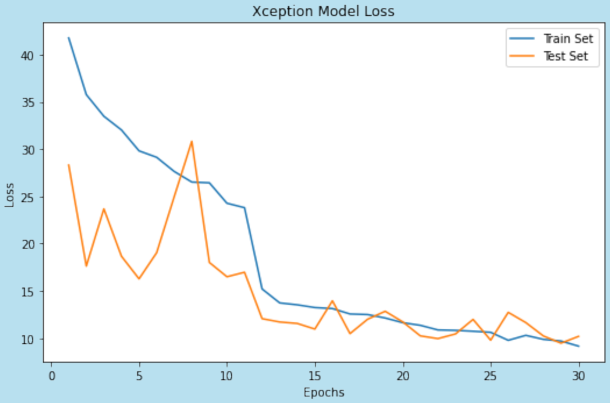
- ✓ The Xception model was also implemented using Keras and pretrained model weights
- ✓ Categorical cross-entropy loss and RMSprop optimizer with an initial learning rate of 0.00004 was used
- ✓ The number of trainable parameters for this model was 702,471 whereas the total number of parameters was 21,563,951
- ✓ The loss for both training and testing sets were very high (>= 9.1564).

```
Model: "sequential_2"
_________________________________________________________________
______
Layer (type)                 Output Shape              Para
m #
=================================================================
======
xception (Functional)        (None, 7, 7, 2048)        2086
1480
_________________________________________________________________
______
flatten_2 (Flatten)          (None, 100352)            0
_________________________________________________________________
______
dropout_3 (Dropout)          (None, 100352)            0
_________________________________________________________________
______
dense_4 (Dense)              (None, 7)                 7024
71
=================================================================
======
Total params: 21,563,951
Trainable params: 702,471
Non-trainable params: 20,861,480
_________________________________________________________________
```

OUTPUT

- ✓ The highest training and testing accuracies achieved by this model was 72% and 73% respectively.





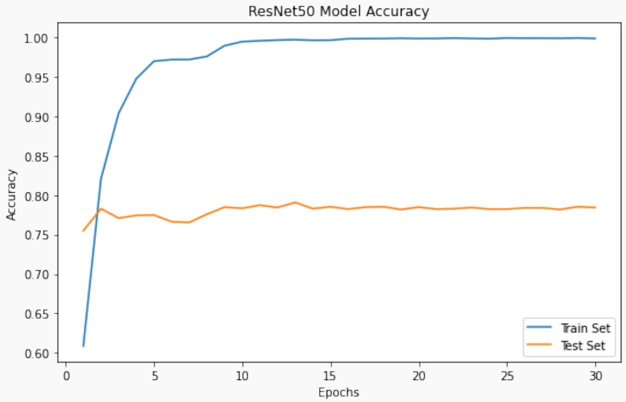
ResNet-50 training and evaluation

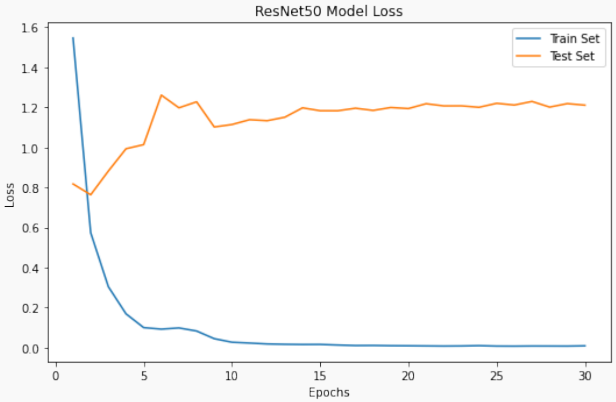
- ✓ The ResNet-50 model was fine tuned with two normalization and fully connected layer blocks
- ✓ The hidden fully connected layer had 256 neurons which after passing through a dropout layer arrives at the final classification layer
- ✓ The total number of parameters for this model was 49,682,311 of which 25,893,383 were trainable
- ✓ The model was trained for 30 epochs. Adam optimizer with an initial learning rate of 0.001 was used
- ✓ During training the learning rate reduced four times on 8th, 14th , 20th and 26th epochs respectively and the final learning rate was 1.6e-06.

