Project Name: - Lice classification using transfer learning models

Name: - Aarti Darji

1. Introduction

   My project focuses on image classification using transfer learning models. A transfer learning model is when we use a pre-trained model rather than a new model that is trained on small dataset. So, what is the challenge with classifying lice?

   a. Size and morphology: Lice are typically less than 3 mm long and have a flattened, elongated body shape that allows them to move quickly and easily through the hair.

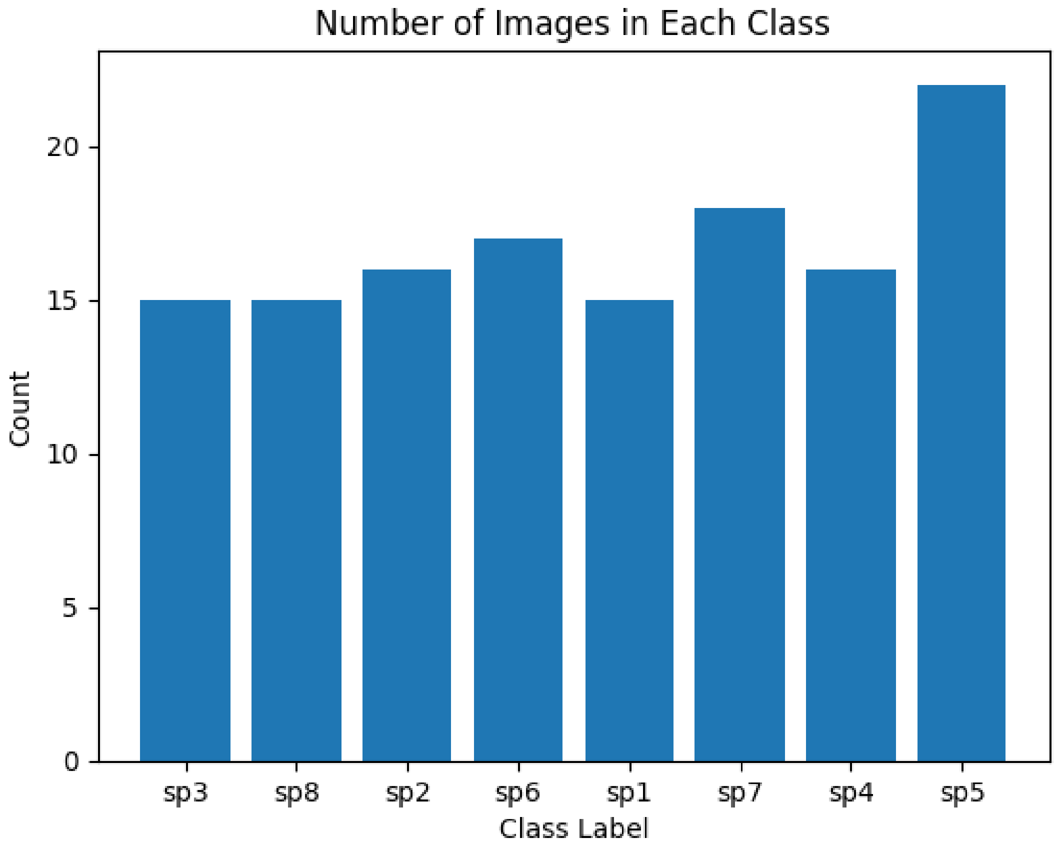
   b. Similarity: There are several species of lice that can infest humans, including head lice, body lice, and pubic lice.

   c. Manual identification: Manual identification of lice requires a high level of expertise and often involves the use of specialized equipment, such as microscopes and magnifying lenses.

   By using a transfer learning model, we can achieve higher performance.

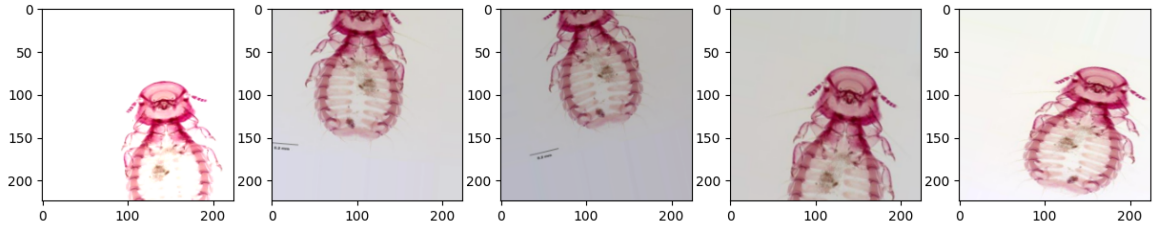
2. Dataset

   The dataset I'm using is the lice data set provided by Dr. Rostami. The lice are classified by 8 species depending on their size and morphology. It is of JPG format and has 134 images of 8 classes.



