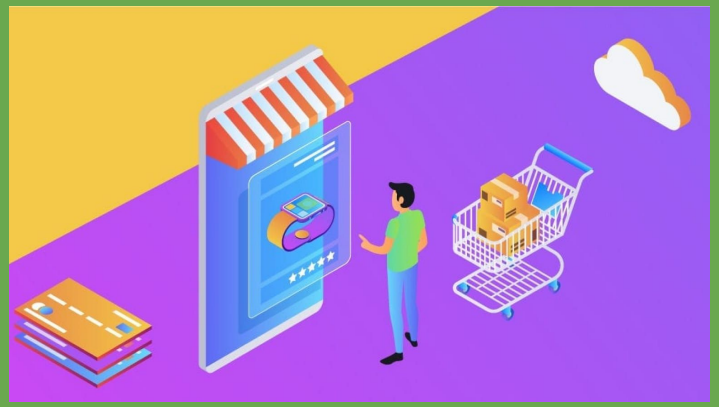
CS5542 Big Data Analytics and App

Smart Shopping Check-Out



SMART SHOPPING CHECKOUT:

- Standing in lengthy crowds and waiting for checkout in any store has become a major issue in our busy daily lives.
- This project is about detecting objects when a customer picks up an item at a store.
- We are working to solve the problem of customers standing in massive queues, as well as consumers paying the bills for the products they have purchased, and store management will not have to tackle the difficulty of customers stealing a few items or taking them away without paying the bills.

Dataset

Collected few images of the below mentioned categories and created a dataset.

Manually defined the bounding boxes of the image using python labelImg

Categories:

- Men pants
- Men shirts
- Men footwear
- Women pants
- Women shirts
- Women footwear

Implementation:

- Used tensor flow model for object detection.
- The TensorFlow Object Detection API is an open-source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models.
- For model configuration, we have used efficientdet configuration
- Used data_augmentation.
- Before running the code, it is important to check the GPU load.

- Used protocol buffers to allow serialization and deserialization of structured data and also to provide a better way, compared to XML, to make systems communicate.
- We have used the model efficientdet d2 for training purposes.
- we have used the concept of labeling the images with Men pants, Men shirts, Men Footwear, Women’s pants, Women’s shirts, Women footwear and defining there Id’s

+ Code + Text Connect Editing

```
[ ] def convert_classes(classes, start=1):
        msg = StringIntLabelMap()
        for id, name in enumerate(classes, start=start):
            msg.item.append(StringIntLabelMapItem(id=id, name=name))
        text = str(text_format.MessageToBytes(msg, as_utf8=True), 'utf-8')
        return text

    labels =  [
                "BoysShirt",
                "GirlsShirt",
                "BoysPant",
                "GirlsPant",
                "MenFootWear",
                "WomenFootWear"
                ]

    txt = convert_classes(labels)
    print(txt)
    with open('labelmap.pbtxt', 'w') as f:
        f.write(txt)
```

Training the Dataset

+ Code + Text Connect Editing

```
[ ] ## Training Configurations
    num_epochs = 10
    num_steps = 4000
    num_eval_steps = 100
    batch_size = 4 # Change this to 4
    print("Number of Steps:", num_steps)
```

```
Number of Steps: 4000
```

Save the trained dataset model

```
!zip -r saved_model.zip inference_graph/saved_model/
```

```
  adding: inference_graph/saved_model/ (stored 0%)
  adding: inference_graph/saved_model/saved_model.pb (deflated 93%)
  adding: inference_graph/saved_model/variables/ (stored 0%)
  adding: inference_graph/saved_model/variables/variables.data-00000-of-00001 (deflated 25%)
  adding: inference_graph/saved_model/variables/variables.index (deflated 78%)
  adding: inference_graph/saved_model/assets/ (stored 0%)
```

+ Code + Text

Main : Donwload saved_model.zip

- We are detecting the object with the label and the boundary box

- We have also added the price tag for each object so that we can get the price for that object which is detected.

WHAT ELSE IT DOES:

Some items may not have barcodes and in that case, it is a time-taking process for the customer to get his bill and this object detection is helpful for the end users to a smarter way to checkout the products.

Detecting the objects all at a time rather than scanning the barcode for each product, this helps users.

Team number:17

Team Members Names:

Haritha Thumukuntla - htmrc@umsystem.edu
Shambhavi Sompally - ss472@umsystem.edu
SaiDurga Maheshwari Kanuganti - skbfz@umsystem.edu
Sandeep Reddy chevula - scv7n@umsyste.edu

THANKYOU