Pattern lab1

## Done By: Ahmed Nasser – 8350 Baher Ahmed Hosny – 8165

**Code:**

A screen shot of a computer program

AI-generated content may be incorrect.

**Step1:** We load the data and resize all the images to the same size so we can use neural networks to create classify the data. Rotation and flipping of the images are to improve the diversity of the training set which causes the model to return better results.

A screen shot of a computer program

AI-generated content may be incorrect.

**Step2:** Cropping the image is also part of the augmentation process which improves diversity in the dataset. The first loop loads the images and labels. There is an extra part in the first loop, we added testing which chooses a random picture and displays it after converting to RGB and resizing it. The second loop loads all images into the dataset after resizing them and converting image to RGB. There are 2 augmented versions of each image added to help the model train, if we add more versions, it could lead to overfitting if the versions aren’t that different to the original version.

A screen shot of a computer program

AI-generated content may be incorrect.

**Step3:** We normalize the data by dividing the array which is cast to float. This is also part of the preprocessing so we can avoid outliers and because neural networks work better with smaller ranges. We shuffle the data so when we split, the data isn’t skewed. Then we split the dataset 80% training and 20% validation.

A screenshot of a computer

AI-generated content may be incorrect.**A screenshot of a computer program

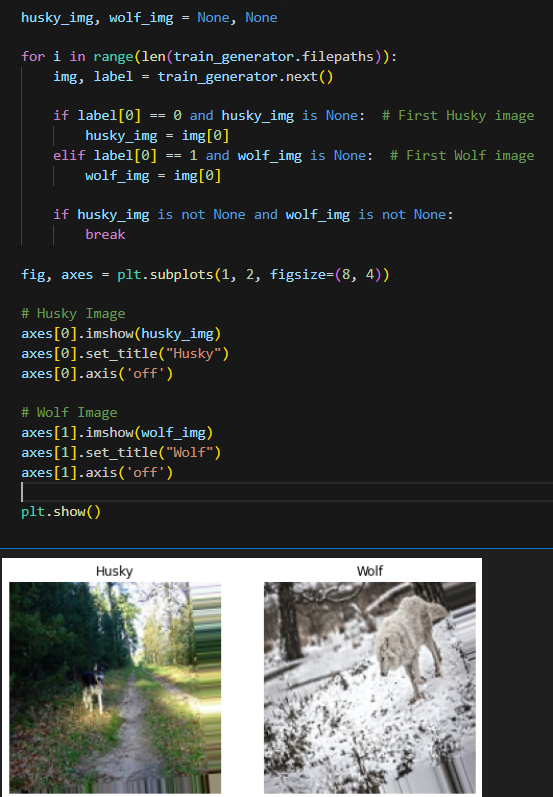
AI-generated content may be incorrect.Step4:** We implemented convolution neural networks, we used RELU as the activation function so model can learn complex features faster. The input shape determines the size of the image with the number of color channels (3). The number of filters is to help us extract features from the image. Maxpooling is to help prevent overfitting by reducing spatial dimensions of the feature map. We flatten the layers to a 1D vector so we can use the dense function can work on the data. Dense function has 128 neurons and helps learn about relationships between features. Dropout layers also helps prevent overfitting. We compile using crossentropy as loss.

**Step5:** We use ImageDataGenerator to augment, preprocess, and split image data.

A screen shot of a computer program

AI-generated content may be incorrect.

**Step6:** We start setting up data generators for the ImageDataGenerator function. We use flow\_from\_directory() because it automatically labels the images and reduces memory usage.



**Step7:** Here we load an image from the training dataset using the train\_generator.

A screenshot of a computer screen

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.A screenshot of a computer

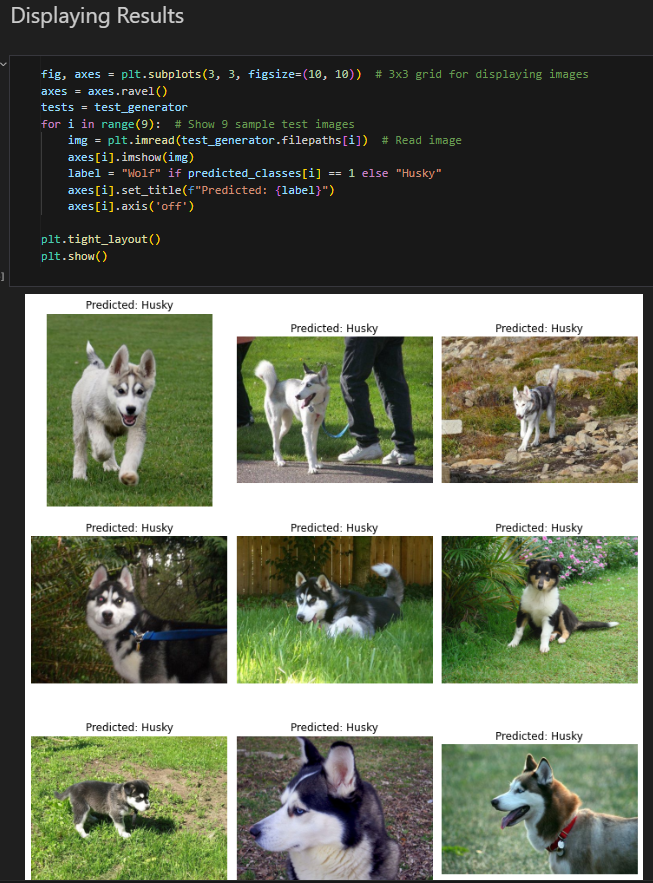
AI-generated content may be incorrect.