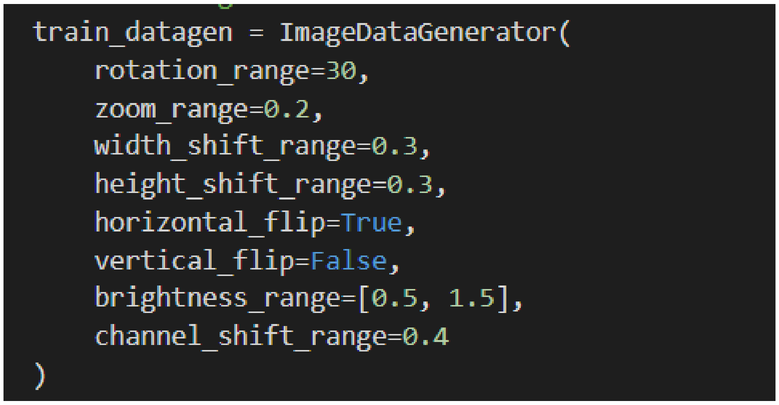
3. Data Preparation

   For the data preparation part of my project, I used data augmentation to get multiple data points of the lice images with different alterations, this reduces the overfitting problem. Then, I reshaped the images to fit the model. After increasing the size of the dataset, I can split it into a training dataset and test dataset.



4. Exploratory Data Analysis (EDA)

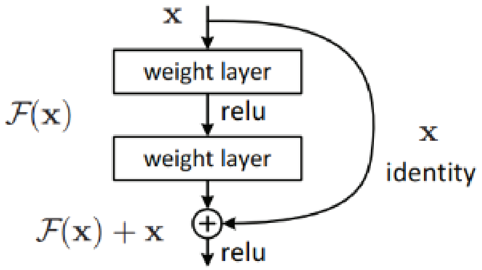
   For my EDA, since my data was already provided to me processed I just worked on the data augmentation part and using my image generator to create new images with alterations to them. In the below image, you can see the metrics I have specified.

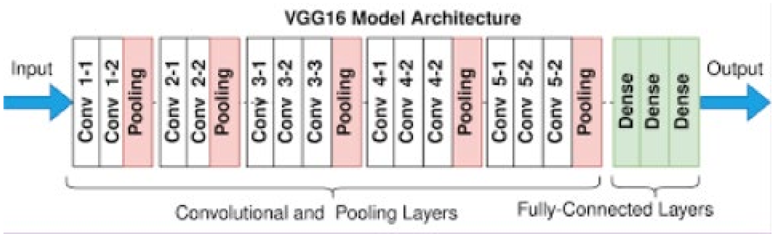


```
train_datagen = ImageDataGenerator(
    rotation_range=30,
    zoom_range=0.2,
    width_shift_range=0.3,
    height_shift_range=0.3,
    horizontal_flip=True,
    vertical_flip=False,
    brightness_range=[0.5, 1.5],
    channel_shift_range=0.4
)
```

5. Model selection and training

I used pre-trained neural network models (transfer learning models) for faster results and maximum optimizations. The ones I used are VGG and ResNet. VGG is deep convolutional neural network that can support either 16 or 19 layers and can have up to 1000 categories/classes. It is one of the most popular model used for image recognition. ResNet is also a convolutional neural network used for deep learning. It is different in the sense that it has something called 'skip connections' which serve as a solution to the vanishing gradient problem.



