

# Project Name : Melanoma-Detection-Assignment

> Outline a brief description of your project.

#### 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 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?

#### ## Table of Contents

- \* [General Info](#general-information)

- \* [Conclusions](#conclusions)

#### ## General Information

- What is the background of your project?

This project comprises of building a multiclass classification model using a custom

convolutional neural network (CNN) in TensorFlow. The objective was to develop a CNN model capable of accurately detecting melanoma, a type of cancer that can be life-threatening if not detected early.

-What is the business problem that your project is trying to solve?

Melanoma accounts for 75% of skin cancer deaths, and the project aimed at providing a solution which could evaluate images and alert dermatologists of the presence of melanoma, and hence reduce the manual efforts required for diagnosis.

- What is the dataset that is being used?

The dataset contains 2,357 images of malignant and benign oncological diseases, obtained from the International Skin Imaging Collaboration (ISIC). The images were categorized into various diseases, including actinic keratosis, basal cell carcinoma, dermatofibroma, melanoma, nevus, pigmented benign keratosis, seborrheic keratosis, squamous cell carcinoma, and vascular lesion.

Throughout the project, best practices and guidelines were followed and has built the CNN model from scratch without relying on any pre-trained or transfer learning models.

## ## Conclusions

The project pipeline involved several key steps. First was to understand the data and defining the paths for the train and test images. It is followed by creating the training and validation datasets, ensuring the images to be resized to 180x180 pixels. All of the 9 classes (categorized based on the diseases) is visualized using the custom code, to gain more insight into the dataset.

During the model building and training phase, CNN model has been built and normalized the pixel values of the images between 0 and 1. Appropriate optimizers and loss functions were selected, and the model was trained for approximately 20 epochs. After fitting the model, the results were analyzed to identify any signs of overfitting or underfitting.

In order to address potential issues of underfitting or overfitting, data augmentation strategy has been implemented. It involved augmenting the training dataset to provide additional variations of the images. The augmented data is then used to train another CNN model for approximately 20 epochs. The model's performance has been evaluated and compared with the earlier results to identify whether the augmentation strategy resolved the issues.

Taking into consideration the class distribution in the training dataset, the class with the least number of samples and dominant classes has been identified has utilized the Augmentor library to rectify class imbalances. With this rectified dataset trained a CNN model for approximately 50 epochs. Finally, the model is

analyzed to be fit and assessed if the issues identified earlier were successfully addressed.

It gained hands-on experience in data preprocessing, CNN architecture design, model training, and evaluation, through this project. It provided valuable insights to work with image classification tasks and handling class imbalances in datasets.

## Contact

Created by [@shehands] - !