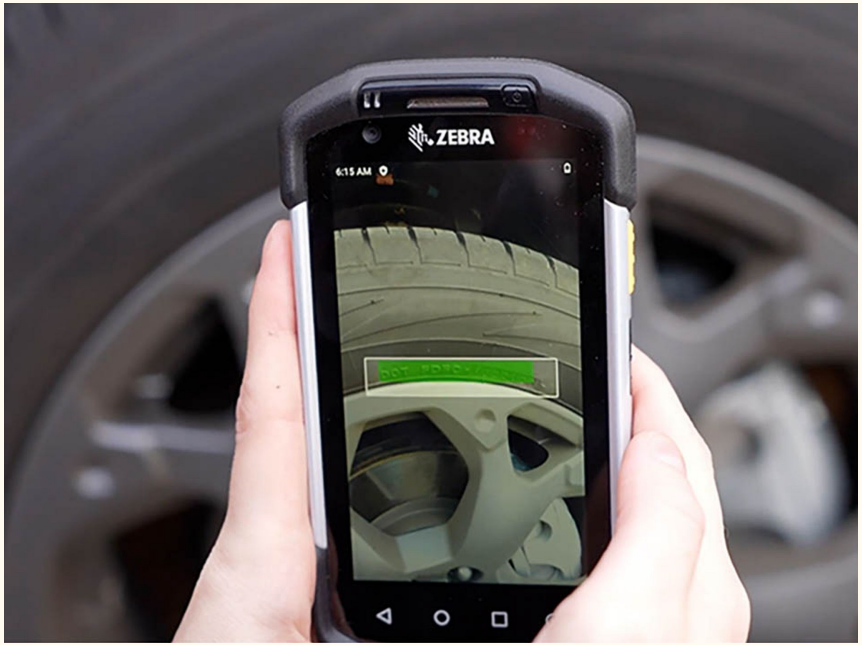
COGS 118B Final Project

Aniket Bhosale

Optical Character Recognition (OCR) using Spectral Clustering - based segmentation

What is OCR?

- OCR is the process of converting an image of a text into machine-readable format.
- Popular in workspaces where a large amount of printed information is handled, like forms, or documents.
- Two common ways it is done - **Pattern Matching** and **Feature Extraction**

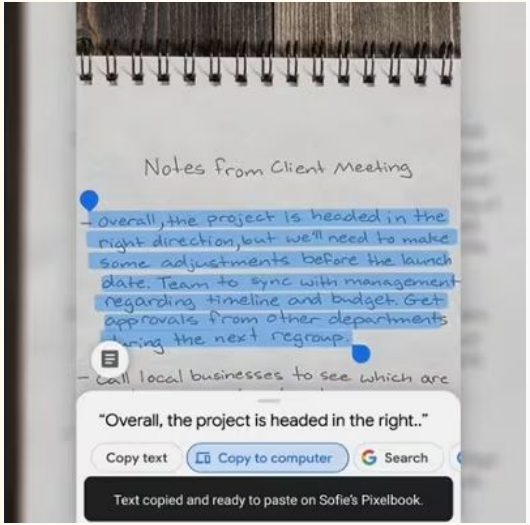


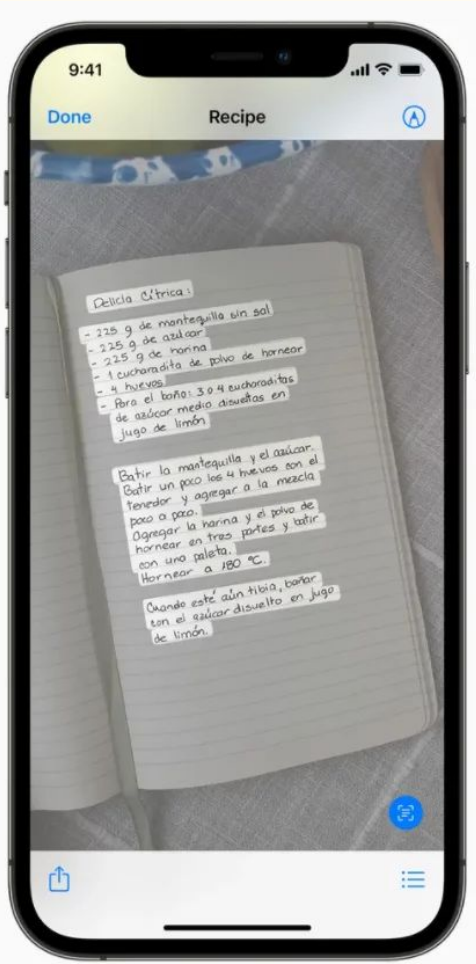
https://www.zebra.com/us/en/resource-library/faq/what-is-ocr.html

https://aws.amazon.com/what-is/ocr/#:~:text=Optical%20Character%20Recognition%20(OCR)%20is,words%20in%20the%20image%20file.

