Cats vs Dogs Detector (CaDoD)

Abstract:

Object detection, one of the basic tasks of computer vision, deals with classifying the image based on its content. In recent years, the approach of object detection has evolved rapidly and is being embraced in all fields ranging from healthcare to automotive industries.

The objective of our project is to classify images as either dogs or cats. Additionally, we also plan to find where the cat/dog is in the image. Although the task is simple to human eyes, computers find it hard to distinguish between images because of a plethora of factors including cluttered background, illumination conditions, deformations, occlusions among several others. We plan to build an end-to-end machine learning model which will help computers differentiate between cat and dog images with better accuracy.

The data that we plan to use is the CaDoD Kaggle data set. It contains a total of ~13k images of dogs and cats. We plan to classify the images using our baseline model, Logistic Regression, and compare it with other learning models like Random Forest and Convolution Neural Networks. In the above classification models, our goal is to minimize the cross-entropy loss. Furthermore, we plan to use linear regression for the prediction of the position of cats and dogs. In all the above models, we plan to use k-fold cross-validation to fine-tune the hyperparameters.

To measure the model's performance, we plan to use accuracy and Mean F1 score (harmonic mean of recall and precision)

Data Description:

The data we plan to use is the Kaggle data set cadod.csv. There are about ~13k images in the data set. There is a good balance of classes with ~6.8K cat images and ~6.1K dog images in the data set.

Attributes:

- There are about 21 features in our data set:16 numerical features
- 5 categorical features

Preprocessing steps:

- Defining the image properties
- Rescaling the images
- Preprocessing the images using data augmentation (flip, zoom, rotation)

Machine Learning Algorithms and Metrics:

The following ML models will be used:

1. Object Detection: We will use classification models for object detection
    a. Logistic Regression - Baseline Model
        i. Implementation – Logistic regression will help us get the likelihood of events by looking at the historical data. We use the sigmoid activation function to convert the number line to a [0,1] range values. We then use these values (probabilities) to decide the classes
            1. Package Used - sklearn.linear_model.LogisticRegression
        ii. Optimization using Batch Gradient Descent and Stochastic Gradient Descent
        iii. Loss function to be minimized - Log loss
    b. Random Forest
        i. Implementation - Uses an ensemble of decision trees and classifies the images based on the class selected by most trees
            1. Package used - sklearn.ensemble.RandomForestClassifier
        ii. Hyperparameters to be tuned - Bootstrap, maximum depth, maximum features, number of estimators, minimum number of sample splits
        iii. Loss function to be minimized - Gini impurity for each decision tree
    c. Convolution Neural Network
        i. Implementation - CNN (Convolutional Neural Networks) is a class of deep neural networks that is used for image classification. It consists of multiple neural layers that use a mathematical function to calculate the weighted sum of input and produce an output. The image is converted to a matrix and then passed into the neural network for classification.
            1. Package used - Tensorflow and Keras
        ii. Hyperparameters to be tuned - Number of epochs, learning rate
        iii. Loss functions to be minimized - Binary Cross Entropy

2. Position Detection
    a. Linear Regression
        i. Optimization using Batch Gradient Descent or Stochastic Gradient Descent
        ii. Loss functions to be minimized - Mean Square Error (MSE)

The following evaluation metrics will be used in the project:

1. Accuracy - the ratio of correction predictions to the total predictions

    Accuracy = (True Cats + True Dogs) / (True Cats + False Cats + True Dogs + False Dogs)