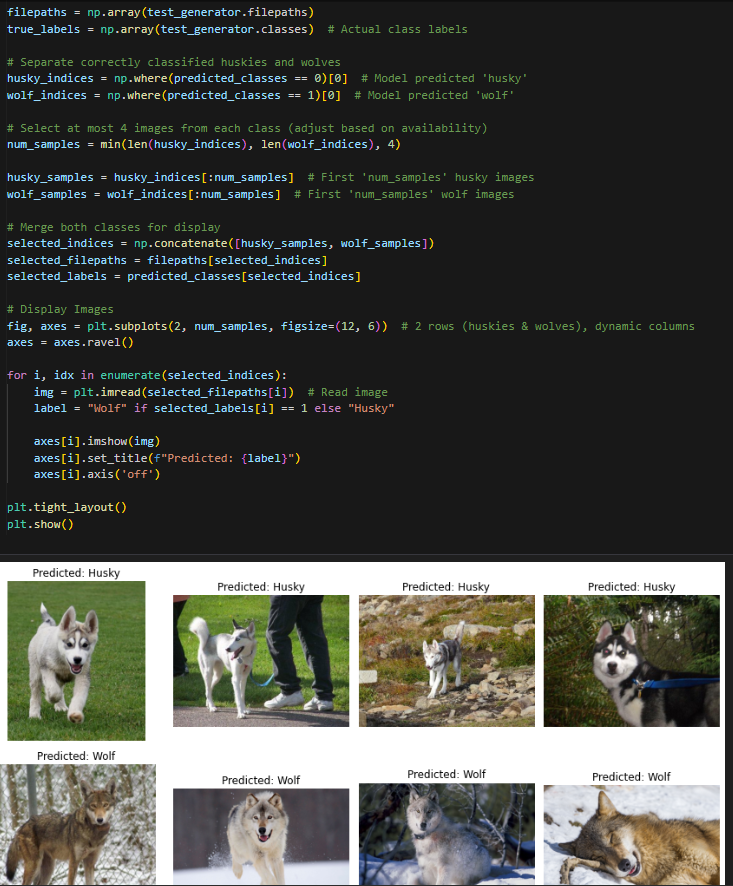
A screenshot of a computer

AI-generated content may be incorrect.**Step8:** We do hyperparameter tuning on the kernel size, we try different values for the kernel size and calculate the accuracy and loss and we found out that kernel size (3,3) produces the best results.

A screenshot of a computer

AI-generated content may be incorrect.

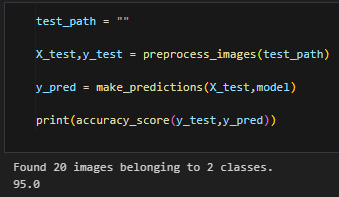
**Step9:** We train the CNN model using the training and validation data.



**Step10:** We use the test data to display some images

**A screen shot of a computer program

AI-generated content may be incorrect.**



**Step11:** We calculate the accuracy using the CNN model and the test images. We loop through the batches and convert probabilities to class labels.

**A screenshot of a computer program

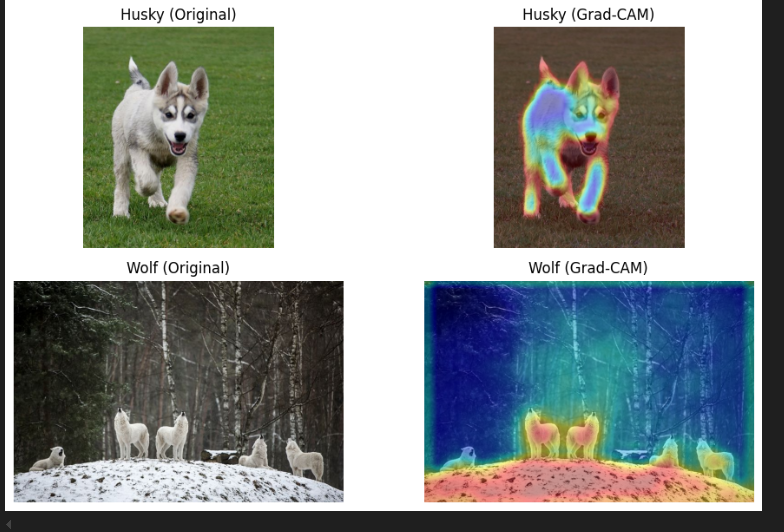
AI-generated content may be incorrect.Bonus:**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Data Leakage:** This occurred because near duplicate images exist between train and test data. This happens when images from the same photoshoot are split between train and test data. The same photoshoot causes most pictures to have the same lighting, background, etc. This is a bad because the model might memorize specific features like the background or lighting. In the wolf data, since all the pictures have snow, it is used to help identify wolves even though the background has nothing to do with wolves. This might cause the model to overfit.