Motivation

- With IOS 15, Apple released live text OCR using the camera app
- Google also has their version of OCR in Google Lens
- Translate App also has had a similar feature for a while now
- Continue the application of past work with text-based data





https://www.theverge.com/2021/6/7/22522859/apple-live-text-photos-notes-ios-15-wwdc

Relation to Class

- Covered spectral clustering in class and for a homework assignment
- Similarity matrix has an important relation here
- Creating the right similarity matrix can make segmentation easy
- NIPS and Shi & Malik were also two important pieces of literature covered in class

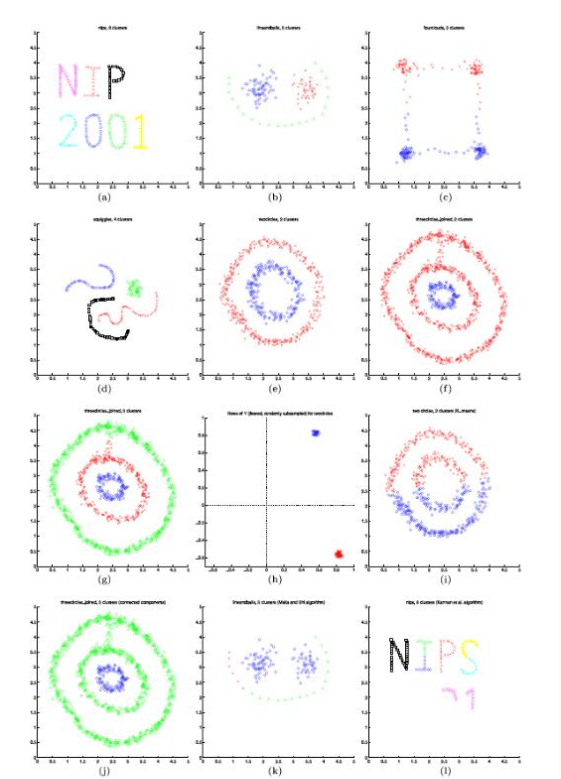
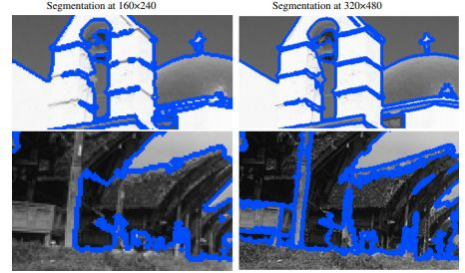
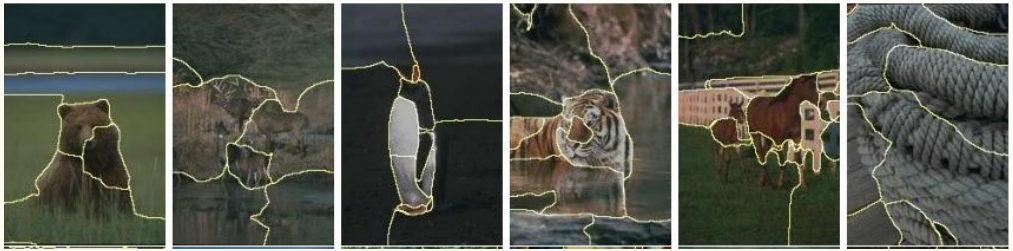


Figure 1: Clustering examples, with clusters indicated by different symbols (and colors, where available). (a-g) Results from our algorithm, where the only parameter varied across runs was k. (h) Rows of Y (jittered, subsampled) for `twocircles` dataset. (i) K-means. (j) A "connected components" algorithm. (k) Meila and Shi algorithm. (l) Kannan et al. Spectral Algorithm I. (See text.)

