GALAXY INTERNATIONAL INTERDISCIPLINARY RESEARCH JOURNAL (GIIRJ)
ISSN (E): 2347-6915
Vol. 11, Issue 11, November (2023)

RECOGNIZING OBJECTS IN IMAGES USING ARTIFICIAL INTELLIGENCE AND PYTHON

Soliev Bakhromjon Nabijonovich ,
Assistant Fergana Branch Tashkent University
Information Technologies name Muhammad al-Khwarizmi ,
bahromjonsoliev@gmail.com

Ermatova Zarina Kakhramonovna ,
Assistant of the Fergana Branch of the Tashkent University of Information
Technologies Named After Muhammad al-Khorezmi,

ABSTRACT

This article explores methods and technologies for recognizing objects in images using artificial intelligence (AI) and the Python programming language. We will consider various approaches to solving the problem of object recognition, present a literature review, and offer practical examples of software solutions.

Keywords: Artificial intelligence, object recognition, Python, computer vision, machine learning.

INTRODUCTION

Recognizing objects in images is an important task in the field of computer vision and artificial intelligence. This has a wide range of applications including automatic face detection, vehicle detection on roads, medical image classification, and more. In this article, we'll look at how to use the Python programming language to solve this problem. In the era of advanced technology, the intersection of artificial intelligence (AI) and computer vision has paved the way for groundbreaking applications. One such application is the recognition of objects in images, a task that has witnessed tremendous advancements with the help of Python programming and AI algorithms. This article explores the key concepts, tools, and techniques involved in recognizing objects in images using AI and Python.

LITERATURE REVIEW AND METHODOLOGY

To solve the problem of object recognition, we use machine learning methods, in particular, deep neural networks. This article covers the following steps:

Data Collection: We need labeled datasets containing images of the objects we want to recognize.

Data Preprocessing: We perform image preprocessing, including resizing, normalization, and noise reduction.

Model selection: We choose a deep neural network such as Convolutional Neural Network (CNN) to train our model.

Model training: We train the model on our data using various optimizers and loss functions. Artificial intelligence (AI) training involves training a model on a large amount of data to create the model's ability to generalize and make predictions on new, previously unseen data. Here is the general AI training process:

Data collection : It is necessary to create a data set that reflects the variety of scenarios and variations that the model may encounter in the real world.

Data Preparation: Data is usually subjected to pre-processing such as scaling, normalization and augmentation to enable more efficient learning.

Selection of model architecture : The model architecture that corresponds to the task is selected . For example, a convolutional neural network (CNN) can be used for an image recognition task

Definition of loss function and optimizer: The loss function determines how well the model performs the task, and the optimizer is used to minimize this function.

Model training: The model is trained on training data, where the input data and its corresponding output data are used to adjust the model parameters.

Validation and hyperparameter tuning : The model is tested on data it has not seen before (validation) and the model's hyperparameters are tuned to achieve better performance.

Performance Evaluation: The model is evaluated on test data to test its ability to generalize to new data.

Tuning and optimization: Depending on the test results, changes may be made to the model or its hyperparameters to achieve better performance.

Deployment : After successful training and evaluation, the model can be deployed for real-world use.

Training AI is computationally intensive, data intensive, and often time consuming. The training process can be repeated several times to achieve optimal results.

Implementation Steps

1. Data Collection and Preparation

Building a robust object recognition model starts with acquiring a well-curated dataset. Websites like ImageNet and COCO provide extensive datasets for various object categories. Once the dataset is obtained, Python scripts can be used to preprocess and organize the data for training.

2. Model Architecture

Selecting an appropriate CNN architecture is critical. Popular choices include VGG, ResNet, and MobileNet. Python frameworks like TensorFlow and PyTorch simplify the implementation of these architectures, allowing developers to focus on model customization and training.

3. Training the Model

Training the model involves feeding the preprocessed dataset into the selected CNN architecture. Python scripts using high-level APIs provided by TensorFlow or PyTorch can facilitate this process. The model is iteratively adjusted to minimize the difference between predicted and actual labels.

4. Evaluation and Fine-Tuning

After training, the model's performance needs to be evaluated using a separate test dataset. Metrics like accuracy, precision, and recall provide insights into the model's effectiveness. Fine-tuning the model based on evaluation results is a crucial step in achieving optimal performance.

RESULTS

After training the model, we evaluate its performance on a test dataset. Examples of program code for training the model and its use for object recognition will be presented in this article. Below are examples of Python code for some stages of object recognition using the TensorFlow library :

1. Installation of libraries:

```
# Installing TensorFlow and TensorFlow Hub
! pip install tensorflow tensorflow-hub
```

2. Importing libraries:

```
import tensorflow as tf
import tensorflow_hub as hub
from tensorflow.keras.preprocessing _ import image
from tensorflow.keras.applications.mobilenet_v2 import _ preprocess_input ,
decode_predictions
import numpy as n.p.
```

3. Loading the pretrained model:

```
# Loading the pre-trained MobileNetV2 model
model = tf.keras .applications.MobileNetV2( weights =' imagenet ')
```

4. Image preprocessing:

```
# Image loading and preprocessing
```

```
img_path = 'path_to_your_image.jpg'
img = image.load_img ( img_path , target_size =(224 , 224))
img_array = image.img_to_array ( img )
img_array = np.expand _dims ( img_array , axis =0)
img_array = preprocess_input ( img_array )
```

5. Predicting an object in an image:

```
# Getting predictions
predictions = model.predict ( img_array )
decoded_predictions = decode_predictions ( predictions )
print ( " Predictions :", decoded_predictions [0])
```

These examples demonstrate the basic steps that can be used to recognize objects in images using TensorFlow and pre-trained models. Please note that for full object recognition on real data, additional tuning and optimization of the model will be required depending on the specific task.

