Final Project Report

CSC480-05
Artificial Intelligence
Dog Breed Identification

Team Dogs

Group Members:
Alvin Chui
Robert Hensley
Justin Cho
Toan Pham

Instructor:
Professor Hisham Assal

- view runnable project code in this Google Drive directory
    - Jupyter notebooks (.ipynb files) can be run in browser by double clicking the file and selecting 'Open in Google Colaboratory'
    - Additional code can be found in the research directory

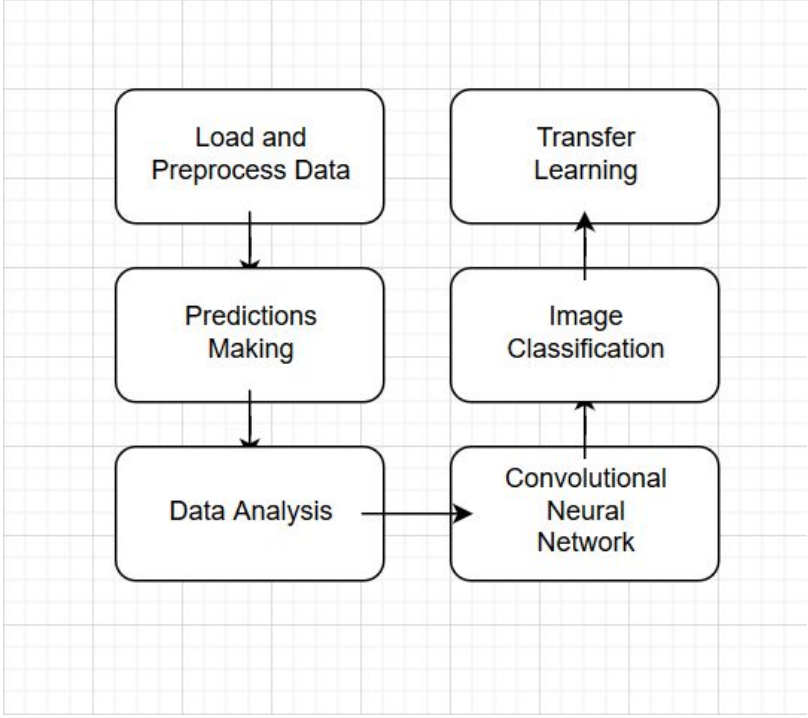
Project Description:

(Main Idea) The goal of the project is to study machine learning by exploring convolutional neural networks (CNN) and pretrained models. Thousands of dog images with corresponding dog breeds will be obtained from Kaggle, and the modeling will be performed on Google Colab. The team will start from a simple model, make some predictions, and improve the dog breed's test accuracy as it progresses.

(AI Relations) Machine Learning is an application of AI that provides system the ability to perform and execute tasks "smartly" and efficiently. It mimics the human's ability to learn and see the world. In addition, the use of Image Classification allows the system to read and "view" its environment, which is the interface of an intelligence.

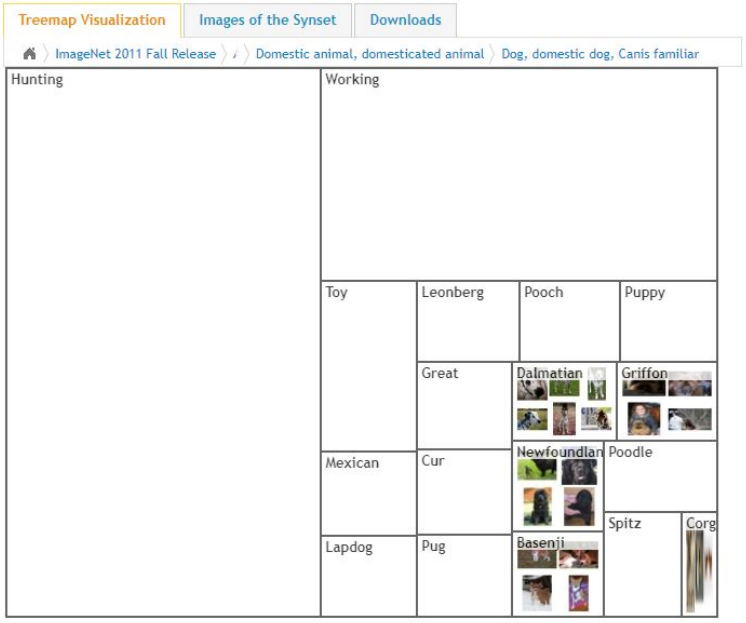
(Key Terms) CNN - Convolutional Neural Network,
ML - Machine Learning, TL - Transfer Learning