Similar Literature

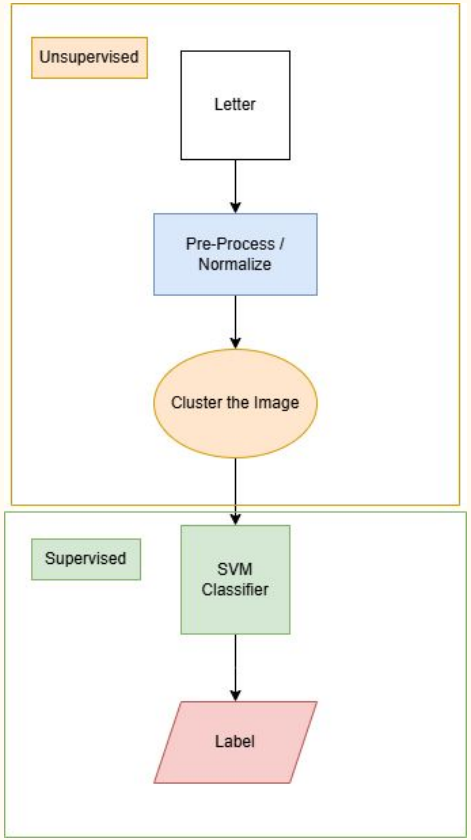


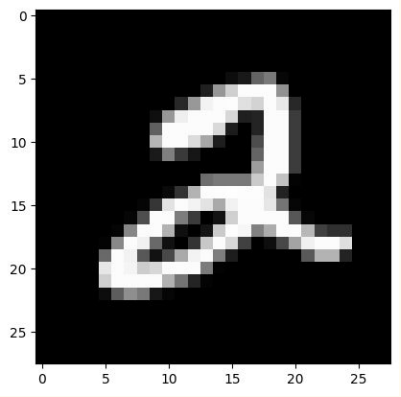
- Application of image segmentation through spectral clustering have been explored before.
- *How Much Does Globalization Help Segmentation?* By Fowles and Malik explore segmentation using spectral clustering - also shown in class
- *Enabling scalable spectral clustering for image segmentation* - By Tung et. al also explores segmentation using the same dataset but with blockwise processing and Stochastic Ensemble Consensus
- Fowles and Malik also use pixel affinities but in addition use more features like texture color and contours

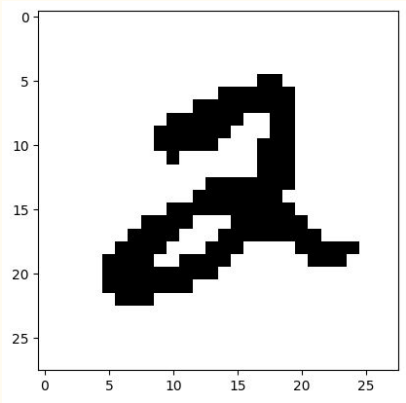


Process

- Start out with MNIST as a benchmark of single-character OCR
- Main idea architecture involves a combined supervised and unsupervised learning approach
- Use spectral-clustering to ‘isolate’ the correct digit before classification



