CASE STUDY - SKIN CANCER DETECTION

Submitted by : Ramya Mercy Rajan

Roll Number :AA.SC.P2MCA2107434

Subject :Deep Learning for Artificial Intelligence

University :Amrita Vishwa Vidyapeetham

Submitted to : Dr. Manju B.R and Ms.Remya Shaji

1. INTRODUCTION

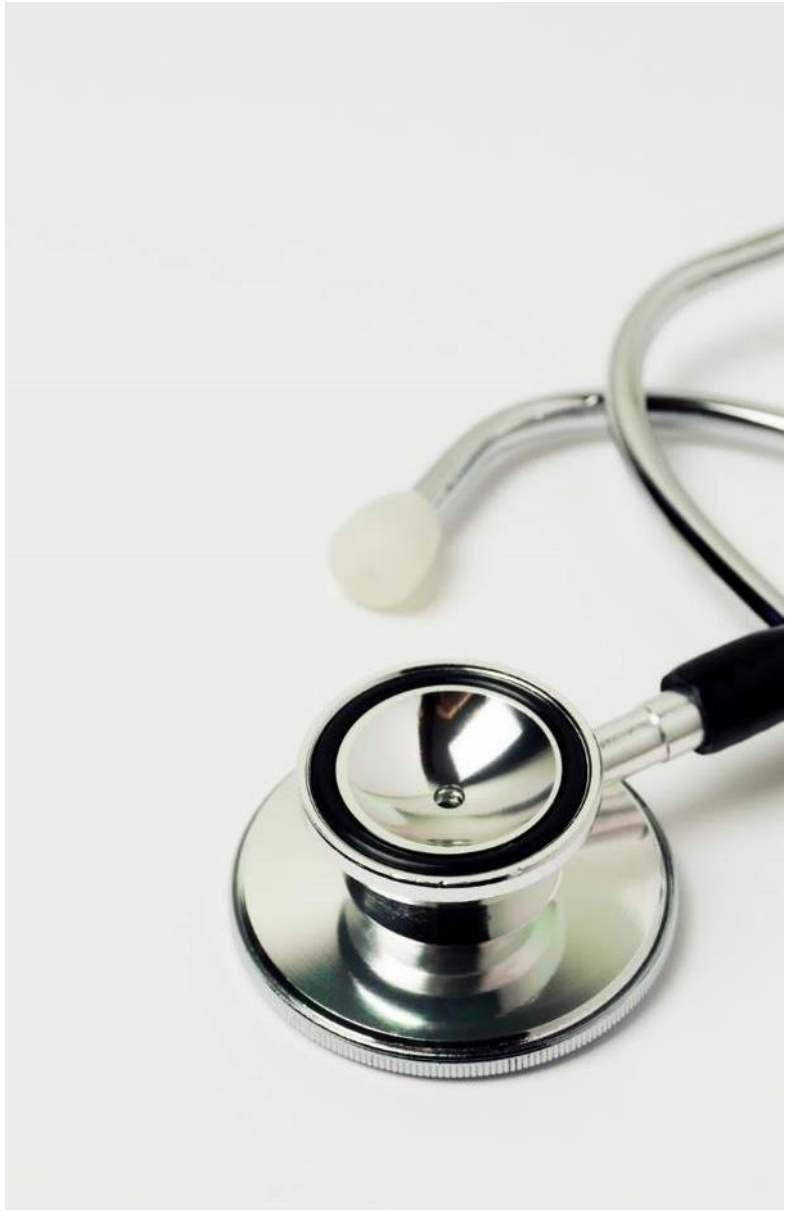
Background Information

Skin cancer is the most common cancer in the United States. It is estimated that approximately 9,500 people in the U.S. are diagnosed with skin cancer every day. More than three million cases of basal cell carcinoma are diagnosed in the United States every year, and nearly 20 Americans die from melanoma every day.

Melanoma and nonmelanoma are the two main types of skin cancer. Nonmelanoma is of lesser concern since it usually can be cured by surgery and is nonlethal. Melanoma, however, is the most dangerous skin cancer type.

Despite advances in therapeutics, the factors that most impact prognosis remain early recognition and removal of neoplasms before deep invasion or metastatic disease can occur.

With the increasing incidence of skin cancers, low awareness among a growing population, and a lack of adequate clinical expertise and services, there is an immediate need for AI systems to assist clinicians in this domain.



