COMPSCI 9542B Advanced Artificial Intelligence - Object Character Recognition

Sihui Zhao
Faculty of Science
The University of Western Ontario
London, Canada
szhao444@uwo.ca

Aesha Gabhawala
Faculty of Science
The University of Western Ontario
City, Country
agabhawa@uwo.ca

Parthavi Madaan
Faculty of Science
The University of Western Ontario
London, Canada
pmadaan@uwo.ca

***Abstract*—With the advancement in technology and digitalisation, the world is switching to paperless information. This change can be seen in the majority of major sectors - banks, legal documents, travel industry, governments, etc. All this is possible using Character Recognition. Google Lens, Apple Photos, cameras used for tracking speed and tolls, generating music from handwritten notes - the applications are endless. This demand of retrieving textual information from images and videos has been our motivation behind this project. The current trend focuses on using object detection models and using it to run accurate predictions using recognition systems. Huge amounts of labeled data is used as input for these models. We are using an object detection technique which considers text as objects. A proven efficient method using YOLO architecture has been implemented from scratch to find text in the ICDAR 2013 dataset. This detected text is then broken down into word level segments, which are then used to recognise the text in images.**

***Index Terms*—OCR, YOLO, SSD**

I. Introduction

Optical Character Recognition (OCR), sometimes referred to as text recognition, is a system that converts the physical, printed documents into machine-readable text. Data is extracted and repurposed from scanned papers, camera photos, and image-only pdfs using an OCR tool. OCR systems extracts letters from images, converts them to words, and then assembles sentences, allowing access to and alteration of the original text. It also removes the necessity for manual data entry. It serves as a back-up, commonly implemented in smartphone applications, to store all the information obtained from the OCR system and copy text from images. Several applications also provide translation services using. Lastly, the OCR method has enabled the physical documents digitization process to automate the working process, and improve the working efficiency.

Many well-known systems and services in our daily lives rely on OCR as a hidden technology. Data-entry automation, assisting blind and visually impaired people, and indexing documents for search engines, such as passports, license plates, invoices, bank accounts, business cards, and automatic number plate recognition, are all important use cases for OCR technology. OCR enables the optimization of big-data modeling by converting paper and scanned image documents into machine-readable, searchable pdf files. Processing and retrieving valuable information cannot be automated without first applying OCR in documents where text layers are not already present. Scanned papers may now be incorporated into a big-data system that can read customer data from bank statements, contracts, and other essential printed documents thanks to OCR text recognition. Organizations can use OCR to automate the data mining input stage rather than having personnel examine innumerable picture documents and manually feed inputs into an automated big-data processing workflow. OCR software can recognise text in images, extract text from photographs, and save text files in the following formats: jpg, jpeg, png, bmp, tiff, pdf, and others. [1]

In this project, we have focussed on developing the OCR system for detecting and recognizing text from real scenes. There are several challenges in identifying characters in scene images as the images can have different scenarios and types making it hard to detect. Unlike monotone characters on fixed backgrounds in classical OCR tasks, the character recognition challenge in scene images is more complicated because of the variations in the images such as: background, texture, text font, text size. We have used the ICDAR 2013 dataset which is a benchmark for applications focussed on reading text from images.

This project is divided into three modules: Text localization, text segmentation and text recognition. Text localization detects bounded boxes for each image. Text segmentation saves the bounded words for each image separately. Text recognition detects and recognizes the character for each segmented word.

II. Related Work

Upon investigation, we found out that there exist numerous datasets to be used for Object Character Recognition. These datasets are being used in accordance with the objective - ICDAR2013 [2] to train the recognition model since the text inside the images is horizontally aligned words, which is better suited for text recognition. If the goal is text detection, a more complex dataset, ICDAR2015 [3] has been used. It contains multiple transformations so that the model can learn better. Datasets with multiple scripts, like ICDAR19-ArT [4] are also in use to achieve text detection and recognition in multiple languages. While numerous datasets exist, one main problem still pertaining is the lack of diversity. Because of

this, the model is not efficient enough to perform well on real-life image testing datasets. To overcome this, Baek et al. [5] suggests using wild images as input data to better train the model.