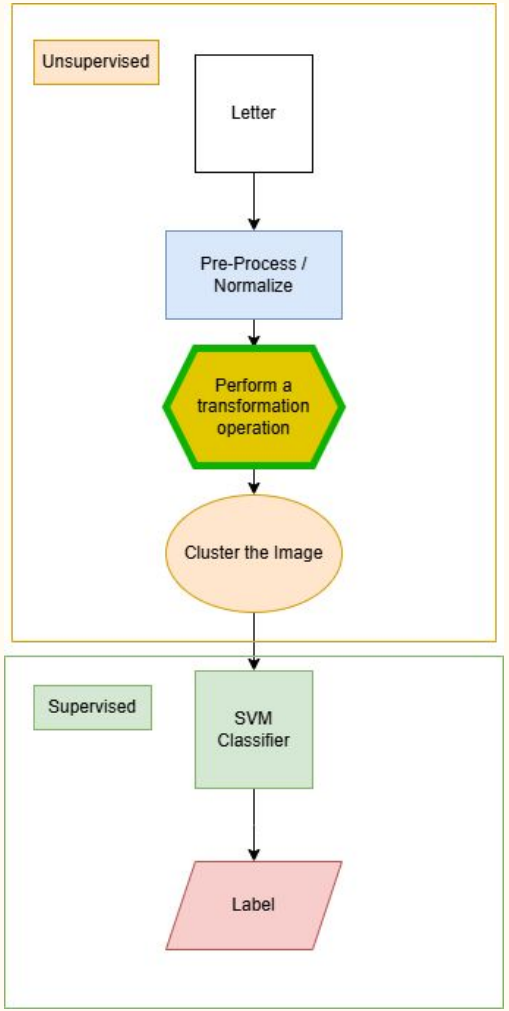




Process

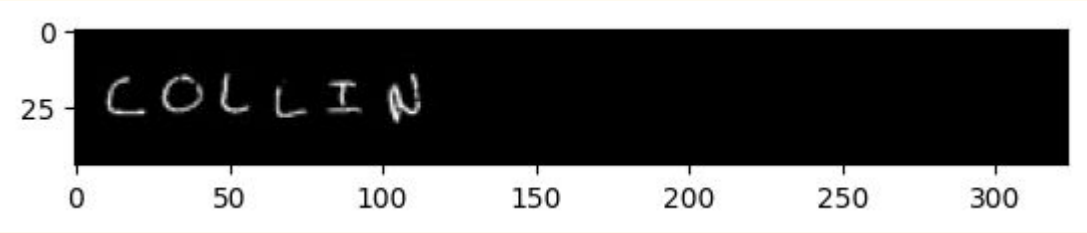
- Key issue with this approach is that it only accounts for intensities.
- To adjust for this, a new pre-processing transformation step is introduced where the X, Y coordinates of a pixel are also provided during similarity matrix construction

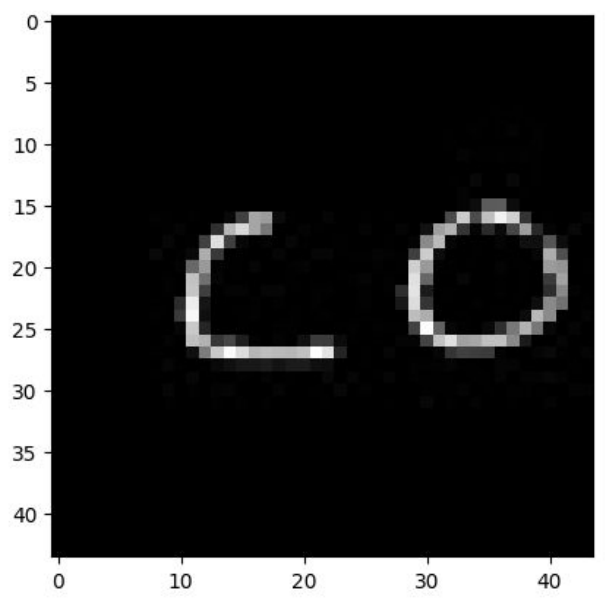
```python
def project_image(img):
    # Create X and Y coordinate matrices
    x_coords, y_coords = np.meshgrid(np.arange(img.shape[0]), np.arange(img.shape[1]))

    image_data = img.flatten()

    # Flatten X and Y matrices to match the flattened image
    x_flat = x_coords.flatten()
    y_flat = y_coords.flatten()

    # Stack the image data with x and y coordinates
    image_with_coords = np.vstack([image_data/15, x_flat, y_flat])
    return image_with_coords
```



Process

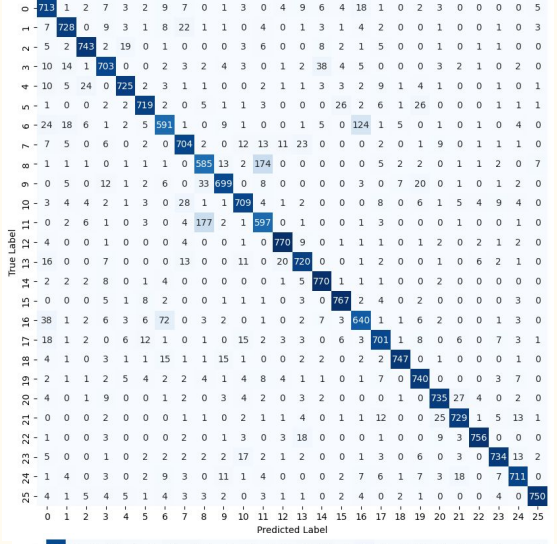
- To extract data, from the image, we need to run the spectral clustering on the image.
- Then isolate each letter from the cluster into its own image and run it through our classifier.

