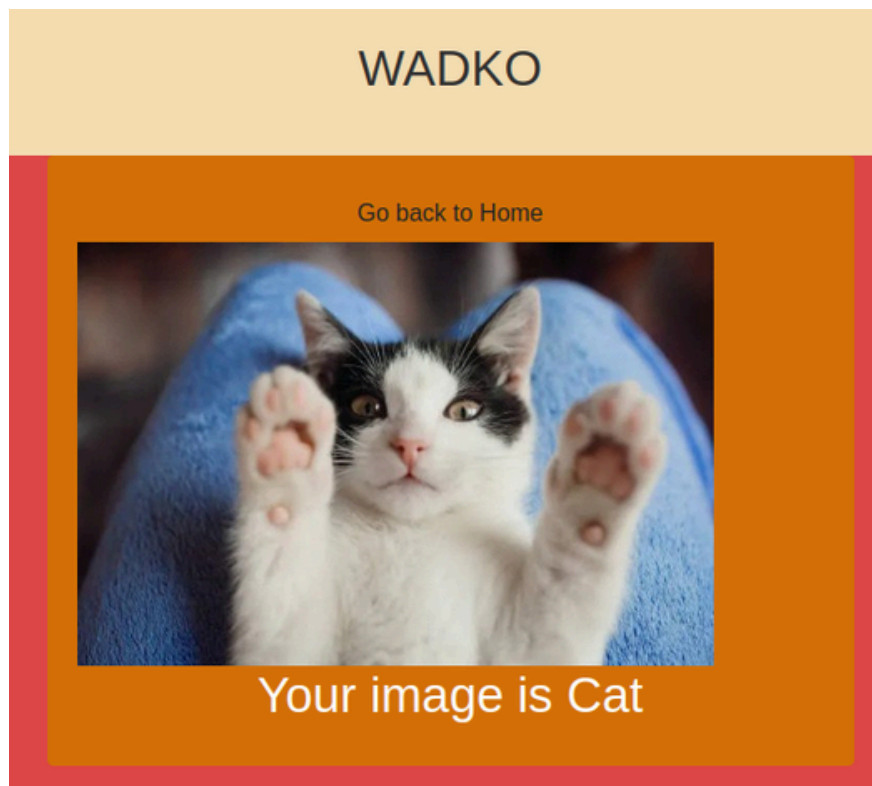


Virtual Assistant for Image Classification (VAfIC/WAdKO)



How to use?

Before getting started, please review our API documentation on GitHub:

<https://github.com/VitaliiYorza/WAdKO/tree/main/wadko-api>

<https://github.com/VitaliiYorza/WAdKO/tree/main/wadko-frontend>

<https://github.com/VitaliiYorza/WAdKO/tree/main/wadko-ml-models>

CAREFULL!

Ensure you have the necessary dependencies installed as mentioned in the respective repositories

Prequesities

```
install python3
```

```
pip install python3-venv
```

```
python3 -m venv flask-venv
```

```
source flask-venv/bin/activate
```

```
pip install Flask
```

```
pip install Pillow
```

```
pip install Flask-CORS
```

```
pip install fastai
```

```
pip install opencv-python
```

```
pip install keras
```

```
pip install tensorflow
```

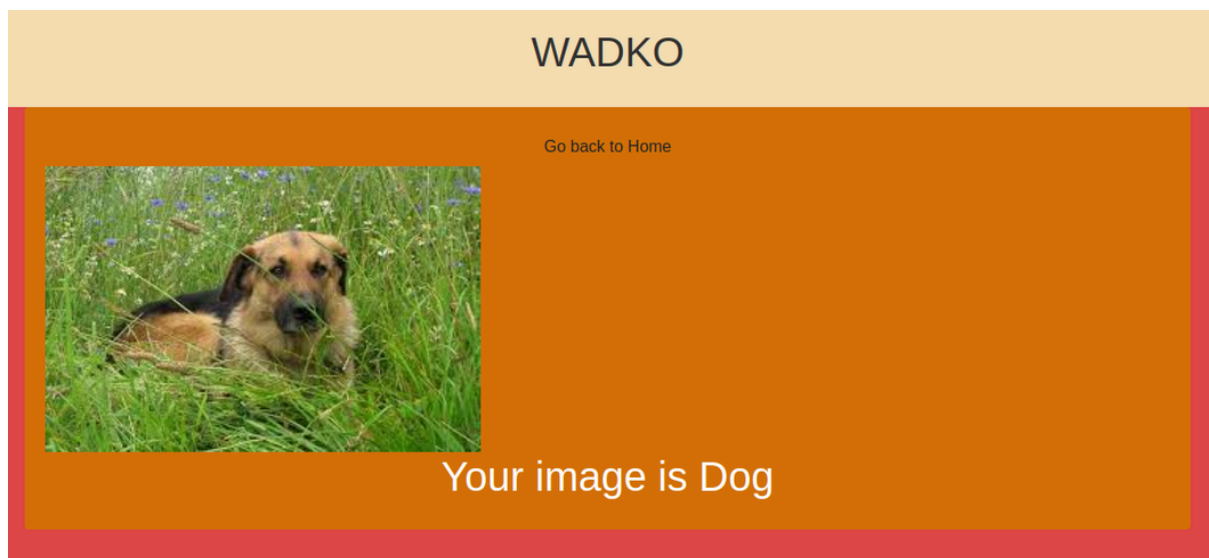
Most of them are quite obvious and popular. It is worth bearing in mind that the FASTAI library installs the TORCH and CUDA libraries, which can take a while to install, particularly on slower internet connections.

