DATA MINING FINAL PROJECT IMAGE PROCESSING TO PREDICT AS CAT OR DOG

CENG 3521, DATA MINING

Team Name: Petbenders

Furkan Baldır
Mehmet Pekcan

Report date: Friday 22nd January, 2021

Abstract

In this project, the goal is predicting as cat or dog for given image. To train data, the Support Vector Machine model was created with using Python language. We did single and multiple predict operations to get results to calculate accuracy and also get the exact result which is cat or dog. On the other hand of project, there are front end part written with javascript library which is React, and API part written with Python again.

1 Introduction

This project can work on local server as a web application. Also it can work on command line by using Python file.

1.1 Goal

The goal of this project is that using machine learning algorithm to make image processing. Generally, image processing projects made with using deep learning algorithms to get high accuracy values. However, machine learning algorithm can do the same things with less accuracy. This project aims to calculate accuracy value with using the SVC(Support Vector Machine) algorithm.

On the other hand, in this project, it is aimed that the end user can make predictions. To do that, our project deployed on web server to upload any image and to predict the animal. We used Javascript language to make a modern progressive web app to run this project.

1.2 Used Programming Languages

- Python 3.8.7
- Javascript

1.3 Used Libraries

Python

Flask_Cors 3.0.10: To make request to api without crossorigin problem.
Flask 1.1.2: To create and deploy an API.
Numpy 1.19.5 To convert image to mathematical array.
Pillow 8.1.0 To open and to manipulate images.
Scikit_learn 0.24.1 To using Support Vector Machine algorithm.

Javascript

React: To build Progrssive Web App
Redux: To handle app data on central place
Redux Saga: To handle asynchronous actions like API request
Redux Sauce: To handle better and understandable with Saga pattern
Reselect: To handle data without side effects
Styled Components: To handle styles better maintainable way when app grows
Immer: To handle saving data to Redux more clear way

2 Preparing the Datasets

This project needs a trained machine learning model to predict a result of given image. That's why there are some datasets used in this project.

2.1 Training Dataset

This is our training dataset. This dataset completely used for training, not testing.
Training Dataset

To use these dataset in this project, the images should convert to numpy arrays to use in our machine learning algorithms. That's why in source code, *backend/training_model_creation.py* script has *pick_training_data*() function to store dataset as "data.pickle".

2.2 Testing Dataset

This is our testing data. This dataset completely used for testing, not training.
Testing Dataset

To use these dataset in this project, the images should convert again to numpy arrays to use in our machine learning algorithms. That's why in source code, *backend/training_model_creation.py* script has *pick_test_data*() function to store dataset as "test.pickle".

3 Creating Machine Learning Model and Predicting

In this project, we used Support Vector Machine algorithm that it exists in scikit-learn Python module to create machine learning model. In this project, our goal is creating a machine learning model, storing the model as a file, then using the file as a model while we are predicting an image.

3.1 Converting Images to Numpy Array

Firstly, images needs to be opened. To do that, $\boldsymbol{Pillow}$ module used. Then, opened images converted to numpy array with using $\boldsymbol{numpy}$ module.

3.2 Creating Machine Learning Model

To do that, from scikit-learn SVC() which is Support Vector Machine Classifier used to fit a model. In source code, $\boldsymbol{backend/training\_model\_creation.py}$ has $\boldsymbol{train\_data}()$ function.

3.3 Storing Machine Learning Model

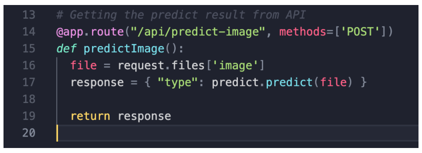
To do that, $\boldsymbol{Pickle}$ module used. When fitting the model finished, $\boldsymbol{backend/file\_management.py}$ script used to call $\boldsymbol{save\_file}(\boldsymbol{filename, data})$ function to save model as 'model.sav'.

3.4 Predicting a Given Image

The final part is predicting a given image. To do that, $\boldsymbol{backend/predict.py}$ script used to call $\boldsymbol{predict}(\boldsymbol{img\_path})$ function to get predicted result. It returns an integer(cat is 0, dog is 1)

4 Bridge of Backend and Frontend: API

To make a better user experience we should move our program from command line interfaces to graphical user interfaces. The first condition to do this movement, we created an API server using Flask. Its duty is basically take the request comes from frontend service then send it to our backend service.



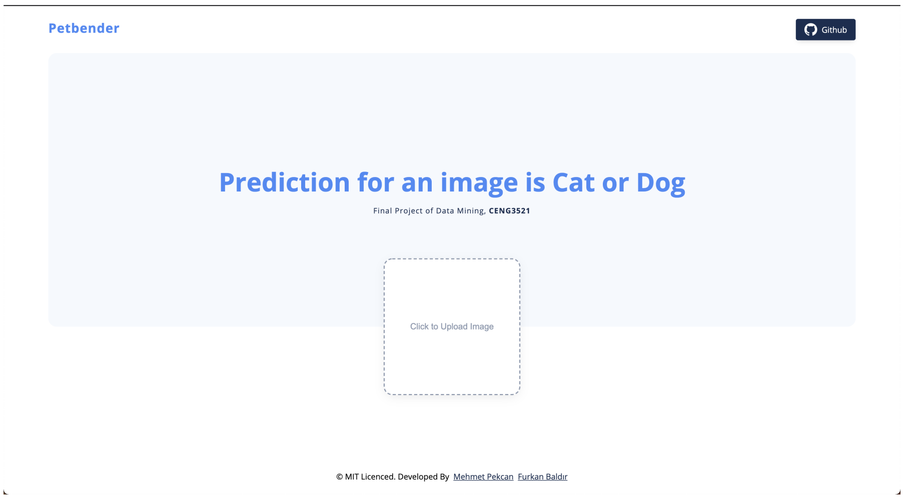
```python
# Getting the predict result from API
@app.route("/api/predict-image", methods=['POST'])
def predictImage():
  file = request.files['image']
  response = { "type": predict.predict(file) }

  return response
```

Figure 1: Predict route

Our API takes the image upload request with file data, then makes a backend call then return it as a JSON.