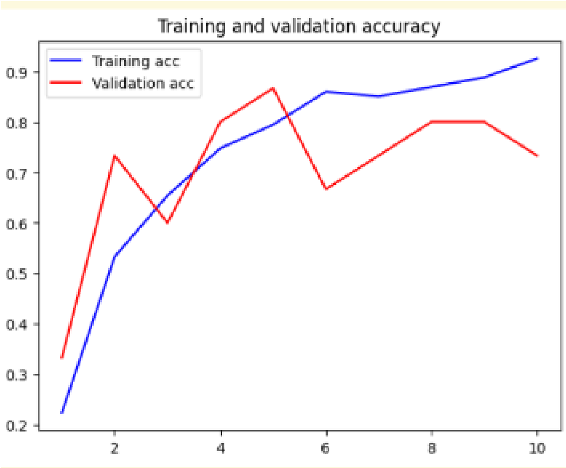


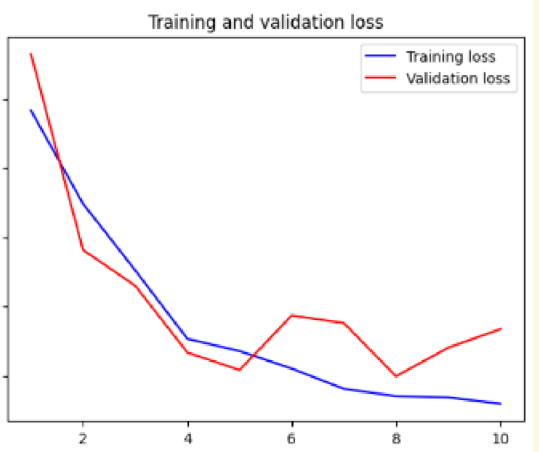
I trained each model for 10 epochs and used the Adam optimizer and the cross categorical entropy for my loss function.

6. Model Evaluation

For evaluating the model, I plotted the training and validation loss and accuracy to visualize my model's performance. Furthermore, I tested my model on my test data for its loss and accuracy.

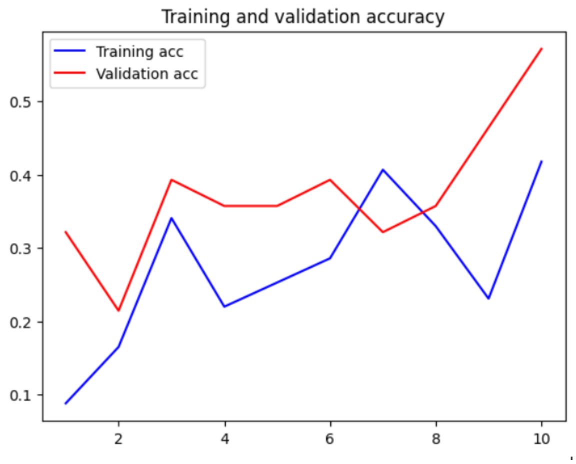
a. Resnet Performance

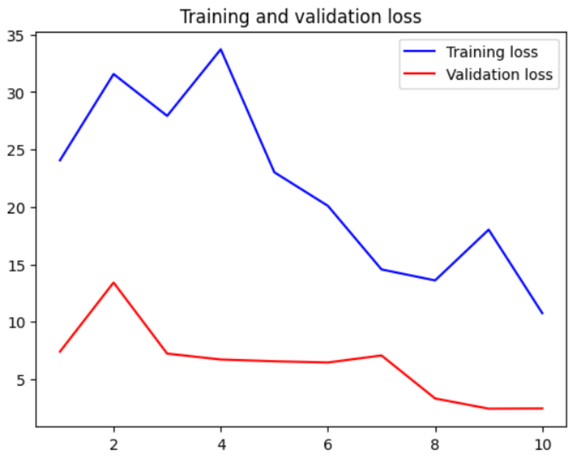




I got a test loss of 0.0363 and test accuracy of 0.0833.

b. VGG-16 Performance





I got a test loss of 4.69 and test accuracy of 0.53.

7. Limitations and future work

While my ResNet model had a good performance I wasn't able to get the same from VGG-16. I believe some of the reasons why I faced this error was due to bias in the model and the size of my dataset. I believe moving on some of the things I can do are training the dataset for longer epochs. I only trained them for 10 but I can increase that to 20 or 30 if needed. Second, I can increase the complexity of my model or perhaps unfreeze some of the top layers to get better accuracy. I can also try obtaining a bigger dataset if available and see how that performs. I would also like to use some different pretrained models such as VGG-19 or AlexNet.