2. Mean F1 score - the harmonic means of recall and precision value

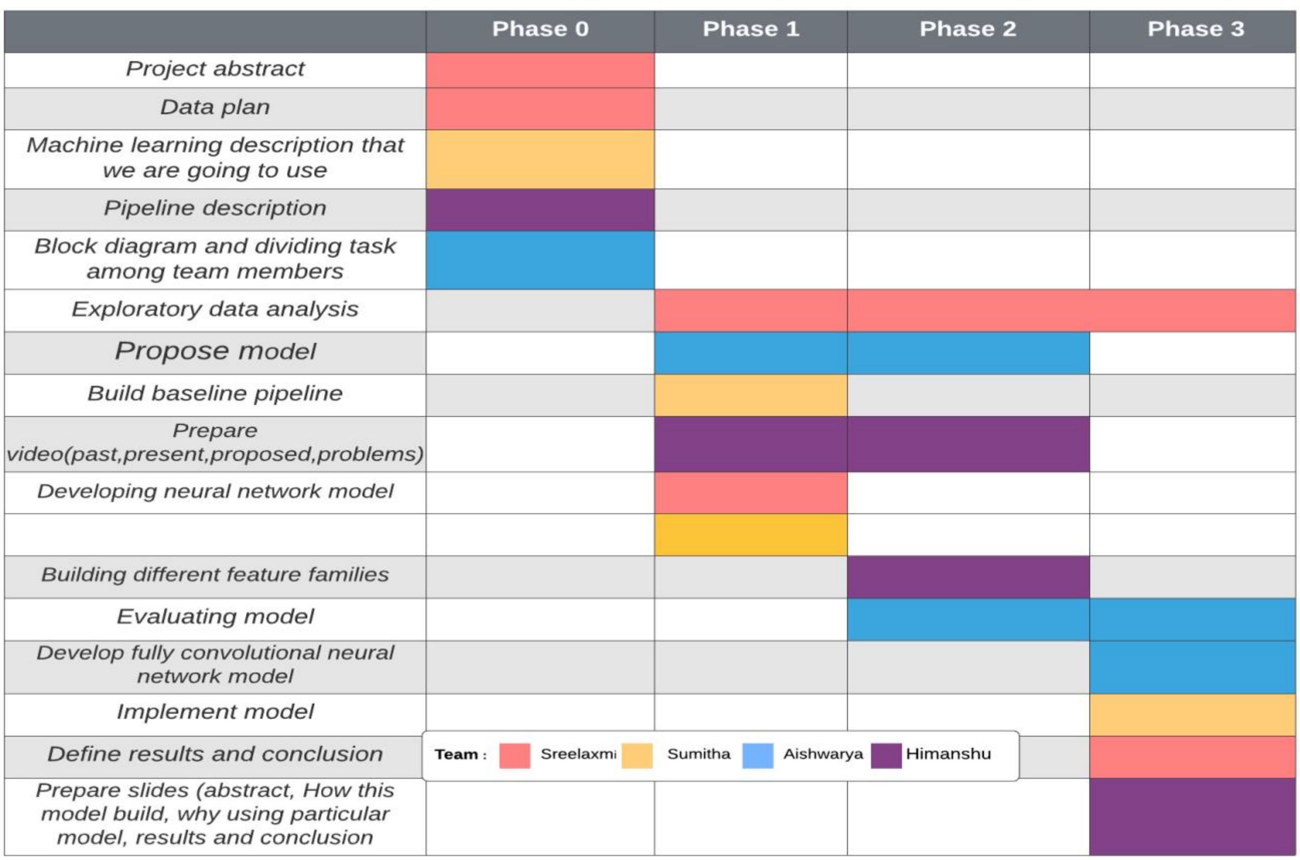
    F1 score = 2*precision * recall / (precision + recall)

    a. Recall is the ratio of true positives to all actual positives
        i. Recall = True positives / (True positives + False negatives)

    b. Precision is the ratio of true positives to all predicted positives
        i. Precision = True positives / (True positives + False positives)

    c. F1 score metric weights both recall and precision equally and thereby a higher value of recall and precision will ensure superior performance of the model

Gantt chart:



Tentative work distribution:

1. Aishwarya:

Will work on proposing a model, evaluating and developing a fully developed neural network model.

2. Himanshu:

Preparing video, feature engineering (developing new features) and preparing slides for the final presentation.

