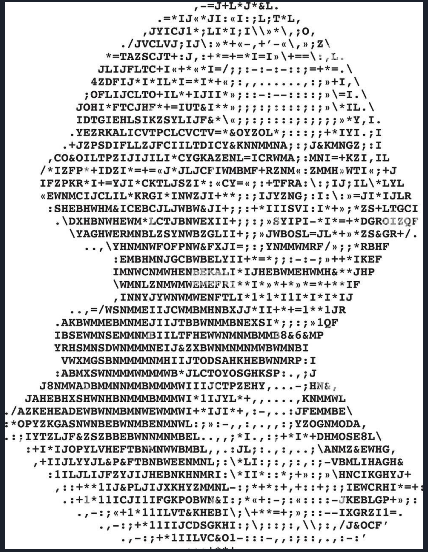
Evaluating Machine Learning Approaches for ASCII Art Generation

Pranav Akiri

Background

- ASCII → American Standard Code for Information Interchange
- ASCII art is typically created by mapping grayscale pixel values to ASCII characters
  - @ is "darker" than -
- ML models are used to classify image segments into ASCII characters instead

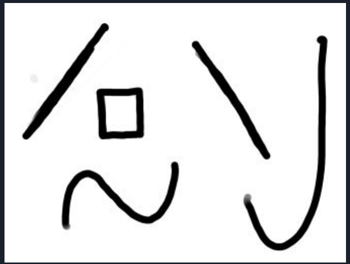


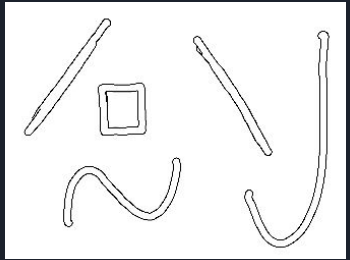
Dataset

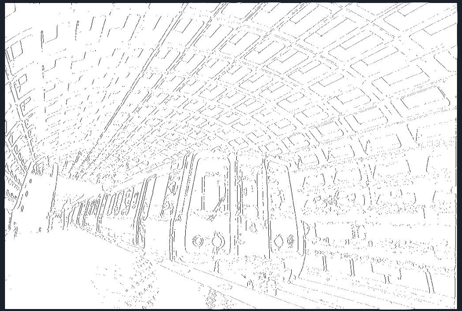
- Past research (Akiyama 2017) has entirely recreated line structures of numerous ASCII art images and further segmenting them into smaller tiles to make the dataset, which can be time-intensive
- Dataset was created by taking random samples of ASCII character image tiles, applying transformations, and assigning each character to a corresponding image
  - Gaussian blur, random noise
- Numerous classes corresponding to each possible ASCII character

Preprocessing

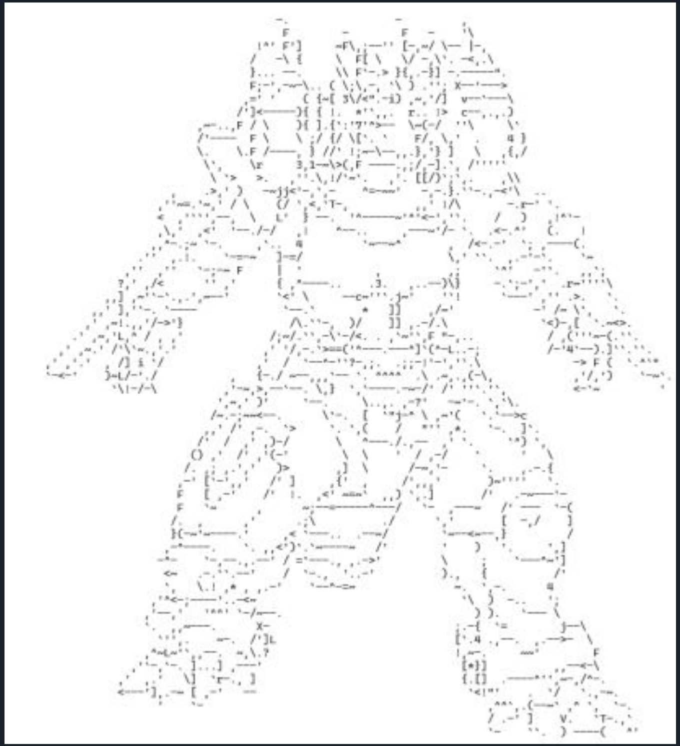
- Images are blurred using a Gaussian blur then Canny edge detection is used to mark lines
- Resizing to ensure resulting ASCII image doesn't look stretched
- Image is divided into 10 x 10 tiles (past research (Akiyama 2017) has used 64 x 64 tiles)
  - Visual outputs were better with the smaller tile size