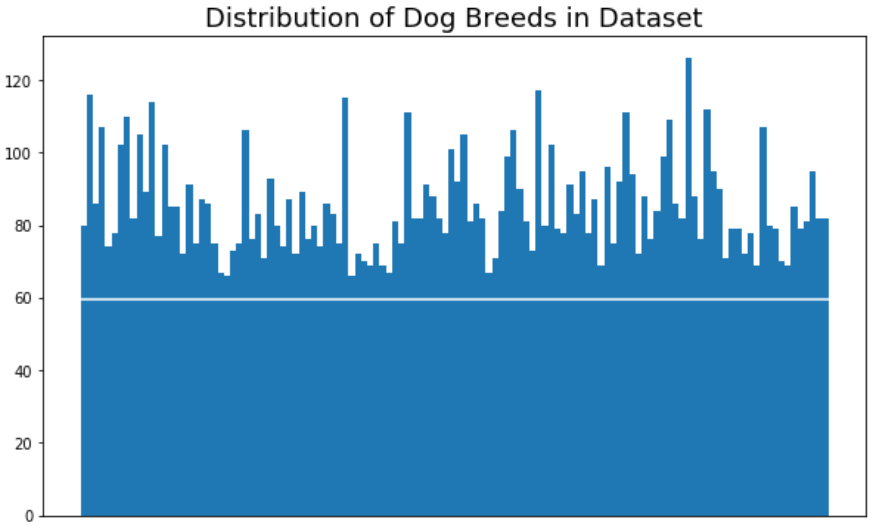
(Learning Process)



Data:

(Dataset) The dataset we used for this project is part of the Dog Breed Identification Challenge on Kaggle. It's a challenge created at Stanford with about 10,000 training images and 120 labels (unique breeds) and is a subset of the ImageNet classification challenge, which contains several million images and many more labels.





(Tensors) In order for our machine learning model to process our data, we needed to convert the images into n-dimensional arrays of float values. These are called tensors. We explored this in our Data Analysis notebook.

We created numpy arrays of shape (n, 60, 60, 3) in our data analysis notebook.

- the variable *n* represents the amount of images in the dataset
- the two 60's represent the height and width of the picture in pixels
- the number 3 represent the RGB values of the pixel, defining its color

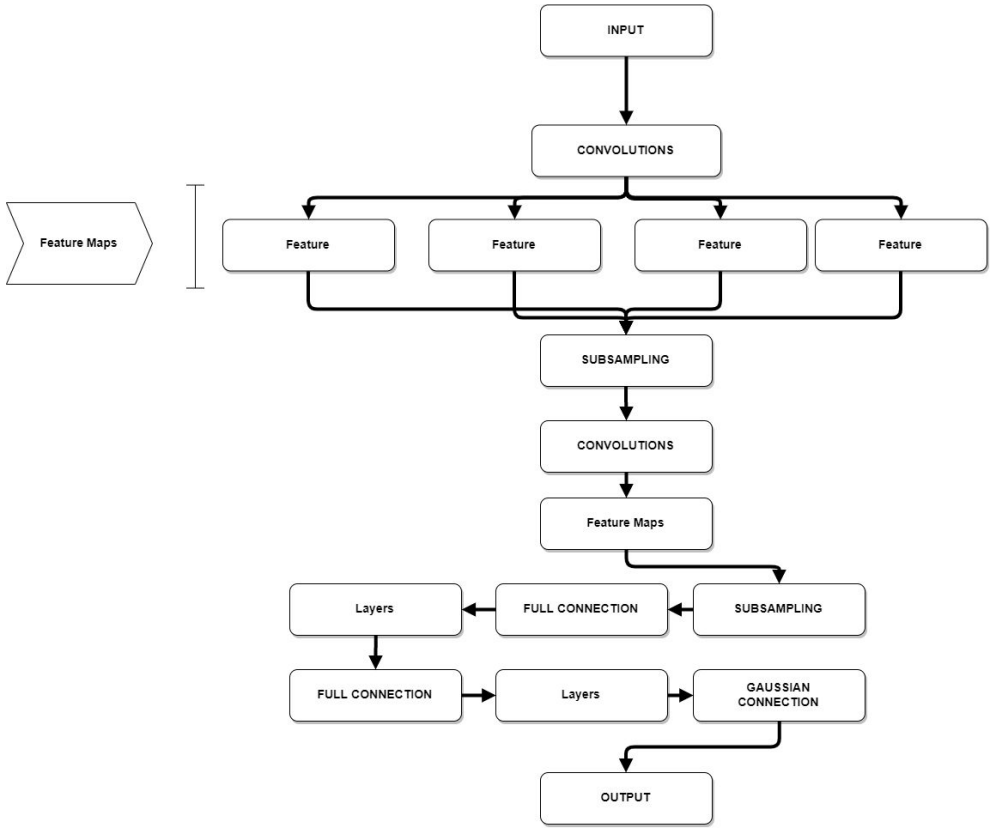
Here is what a 60x60 tensor looks like when it is plotted on a graph next to its original higher resolution image. In our data augmentation, we stretch the image into a square. With more time, we could improve our data augmentation by cropping the image around the dog rather than stretching the entire image.