CONCLUSION

Artificial Intelligence and Python provide powerful tools to solve the problem of recognizing objects in images. We covered the key steps in this process, including data collection, preprocessing, model selection, and model training. Our programming examples demonstrate how to put these techniques into practice. This article can serve as a starting point for exploring more complex and interesting applications in object recognition using AI and Python. Recognizing objects in images using artificial intelligence and Python has evolved into a dynamic and accessible field. The combination of powerful CNN architectures, extensive datasets, and Python's versatile libraries has democratized the development of sophisticated object recognition systems. As technology continues to advance, the applications of image recognition are likely to expand, contributing to a myriad of industries and enhancing our daily lives.

LITERATURE

1. Asrayev M. 0-TARTIBLI BIR JINSLI FUNKSIONALLAR KO 'RINISHIDAGI SODDA MEZONLAR UCHUN 1 INFORMATIV BELGILAR MAJMUASINI ANIQLASH USULLARI //Descendants of Al- Fargani . – 2023. – T. 1. – No. 2. – P. 9-1
2. Samijonov A. et al . Gradient method for determining non-informative features on the basis of a homogeneous criterion with a positive degree //IOP Conference Series: Materials Science and Engineering. – IOP Publishing, 2020. – T. 919. – No. 4. – P. 042011.
3. Python- Powered E-Commerce Solutions in Uzbekistan. (2023). Journal of Technical Research and Development, 1(1), 51-54. https://jtrd.mcdir.me/index.php/jtrd/article/view/3
4. Python's Role in Revolutionizing E-Commerce in Uzbekistan. (2023). Journal of Technical Research and Development, 1(1), 51-54. https://jtrd.mcdir.me/index.php/jtrd/article/view/4

5. Uzbekistan's Digital Market: Python's E-Commerce Impact. (2023). Journal of Technical Research and Development, 1(1), 58-61. https://jtrd.mcdir.me/index.php/jtrd/article/view/5

6.Navigating _ the E-Commerce Landscape in Uzbekistan with Python. (2023). Journal of Technical Research and Development, 1(1), 46-50. https://jtrd.mcdir.me/index.php/jtrd/article/view/1

7. Elevating E-Commerce in Uzbekistan with Python. (2023). Journal of Technical Research and Development, 1(1), 43-45. https://jtrd.mcdir.me/index.php/jtrd/article/view/2

8 . Zulunov RM, Ermatova ZQ PYTHON VA TENSORFLOW YORDAMIDA SUN'IY INTELLEKT DASTURINI YARATISH TEXNOLOGIYASI //Journal of Integrated Education and Research. – 2023. – T. 2. – No. 3. – pp. 53-56.

9 . Kayumov A. et al . PYTHON DASTURLASH TILIDA RASMLAR BILAN ISHLASH. PILLOW MODULI //Research and implementation _ – 2023 4. Ermatova Z. ALTERNATIVE APPROACHES TO ERROR HANDLING: COMPARISON OF EXCEPTIONS AND STD:: EXPECTED IN C++ // Descendants of Al- Fargani . – 2023. – T. 1. – No. 2. – pp. 67-73. 10 . Zulunov RM, Ermatova ZQ PYTHON VA TENSORFLOW YORDAMIDA SUN'IY INTELLEKT DASTURINI YARATISH TEXNOLOGIYASI //Journal of Integrated Education and Research. – 2023. – T. 2. – No. 3. – pp. 53-56.

11. Urinboev Abdushukur Abdurakhimovich . (2023). The Vital Role of Web Programming in the Digital Age . Journal of Science- Innovative Research in Uzbekistan, 1(6), 42–51. Retrieved from https://universalpublishings.com/index.php/jsiru/article/view/1933

12. O'rinboev A. ANALYZING THE EFFICIENCY AND PERFORMANCE OPTIMIZATION TECHNIQUES OF REACT. JS IN MODERN WEB DEVELOPMENT //Innovative research in the modern world: theory and practice. – 2023. – T. 2. – No. 24. – pp. 54-57.

13. Abdullayev A., Kadirova LA INNOVATIVE APPROACH TO FORECASTING AND MANAGING THE EMPLOYMENT OF GRADUATES OF HIGHER EDUCATION INSTITUTIONS // Economics : analyzes and forecasts . – 2021. – No. 2. – pp. 6-12.

14. Abdullaev A., Kadirova L. A., Turgunov M. A. Technology applications electronic funds training in the system regional education // Innovative pedagogical technology _ – 2018. – P. 120-123.

15. Sodikova M. EFFECTIVE METHODS OF TEACHING HISTORY // SCIENCE AND TECHNOLOGY. WORLD RESEARCH. – 2020. – pp. 29-31.

16. Sadikova M. OPTIMIZATION OF THE BUSINESS PROCESS AS ONE OF THE MAIN TASKS IN MODERN MANAGEMENT // Theory and practice modern science _ – 2022. – No. 9 (87). – P. 3-7.

17. Sadikova Munira Alisherovna . (2023). DISADVANTAGES OF TEACHING PROGRAMMING IN DISTANCE EDUCATION. Intent Research Scientific Journal, 2(10), 26–33.