Optical Recognition of Hand-Drawn Chemical Structures

Bradley Emi
Stanford University
Dept. of Computer Science

Abstract

Optical chemical structure recognition is the task of converting a graphical image of a chemical molecule into its standard structural representation. Specifically, the chemical structure recognition algorithm should correctly identify the graph structure with correct atomic/group labels for each node, and the correct type of bond label for each vertex. We introduce a novel method to improve upon state-of-the-art methods with an eye towards solving the problem in the face of the additional difficulties when molecules are hand-drawn. We employ basic text recognition and corner detection methods to first label the atoms and groups that form the nodes of the chemical structure graph, and conclude that our approach to corner detection outperforms the line vectorization algorithms typically used in other systems. A Hough transform is used to recognize the presence of bonds between the nodes. The major difference in our approach is to use a new technique that classifies bonds according to various feature descriptors of sliding-window cross-sections of bonds using a supervised machine learning approach. In addition to the baseline method of using the Hough transform to also classify bonds, we use local maxima detectors on single-pixel slices of bond cross-sections and histogram of oriented gradients (HOG) features of wider bond cross-sections coupled with support vector machine (SVM), logistic regression, decision tree, and neural network classifiers. We compare the results of these feature descriptors, analyzing our pipeline on a hand-drawn dataset of 360 simple molecules and conclude that these new bond recognition technique leads to major improvements in recognition performance over the baseline.

1 Introduction

The standard presentation of organic chemical data in a wide variety of fields, such as biology, chemistry, and medicine, remains the structural diagram, which contains all the chemical information of a given molecule, but is unsuitable for computational analysis. The problem of optical structure recognition, the conversion of these images of structures into the usable, machine-readable labeled graph data formats, remains highly inconvenient and inaccurate in many cases. Wide availability of this kind of data from scientific patents, journal articles, textbooks and other printed sources would lead to major progress in not only chemistry, but also pharmaceuticals [1], chemical biology [2], medicine [3], and several other fields of scientific research. A tool for chemical structure recognition would also open up new possibilities for modern artificial intelligence and data mining applied to already existing datasets that currently only exist in image format. Furthermore, virtually no work has been done on the generalization of optical structure recognition to hand-drawn chemical molecules. Current research approaches focus on optimizing computer-generated algorithms to improve accuracy on extremely large molecules typically presented in scientific patents. (Fig. 1) Little work has been done on smaller, more common hand-drawn molecules. (Fig. 2) While computer generated molecules are difficult enough to recognize, smaller hand-drawn molecules present different and unique challenges from a computer vision perspective.

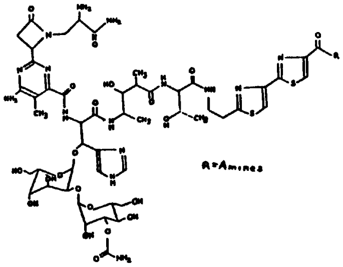


Fig. 1: A molecule from U.S. Patent Class #435; most current optical structure recognition research focuses on similar molecules.

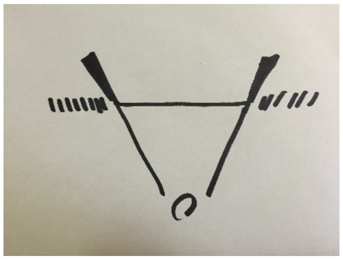


Fig. 2: A molecule from our dataset. Its overall structure is much simpler, but its bonds and labels require different treatment.

There are several additional advantages of being able to recognize handwritten molecules in addition to computer-generated molecules. For example, the computer generation of molecular structures is currently quite tedious, and an application to perform real-time recognition of small components of hand-drawn structures does not yet exist.

2 Review of Previous Work

2.1 Summary of Previous Work

Previous work on the optical structure recognition problem has to date focused exclusively on computer-generated structures. Early research began in the 1990s, with IBM receiving a patent for recognition of chemical graphics among other printed material on a page as well as basic line tracing techniques (Fig. 3) to recognize structures. [4] A similar approach was developed by University of Leeds researchers and called CLiDE in the same year. [5]

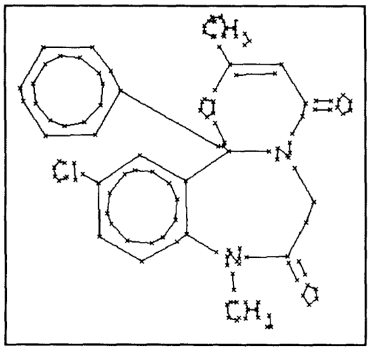


Fig. 3: IBM's line-tracing algorithm.

