FINAL REPORT

Retail Product Image Recognition

Problem Statement

With the recent times where digital transformation has been immense and there has been a significant shift towards online shopping. It still hasn't eradicated or has brought down the offline shopping experience.

People still like to go out to and buy the products, providing them with a sense of control and let them have the feel for their buy. And this is not changing even with the ease of online shopping.

But the problem still exists at the offline stores. The problem of time-consuming billing counters at place.

For the problem in hand, we are going to build an image recognition model that will help us redefine this situation and help reduce the time the customer has to spend at billing.

The model will automatically identify and count the items the user has selected and provide him with a bill of the respected items.

NOTE: At the initial stage of project we only look to identify products to their respective super-category and not to the item itself

Stage 1 – Data Wrangling & EDA

The initial stage of the project was focused at collecting ample number of images for different products ranging to a variety of categories.

The data can be found at the following Kaggle repository.

https://www.kaggle.com/diyer22/retail-product-checkout-dataset

The images were divided into three directories, as follows-

Train = It contains 52,000 images of single items present within it

Test = It contains 24000 images of multiple items in a single image

Validation = This is to test the model performance in a more real-world examples where images contain number of items in a single image.

The data is provided to us in the json format having the following information.

```
├── info: dict  7
├── licenses: list  1
├── categories: list  200
├── __raw_Chinese_name_df: list  200
├── images: list  53739
└── annotations: list  53739
```

Diagram1: Train json

While doing some analysis it revealed that there are a total of 17 super-categories. These super –categories then in total comprised of 200 different products in them.

The 17 super-categories and their respective product count we have with us are-

	supercategory	Count
0	alcohol	11
1	candy	10
2	canned_food	14
3	chocolate	12
4	dessert	17
5	dried_food	9
6	dried_fruit	9
7	drink	15
8	gum	8
9	instant_drink	11
10	instant_noodles	12
11	milk	11
12	personal_hygiene	10
13	puffed_food	12
14	seasoner	12
15	stationery	7
16	tissue	20

The images in the dataset are of these products themselves and are named by a unique code assigned to each product.

Using these image codes I segregated the images from 1 directory to assigning them into respective super category folders.

Stage 2 – Pre-Processing and Training

In the pre-processing and training stage we converted each image to its respective array so that our model could interpret it.

We also integer encoded each of the super-category respective to its product.

The range of target variable is 0-16

We got our feature variable in array form as X and target variable in integer form as y.

Now since the array data ranged from0 to 255 we had to normalize the data which we simply achieved my dividing X by 255.

Now once the data was normalised we began building a simple CNN architecture for our project.

We chose Sequential model for this project.

In the model we added 2 64 feature densely connected convolution layer with relu activation function.

Each of the densely connected conv layer was followed by a max pooling layer to reduce the spatial dimensionality, using a 2*2 step.

After that we connected to a dense layer with 64 features to it again having a relu activation function.

Finally since we had 17 target feature we added our final dense layer with 17 nodes and having the activation function as softmax.

Features of model compilation -

Optimizer – Adam

Loss function – sparse-categorical cross entropy

Metrics – Accuracy

After training the model with the dataset we achieved an accuracy_score and f1 score both to be 90 percent.

Future Scope-

1) We aim to detect more than one object in the image and successfully able to identify the category of it.

2) After detecting multiple super-category in one image we aim to improve the model to identify individual products themselves

Conclusion

We were able to build a simple CNN model to detect an image with single item and successfully identify the category it belonged to.