**SIIM ISIC MELANOMA CLASSIFICATION**

**AIM:** Identify melanoma in lesion images using Ensemble learning under Deep Learning models. Our model should predict the probability (floating point) between 0.0 and 1.0 that the lesion in the image is malignant (the target). In the training data, train.csv, the value 0 denotes benign, and 1 indicates malignant.

**DATASET:** I downloaded the dataset from Kaggle platform under their Featured Predcition Competiton to predict a binary target for every image.

The ISIC 2020 Challenge Dataset <https://doi.org/10.34970/2020-ds01> (c) by ISDIS, 2020

The dataset was generated by the International Skin Imaging Collaboration (ISIC) and images are from the following sources: Hospital Clínic de Barcelona, Medical University of Vienna, Memorial Sloan Kettering Cancer Center, Melanoma Institute Australia, The University of Queensland, and the University of Athens Medical School.

**DATA DESCRIPTION:** The images are provided in DICOM format. This can be accessed using commonly-available libraries like pydicom, and contains both image and metadata. It is a commonly used medical imaging data format.

Images are also provided in JPEG and TFRecord format (in the jpeg and tfrecords directories, respectively). Images in TFRecord format have been resized to a uniform 1024x1024.

Metadata is also provided outside of the DICOM format, in CSV files. See the Columns section for a description.

## Files

* train.csv - the training set
* test.csv - the test set
* sample\_submission.csv - a sample submission file in the correct format

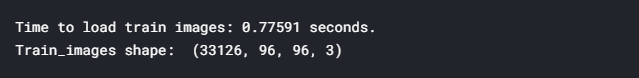
## Columns

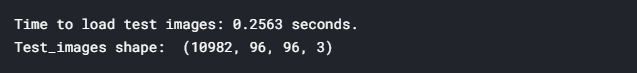
* image\_name - unique identifier, points to filename of related DICOM image
* patient\_id - unique patient identifier
* sex - the sex of the patient (when unknown, will be blank)
* age\_approx - approximate patient age at time of imaging
* anatom\_site\_general\_challenge - location of imaged site
* diagnosis - detailed diagnosis information (train only)
* benign\_malignant - indicator of malignancy of imaged lesion
* target - binarized version of the target variable



**TRAINING OF DATA:** I choose the jpeg data to work with which contains **11000 test** files and **33100 train** files .

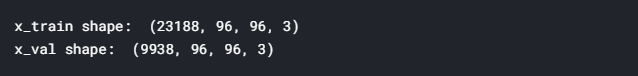
* train.csv contains : 33126 unique values and 8 columns.
* test.csv contains : 10982 unique values and 5 columns.

Importing image data into numpy arrays:Due to large data set size we have,You will need a lot of images (possibly >10,000) to perform deep learning, and loading those images can take a long time (possibly >1 min). Particularly when you want to make changes to your model over and over, image loading time cannot be negligible. Here, I will introduce how to convert images to a .npy file which will not only reduce the loading time but also memory space and data transfer time when you want to copy the image data to somewhere else. So I convert them into .npy files for faster loading of data and uses size 96 x 96 for loading train and test images. I therefore used **x\_train\_96.npy** file to train my data for better results.

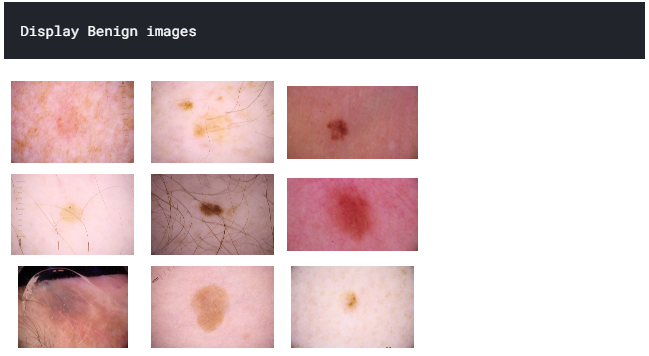
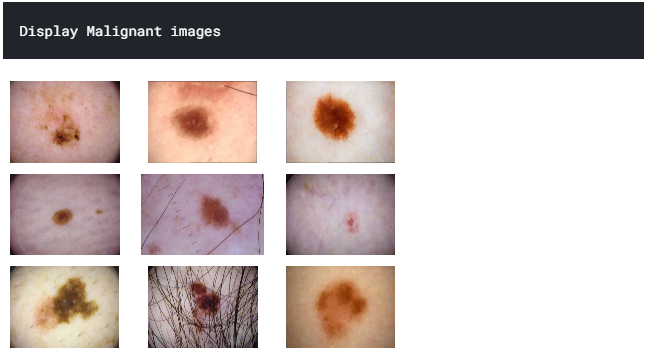




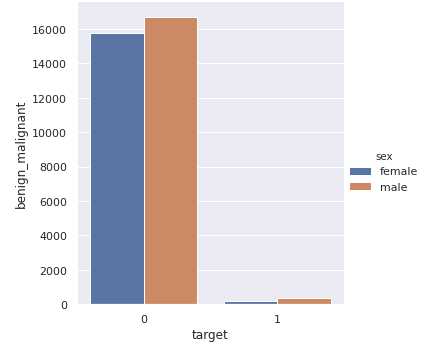
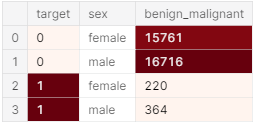
Splitting of Train data: I split my data with keeping test\_size = **0.3**.



Data Augmentation is performed using Keras ImageDataGenerator to increase the generalizability of the model.

**VISUALIZATION OF DATA: **

Target Vs Sex Distribution:

**MODELLING:** I used Ensemble Learning technique to improve the accuracy of my model. Ensemble Building is the leading winning strategy for machine learning competitions and often the technique used for solving real-world problems. What often happens is that while solving a problem or participating in a competition you end up with several trained models, each one with some differences to another — and you end up picking up your best model based on your best evaluation score. The truth is that your best model “knows less” about the data, than all others “weak models” combined. Combining several base models together to create a more powerful ensemble model then rises as a natural by-product of this workflow.The Convolutional Neural Network models I used are:

* VGG-16
* XCEPTION
* DenseNet-201

**Observations:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODEL** | **BATCH SIZE** | **EPOCH NO.** | **TEST LOSS** | **TEST ACCURACY** |
| VGG-16 | 128 | 30 | **0.117** | **0.981** |
| XCEPTION | 128 | 15 | **0.117** | **0.981** |
| DenseNet | 128 | 30 | **0.151** | **0.952** |

**Result :** Total Accuracy attained is 0.97 with a Kaggle score of 0.86 in public leaderboard and melanoma test images have been predicted with malignant or benign.