For the machine learning algorithm we wrote in our *Machine Learning* notebook, we use slightly more abbreviated image processing techniques and create a higher resolution image (96 x 96).

Project Design:

To train and test our model, we used the Dog Breed Identification dataset on Kaggle, which contains 10223 labelled dog images, with 120 unique breeds. We used a 90-10 split to train and test our model.

(General Architecture for CNN) Convolutional layers filter and reduce computation, forcing a break of symmetry in the network



Implementation:

(Language) Python

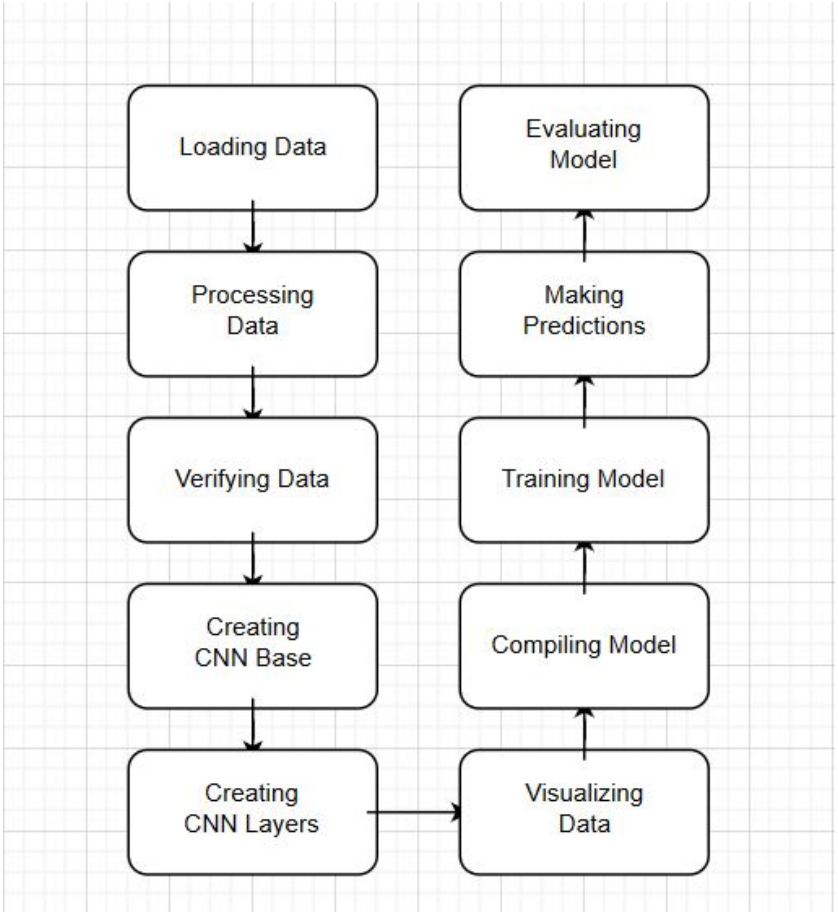
(Drive Mount) GoogleColab notebook allows users to mount data from Google Drive

(TensorFlow) TensorFlow is an open source machine learning framework for dataflow programming that is developed by Google. The word “Tensor” represents multidimensional arrays while the word “Flow” contributes to the system control flow. TensorFlow provides open source library that would allow users to develop and train their machine learning models.