More modern approaches, such as ChemReader from the University of Michigan [6], and the National Cancer Institute's open-source OSRA [7] have both employed more sophisticated text recognition (OCR) and line detection algorithms; ChemReader uses a generalized Hough Transform and OSRA uses the Potrace library.

State-of-the-art approaches, such as the approach of MLOCSR, developed by Italian researchers Frasconi, Gabbrielli, Lippi, and Marinai, generally recognize molecules in two stages. The first stage is a low-level processing module, which detects edges, corners, and text; and a high-level reasoning engine, which uses Markov logic networks, utilizing prior chemical and graphical knowledge to correct errors in the low-level module. [8] Modern approaches, such as a more recent iteration of CLiDE, also use a specialized artificial neural network to classify text labels.

2.2 Improvements to Existing Approaches

The focus of this paper is the novel approach to the correct identification of hand-drawn bonds in the low-level module, the correct identification of atoms and edges without the use of high-level correction using chemical and graphical knowledge. Previous attempts at optical structure recognition, even state-of-the-art approaches, are heavily dependent on the correct identification of fine lines (the individual thin lines constituting double, triple, and dashed bones), which fails in the case of imperfect hand-drawn bonds. Frasconi et. al.'s algorithm, MLOCSR, uses the Douglas-Peucker algorithm [9] to approximate the contour of the molecule with a polygon which fits the least-vertex polygon to the contour within a certain precision. We hypothesize that line detection-only vectorization algorithms such as the Douglas-Peucker algorithm may fail in cases where bonds are not straight (Fig. 4), assigning too many vertices to the molecule. Furthermore, classification algorithms can fail when dashes follow an irregular pattern and/or touch (Fig. 5).

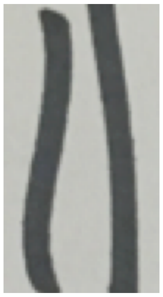


Fig. 4: Bonds are not straight, a difficulty for line-only vectorization

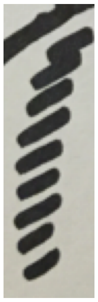


Fig. 5: Dashes touch and are irregular

Several algorithms use the Hough transform to detect lines and line segments. However, Fig. 6 displays the difficulty of using the Hough transform on a hand-drawn image. Even when the threshold for the required number of votes for the Hough transform to detect a line is optimized, both false positives and false negatives still occur for reasons specific to hand-drawn images.

The solution to these difficulties is by using the Hough transform to only *recognize* bonds. Bonds are then classified according to the features of their horizontal cross-sections. In this paper, we experiment with a number of features and classifiers to optimize the accuracy of bond type recognition. These experiments constitute a majority of the current paper, with enhancements to text recognition and further experimentation on the higher-level module incorporating chemical knowledge representations forthcoming in future work.

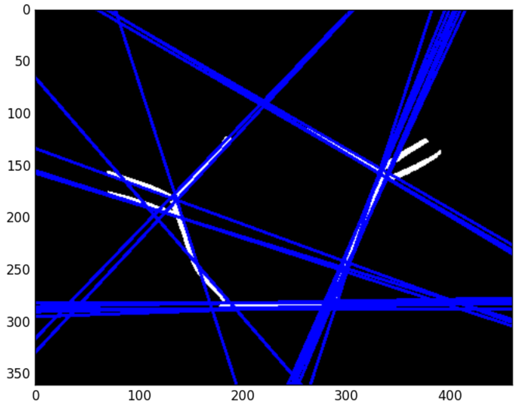


Fig 6: Difficulty of using the Hough transform on a hand-drawn molecule. A false positive is detected on the bottom left because of the bend in the single bond. The double bond on the right, meanwhile, is undetected as a false negative.

