**Project Name - Melanoma Detection Assignment**

To build a multiclass classification model using a custom convolutional neural network in TensorFlow

**Table of Contents**

* [General Info](https://github.com/Karishma6636/Bike-Sharing-Assignment#general-information)
* Project Pipeline
* [Technologies Used](https://github.com/Karishma6636/Bike-Sharing-Assignment#technologies-used)

**General Information**

To build a CNN based model which can accurately detect **melanoma**. Melanoma is a type of cancer that can be deadly if not detected early. It accounts for 75% of skin cancer deaths. A solution that can evaluate images and alert dermatologists about the presence of melanoma has the potential to reduce a lot of manual effort needed in diagnosis.

**Project Pipeline**

* Data Reading/Data Understanding → Defining the path for train and test images
* Dataset Creation→ Create train & validation dataset from the train directory with a batch size of 32. Also, make sure you resize your images to 180\*180.
* Dataset visualisation → Create a code to visualize one instance of all the nine classes present in the dataset
* Model Building & training :
  + Create a CNN model, which can accurately detect 9 classes present in the dataset. While building the model, rescale images to normalize pixel values between (0,1).
  + Choose an appropriate optimiser and loss function for model training
  + Train the model for ~20 epochs
  + Write your findings after the model fit. You must check if there is any evidence of model overfit or underfit.
* Chose an appropriate data augmentation strategy to resolve underfitting/overfitting
* Model Building & training on the augmented data :
  + Create a CNN model, which can accurately detect 9 classes present in the dataset. While building the model rescale images to normalize pixel values between (0,1).
  + Choose an appropriate optimiser and loss function for model training
  + Train the model for ~20 epochs
  + Write your findings after the model fit, see if the earlier issue is resolved or not?
* Class distribution: Examine the current class distribution in the training dataset
  + - Which class has the least number of samples?
  + - Which classes dominate the data in terms of the proportionate number of samples?
* Handling class imbalances: Rectify class imbalances present in the training dataset with [Augmentor](https://augmentor.readthedocs.io/en/master/" \t "_blank) library.
* Model Building & training on the rectified class imbalance data :
  + Create a CNN model, which can accurately detect 9 classes present in the dataset. While building the model, rescale images to normalize pixel values between (0,1).
  + Choose an appropriate optimiser and loss function for model training
  + Train the model for ~30 epochs
  + Write your findings after the model fit, see if the issues are resolved or not?

**Technologies Used**

* Pandas - version 1.4.4
* Numpy - version 1.21.5
* Matplotlib - version 3.5.2
* tensorflow

**Contact**

Created by [@karishma6636]- feel free to contact me!