Problem Statement

Cancer cell detection and preparing the model based on the dataset

Dataset

HAM10000 ("Human Against Machine with 10000 training images") consists of 10015 dermatoscopic images which can serve as a training set for academic machine learning purposes. Cases include a representative collection of all important diagnostic categories in the realm of pigmented lesions More than 50% of lesions are confirmed through histopathology (histo), the ground truth for the rest of the cases is either follow-up examination (follow_up), expert consensus (consensus), or confirmation by in-vivo confocal microscopy (confocal). The dataset includes lesions with multiple images, which can be tracked by the lesion_id-column within the HAM10000_metadata file.

Tools used

Python,Numpy,Pandas,CNN

Approach Towards the Problem Statement

A Convolutional Neural Network(CNN) model using keras is built to perform multi-class classification on the dataset.The choice of using a Convolutional Neural Network (CNN) for this purpose is based on the nature of the dataset and the problem at hand. Here are the reasons for using CNN in this code:

1.Image Data: The HAM10000 dataset consists of images of skin lesions. CNNs are well-suited for image classification tasks as they can effectively learn spatial hierarchies and capture local patterns in images.

2.Translation Invariance: CNNs are designed to be translation invariant, meaning they can recognize patterns in an image regardless of their position. In the case of skin lesion classification, the position of the lesion within the image is not important for classification, and CNNs can learn to extract features regardless of their location.

3.Local Feature Extraction: CNNs use convolutional layers to convolve small filters over the input image, capturing local features. This property is beneficial in identifying distinctive patterns and textures present in skin lesions, which are crucial for accurate classification.

4.Parameter Sharing: CNNs leverage parameter sharing, meaning that the same weights are used across different regions of the image. This reduces the number of parameters in the model, allowing it to generalize better and handle variations in lesion appearance.

5.Hierarchical Representation: CNNs are capable of learning hierarchical representations of images. The initial layers capture low-level features such as edges and textures, while deeper layers learn high-level representations that correspond to more complex structures and concepts. This hierarchical approach aligns well with the complexity of skin lesion classification.

Overall, CNNs have demonstrated remarkable performance in image classification tasks, making them a suitable choice for the multi-class classification problem of identifying different types of skin lesions in the HAM10000 dataset.

Code

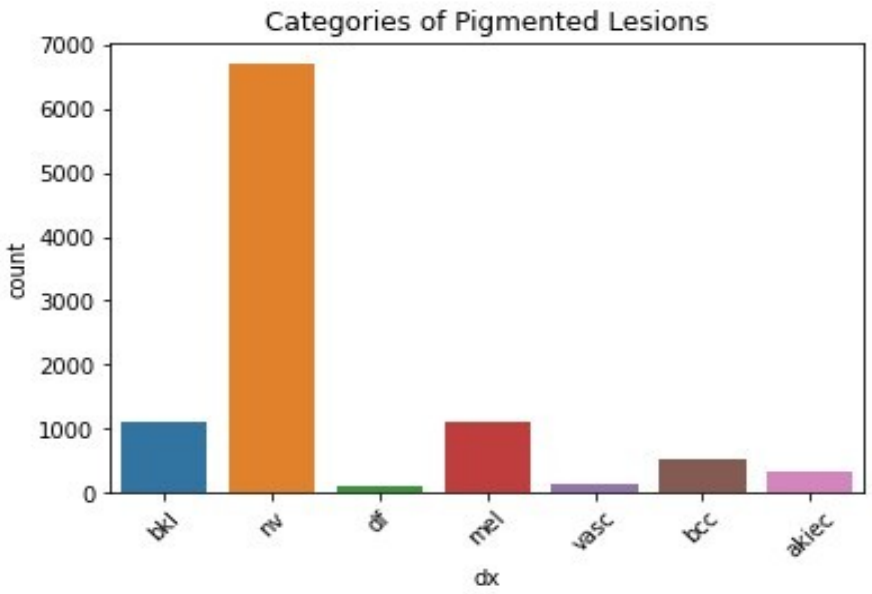
After importing all the required libraries metadata is loaded into a pandas dataframe called,df.The dataframe consist of 10015 rows and 7 columns.The 'dx' column

	lesion_id	image_id	dx	dx_type	age	sex	localization
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear
...	...	...	...	...	...	...	...
10010	HAM_0002867	ISIC_0033084	akiec	histo	40.0	male	abdomen
10011	HAM_0002867	ISIC_0033550	akiec	histo	40.0	male	abdomen
10012	HAM_0002867	ISIC_0033536	akiec	histo	40.0	male	abdomen
10013	HAM_0000239	ISIC_0032854	akiec	histo	80.0	male	face
10014	HAM_0003521	ISIC_0032258	mel	histo	70.0	female	back

10015 rows × 7 columns

represents seven classes of Lesions.

akie represents Actinic keratoses and intraepithelial carcinoma / Bowen's disease, bcc represents basal cell carcinoma,bkl represents benign keratosis-like lesions (solar lentigines / seborrheic keratoses and lichen-planus like keratoses,df for dermatofibroma,melanoma (mel),melanocytic nevi (nv) and vascular lesions (angiomas, angiokeratomas, pyogenic granulomas and hemorrhage (vasc).



Out of seven classes of lesions,melanocytic nevi (nv) holds a higher rate(6705) and dermatofibroma (df) have a lowest number(115).After examining the seven classes, it is evident that a significant imbalance exists. To address this issue, resampling techniques are employed. In this approach, 500 samples are collected from each class, and if desired, upsampling or downsampling methods are applied.

```
#fetchhing images and combining them into one folder
image_path={os.path.splitext(os.path.basename(x))[0]:x for x in glob(os.path.join('C:\\softnerve\\HAM100000','*','*.jpg'))}
```

```
df_balanced['path']=df['image_id'].map(image_path.get)
df_balanced['image']=df_balanced['path'].map(lambda x: np.asarray(Image.open(x).resize((size,size))))
```

code fetches the image files from the directory, maps the image IDs to their respective file paths, and then loads and resizes each image into a NumPy array, which is stored in the 'image' column of the new dataframe,'df_balanced'. This allows the images to be easily used as input for training the CNN model.

```
#feature and target selection
x=np.asarray(df_balanced['image'].tolist())   #array of uint8
x=x/255  #max pixel is255   #array of float64
y=df_balanced['label']   #series
# for a multiclass classification structure should be changed to categorical values
y_cat=to_categorical(y,num_classes=7)

#splitting and training data
x_train,x_test,y_train,y_test=train_test_split(x,y_cat,test_size=0.25,random_state=42)
```

this code prepares the feature and target data for training the model. It converts the image data into a NumPy array, normalizes the pixel values, converts the target values into categorical values, and splits the data into training and testing sets.

```
num_classes=7
model=Sequential()
model.add(Conv2D(256,(3,3),activation='relu',input_shape=(size,size,3)))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.3))

model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.3))

model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.3))
model.add(Flatten())

model.add(Dense(32))
model.add(Dense(7,activation='softmax'))
model.summary()

model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['acc'])
```

A CNN model is built with multiple convolutional layers, pooling layers, dropout layers, and fully connected layers, ultimately outputting class probabilities for the input data.

```
28/28 [==============================] - 7s 252ms/step - loss: 0.7107 - acc: 0.7771
test accuracy 0.7771428823471069
```

After fitting the model,shown a good test accuracy of 77.71.

After plotting the training and validation loss ,there is no bigger overfitting going on . Both the training and validation are performing well.