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Functional)        (None, 7, 7, 2048)        23587712
_________________________________________________________________
flatten_3 (Flatten)          (None, 100352)            0
_________________________________________________________________
batch_normalization_10 (Batc (None, 100352)            401408
_________________________________________________________________
dense_5 (Dense)              (None, 256)               25690368
_________________________________________________________________
batch_normalization_11 (Batc (None, 256)               1024
_________________________________________________________________
dropout_4 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_6 (Dense)              (None, 7)                 1799
=================================================================
Total params: 49,682,311
Trainable params: 25,893,383
Non-trainable params: 23,788,928
_________________________________________________________________
```

OUTPUT

The highest training and testing accuracies for this model were 99% and 79% respectively





PERFOMANCE COMPARISON

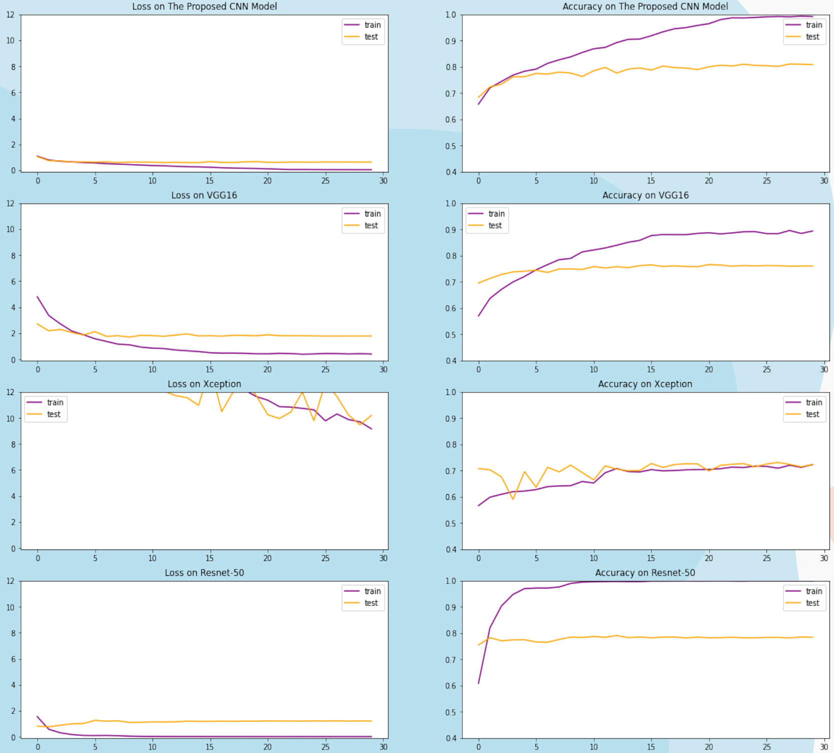
- ✓ The performance of four deep learning models for skin cancer classification was evaluated based on input data size, model complexity, accuracy, model loss, and accuracy curves
- ✓ The complexity of the proposed CNN model is lower than other pre-trained models in terms of its number of parameters
- ✓ the depth and number of network parameters of the Xpection and ResNet-50 models are higher than that of proposed model
- ✓ the proposed model has more parameters than the VGG-16 model

Deep Learning Model	Number of convolutional layers	Number of parameters
Proposed CNN Model	6	16,717,063
VGG-16	13	14,890,311
Xception	71	21,563,951
ResNet-50	48	49,682,311

Deep Learning Model	Training Accuracy	Testing Accuracy
Proposed CNN Model	99%	81%
VGG-16	89%	76%
Xception	72%	73%
ResNet-50	99%	79%

Table: Models Result after Evaluation

Deep Learning Model	Loss	Accuracy
Proposed CNN Model	0.6314	81%
VGG-16	1.7814	76%
Xception	11.6497	73%
ResNet-50	1.1508	79%



CONCLUSION AND FUTURE WORK

In conclusion,

- ✓ this paper represents a new approach to classifying skin cancer lesions from images
- ✓ First, we find the region of interest in dermoscopy images and cropped them using the image edge detection technique
- ✓ Second, it provided an efficient methodology by proposing a CNN model
- ✓ For sophisticated and accurate results, the neural network requires a large amount of data to train
- ✓ However, the experimental result shows that even with low computational power, this model can attain satisfactory accuracy, and the accuracy rate is excellent compared to VGG-16, Xception and ResNet-50 models
- ✓ The proposed CNN model training time is faster than the VGG-16, Xception and ResNet50 models
- ✓ Subsequently, the proposed CNN model needs fewer computational specifications as it takes less execution time.
- ✓ Moreover, the proposed CNN model accuracy is better than VGG-16, Xception and ResNet-50 models
- ✓ The proposed system can play a prognostic significance in the classification of skin cancer for both patients and physicians
- ✓ The proposed system is for categorical classification problem, which can classify the cancer type into seven classes
- ✓ Comparatively, the proposed CNN model can detect and classify skin cancer better than other proposed pre-trained models
- ✓ Also, this proposed system can play an influential role in the early diagnosis of dangerous diseases in other clinical domains related to medical imaging, particularly lung cancer and breast cancer, whose mortality rate is very high globally

FUTURE WORK

- ✓ I will combine multiple models to build a meta-model for better performance
- ✓ This process is known as Ensemble Learning
- ✓ The prediction levels are much more accurate with the use of the Ensemble Learning technique
- ✓ However, the implementation of this meta-model requires a high-performance Graphics Processing Unit (GPU) in addition to a CPU that demands high computational power, cost, and time
- ✓ The future work lies in the implementation of Skin Cancer Classification using Ensemble learning.