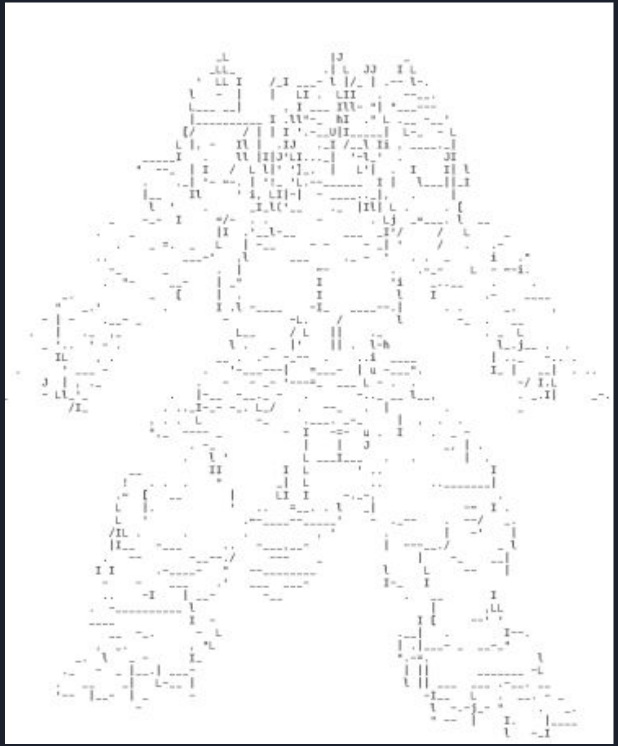




Outputs with Different Tile Sizes



KNN with 10 x 10 image tiles

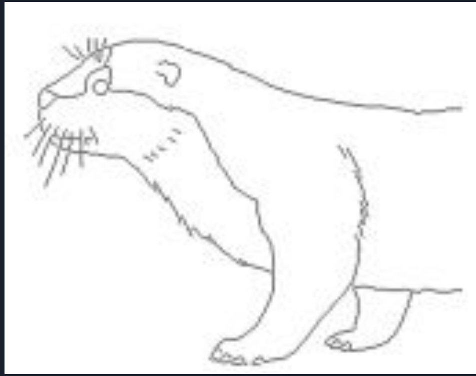


KNN with 64 x 64 image tiles

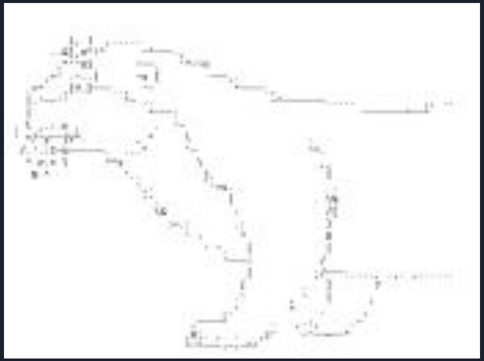
Models

- KNN
  - Data point is classified based on the 5 nearest neighbors (k = 5)
  - Each data point consists of image pixel values with the associated ASCII character as the label
- SVM
  - Kernel trick is used to separate overlapping data
  - Hyperplane classifies data points between the two closest matches
- Random Forest
  - 100 decision trees, each running through several possible ASCII character options
- All models were run using scikit-learn with default parameters

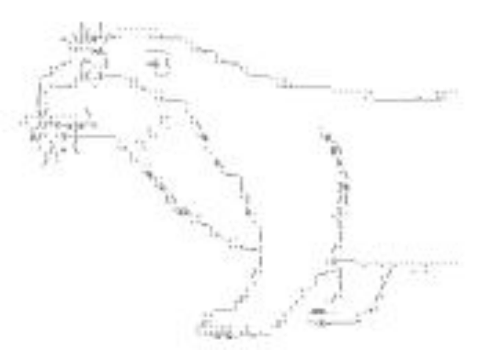
Results



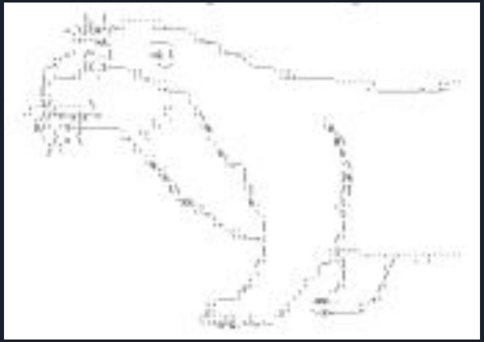
Original Image with Canny



KNN



SVM



Random Forest

Performance Metric Results

Model	F1 Score	Recall
k-NN	**0.95**	**0.96**
SVM	0.93	0.94
Random Forest	0.91	0.91

- KNN outperforms SVM and Random Forest in both F1 Score and Recall
  - Relies on majority voting instead of binary separation

Novelty & Conclusion

- Combines computer vision with machine learning to produce well-performing results
- Use of machine learning models for image processing tasks
- Classical machine learning methods, like Random Forest, are applied to a context they're not used as much in yet perform well
- Deep learning models do not always give better results than regular machine learning models and machine learning models like KNN, SVM, and Random Forest are well equipped for certain problems in other fields of artificial intelligence

Related Work

- Akiyama 2017 - "ASCII Art Synthesis with Convolutional Networks"
  - Focuses on the use of convolutional neural networks (CNNs) specifically to make ASCII art images
  - CNN performance was somewhat worse, perhaps due to using larger image tiles
- Xuemiao et al. 2016 - "ASCII Art Synthesis from Natural Photographs"
  - Utilizes a non-CRF (classical receptive field) model and separates data that perceptrons might find hard to distinguish with normal data for better results
  - This approach does not use any of the ML models we learned but it does show that there are more bespoke ways to create ASCII art

Sources

Akiyama, O. (2017). ASCII Art Synthesis with Convolutional Networks. https://www.semanticscholar.org/paper/ASCII-Art-Synthesis-with-Convolutional-Networks-Akiyama/eb23e948e576e96fd138e85e13bcec9f93aacda7

Coumar, S., & Kingston, Z. (2025, March 18). Evaluating Machine Learning Approaches for ASCII Art Generation. https://doi.org/10.48550/arXiv.2503.14375

Xu, X., Zhong, L., Xie, M., Liu, X., Qin, J., & Wong, T.-T. (2016, June 14). ASCII Art Synthesis from Natural Photographs. https://ieeexplore.ieee.org/document/7491376