SKIN CANCER DETECTION

My aim is to predict the given skin lesion image taken by dermatoscopy and help medical professionals for early diagnosis and treatment of skin cancers. To train the deep learning model, I have used DermaMNIST by MedMNIST; this dataset contains multi-class images of skin lesions taken by dermatoscopy from real-life patients. It has 10000 images with seven classes that could be divided into malignant (Benign keratosis-like lesions, Dermatofibroma, Melanocytic nevi, Vascular lesions) and benign groups(Actinic keratoses and intraepithelial carcinoma, Basal cell carcinoma, Melanoma).

2. PROJECT EXECUTION

2.1 Data Acquisition

I will be utilizing the DermaMNST by MedMNIST. This is a dataset containing multiclass images as (28 x 28) (2D) numpy arrays in the form of an npz file. The images in the dataset have been collected from real life patients via dermascope. The dataset was downloaded from the official website: Available at <https://medmnist.com/>

2.2 Training Methodology

It is a vast dataset of 10,015 pictures of skin lesions which can be used to identify early stages of skin cancer by classifying them as malignant or benign. This data set is divided into training, validation and testing dataset; 7007, 1003 and 2005 pictures respectively. I will be building a deep learning neural network which will process an image and give a prediction output.

2. PROJECT EXECUTION

- Model Architecture

The model will be built using keras and tensorflow version 2.7.0. The model is the Convolutional Neural Network which is often used to analyze visual input. The model is made up of several types of layers like Flatten, Dense, Dropout, Batch Normalization, Pooling and 2D convolutional layers using activation functions 'relu' and 'softmax'.

I have created a sequential model which is a deep learning model where model layers are created and added to sequentially.

The 2D Convolutional layer is the main layer in the CNN which contains a set of kernels which help the model learn parameters through training. The filters convolve with the input images to create a feature map that is used to see which regions in the input image are relevant to the class.

The Batch Normalization layer is used to standardize the inputs to a layer for each mini-batch. This will help stabilize the learning process, and reduce the number of training epochs required to train deep networks.

The Pooling Layer is added to reduce the dimensions of the feature maps. It reduces the number of parameters to learn and the amount of computation performed in the network. The pooling layer summarizes the features present in a region of the feature map generated by a convolution layer.

2. PROJECT EXECUTION

- The Flatten Layer will be converting the data into a 1D array for inputting it to the next layer. We flatten the output of the convolutional layers to create a single long feature vector.
- The Dropout Layer will be used to prevent the possibility of overfitting of the model by randomly setting input units to 0 with a frequency of rate at each step during training time.
- The Dense Layer will ensure that all outputs from the previous layer are input to all its neurons, each neuron providing one output to the next layer. It is the fully connected layer in Keras and will have the neurons and activation function specified in this layer.
- The activation function is used to apply a certain degree of nonlinearity into the output of a neuron. The softmax function is used as the activation function in the output layer of neural network models that predict a multiclass probability.
- The ReLU is a function that will output the input directly if it is positive, otherwise, it will output zero. It will prevent the vanishing gradient problem, allowing models to learn faster and perform better

PROJECT EXECUTION

Model Training

- The model will be trained with an optimum number of epochs to achieve a high accuracy. The epochs determine the number of times the dataset will run through the CNN to train. If there are way too many epochs it could result in overfitting hence it is best to use an optimal number of epochs with early stopping call back function which will stop the training once the model performance stops improving on a hold out validation dataset.
- The model will be trained with a low learning rate which will ensure that the model is able to learn optimally by globally setting suitable weights. This will take longer to train but it will produce better results than if the learning rate is high.
- The model performed better with focal loss function which could assist in tackling the issue faced by the imbalanced dataset. Optimizers are functions that are used to modify the attributes such as weights and learning rate of a neural network. This assists in reducing the overall loss and improves the accuracy.
- The model will utilize Adam Optimizer by Keras. This algorithm will take into consideration the 'exponentially weighted average' of the gradients and use this concept to get an average to accelerate the gradient descent.