3 Methods

3.1 Summary

The abstract general structure of the pipeline we use in order to recognize the chemical structure of our molecules is as follows:

- Recognize text labels using scale-invariant template matching
- Removal of text from image
- Bond and corner detection
- Bond detection
- Bond classification
- Association of corners to atoms and groups

Due to the lack of a large amount of hand-drawn chemical data and the small number of labels, a simpler scale-invariant template-matching approach was used to detect text in images with reasonable accuracy. Other approaches, such as Google Tesseract [9] and other supervised learning classifiers with histogram of oriented gradients (HOG) [10] features were attempted. Tesseract was exceedingly difficult to configure due to its more general use parameters, which use a number of language models which make a large number of assumptions in order to more accurately recognize more structured sources of text. However, these assumptions, which do not apply to our purposes, were difficult to remove programmatically and debug. Future work may incorporate a state-of-the-art OCR engine; however, this approach was ultimately unsuccessful. HOG classifiers were found to suffer from a large number of false positives due to a lack of negative training examples. With more data, this approach could also prove to be more successful, but was inapplicable with our limited amount of training data.

Bond and corner detection was used to identify various points of interest of the coarse line features; intersections of lines that represent a carbon, or lines which end in a connection to a text box, representing a different atom or group. A coarse Gaussian filter was applied to the image before applying a Hough transform to find the lines and points of interest.

Finally, these points of interest were assembled to locate the atoms and groups of the molecule as well as the bonds, leaving bond classification as the final task. Cross-sections of the bonds were analyzed using various feature descriptors and classified according to several machine-learning classifiers trained on 45 hand-drawn molecules. The success of our approach with a small training set size is again reason to believe this approach will become more accurate as more data is acquired. More details on the bond classification algorithm is described in the methods section.

3.2 Technical Solution

3.2.1 Data

Our data comprises 360 images of 9 different simple hand-drawn molecules (40 images of each molecule) drawn on standard white printer paper in fine point black Sharpie marker. Each image was taken with an iPhone 6 camera at 3264 x 2448 resolution with three color channels (no alpha channel) downsampled to 400 x 300 grayscale using bilinear interpolation. All the images were taken in identical lighting, and were drawn by three different people, so our model would not overfit to a particular person's drawing style. Each image was preprocessed with binarization using a 40% threshold. No other preprocessing stages were applied.

45 of the images (5 per molecule) were set aside as a training set.

3.2.2 Text Recognition

For text recognition, we tried a number of approaches, including scale-invariant template matching using 5 images of each of 6 templates ("O," "H", "OR", "RO", "N", and "OH"), and a number of supervised learning classifiers using 4 templates ("O," "H," "R," and "N"). By visual of the dataset, we estimated a minimum and maximum scale for the images at 20x20 pixels and 60x60 pixels respectively and implemented a spatial pyramid sliding window with the length of each square window increasing from 20 to 60 pixels in steps of 5 pixels. This was a very conservative estimate, and for reimplementation on different image sizes, we recommend scaling from 0.3% of the total area of the image to 3%.

For the supervised learning classifiers, to collect negative training examples, we randomly selected 1200 of these windows that were verified by hand to have no text from the training set. We then collected 5 of each of the 4 templates from the training set. To augment the number of positive examples, we additionally used 55 images for each template from the open source Chars74K handwritten dataset [11]. We cropped each image to eliminate whitespace and extracted histogram of oriented gradients features from each using 64 bins. We then compared the performance of a logistic regression classifier, a linear SVM classifier, and a neural network with one hidden layer with 30 nodes. Results are presented in section 4.

For the scale-invariant template-matching, we applied a Gaussian filter with size equal to half the width of the measured strokes to all training templates and the image for matching, and then used the spatial pyramid sliding window described above to match the images. We then chose the tolerance level, 0.77, for which the F1 score was maximized. Non-maximal suppression is used to remove overlapping bounding boxes. A sample of the output of this stage is presented in Fig. 7. We used the results of this algorithm for the next stages of the pipeline. More details are presented in section 4.

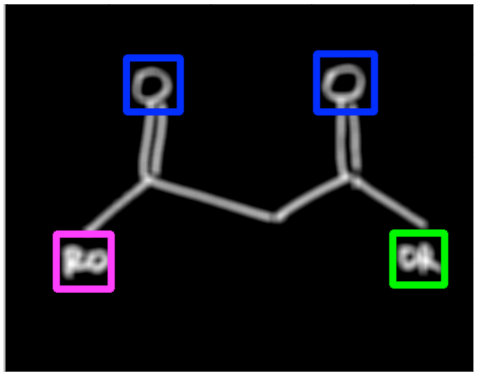


Fig. 7: Sample results of template matching OCR.

3.2.3 Corner Detection Overview

We reimplement the Douglas-Peucker algorithm on our dataset for comparison with MLOCSR, and we also implement a corner detection algorithm based on a broad Harris corner detector.

