THE UNIVERSITY OF TEXAS AT DALLAS

Itemization of Receipts Using Computer Vision Techniques

V Jayasuryaa Govindraj, Manasi Barhanpurkar, Raj Mishra, Saahil Gilani

CS6384.002 Computer Vision

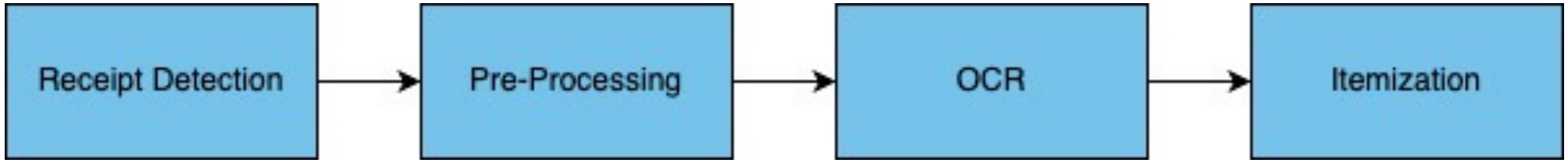
Group - 13

Introduction

Goals

- We tried to implement one of the major use cases of Computer Vision
- An application which focusses on object detection and then extracting useful data using OCR
- The implemented solution is over receipts which we get from restaurants or groceries
- Finally, the implementation should be able to detect receipts from an image and then fetch us the item-price mapping for itemization purposes

Implemented Solution



- We have four phases to our project:
  - Receipt Detection: We used YOLO object detection techniques to achieve this phase
  - Pre-Processing: This will involve fetching the cropped image from the original image and enhancing the image to feed into the next stage
  - The next stage is the optical character recognition stage where we used Google Tesseract to extract text from images
  - The final stage is the itemization stage where we build an item-price mapping in JSON format using the data we received from the previous stage

Receipt Detection

You Look Only Once (YOLOv3)

- Yolo is a real time object detection algorithm first introduced by J Redmon et al. in a research paper in 2016
- YOLO is a one stage detection algorithm unlike two stage algorithms like R-CNN, Fast R-CNN
- We are using YOLOv3 for our project and training purpose
- Key Components of YOLOv3:
  - Determining confidence of bounding box using IOU
  - CNN architecture with 53 convolutional layers – Darknet-53
  - Binary cross entropy loss function
- We used open source ImageAI library for the purpose of YOLOv3
- Within ImageAI we used transfer learning by using a pre-trained YOLOv3 model, this model was pre-trained on Coco dataset

Dataset Creation

- It was difficult to fetch dataset which sufficed our requirement, we were able to find dataset of only receipts (Sample Receipt Dataset)
- Using this dataset, we created our own dataset for object detection by overlaying the receipt images on random background images
- Dataset structure:

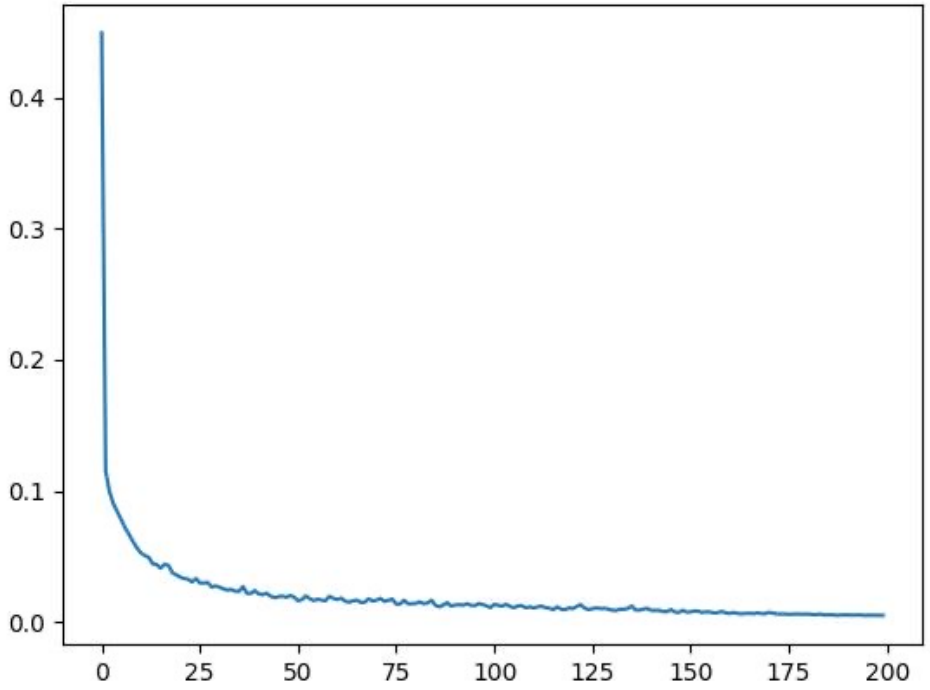
```
receipt/  - total 192 images
      train/ - total 146 images
            annotations/ - YOLO annotation format
                  classes.txt
                  image (i).txt
            images/
                  image (i).txt

      validation/ - similar structure as train - total 46 images
```