(Frameworks) Using Keras on top of Tensorflow

(Imagine Classification) Imagine Classification refers to the process where an image is extracted into information that can be processed by machines or computers.

(Flow Diagrams) General implementation steps and processes



(Model Architecture)

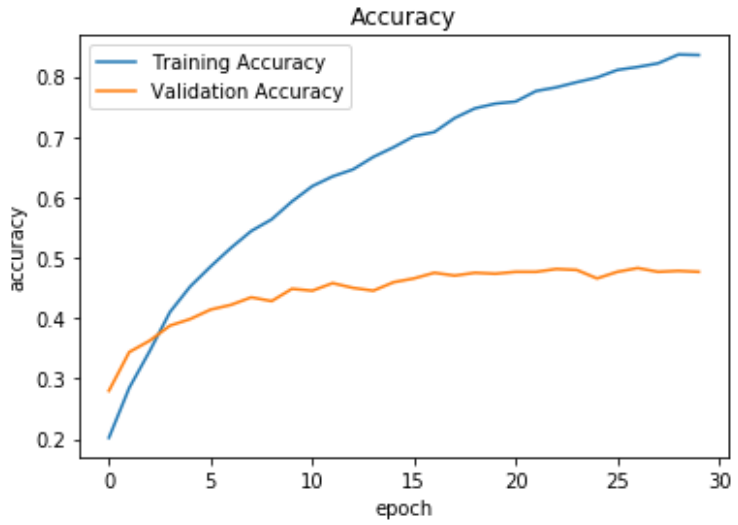
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
mobilenetv2_1.00_96 (Model)  (None, 3, 3, 1280)        2257984
_________________________________________________________________
global_max_pooling2d (Global (None, 1280)              0
_________________________________________________________________
dense (Dense)                (None, 120)               153720
=================================================================
Total params: 2,411,704
Trainable params: 153,720
Non-trainable params: 2,257,984
_________________________________________________________________
```

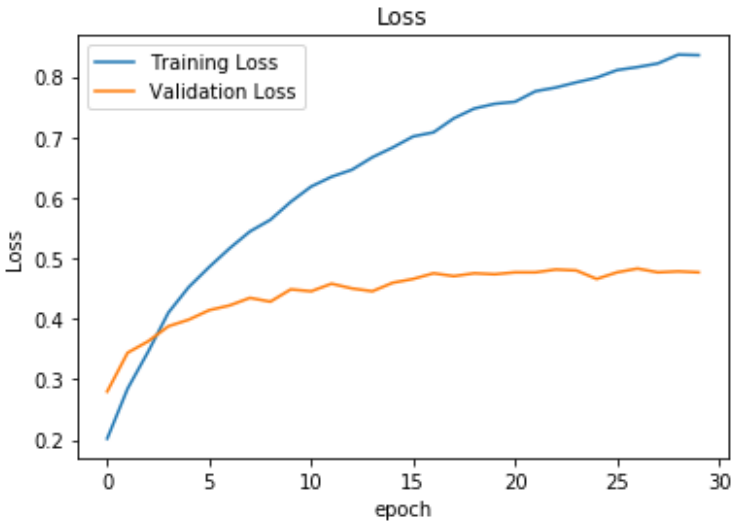
For the sake of this project, we would like to create a machine learning model that is a bit more lightweight in terms of required computing resources. So we created a simple Neural Network architecture that uses a mobilenet architecture with ImageNet weights.

We chose a mobilenet architecture because it is known for its compute efficiency. We also choose to use transfer learning with ImageNet weights because the Stanford Dogs Dataset is a subset of the ImageNet challenge, so it's appropriate to use weights that work well with the features of that dataset.

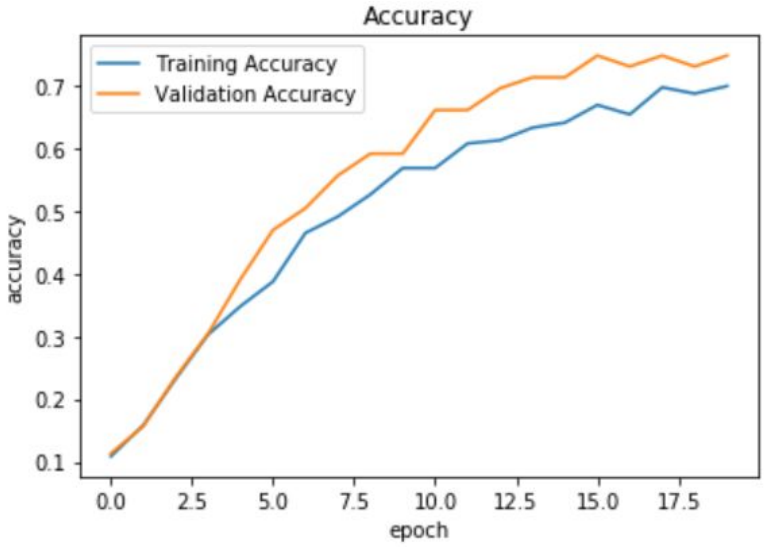
Our model ends with a MaxPooling layer that feeds into a dense layer with 120 neurons. These 120 neurons represent each class in the dataset and their activation represents the probability that an image is that particular class.