Usage

First of all run the Flask application on host "0.0.0.0" and port 8081

The app.py file acts as the main Flask application responsible for image scanning and prediction in the Virtual Assistant for Image Classification (VAfIC) project. This file serves as the backend, leveraging various dependencies to provide a seamless and efficient image processing experience.

Once the server is set up, you can upload any image of your cat or dog and then select the appropriate model/function to classify the image



ABOUT FUNCTIONS

In our project there are 3 main files.

- `image_scan_model.py`
- `predict_image.py`
- `app.py`

These three files collectively form the backbone of our image processing project, offering a comprehensive solution for anyone seeking efficient image scanning, prediction, and classification.

Their cohesion and functionality are pivotal to the success of the Virtual Assistant for Image Classification (VAfIC) project. Below you can find a short description about them

DESCRIPTION

image_scan_model.py

This file, `image_scan_model.py`, serves as the core module for image scanning and prediction within the Virtual Assistant for Image Classification (VAfIC) project. It defines the `ImageScanModel` class, acting as a model for image-related tasks. This file plays a crucial role in handling image-related information and acts as the backbone for image scanning and prediction tasks in the VAfIC project.

Class: `ImageScanModel`

Represents a model for image scanning and prediction.

Attributes:

- `imageBase64URL` (str): Base64-encoded image data.
- `imageItemType` (str): Type of the image item ('Number', 'Cat', etc.).
- `result` (str): Result of the image scan or prediction.

Methods:

- `__init__(self, imageBase64URL, imageItemType, result)`:
Initializes an instance of the `ImageScanModel` class.

- `__str__(self)`: Returns a string representation of the ImageScanModel instance.
- `to_dict(self)`: Converts the instance to a dictionary.
- `from_dict(cls, data)`: Creates an instance of ImageScanModel from a dictionary.
- `to_json(self)`: Converts the instance to a JSON object.
- `from_json(cls, json_data)`: Creates an instance of ImageScanModel from a JSON object.

predict_image.py

The `predict_image.py` file is a script designed for making predictions using a pre-trained fastai vision model. This script is a valuable component within the Virtual Assistant for Image Classification (VAfIC) project. The primary purpose is to predict the class of an image given its path, utilizing a pre-trained model specified by the `model_path` parameter, which defaults to „export.pkl” This file enables seamless integration of image prediction capabilities into the VAfIC project, offering a straightforward way to obtain predictions for various image classes.

Description:

Prediction script using a pre-trained fastai vision model.

Functions:

- `predict_image(image_path, model_path='export.pkl')`: Predicts the class of an image using a pre-trained fastai vision model.

Example Usage:

```
image_path = 'kot.jpg'

# Upload new Img
img = PILImage.create(image_path)

# Prediction
pred, _, _ = model_export.predict(img)
print(pred)
```

app.py

This file provides the core functionalities for handling incoming image data, performing predictions based on the specified category, and delivering the results through the defined API endpoint. It is the backbone of the VAfIC project's image processing capabilities.

Description:

Flask application for image scanning and prediction.

Dependencies:

- Flask
- flask_cors
- image_scan_model
- PIL
- fastai.vision.all
- os
- cv2
- keras.models
- tensorflow
- numpy

Functions and Methods:

- process_data(soup, image): Process image data based on the specified category.
- predict_cat(data): Predict whether the image contains a cat or a dog.
- predict_digit(data): Predict the digit from the image.

- `decode_base64_image(data_uri)`: Decode a Base64-encoded image.
- `getPath(path)`: Get the absolute path of the specified file.
- `scan_image()`: Endpoint to receive image data, process it, and return the prediction result.

Usage:

- Run the Flask application on host "0.0.0.0" and port 8081.