**Image Classification with Convolutional Neural Network**

**Overview**

This project focuses on classifying fashion product images using a Convolutional Neural Network (CNN). The dataset, sourced from Kaggle, contains 44k+ product images with corresponding catalog information. The goal is to categorize products into master categories, such as Apparel, Accessories, Footwear, Personal Care, Free Items, Sporting Goods, and Home.

**Table of Contents**

1. Data Wrangling
2. Data Processing
3. Modeling
4. Hyperparameter Tuning
5. Recommendation System
6. What's Next
7. How to Use
8. Contributors
9. Data Wrangling

* The dataset was obtained from Kaggle, comprising product images and a catalog spreadsheet.
* A subset of the dataset was chosen due to computing constraints, with a focus on majority and minority categories.
* Catalog information was used to label images with their respective master categories.

1. Data Processing

* Images were resized and converted into a 4D numpy array compatible with the VGG16 model.
* Target features were transformed into categorical arrays for training.

1. Modeling

* Transfer learning was applied using the VGG16 pre-trained CNN model.
* Two fully connected dense layers were added, followed by the output layer for classification.
* The model was trained with an 80/20 train-test split, achieving an overall accuracy of 95.85%.

1. Hyperparameter Tuning

* Hyperparameters, including the number of neurons and learning rate, were tuned using Keras Tuner.
* Despite good performance, the tuned model showed a decrease in accuracy and misclassification of the "Sporting Goods" class.

1. Recommendation System

* A recommendation system was built using feature extraction from the VGG16 model.
* Cosine similarity scores were calculated to provide recommendations based on image similarities.

1. What's Next

* Future plans include testing the model on the entire dataset using cloud-based resources.
* Improving model robustness, modularity, and portability for broader adoption.
* Experimenting with additional hyperparameters and model architectures.
* Comparing the performance of different pre-trained models.
* Starting a new project involving regression with deep neural networks.

1. How to Use

* Clone the repository: `git clone https://github.com/your-username/your-repo.git`
* Install dependencies: `pip install -r requirements.txt`
* Open and run the Jupyter Notebook for detailed analysis and model implementation.

1. Contributors

* Hongling Yang (https://github.com/hyang78227)