DOG BREED IDENTIFICATION

UKU TONSIVER

Some background

In 2017, a data science community website Kaggle hosted a machine learning competition called Dog Breed Identification. The aim of the participants was to build a machine learning model that can accurately determine the breed of the dog from an image. Although the competition concluded several years ago, I decided to also give it my best shot.

Model introduction

I decided to merge three pretrained neural network architectures together and train the model on my datasets. After considerable research, I chose to use the following architectures:

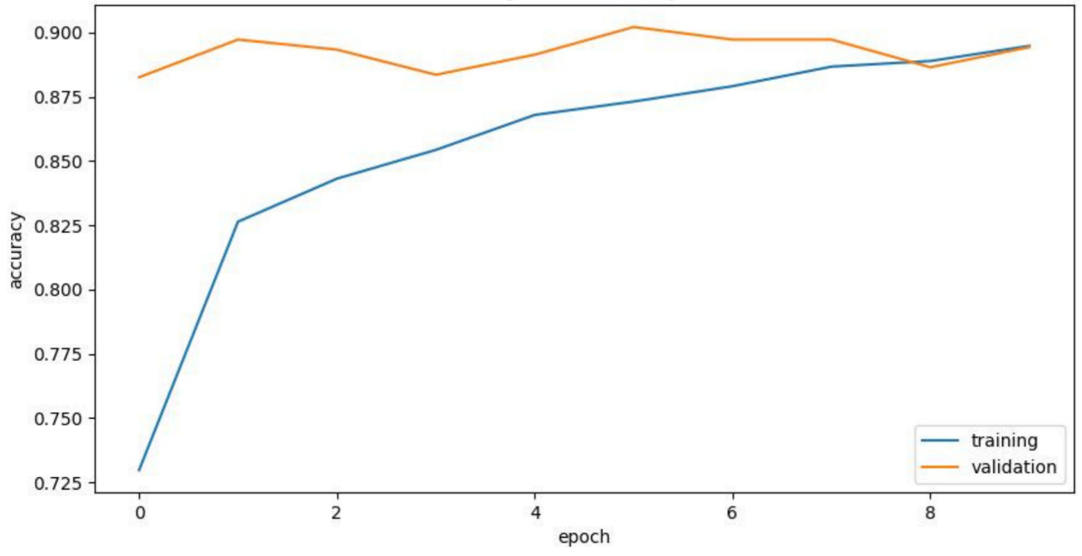
- Inception v3 - the latest convolutional deep learning neural network architecture in the Inception family
- Xception - a convolutional deep learning neural network architecture that uses depthwise separable convolutions, further development of the Inception family
- NASNet - a machine learning model that searches for the best neural network architecture for a given task

All of the abovementioned architectures have shown useful in image classification. The models were pretrained on ImageNet databases.

Results

As to quantify the results of training the model, I measured the final accuracy and and Cross-Entropy loss of the validation set. I did not have access to the correct labels of the test set, but Kaggle calculated the Multi Class Log loss of my submission to be approximately 0.606. The accuracy and loss on the validation dataset reached 0.89 and 0.65 respectively (see figures 2 and 3).

