Talia Leitner

1.

The goal of my project was to design an algorithm that could read expiration dates. The idea was for my program to take a picture of a food product's expiration date as input and convert the date into a string. The expiration date would then be returned to the user and stored in a database for restaurant or grocery store owners to keep track of their inventory.

2.

I used the OpenCV text detection model. It is an example of an efficient and accurate scene text (EAST) detection model. This neural network takes an image as input and returns two arrays. The arrays contain the coordinates of the areas that it finds to have text and the probability that there is text in those areas. This is an example of a supervised model that uses regression and is not explainable.

Additionally, I used the Pytesseract optical character recognition (OCR) model. This neural network takes an image as input and returns all detected text as a string. It is an example of a supervised model that uses classification and is not explainable.

I chose to use these models because many times food product labeling has many colors and designs. The first model indicates which areas of the image contain text so that it is easier for the second model to read text from the image.

I had also sought out to build a supervised model that uses classification to detect the angle of rotation of an image (0 through 180 degrees). This would be useful since both of the above models fail to detect text in rotated images. Unfortunately, I did not have enough data to train this model.

3.

The product I was building is a web application that takes an image of a food product's expiration date and returns the date as text. This is useful for restaurant and grocery store owners because their employees can take pictures of their inventory on their smartphones. These photos can then be uploaded to the app. The app returns the digitized expiration dates that can then be stored in a database. This database can be used by the owners to keep track of when their food expires.

4.

The data used during this exercise were images of expiration dates on food products. Some of the images were taken by me and Debbie Cohen, others were found online, and the rest were collected by the company we worked with. The limitations of our data were that we were unable to collect enough images and the images did not come labeled. We worked with a company based in Israel, so we tried to find expiration dates that would be seen on food products there. We noticed that there was measurement bias since our training data was composed of images in JPG format. The images that would likely be uploaded to our application by users would be smartphone photos in HEIF format. We therefore added a method in our program that converts the HEIF photos into JPG format to resolve this issue.

The model we sought out to build to detect rotation angles required images of all different angles. We selected the images from our existing dataset that were at a 0 degree angle and rotated them in 9 different directions. We saved these images in an array and saved the rotation angles in a separate label array.

5.

The models that I used were pre-trained, but I tried out different techniques of preprocessing to enhance the text detection. The preprocessing included gray scaling, thresholding, blurring, and sharpening the images.

6.

I evaluated the performance of my program based on the accuracy of the two models. First, I checked whether the OpenCV text detection model accurately found all of the areas in the image with text. Next, I checked whether Pytesseract accurately converted the text from the image into a string. I compared the results of the images that had been processed by the first model, and cropped accordingly, with those that had not. I also compared the results of the images that had been preprocessed, using methods from OpenCV, with those that had not.

7.

The limitations of my model are that it does not work well with text that is in dot matrix font and with images that are rotated. I attempted to fix the issue of the dot matrix font by blurring the dots together and then sharpening the image. I attempted to fix the issue of the rotations by designing a model that detects the rotation angle of an image.

8.

It would have been useful to have a complete dataset with images of expiration dates and labels with their text. A dataset with rotated images and labels with their rotation angles would have also been useful in training the rotation model.

9.

If I had more time, I would work on an algorithm that takes the string returned by the program and converts it into standard date format. The algorithm would remove any text that is not part of the expiration date and properly format the month, day and year.

10.

The main challenge that I encountered during the course of this project was preprocessing. It was very difficult to find a method of preprocessing that worked for all of the images in my dataset. The images with text in dot matrix font needed to be blurred and sharpened. Other images did not need as much preprocessing. It was also very difficult to determine the angle that an image needed to be rotated so that it could be used by the program. I found that the text could only be detected if the image was taken at an angle between 3 and -6 degrees. If I could have successfully trained a model to detect the angle of rotation of an image then this problem could have easily been resolved.