Skin Cancer Classification

After loading the skin cancer data, we observed a significant class imbalance, where certain classes had a much higher number of samples compared to others. To address this issue, we utilized the resample function from the sklearn.utils module. For the classes with a higher number of samples, we randomly reduced the sample size to ensure a more balanced distribution. Conversely, for the classes with a lower number of samples, we performed oversampling by creating duplicate instances to increase their representation in the dataset. While this approach is not considered ideal, it has yielded favourable results in our experiments. By mitigating the impact of class imbalance, we have improved the model's ability to learn and classify skin cancer across all categories, contributing to more reliable and accurate predictions.

We used the train_test_split() method to efficiently split the dataset into distinct training and test sets before performing any preprocessing on the skin cancer data. To achieve representative distribution in both sets, this approach distributed samples from each class in a random and proportionate manner. We created a solid base for further preprocessing procedures by doing this. This first division enables us to properly preprocess the data while preserving the integrity of the training and test sets, including picture resizing, normalisation, and enhancement. In the end, this method enables the creation of a strong skin cancer classification model that can successfully generalise to unknown data and produce precise predictions.

We made use of the well-known computer vision toolkit OpenCV to preprocess the skin cancer photographs. First, we equally scaled all the photos to a 150x150 pixel dimension. During model training, this scaling assures consistency and minimises computing complexity. In order to improve the efficiency of the model, we also normalised the picture array's pixel values. The pixel intensity range is standardised through normalisation, which makes it simpler for the model to discover significant patterns.
The picture preparation methods work as a "black box" in our system. We provide the raw skin cancer picture into this preprocessing tool as input. The module then does the image array's resizing and normalisation operations. Our skin cancer classification model then receives the processed picture array as its input.
We guarantee that the input to the model is in the form of a preprocessed picture array by using this well specified image preprocessing pipeline, which is designed for efficient feature extraction and classification. With this method, the model is better able to learn from and anticipate outcomes based on the processed skin cancer pictures, improving patient care and diagnostic precision.

Skin Cancer Classification

MODEL EXPLANATION:

The model offered is a sequential CNN architecture created for the categorization of skin cancer. An overview of the model's design and how it links to the earlier background is given below:

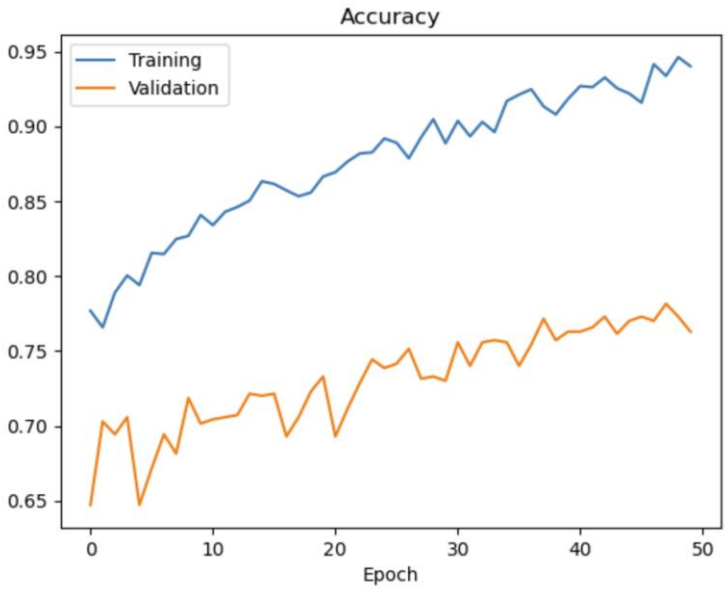
The model offered is a sequential CNN architecture created for the categorization of skin cancer. An overview of the model's design and how it links to the earlier background is given below:

1. Input Layer: The model accepts input pictures of the shape (150, 150, 3), which stands for 150-pixel width and height and three RGB colour channels.
2. Starting with a Conv2D layer with 256 filters, a kernel size of (3, 3), and ReLU activation, the model moves on to convolutional layers. By combining the learnt filters with the input pictures, this layer extracts features. Then, to limit the number of spatial dimensions and incorporate translation invariance, max pooling is used with a pool size of (2, 2). To avoid overfitting, dropout regularisation at a rate of 0.3 is also used.
3. More Convolutional Layers: Two additional Conv2D layers with 128 and 64 filters each are added. MaxPooling2D and Dropout layers are placed after each layer. From the input photos, these layers further collect and extract significant characteristics.
4. In order to prepare it for the fully connected layers, the flatten layer reshapes the output of the earlier convolutional layers into a flat vector.
5. Two Dense fully connected layers make up the model. The network can learn intricate nonlinear representations thanks to the first Dense layer's 32 units and ReLU activation function. These layers don't explicitly call for dropout regularisation, although it may be added if required. Seven units, or the seven kinds of skin cancer, are represented in the final Dense layer. Class probabilities for multi-class classification are produced using the softmax activation function.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)           (None, 148, 148, 256)     7168

 max_pooling2d_3 (MaxPooling  (None, 74, 74, 256)      0
 2D)

 dropout_3 (Dropout)         (None, 74, 74, 256)       0

 conv2d_4 (Conv2D)           (None, 72, 72, 128)       295040

 max_pooling2d_4 (MaxPooling  (None, 36, 36, 128)      0
 2D)

 dropout_4 (Dropout)         (None, 36, 36, 128)       0

 conv2d_5 (Conv2D)           (None, 34, 34, 64)        73792

 max_pooling2d_5 (MaxPooling  (None, 17, 17, 64)       0
 2D)

 dropout_5 (Dropout)         (None, 17, 17, 64)        0

 flatten_1 (Flatten)         (None, 18496)             0

 dense_2 (Dense)             (None, 32)                591904

 dense_3 (Dense)             (None, 7)                 231

=================================================================
Total params: 968,135
Trainable params: 968,135
Non-trainable params: 0
```

Skin Cancer Classification

RESULTS:



A validation accuracy of 76% is a commendable accomplishment, especially in light of the quantity of the dataset and the model's complexity. It illustrates that the CNN model you used can identify significant trends and forecast skin cancer categorization with high accuracy.

Your insight on the Xception model is also important. For huge datasets and more difficult image identification tasks, the Xception model, noted for its depth and complexity, is intended. However, it appears that the Xception model may be overly sophisticated in this situation, with a very limited dataset, and might result in overfitting or difficulty with convergence. The first results showed a 66.06% accuracy, which suggests that the model might not be appropriate for the amount of the dataset.

The appropriate balance between model complexity and dataset quantity must be taken into account. When the dataset is small, simpler models may perform similarly to or even better than more sophisticated ones. Your findings imply that the basic CNN model you used is already successful in collecting elements important for classifying skin cancer.

You may prevent overfitting and improve generalisation performance by choosing a model complexity that is appropriate for the data that is available. It is always important to take into account the unique features of the dataset and test out several models to find the most effective strategy.