For clarity, we use the terminology of MLOCSR, defining a *C-point* to be a corner corresponding to the intersection of the main bonds of a carbon, a *D-point* to be the endpoint of a line segment not connected to the main bond to represent a double or triple bond, and a *T-point* to be the end of a line segment drawn to a text box to indicate a bond to a non-carbon.

3.2.4 Best-Fit Polygon Reimplementation

As in MLOCSR, we use the Douglas-Peucker algorithm to detect the vicinity of C-points and T-points, and look for D-points later once the main corners are located. This algorithm for each contour iteratively tries to fit *n*-vertex polygons to the contour, increasing *n* until no point on a contour is further than a threshold distance away from the polygon. The algorithm then returns the vertices of the polygon.

We search for clusters of all points of polygons that fit the opposite contours of the image after a Canny edge detector is applied in order to accomplish this goal. We use the threshold of $\sqrt{2}$ times the edge length as prescribed in MLOSCR. Fig. 8 shows the results of this stage.

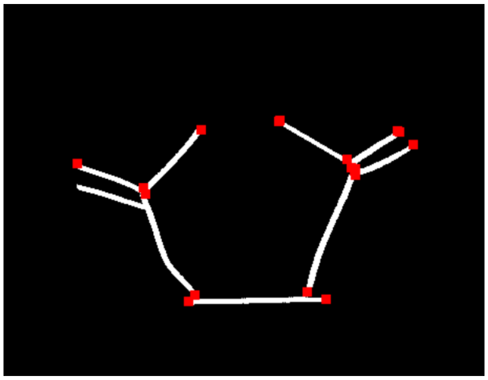


Fig. 8: Result of the Douglas-Peucker algorithm to fit a polygon to the contours of the Canny edges of the molecule image.

We then use a basic agglomerative clustering algorithm, setting a maximum distance between clusters to 50 pixels.. If a polygon vertex is less than 50 pixels away from an existing cluster center, we assign it to that cluster, updating the center point of that cluster. Otherwise, we initialize a new cluster. The results of this stage applied correctly to a molecule image are shown below in Fig. 9, with the blue points representing final cluster centers. Testing results are shown in section 4.

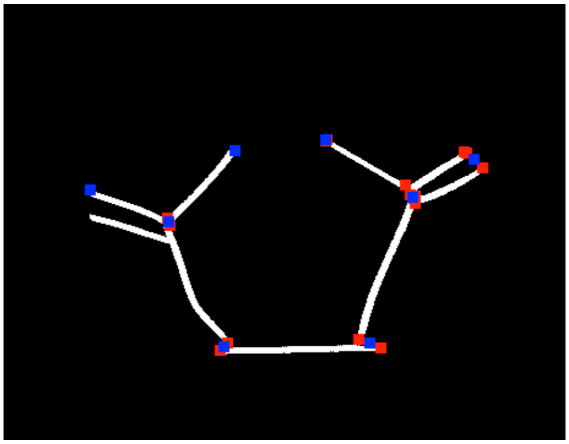


Fig. 9: Results of agglomerative clustering stage.

3.2.5 Harris Corner Detector

The goal of the Harris detector [12] in this context is the same; to identify the C- and T-points but not necessarily the finer D-points that distinguish double and triple bonds. The Harris corner detector looks for a high variation in the gradient of an image in two directions.

We first apply a coarse Gaussian filter to the image with the size of the estimated stroke width. We then run the Harris corner detector, once again requiring corners to be a threshold distance apart.

A sample result on the same molecule after filtering is presented in Fig. 10.

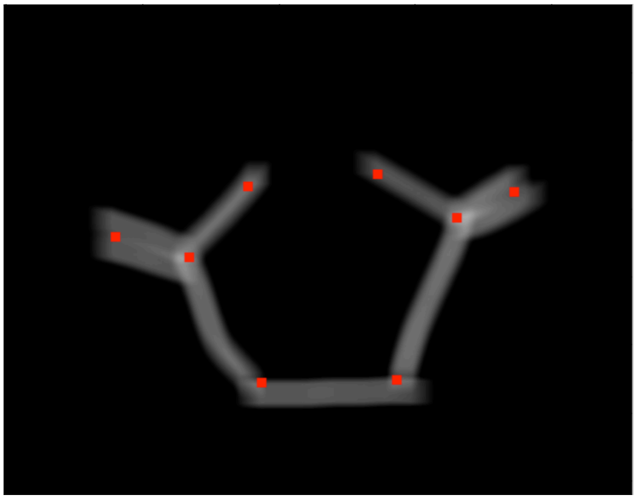


Fig. 10: Results of Harris corner detector.