PROJECT EXECUTION

- Focal Loss

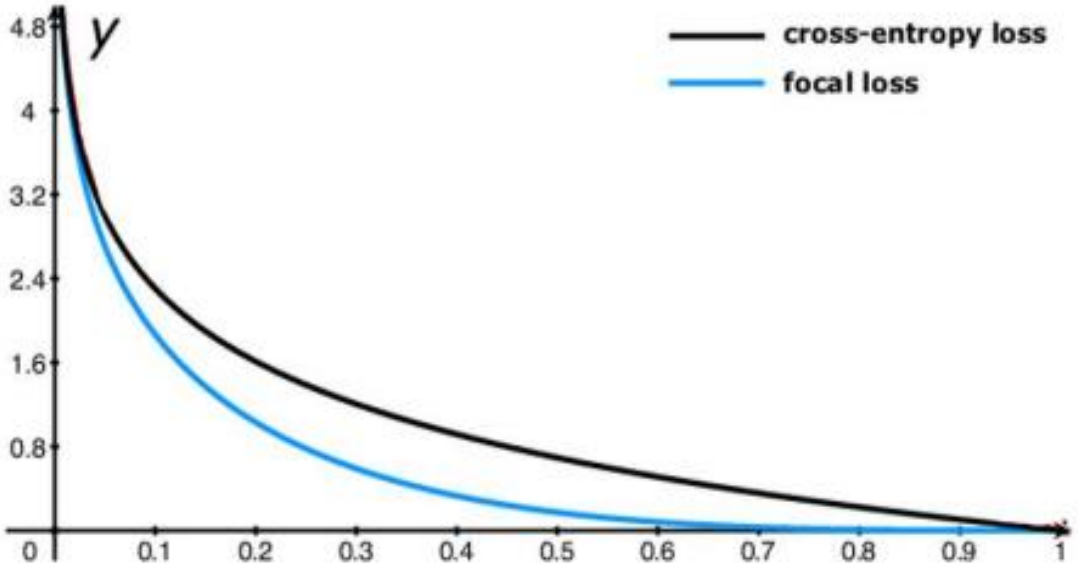
Unbalanced data like what I have here can cause some problems to the model learning procedure. To address this problem, I have used a new loss function named focal loss. With the focal loss, we add a modulating factor multiplied by the Cross-Entropy Loss. By this modulating factor, when a sample is misclassified, p is low and modulating factor is near one, so the loss is unaffected. But as p increases, the modulating factor approaches 0, and the loss for well-classified examples is down-weighted.

$$\mathrm{FL}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t).$$

PROJECT EXECUTION

$$\mathrm{FL}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t).$$

This modulating factor reduces the loss function contribution from easy examples and extends the range in which an example receives low loss. So the model will correctly learn the data and save from overfitting.



2.3 SERVICE IMPLEMENTATION

- A “Benign” tumor refers to a condition, tumor, or growth that is not cancerous and it means that it does not spread to other parts of the body. It does not invade nearby tissue. A “Malignant” tumor means it's cancerous. Malignant tumors contain cells that are cancerous, growing out of control and capable of metastasizing. Metastasize denotes that the cells of the tumor are able to leave the original tumor and travel to other parts of the body.
- Intervention and stopping the cancer growth is now required. Research indicates that most experienced physicians can diagnose cancer with 76% accuracy. I have executed all these images in order to specify if that cancer out of these images is malignant or benign. I add up all these features into our machine learning model and the model teaches the machine how to basically classify images or classify data and report us if it is malignant or benign. This data set is divided into training, validation, and testing dataset. I have built a deep learning neural network which processed an image and gave a prediction output.
- The model is the Convolutional Neural Network which is often used to analyze visual input. The model is trained with an optimum number of epochs to achieve a high accuracy.

OUTCOMES

- 3.1. Data Preprocessing

All images in our dataset were pre-processed into (28 x 28 x 28) (3D) NumPy arrays in the form of an npz file. However, given that the original images consist of RGB coefficients in the range of [0, 255], more importantly, the sample size and the imbalanced nature of outcomes, i.e., the distribution of classes in our dataset. We implemented data pre-processing and augmentation methods concerning the data role. These methods were applied for all training, validation, and testing datasets.

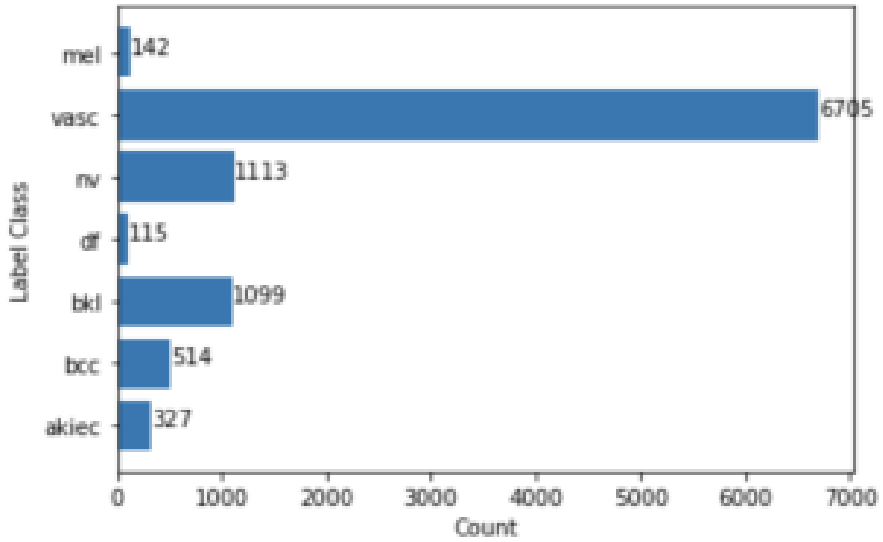
Govern by the domain knowledge, model performance and accuracy all of the following were applied for the training images using ImageDataGenerator API from tensorflow :

