**Data Set:**

This Machine Learning code deals with analyzing and preprocessing images of Skin Cancer dataset from Kaggle. You can find the dataset from the following link: (Which is of 6Gb)

<https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000>

You can find the code lines in the Skin\_Cancer notebook which I provided. I have learned the TensorFlow framework on Coursera, so I used this framework which I know and tried to solve the problem. The first block of the code contains the imported functions required to run the model. I have downloaded the 6Gb datafile which contains image files as part 1 and part2 and the required csv file into my personnel drive and mounted the drive in the google colab notebook. In the next block I have printed the images along with their labels as output. The next block of the code visualizes the images. They are 7 types of skin type cancers represented during visualization, in the next block of code I have paired the types of skin cancers along with base directories part 1 and part 2. In the next block I imported the csv meta data and printed out the head using df.head() function, next I mapped the path image\_id , lesson\_id.

**Data Processing:**

Here in this session, I have checked the csv file in which I have found the column ‘age’ has 57 causalities, so I filled the empty places using the statistical mean method. Next, I attempted to resize the images using the read\_and\_resize\_image function, unfortunately I have received some error in reshaping I tried to figure out as a fresher it’s not up to my knowledge. So, I continued further by checking the shape count at first, I have received only 5015 images as output count. Later I checked the missing count so that the total shape count equal to 10015 images. Later I splited the train and test tests using train\_test\_split inbuild function. Later I checked out the shape of train\_df as I got error previously just making sure that all have the same shape. Next, I extracted the image data and target labels for train and test sets.

**Machine Model:**

I have created a model using two CNN layers along with max pooling, followed by the flatten and LSTM layers and used SoftMax as activation function. I compiled the model using Adam optimizer along with ‘sparse\_categorical\_crossentropy’. Finally, I used model.fit with 100 epochs.

**Conclusion:**

My model got a training accuracy of 67 and 62 percent validation accuracy. So later I appended and tried with Bidirectional layer for which I have got the same accuracy metric, by using both LSTM and bidirectional LSTM layers the accuracy has decreased a little amount, later by adding learning rate also haven’t changed anything. So, by experimenting many times my model accuracy maintained constant, but the loss values changed in decimal throughout the 100 epochs by using the single LSTM layer before the final output layer.