Sample Dataset Images

Images not to scale

Results

- We trained for 200 iterations on 146 images using ImageAI
- Observed loss curve and AP value:



AP: 0.45

Recall: 0.46

Precision: 0.75

Sample Output - 1

YOLO

Receipt Detection

Cropping

Images not to scale

Sample Output - 2

YOLO

Receipt Detection

Cropping

Images not to scale

OCR and Itemization

Pre-Processing, Tesseract and Itemization

- In this phase of the project we receive the cropped image from the object detection model
- We pre-process and enhance the image in order to fetch the best output from Tesseract, few pre-processing steps:
  - Converting the image from RGB to gray
  - Median blurring
  - Thresholding
- The next step involves feeding the processed image to Google Tesseract to extract text data from the images
- The extracted text is then processed to itemize the data and output a item-price mapping in JSON format

Pre-Processing Output

RGB to Gray
And median
blurring

Thresholding

Images not to scale

Image to Itemized output

Contours

Tesseract Engine


```
{
    "result": [
        {
            "price": "10.98",
            "item_name": "1 Side Craw Etouffee 4oz",
            "item_id": "8fc09db2e85a11ed832a2e0d6b9cfb45"
        },
        {
            "price": "21.99",
            "item_name": "me 1 Pan Seared Tilapia",
            "item_id": "8fc09f88e85a11ed832a2e0d6b9cfb45"
        },
        {
            "price": "5.98",
            "item_name": "2 Tea",
            "item_id": "8fc09fc4e85a11ed832a2e0d6b9cfb45"
        },
        {
            "price": "38.95",
            "item_name": "Sub Total:",
            "item_id": "8fc09ff6e85a11ed832a2e0d6b9cfb45"
        },
        {
            "price": "3.21",
            "item_name": "Tax:",
            "item_id": "8fc0a050e85a11ed832a2e0d6b9cfb45"
        },
        {
            "price": "42.16",
            "item_name": "Sub Total:",
            "item_id": "8fc0a082e85a11ed832a2e0d6b9cfb45"
        },
        {
            "price": "42.16",
            "item_name": "01/07 13:29 TOTAL:",
            "item_id": "8fc0a0bee85a11ed832a2e0d6b9cfb45"
        }
    ]
}
```


Output in JSON Format

Images not to scale

Conclusion & Future Work

- This is a great tool, and the underlying concept of Object detection and OCR can be used to automate many other real-world problems, some of like:
  - License plate detection on cars and extracting registration number
  - Vehicles can use it to detect roadside signs and highway markings
  - Inventory management at supermarkets
- Few of the work we would like to take up in the future would be to:
  - Experiment and compare performances with newer YOLO models
  - Create a more versatile dataset by manually labelling and capturing it through our cameras
  - Integrate with a frontend application to make it more user friendly and market adaptable

References

- YOLO (https://arxiv.org/abs/1506.02640)
- YOLOv3: An Incremental Improvement (https://arxiv.org/abs/1804.02767)
- ImageAI
  - https://github.com/OlafenwaMoses/ImageAI
  - https://imageai.readthedocs.io/en/latest/customdetection/index.html
  - https://github.com/OlafenwaMoses/ImageAI/releases/download/3.0.0-pretrained/yolov3.pt
- Google Tesseract
  - https://storage.googleapis.com/pub-tools-public-publication-data/pdf/33418.pdf
  - https://tesseract-ocr.github.io/
- Sample Receipt Dataset (https://expressexpense.com/blog/free-receipt-images-ocr-machine-learning-dataset/)

Questions?

Thank you