As far as Optical Character Recognition is concerned, there are three general approaches to solve the problem: classical computer vision techniques, standard deep learning detection method and specialized deep learning technique. Classic computer vision makes use of OpenCV and Python. It applies filters to make characters stand out in the background, then uses contour detection to recognise these characters and lastly uses image classification to classify characters. This technique has its limitations - when two characters are very close, this method does not work. Standard deep learning uses artificial neural networks to perform computations on large amounts of data. It includes using architectures such as YOLO, SSD and Mask RCNN. Using these end-to-end deep learning models along with tuning the hyperparameters increases the efficiency and accuracy of models. Specialized deep learning uses specialized networks - EAST is one example, which is a special version of U-Net to extract features from different levels of network.

The most widely used method in scene text detection is the Tesseract method, which utilizes text recognition and line segmentation. This method has also been updated by incorporating the LSTM based text recognition - this allows it to take slices of image across variable with characters and recognise it. There also exists convolutional character network models, like CharNet which directly outputs bounding boxes of words and characters, including corresponding labels. It is widely used to optimize text detection [6]

III. Data

We have used the ICDAR 2013 dataset which is useful for applications focussed on reading text from real scene images. The dataset consists of 461 total images out of which 345 images were used for training, 100 for validation and 16 images were used for testing. Word-level annotations are provided for each image The ground-truth image files contain pixel level assignment for individual characters.

As the dataset is limite, we have implemented data augmentation methods to increase the dataset size. Image augmentation method is used to address the restricted availability of data for better training and testing of the model. It is a procedure used to build the size of the preparation dataset by misleadingly altering pictures in the dataset. In this project, the preparation of pictures is increased with eight particular activities to be specific: shearing, contrasting, flipping on a level plane, rotating, zooming, and obscuring. These processes increased the number of images by 21 times.

A. Data Pre-processing

All the images in the ICDAR 2013 dataset are resized to the dimensions of the 512x512 as that is the input image size expected by the model. Similar transformation is also applied to the ground truth images. These images are then normalized to the range of [-1,1] for more sensible results. Ground truth coordinates are processed to form a matrix of dimensions as (grid height, grid width, 1, 5). All these preprocessed images were then converted and stored in the form of Numpy arrays to ease the process of training.

B. Data Augmentation

This refers to techniques by which one can increase the amount of data by adding modified versions of already existing data. It is used as a regularizer to control overfitting. Since our training dataset has mere 345 images, we decided to use the methods of data augmentation to increase the dataset size. We have used Shear, Vertical Flip, Rotation, Rescaling and Normalization. These techniques increased the dataset size by 21 times.

Shear refers to distorting the image along x-axis. We are distorting the images by 20%.Vertical flipping, as the name suggests, flips the image. We are also rotating the images by an angle of 90 degrees. Then, we ensure that these images are in line with pre-processing steps by rescaling them to dimensions of 512x512 and then normalizing them.

IV. Methods

The method is split into 4 parts, Text Localization, Text Segmentation, Text Recognition. The methods used in each of the part will be explained down below.

A. Text Localization

Text localization detects the bounding boxes of the text in the images. Scene texts differ from scanned documents in a number of ways, necessitating the use of specialized approaches to locate and recognise them. In addition, there is no prior information on the placement, layout, orientation, size, typeface, and color of texts in a scene image. Also, there are various textures and patterns that resemble characters making localization of scene text a challenging and crucial work.

For the text localization, we have implemented the YOLO architecture which is an object detection model from scratch.

1) Network Overview

The YOLO is a single stage real-time object detection model. It is one of the most advanced models for object detection and has been developed by ultralytics.