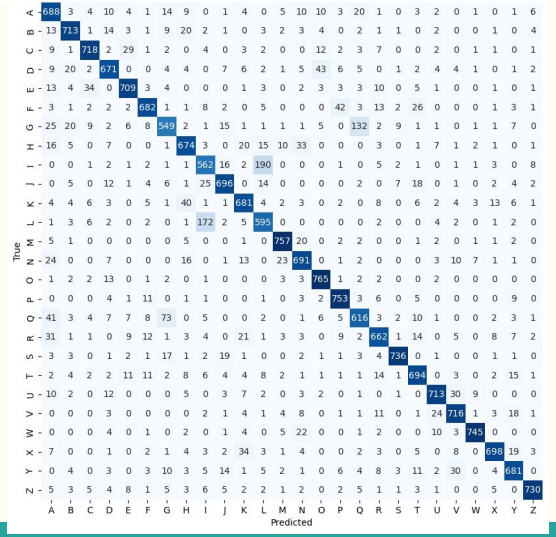




Results

- Originally the primary version of the model was trained on the MNIST and the EMNIST datasets
- From the images shown before, we can see that the model does a good job of identifying the shape.
- One very interesting result was that the accuracy of the model was similar when the training data was run through spectral clustering during pre-processing.
- However, the prediction became more robust in the latter





	Training Data	Validation Data	Testing Data
Without Spectral Clustering	98.92%	97.65%	97.73%
With Spectral Clustering	98.16%	96.5%	96.6%

Accuracy of Classifier with vs. without Spectral Clustering done in pre-processing

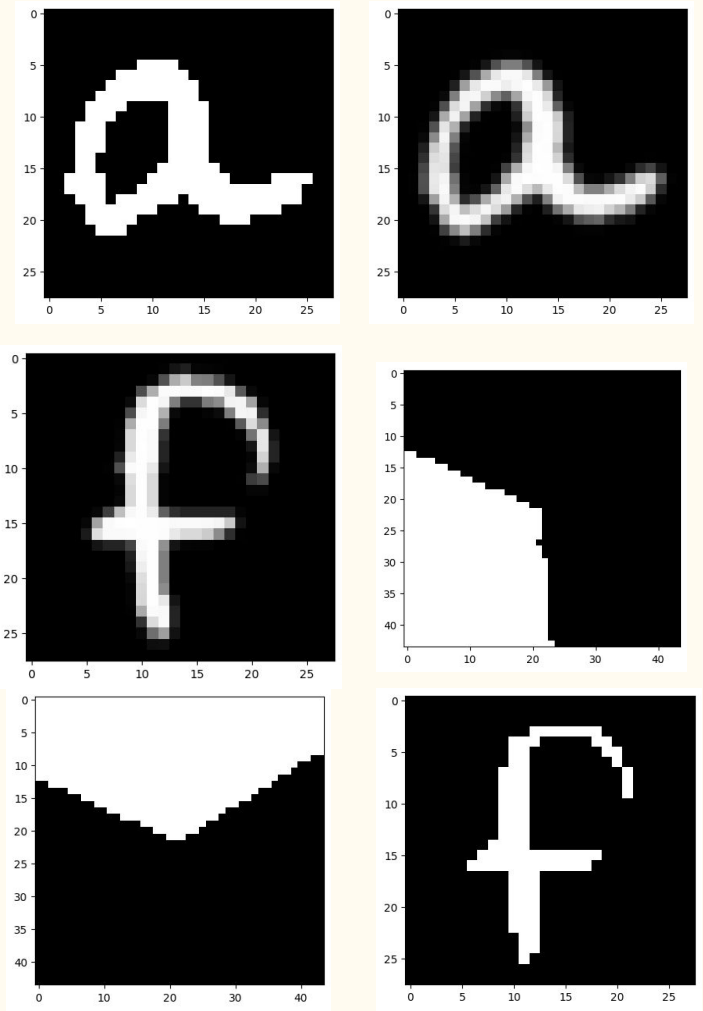
	Training Data	Validation Data	Testing Data
Original Classifier	83.75%	83.16%	82.74%
Classifier w/ Spectral Clustering	96.94%	95.95%	96.31%

Accuracy of Classifier when the inverse data is presented to it (without gets clustered data and vice versa)

Note: A SVM Classifier with a RBF Kernel was used as the classifier.