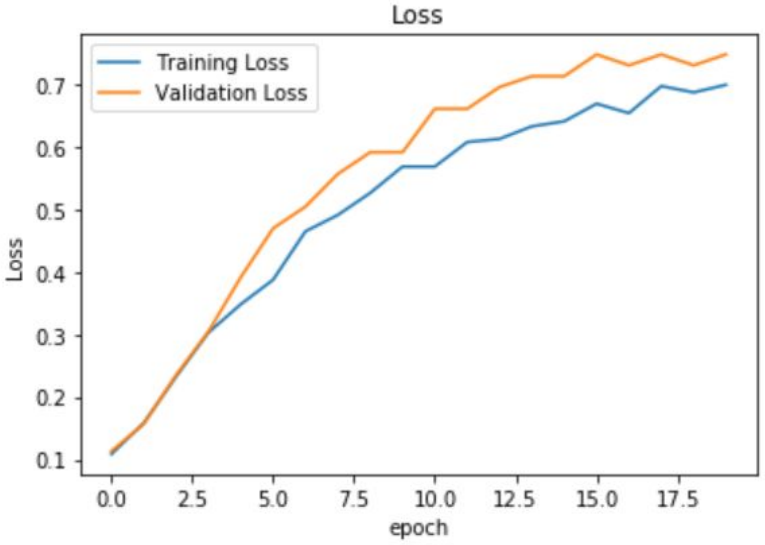
Analysis:





Our machine learning model is pretty simple with few layers so the point at which the training accuracy becomes higher than validation accuracy occurs pretty early on (at around 3 epochs). Validation accuracy hovers around 45%.





After getting a low accuracy rate, we strengthened our model by reducing the number of dog breeds, classifying them into groups. The number of dog breeds is reduced from about 120 to ten groups. Next, we improved our model by changing the layers. As a result, the validation accuracy increases significantly.

Problems:

Difficulty #1

Description:

TensorFlow 1.15.0 version doesn't support casting a string to float during modeling.

Screenshot:

Solution:

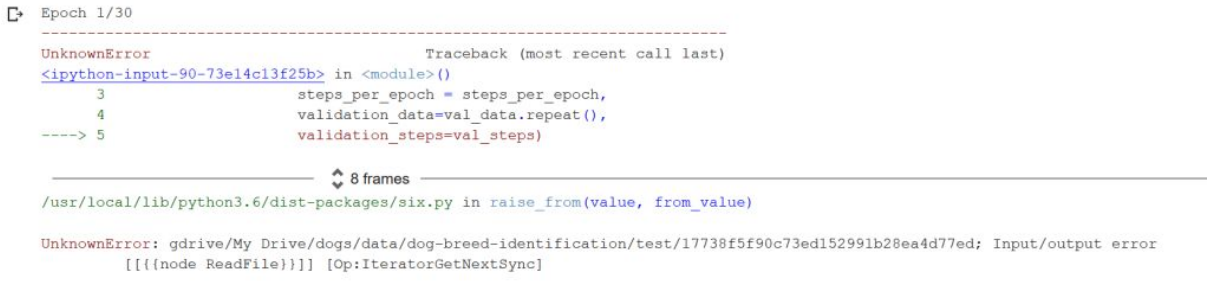
1) Encode your data into a vector or float data type.
2) Upgrade TensorFlow to the newest version 2.0.0. This version would support the casting of string to float during modeling.

Difficulty #2

Description:

An unknown error would occur when the test data are enormous. GoogleColab notebook couldn't process the data and produce a hidden time-out error. Our team spent a lot of time figuring the error.

Screenshot:



Solution:

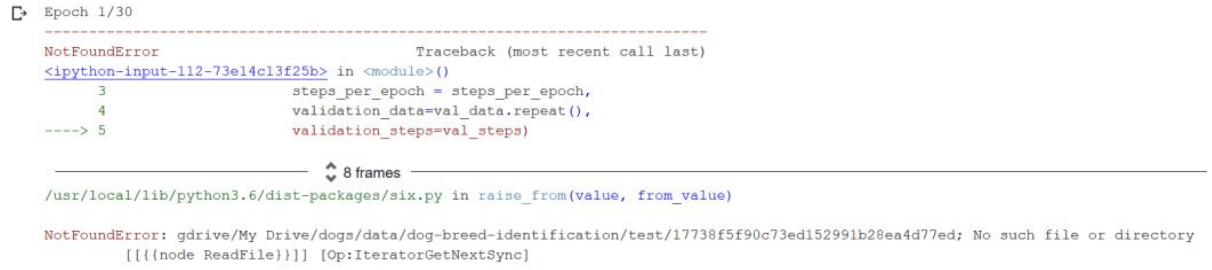
1) Reduce the size of the data
2) Switch to a local machine for learning and testing

Difficulty #3:

Description:

The file couldn't be detected by GoogleColab even though it was there.

Screenshot:



Solution:

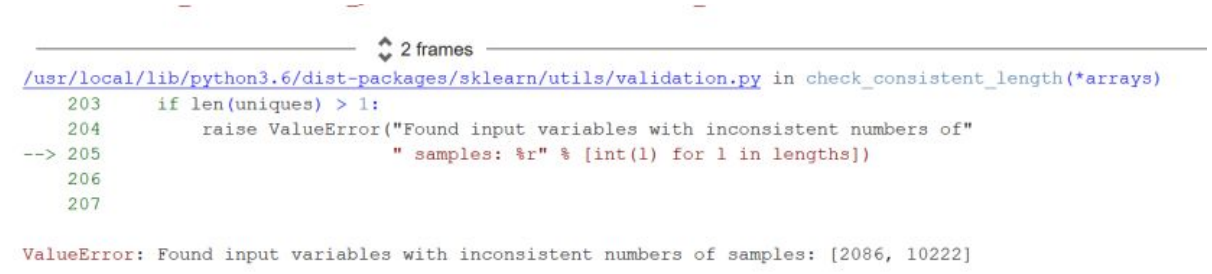
1) Switch to a local machine for learning and testing
2) Reset your data and update your notebook

Difficulty #4