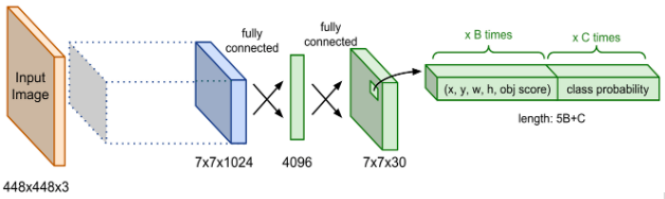


Fig. 1. *YOLO Architecture*

The YOLO model (Figure 1) is made up of three key components: the head, neck, and backbone. The backbone

is the part of the network made up of convolutional layers to detect key features of an image and process them. The backbone is first trained on a classification dataset, such as ImageNet, and typically trained at a lower resolution than the final detection model, as detection requires finer details than classification. The neck uses the features from the convolution layers in the backbone with fully connected layers to make predictions on probabilities and bounding box coordinates. The head is the final output layer of the network which can be interchanged with other layers with the same input shape for transfer learning.

Inspired by the YOLO architecture, we have used the MobileNetV2 architecture pre-trained on ImageNet weights. MobileNetV2 is based on an inverted residual structure where the residual connections are between the bottleneck layers. The intermediate layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. The architecture of MobileNetV2, as a whole, consists of the initial fully connected convolutional layers with 32 filters, followed by 19 residual bottleneck layers.

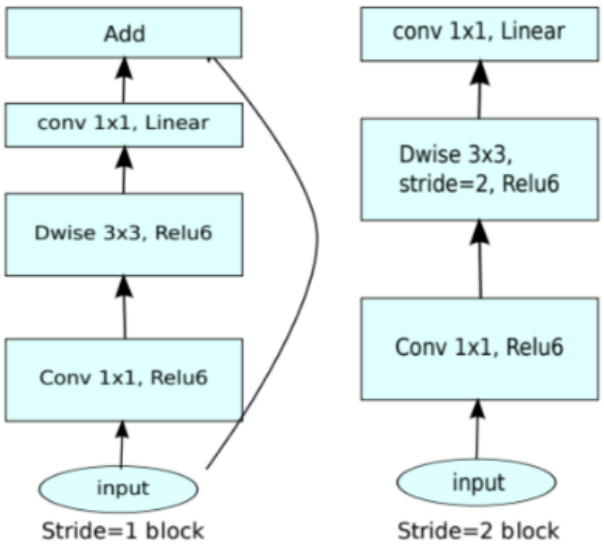


Fig. 2. *MobileNetV2 architecture*

We chose MobileNetV2 (Figure 2) as the backbone for feature extraction since it has demonstrated excellent object detection results. We then added three convolutional layers with ReLU as the activation function. Dropout layers with a dropout rate of 40% are added after each convolution layer to overcome the problem of overfitting. Batch normalization layers are also added to help regularize the model. The model uses Leaky ReLU as its activation layer in the whole architecture except for the last layer where it used a linear activation function. The model uses Adam as the optimizer with a learning rate of 1e-6 to allow for slow learning which is later adjusted automatically by Adam optimizer.

2) Training

This model was then trained on the training dataset (after performing image augmentation) for about 175 epochs until both the training and validation losses converged (Figure 3).

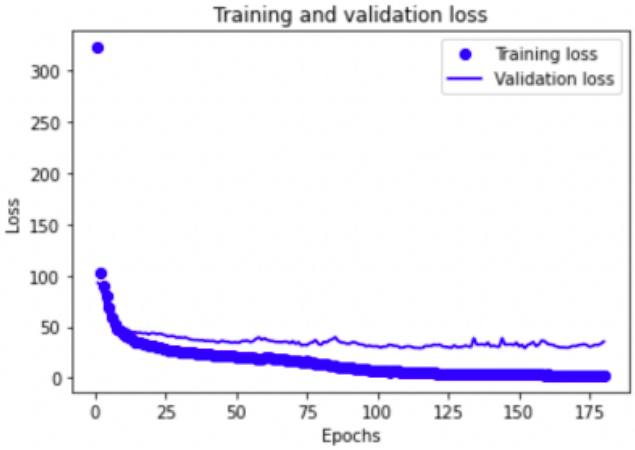


Fig. 3. *Training and validation loss using YOLO model*

All other default parameters were used for transfer learning and custom training the network. The model was tested on a set of images and the performance metrics were calculated.

