

## Dog Breed Classification

### 1. Domain Background

In this project, I'm going to implement and develop a dog breed classification web app. It will be a simple Web application which consists of an upload image option, where the user can upload an image as per his dog's image and the web will correctly classify the breed of the dog. The problem of classification is a very important domain in the field of machine learning and deep learning, and it is a heated topic in this world. My personal motivation for this project has two perspectives, first is I want to do this project to learn classification problems at the deepest level. And second is I love dogs, so when I saw the project option I chose it readily.

### 2. Problem Statement

Classification of dog breeds using the datasets given. We will classify the dog breed using the datasets given and will create a web application using aws sagemaker. We will first get the datasets, then we will process and explore datasets, then we will create a cnn model using pytorch and then we will evaluate our model on different test datasets, and finally we will deploy it on web html page using lambda function and aws api gateway.

### 3. Datasets and inputs

Datasets will be taken from udacity, and then we will process data sets in grid to see labels and dog breed, missing data will be removed from datasets and finally we will split data sets into train, valid and test sets.

### 4. Solution Statement

For this problem we will create a CNN model using pytorch, then we will evaluate the model and finally we will deploy using aws lambda function.

## 5. Benchmark Model

For benchmark model we will use alexnet, resnet, vggnet and other best recognized models in the domain of CNN.

## 6. Evaluation Metrics

We will evaluate the benchmark model based on test accuracy, recall and precision methods.

## 7. Project Design

Workflow will be as follows:

- a. Import datasets
- b. Detect image
- c. Create a CNN model using pytorch
- d. Train the model
- e. Evaluate the model
- f. Create html file to upload images to s3 bucket
- g. Create lambda function
- h. Connect lambda function to api gateway
- i. Put the link from api gateway to html file
- j. Predict the images
- k. Create test metrics to compare our model with other models

Other options will be considered which says if human face is detected we will classify to some breed resemblance.