1. Rescaling: Rescaled all images from [0, 255] to [0, 1].
2. 90 degree range for random rotations.
3. Random horizontal and vertical flip.
4. Random zoom range of 10%.
5. Width and height shift of 10% of the total width/height

Only rescaling was implemented to the validation and testing dataset. Data augmentation is also a very important aspect that will assist in preventing the overfitting of the model by increasing the number of data present in the training dataset.

OUTCOME

```
x = classes_df.index
y = classes_df.Count
plt.barh(x, y)
plt.xlabel('Count')
plt.ylabel('Label Class')
for index, value in enumerate(y):
    plt.text(value, index,
             str(value))
plt.show()
```

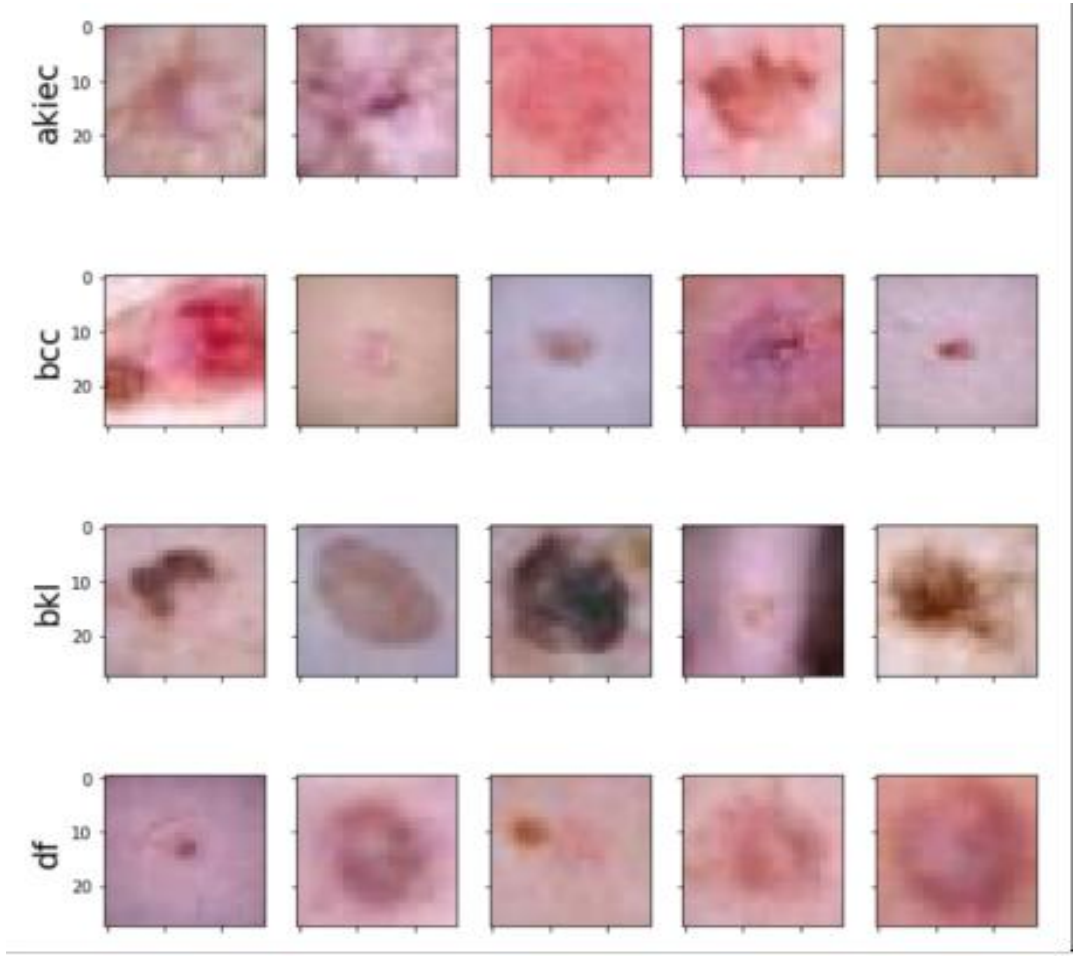


- 3.2 Exploratory Data Analysis (EDA)

The data has been analyzed using EDA to understand how to use the details to our advantage and summarize the vast data provided to us. I first checked the files and their shape and size. Using numpy unique modules we were able to check how many classes we would be dealing with in the dataset.

