Identifying products in grocery catalogs using image classification

Team 129: akshobhy, ialtun, nsaquib2

CSE 676 Final Project

04/03/2024

University at Buffalo The State University of New York

Agenda

- Project Description (1 min)
- Background (2 mins)
- Dataset (3 mins)
- Methods (3 mins)
- Results (3 mins)
- Key Observations (2 mins)
- Q & A

Project Description

Identifying products in grocery catalogs using image classification

- Outputs acts as labels for input images
- Web Scraped data using beautifulsoup as there was no complete dataset
- Multiclass classification on 10 classes
- Used 6 Deep Learning models and achieved highest of 85.19% accuracy

Background

- Makes the products more accessible to the customers by making the process of searching for the product more convenient.
- Easiest way for users to search for products using VR and AR technologies.
- It is going to be a necessary technology in near future.

Dataset

- No publicly available dataset that fits to our purpose.
- Collecting data using Python web scrapers.
- We expect around 1500 images per class (15000 images in total).
- Downloading 10 images in a single machine takes 5 mins.
  (total 125 hrs required for the download which is not feasible).

Dataset — Solution!

- We utilize CCR in a two phase approach.

1) Get the URL links of each image without downloading it using BeautifulSoup and store those links.
2) Generate 750 batch scripts where each batch corresponds to a download set of 20.

- Using CCR, we collected required image dataset (around 1GB) for our models.

```
for cls in classes:
  links = []
  for i in range(5):
     htmldata = getdata("search URL")
     soup = BeautifulSoup(htmldata, 'html.parser')
     links.append(soup.find_all('img'))
  file.write(links)
```

Methods

Data Preprocessing

- We used calculated normalization parameters from our own images.
- Heavy data augmentation including resize, random rotation, cropping, horizontal flip, etc.
- 70 15 15 split for training, testing and validation.
- Used a very big size for data loaders so that we could train our model quicker.

Model architectures

1. Resnet with dropout layer
2. SE blocks with Resnet
3. Resnet with transformer
4. Resnet + SE block + transformer layer
5. Resnet 50 with transformer
6. Resnet 50

Results

Our models gave following testing accuracy:

1. Resnet 18 with batch norm - 85.19
2. Resnet 18 with SE modules - 84.56
3. Resnet 18 with transformer layer - 78.859
4. Resnet 18 + SE block + transformer - 77.65
5. Resnet 50 - 84.61
6. Resnet 50 + transformer - 77.25

*Different results across various iteration, for transformer based models.

Key Observations

1. Importance of data collection scheme for model performance.
2. Importance of label selection.
3. Focus on training efficiency.
4. Preprocessing trade off.

Contribution Table

Name	UBIT	Contribution
Akshobhya Sharma	akshobhy	33%
Ibrahim Bahadir Altun	ialtun	33%
Nazmus Saquib	nsaquib2	33%

Thank you!

Q & A