**TITLE**: Skin Cancer Detection and Classification Using Machine Learning

**INTRODUCTION**: Skin cancer is the growth of abnormal cells in the outermost skin layer (epidermis), caused by unrepaired DNA damage that triggers mutations. This mutation leads the skin cells to multiply rapidly and form malignant tumors.

Just like many other cancer types, skin cancers are highly curable if detected early. However, early detection remains a challenge for the following reasons:

* Clinical examinations are expensive, requires high level of training, and efforts to operate the equipments.
* There are many different classes of skin cancer and they can look different, depending on skin tone, size, location where they occur and skin type of the patient. This makes it harder for even experienced medical practitioners to accurately identify and classify them.

**SOURCE CODE:[LINK-TO-SOURCE-CODE]**

**OBJECTIVES:**

* To use previously classified skin cancer image data sets to train machine learning models which will help to classify skin cancer types.
* To deploy a web app which will enable doctors to quickly classify cancer types by simply uploading the images with their smartphones, instead of going to perform the experiments in the laboratory.

**ABOUT THE DATASET:**

There are 9 classes of skin cancer and the original dataset consists of 2357 images of malignant and benign oncological diseases, which were formed from The International Skin Imaging Collaboration (ISIC). All images were sorted according to the classification taken with ISIC, and all subsets were divided into the same number of images, with the exception of melanomas and moles, whose images are slightly dominant.

The data set contains the following diseases:

* Actinic keratosis
* Basal cell carcinoma
* Dermatofibroma
* Melanoma
* Nevus
* Pigmented benign keratosis
* Seborrheic keratosis
* Squamous cell carcinoma
* Vascular lesion

The dataset can be found here on [**Kaggle**].

**MODEL SUMMARY**

Because this is a multi-class classification problem, we used Convolution Neural Network (CNN) consisting of **[X]** layers to train the model. We also used other Python libraries such as Tensorflow, Pandas, Matplotlib, etc., for image processing, data visualizations and other ancillary statistical analysis, and the resulting model takes about 45 minutes to train.

**[INSERT CNN ILLUSTRATION GRAPHICS]**

**CREATING TRAINING, VALIDATION AND TEST SPLITS**

The original dataset was grouped into two folders: the train set (2239 images) and the test set (118 images). 80% of the train set was used to train the model while the remaining 20% was used to validate it.

**COMPILING AND TRAINING**

The model did overfit with poor validation accuracy!

**CHECKING FOR DISTRIBUTION OF CLASSES IN THE TRAINING DATASET**

As you can see below, the is a high class imbalance. Some classes (Melanomas and Moles) have proportionately higher number of samples than the others, and this resulted in the poor performance of the model.

**[BAR CHART SHOWING CLASS INBALANCE]**

**AUGMENTING THE IMAGE DATASET**

We used the **Augmentation library** to manipulate our original dataset so as to obtain modified versions of the images. We did rotate the images 15 degrees (right and left), and for each class, 2,000 modified images were created, resulting in an overall sample size of 18,000 images.

**CREATING TRAINING AND VALIDATION SET (AUGMENTED)**

We retrained the model but this time around, we used the images created using the augmentation library. This increased the validation accuracy to **90%**!

**HYPERPARAMETER TUNING**

RandomSearch was used on the model **[WHY?].** Also, different layer depths were tested but not much Improvement was recorded.

**TESTING THE MODEL**

Even though we recorded very high validation accuracy, the model performed poorly on the test set with evaluation accuracy score of **44%.** We observed that the model could accurately predict the Melanomas class but failed to predict others. As already pointed out, the melanomas class is one of the predominant classes (second largest).

**RECOMMENDATION AND CONCLUSION**

It is important to note that even though the Augmentation library was used to even out the class imbalance,