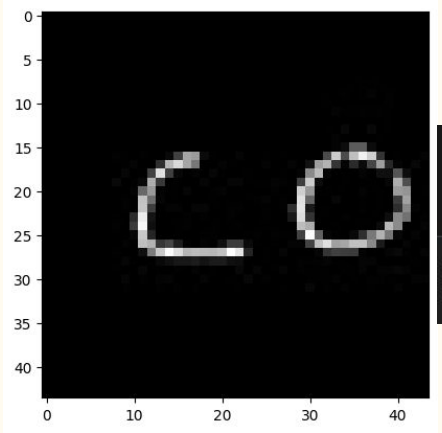
Results

- The results of running the data through the EMNIST classifier were similar.
- The addition of the spatial coordinates caused a heavy bias towards the cuts that favored spatially closer pixels rather than pixels with similar intensities
- This issue was resolved by adding a constant scaling factor to the pixel intensities that projected them higher than the spatial data.



Results

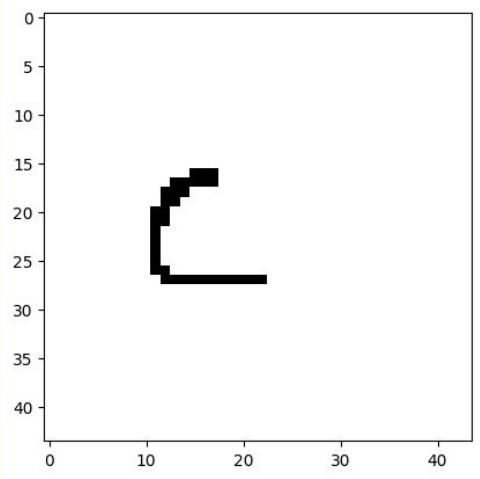
- Sliding window is required to maintain a computationally efficient method
- Training Data lacks scaling so we get a lot of misclassifications.
- Centering and scaling the new images solves the issues and we get correct predictions
- Need to sweep sigma for each different image

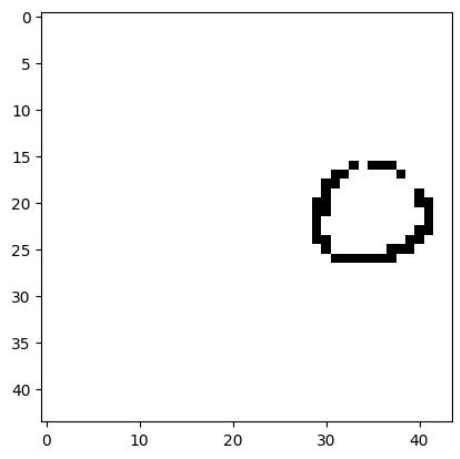


```
print(f'Predicted Letter 1: {image_decoded_1}')
print(f'Predicted Letter 2: {image_decoded_2}')
```

✓ 0.0s

```
Predicted Letter 1: C
Predicted Letter 2: O
```





Discussions

- Having a dataset with different scaling was troublesome and can be improved
- Running a scree plot of the hyperparameters can be done with more computation resources
- Sequencing of the images would be a great addition and can segway into some good NLP work
- Number of eigenvectors suggested by the related work was not working in this case. Noticed that the number of eigenvectors = k worked better
- Obtaining data of the appropriate scaling is hard. Further improvements can also revolve around smarter image scaling and centering.

Future Applications

- This effort has provided motivation that Spectral clustering may not be the right approach for OCR.
- Spectral Clustering is very computationally intensive and needs sequencing which is not viable for real-time OCR like the motivation for the project.
- Pre-clustering data before training is a good take-away and may make models more robust.
- Pre-clustering may be very useful for semantic segmentation. It can provide especially useful to find desired objects in an image very quickly by running clustering and then picking the objects with the desired attributes for predictions.