B. Text Segmentation

Once we have an estimate of the location of text in an image, the next goal is to split the bounded text into single characters. These characters are then stored and used as input for Text Recognition.Some of the results of Text Segmentation are as follows:

C. Text Recognition

Text Recognition, or character classification aims to use the input of Text Segmentation and recognise each character. This is achieved using a fixed-input image, which is then applied to the character images in a set of ground truth labeled images.

While the system is very powerful, it is prone to over-fitting in case of small character datasets, like the english alphabet in our case.

It is advisable to merge various character labeled datasets and use it to train the recognition model. These images should also be passed through a pipeline of augmentation methods to minimize overfitting.

The anticipated results should look like:

Fig 4: Anticipated results from Text Recognition

V. Results

A. Data Augmentation

The original image looks like below:

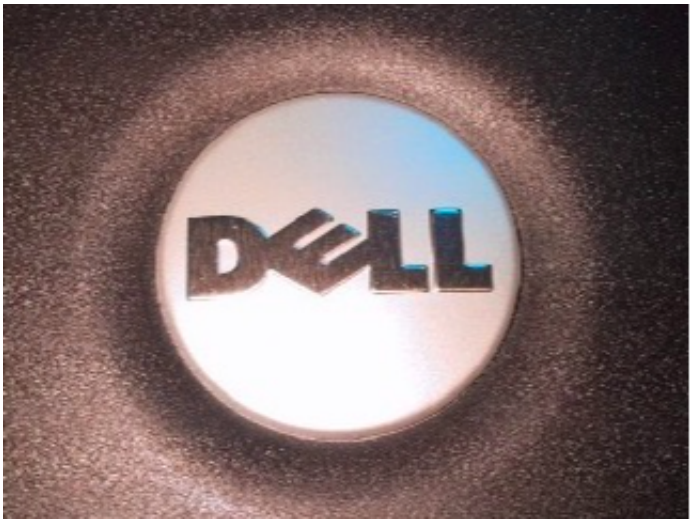


Fig 5: Original image before data augmentation methods

Results from augmentation methods:

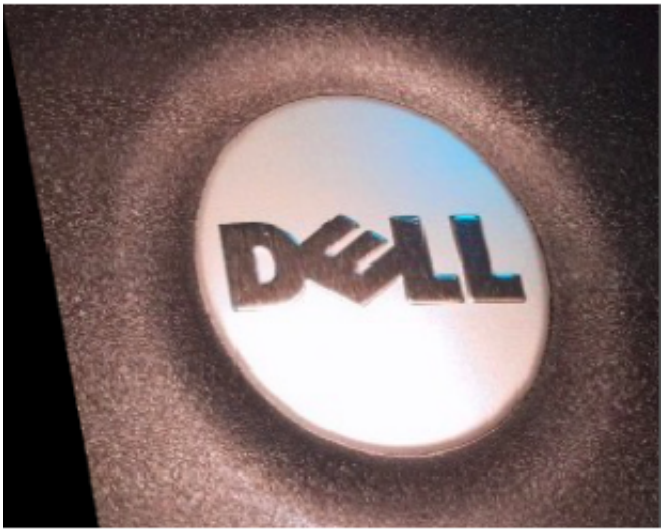


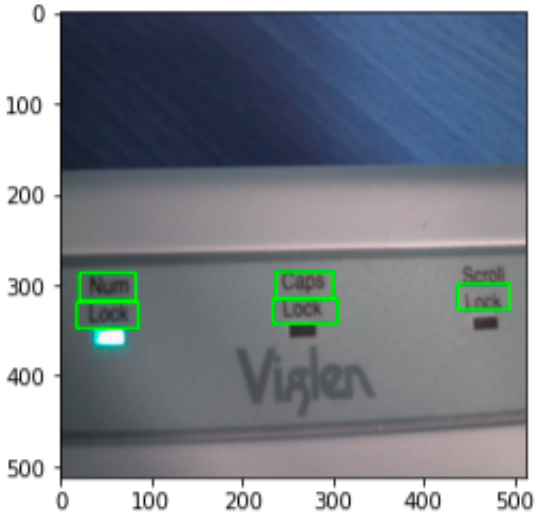
Fig 6: Shear parameter of 0.2

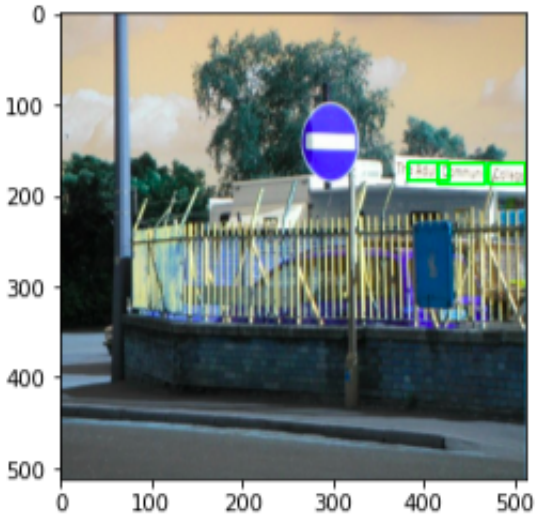
Fig 7: Rotation by 90 degrees

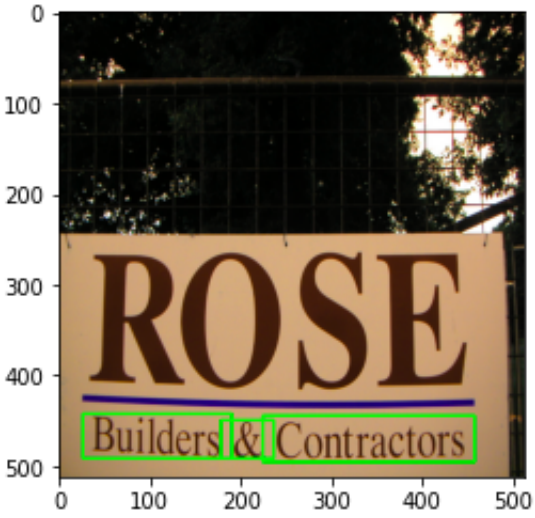
Fig 8: Vertical Flip

B. Text Localization

Text Localization detects the text in images and labels them using bounding boxes. Some of the results are shown below:







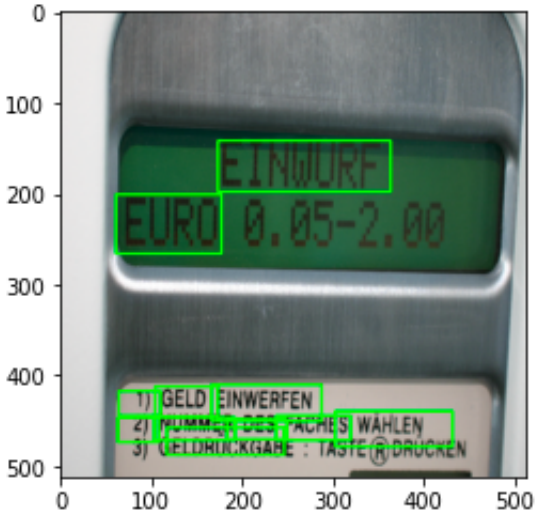
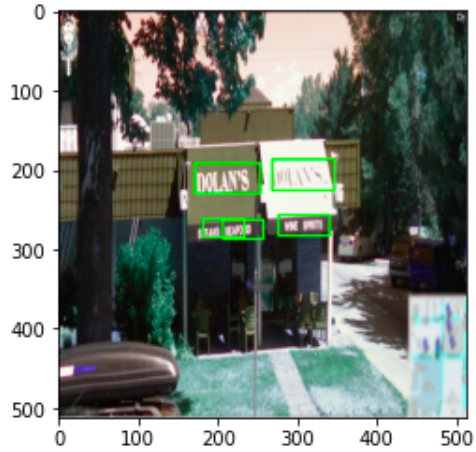
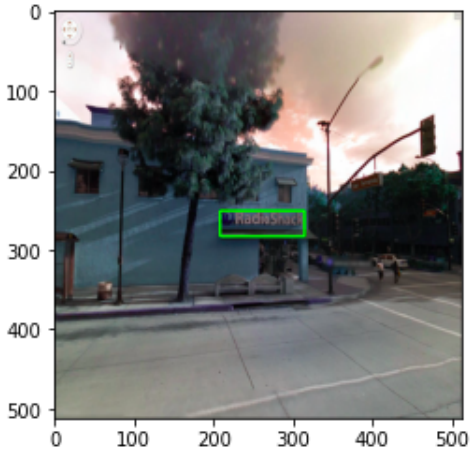


Fig 9: Text Localisation on images on train data

The performance of the trained model on test images produces the following results:





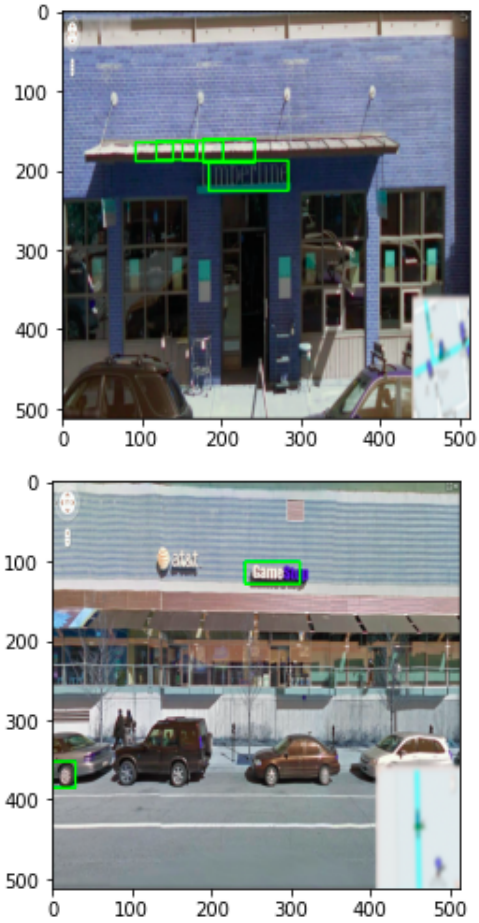


Fig 10: Text Localisation on images on test dataset

We see that the model is not very efficient. It is not able detect some text, and is creating bounding boxes at wrong parts of the image in some cases.

C. Text Segmentation

Text Segmentation divides the output of text localization into individual character images. It looks like below:

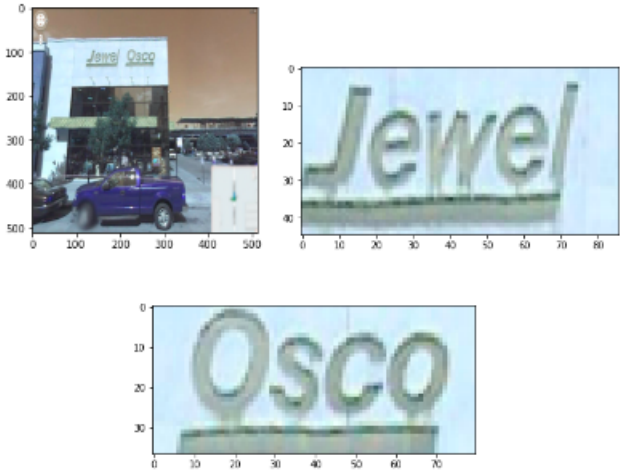


Fig 11 A: Text Segmentation on images

Fig 11 B: Text Segmentation on images

D. Text Recognition

After achieving segmented words for images, we aim to define a list of digits 0-9 and capital letters A-Z. The model attempts to use this information to predict the text in these segmented images, which looks like below:

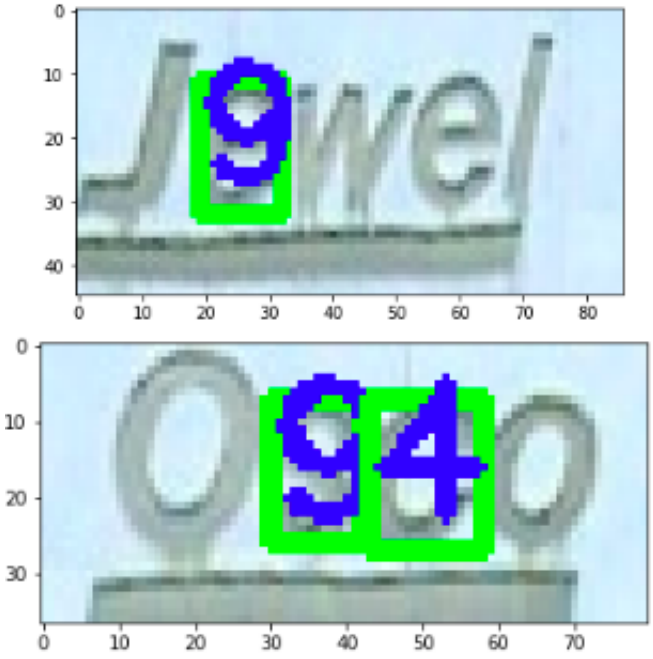


Fig 12: Text Recognition on images

VI. Conclusion

We see that the Yolo model does a good job at successfully localizing the text in images. There are other architectures too that could have been used, like VGG16 and ResNet50. But, upon research, we found that Yolo outperforms these models when it comes to this task in particular. Text segmentation is also achieved to break down the detected text into individual words. These results are input into the model to predict characters, which we see is not performing well. This might be because of insufficient data for training, or the model requires immense time in training.

VII. Discussion

This project is focussed on only identifying words in scene images. It does not detect any spaces or special characters in the image and hence, does not retain the structure of the text. However, this process can also be extended to detect sentences and graphical elements like figures and tables in an image.

To limit the complexity of the project, we have trained the model to only recognize characters of the English language. The model can be further trained on other languages to allow for recognition in other languages as well. It can also be extended further by providing translation services and spell checks.

This project was focussed on detecting and recognising text in scene images. For more generalized or domain specific applications, appropriate datasets can be used to detect and recognize specific details in the images.

While our efforts to implement text recognition were not fruitful, we aim on improving the model in near future.

References

[1] IBM Cloud Education — What Is Optical Character Recognition (OCR)? — IBM. (2022, January 5). — IBM; www.ibm.com. https://www.ibm.com/cloud/blog/optical-character-recognition

[2] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In 2013 12th International Conference on Document Analysis and Recognition, pages 1484–1493. 9 textvqa. ArXiv, abs/2006.00753, 2020. 3, 7 IEEE, 2013.

[3] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pages 1156–1160. IEEE, 2015.

[4] Chee-Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaitao Zhang, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, Chee Seng Chan, and Lianwen Jin. Icdar2019 robust reading challenge on arbitrary-shaped text (rrc-art). In Proc. Int. Conf. on Document Analysis and Recognition, 2019.

[5] Baek, J., Matsui, Y., Aizawa, K.: What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

[6] Linjie Xing, Zhi Tian, Weilin Huang, and Matthew R Scott. Convolutional character networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019