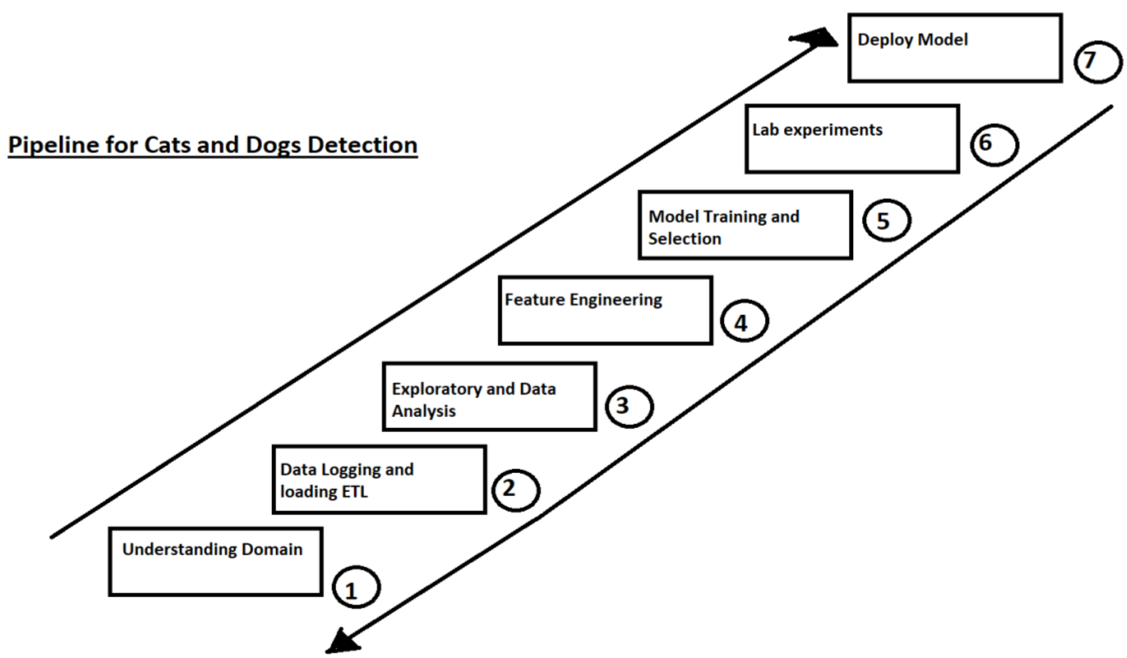
3. Sreelaxmi:

Exploratory data analysis and developing a neural network model, Additionally, she will be working on defining the results and conclusions.

4. Sumitha:

Building a baseline pipeline, developing a neural network model and implementing the model.

Machine learning Pipelines



1. Problem Definition and Data:

We are given a set of images; we need to classify them as either cat or dog and also identify the location of the image.

2. Data Logging and loading (ETL):

The data will consist of images. The position of dog or cat can be anywhere in the given images. We need to detect by various techniques like edge-detection, etc.

3. Exploratory and Data Analysis (EDA):

We need to do exploratory data analysis and pre-process the data and make sure all the images are of the same size. Also, we need to perform image augmentation like flipping, zoom, and rotation. Also, we can try to find correlations between any input features.

4. Feature engineering.

It consists of extracting new features from the given dataset. We can use a combination of existing features to come up with a new feature that will increase the accuracy of our model.

5. Model selection and training:

We can split the data into training and test set. Furthermore, we use k-fold cross validation to split the training set into training and validation set. We will be using grid search along with cross fold validation to tune our hyperparameter.

6. Lab experiments:

We will compare the developed ML models and choose the model with highest accuracy and F1 score.

7. Deploy Model:

This is an important phase. In this phase, we deploy our model to the outside world. In our case, we will do a submission in Kaggle to understand where our model stands.

Group Members:

We are a group of 4 members:

1. Aishwarya Sinhasane - avsinhas@iu.edu (In picture, Left top)
2. Himanshu Joshi - hsjoshi@iu.edu (In picture, Right bottom)
3. Sreelaxmi Chakkadath - schakkad@iu.edu (In picture, Left bottom)
4. Sumitha Vellinalur Thattai - svtranga@iu.edu (In picture, Right top)