Figure 2. Accuracy

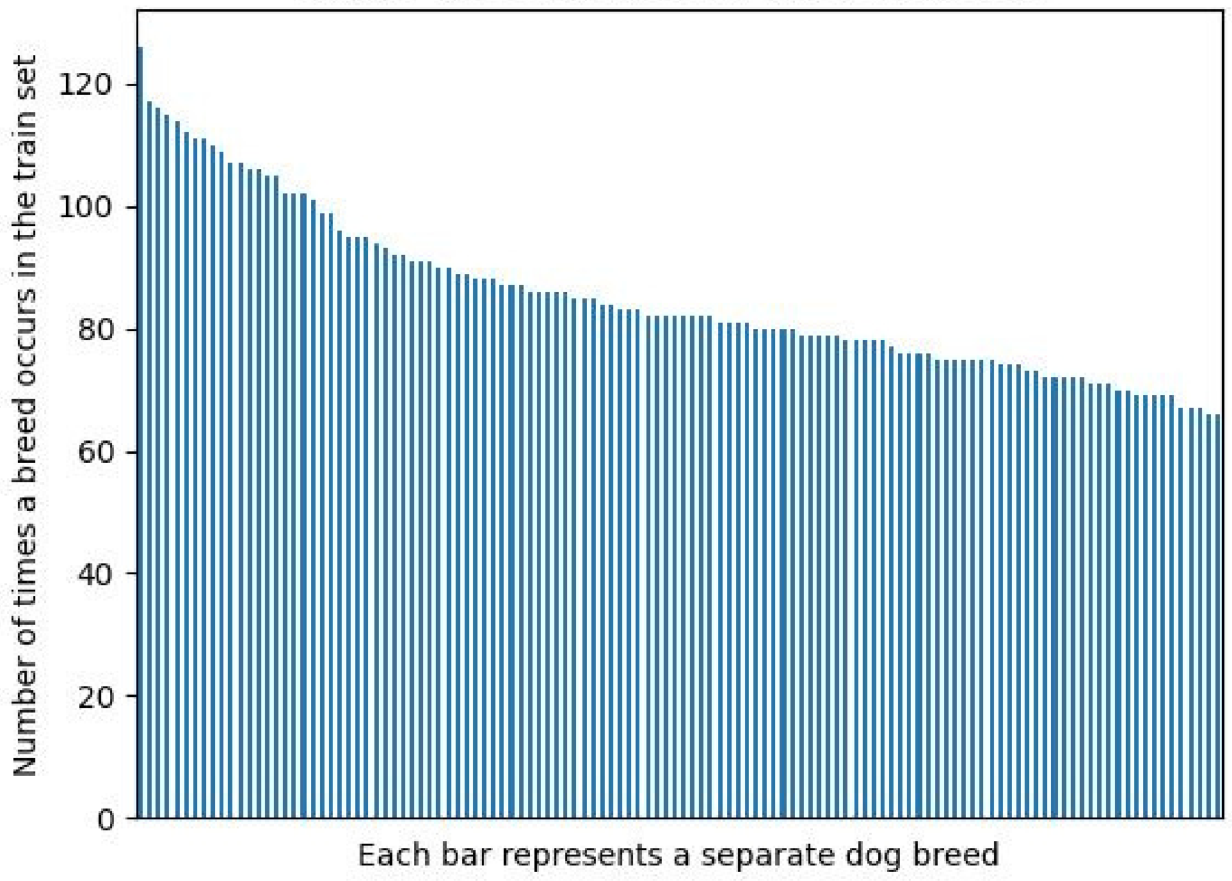


Data and preprocessing

The data provided consisted of train set and test set of images of dogs, each having approximately 10 000 images. There was additionally a .csv file for the labels (dog breeds) of the images in the train set. Preprocessing included dividing training images into directories corresponding to their labels. From there, image datasets were constructed from the directories.

There were 120 different breeds of dogs. There were differences of representations of different breeds (see figure 1), but all breeds were represented sufficiently for training the model.

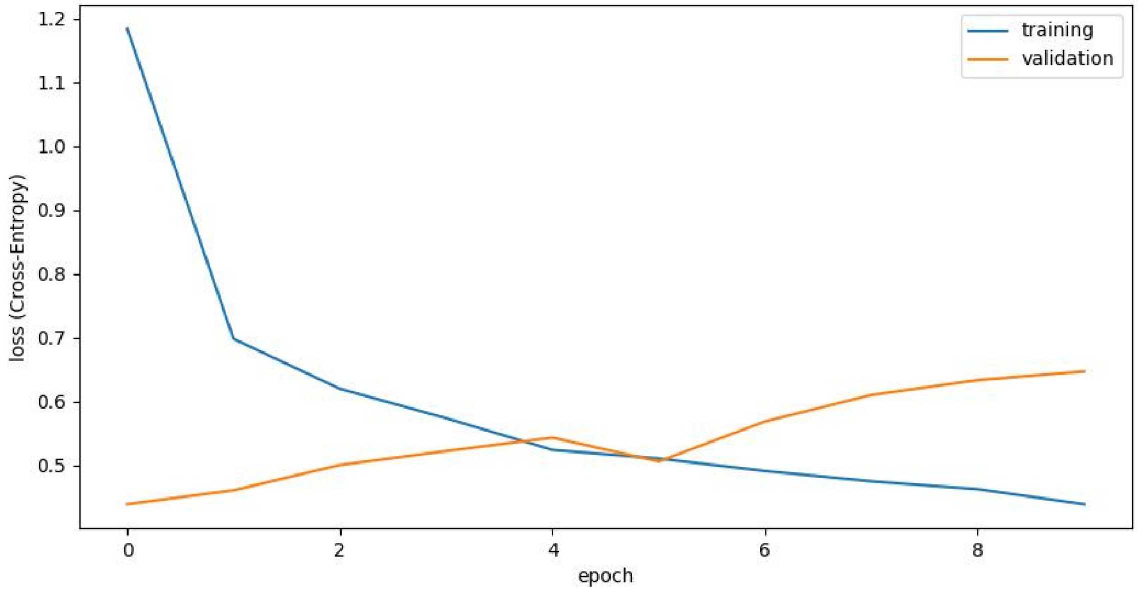
Figure 1. Occurrences of different breeds



Training process

After applying a couple optimization techniques, I trained the model on train set. I used 90% of the images for training and 10% for validation. With the number of epochs set as 10 and batch size set as 32, training the model took approximately 12 hours.

Figure 3. Loss


loss (Cross-Entropy)
epoch
training
validation