5 Frontend

As we mentioned previous section, the way of building a better user experience is passing to showing graphical interfaces to user. Not command line interface. So with this rule, we implemented a Progressive Web App that takes power of React. That is the better way to take image from user and showing the predicted class. We don't mention the whole frontend logic here because it is not the information that directly relevant about this project.

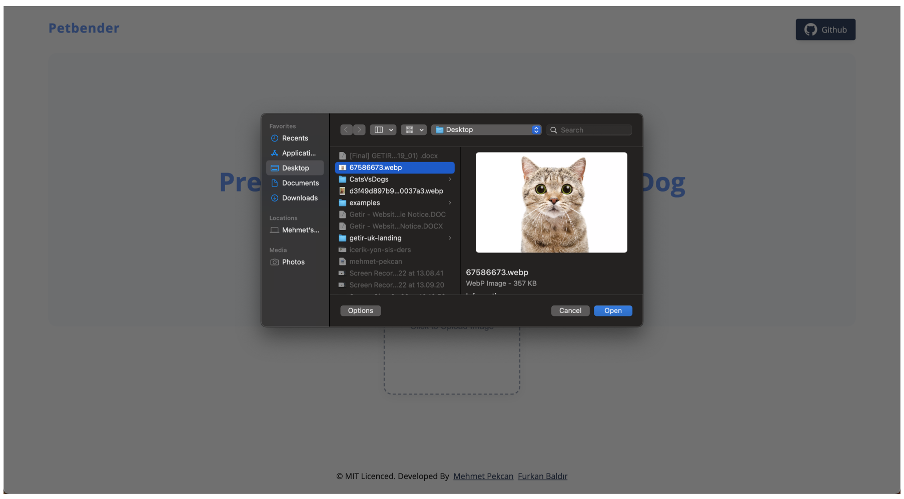


Prediction for an image is Cat or Dog

Final Project of Data Mining, **CENG3521**

© MIT Licenced. Developed By Mehmet Pekcan Furkan Baldır

Figure 2: Main page of the web app



© MIT Licenced. Developed By Mehmet Pekcan Furkan Baldır

Figure 3: Choosing and uploading image

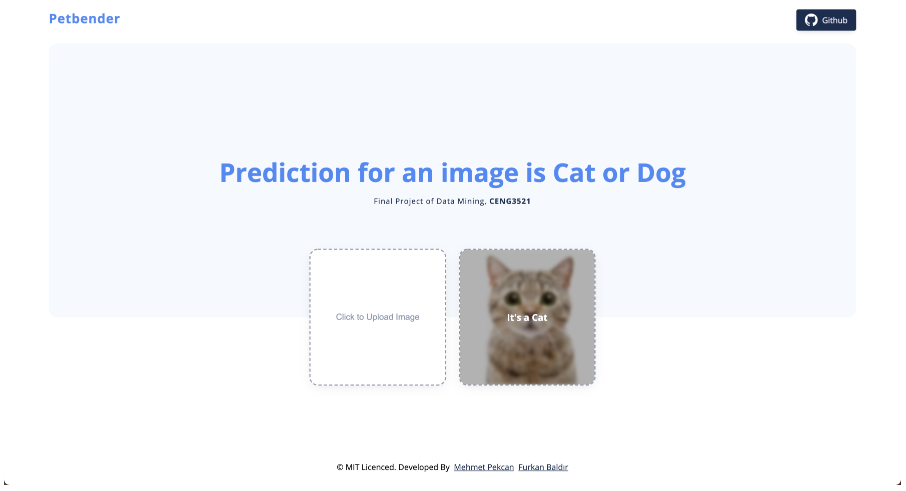

Petbender
Github
Prediction for an image is Cat or Dog
Final Project of Data Mining, CENG3521
Click to Upload Image
It's a Cat
© MIT Licenced. Developed By Mehmet Pekcan Furkan Baldır

Figure 4: Showing the predicted class

6 Results

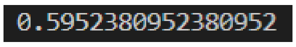
First of all, we have a project website click to go: CatsVsDogs

In machine learning side, there are 2 results to mention:

6.1 The Accuracy of Support Vector Machine Algorithm

As we know, machine learning algorithms have less accuracy than deep learning algorithms to make image processing. That's why we didn't be in expectation to get very high accuracy. Our goal is to calculate how much accuracy we can get with the machine learning algorithm. As we said, we have test dataset to calculate accuracy with using $predict\_with\_test\_data()$ from $backend/predict.py$ script.

The accuracy: **%59**



```
0.5952380952380952
```

Figure 5: Accuracy of model with test dataset

6.2 Deployed Project Works Fine

As we said, our web page is online on https://supremepanda.github.io/CatsVsDogs/. You can upload an image, and you should wait the result which is cat or dog.

Note: Response may come a little late. The reason for this is that we benefit from free server services and as a result we have uploaded our project to a low speed server.

7 Conclusion

First of all, this project is an machine learning project to predict given image. For this work, Python handled machine learning and API parts, Javascript handled frontend part of project.

This project has two main aim to solve:

Being a web application for users to predict given image.
Calculating accuracy of the machine learning algorithm.

As a result, this project works as a web application to predict given image as cat or dog.