

Melanoma-CNN-Prediction

Problem Statement

In this assignment, you will build a multiclass classification model using a custom convolutional neural network in tensorflow.

Problem statement: 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 which can evaluate images and alert the dermatologists about the presence of melanoma has the potential to reduce a lot of manual effort needed in diagnosis.

The 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:

1. Actinic keratosis
2. Basal cell carcinoma
3. Dermatofibroma
4. Melanoma
5. Nevus
6. Pigmented benign keratosis
7. Seborrheic keratosis
8. Squamous cell carcinoma
9. Vascular lesion

NOTE: You don't have to use any pre-trained model using Transfer learning. All the model building process should be based on a custom model.

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, see if there is evidence of model overfit or underfit
- Choose 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?

Conclusion (are present in the coding file also)

As the accuracy of the model increases, the loss decreases. The final model has an accuracy of 87% and a loss of 0.3. The model is able to predict the class of the lesion with a high accuracy. Augmenting the data and countering class imbalance helped in improving the accuracy of the model.