I have concatenated the X and Y datasets to check the frequency of the data. I tried plotting the data frame of frequency of each class with a matplotlib graph. This graph showed the imbalance in the dataset with one class having over 6000 images while the others had fewer images.

OUTCOME



- I also plot a grid to visualize the images in the dataset according to the multiclass labels to get an overview of what kind of images I will be dealing with.

3.3 MODELING

- The model was trained rigorously to ensure that we achieved the highest possible accuracy. I used an early stopping function with a callback to stop model training when the highest possible accuracy had been attained. In a medical problem, accuracy is not the only important metric that needs to be evaluated.
- The precision and recall of the model is of equal importance, hence I had to ensure that the precision, recall and F1 score of all classes were not zero. After multiple attempts I achieved an accuracy of 77% with a decent precision and recall for all classes.
- Previously, I was not able to get precision and recall more than zero for the classes with less images. After attempting with focal loss we achieved a better figure.

OUTCOME

- Below is the classification report of the model which was achieved after multiple attempts:

	precision	recall	f1-score	support
0	0.42	0.55	0.48	66
1	0.45	0.58	0.51	103
2	0.60	0.48	0.53	220
3	0.50	0.17	0.26	23
4	0.53	0.27	0.36	223
5	0.86	0.94	0.89	1341
6	0.80	0.69	0.74	29
accuracy			0.77	2005
macro avg	0.59	0.52	0.54	2005
weighted avg	0.75	0.77	0.75	2005

OUTCOME

- 3.5. Testing and Improvements.

The dataset was split into 70%, 10%, and 20% for training, validation and testing; respectively, before several models were trained with different combinations of hyperparameters. The final model was tested using the 20% of the dataset, which is unseen data that the algorithm never previously encountered.

In Google Colab I printed the classification report and the summary of the training and validation to check the results of the model. In all experiments and due to the unbalanced nature of the dataset, the accuracy (ACC) score and F1-score were used as performance metrics.

In future studies, this model accuracy can be improved by trying one or all of the following; first, include more images for the under sampled classes. This will help tackle the imbalance dataset issue that could be a bottleneck while classifying images that have very few images in the training phase of the model training.

In this study other pretrained models can be tried e.g., U net, ResNet and more using supportive hardware such as faster processor and GPU.

REFERENCES

- https://www.aad.org/media/stats-skin-cancer
- Ray A, Gupta A, Al A. Skin lesion classification with deep convolutional neural network: process development and validation. JMIR Dermatol. 2020 May 7;3(1):e18438. doi: 10.2196/18438
- Glazer AM, Rigel DS, Winkelmann RR, Farberg AS. Clinical Diagnosis of Skin Cancer: Enhancing Inspection and Early Recognition. Dermatol Clin. 2017 Oct;35(4):409-416. doi: 10.1016/j.det.2017.06.001. Epub 2017 Aug 7. PMID: 28886797
- Goyal M, Knackstedt T, Yan S, Hassanpour S. Artificial intelligence-based image classification methods for diagnosis of skin cancer:

REFERENCES

- Omania Health. 2019. Skin Cancer Growing Public Concern UAE. Omania Health. [online]. Available at"
- https://insights.omnia-health.com/medical-specialities/skin-cancer-growing-public-co ncern-uae

- Brownlee, J. 2019. "A Gentle Introduction to the Rectified Linear Unit (ReLU)". Machine Learning Mastery. [online]. Available at:
- https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-lear ning-neural-networks/

- https://towardsdatascience.com/a-loss-function-suitable-for-class-imbalanced-data-fo cal-loss-af1702d75d75

THANKYOU

Submitted by : Ramya Mercy Rajan

Roll Number : AA.SC.P2MCA2107434

Subject :Deep Learning for Artificial Intelligence

University :Amrita Vishwa Vidyapeetham

Submitted to : Dr. Manju B.R and Ms.Remya Shaji