Description:

When the number of data in a directory doesn't match the number of data listed in a csv file, modeling error would occur.

Screenshot:



Solution:

1. Prepare another set of data for learning and testing
2. Go through the dataset and adjust the data
3. Look for an alternative modeling style

Conclusion:

(*Improvements*) Here are a few ways we could improve our model in the future

1. Increase the amount of layers
2. Test several models for transfer learning
3. Include more data augmentation techniques, this includes:
    a. Cropping the image on dogs (this removes background noise and focuses attention on the subject)
    b. Rotating the image

(Suggested Learning Path)

1. Machine Learning Basics (Basic concepts like: Training / Testing / Validation; maybe start with K-Means regression)
2. Neural Network Basics (construct a simple one in Python)
3. Simple Computer Vision Task (MNIST)
4. Binary Image Classification Problems (Cats vs. Dogs)
5. Larger Image Classification Problems (like N-Dog Breeds)

(Suggestions) Here are our suggestions for students who are going to take the Introduction to Artificial Intelligence course before choosing their project:

- Set up the learning and testing environment on a local and reliable machine
- Get data from a reliable source for testing and machine learning
- Set a short-term goal before reaching for a long-term
- Study a concept or definition with coding samples
- Have fun and go through lots of tutorials

(Bibliography)

- https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
- https://keras.io/
- https://www.kaggle.com/c/dog-breed-identification

Notebook:
https://colab.research.google.com/drive/1fDcokwYaNSr9MNM-q-DvYph8f3EgxJqs