A comparison of the results is presented in detail in Sec. 4, but we find that the Harris corner detector (89% accuracy) outperforms the polygon reconstruction method (75% accuracy).

3.2.6 Bond Detection

In contrast to methods based on line vectorization, which comprises not only MLOCSR but also a majority of the existing methods in the literature, we use the Hough transform only to *detect* bonds, rather than to *classify* them. Line vectorization methods make several errors and in the case of hand-drawn molecules, are not precise enough to detect the D-points that can be detected by the polygon reconstruction method when molecules are perfectly straight. As the state-of-the-art method in MLOCSR only recognizes under 80% of the C- and T-points, there is very little hope for such an algorithm to be able to detect the finer D-points given the large amount of variability in hand-drawn bonds.

Since a carbon can only have four bonds, for each of the nodes detected in the previous stage, we look at the four closest nodes to see if there is a bond between them. While further molecules are not strictly forbidden from being connected to a carbon, it is extremely uncommon, and this case does not occur in any of the molecules in our dataset. For more general molecules, more nodes can be examined and spurious matches can be removed using a Markov logic network similar to what is implemented in MLOCSR, but we do not implement that here for simplicity.

The other heuristic we use is that if three nodes are collinear, there is not a bond between the two outer nodes. This situation only occurs when there are two bonds at a 180-degree bond angle, so the outer nodes cannot have a bond between them.

These heuristics leave us with a number of candidate bonds, a list of possible node-node pairs that could contain a bond between them. To refine this list, we create a bounding box of the edge between the two nodes at a fixed width (40 pixels) and split the bounding box into windows of fixed size. On each window, we then apply the Hough transform with a very low threshold to look for lines in the window. We only accept lines that are within 1 degree of the expected direction of the bond. We then require that all of the windows in the bounding box contain a line detected by the Hough transform. We assume that if a node-node pair does not contain a bond, at least one of its windows will not have a matching line in the orientation of the node-node pair. This approach leads to several false negatives, but these can be rectified by a Markov logic network in later steps, because most of the false negatives are simply a missing bond in a ring or another predictable structure. This approach does lead to a relatively low false positive rate, and the false positives are generally simply spurious triangular closings, which are very rare in organic molecules and would also be removed by a Markov logic network. Results are presented in Sec. 4. The process is visualized in Fig. 11.

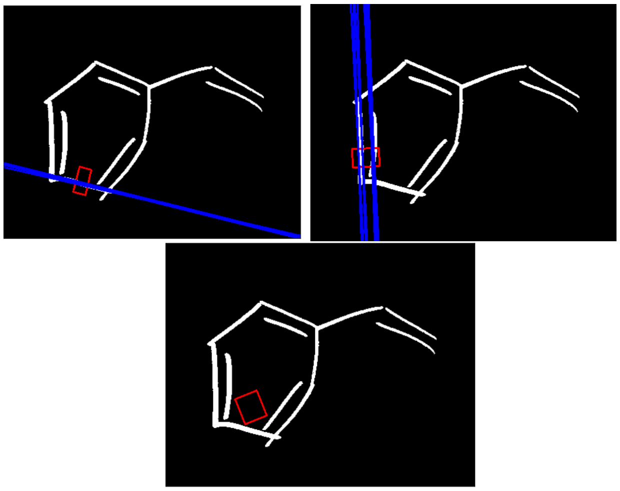


Fig. 11: Top left, top right: Hough line detections (blue) for the node-node pairs with a bond for a given window (red) in the bounding box. Bottom: A window between two opposite nodes that will not have a Hough line detection, so the algorithm will not assign it a bond, even though there is contamination elsewhere in the bounding box.

3.2.7 Bond Classification

The final part of the pipeline is bond classification, before higher-level modules use chemical knowledge to correct errors (future work).

We use a sliding window moving along the cross-sections of bonds extracted from our training set via screenshots, using a window size of 10 pixels down the length of the bond and 40 pixels across. Typically this will result in around 3 to 10 windows per bond. Our training set contains 62 single bonds, 33 double bonds, 10 wedge bonds, 10 dashed bonds, and 5 triple bonds.

We then use HOG features on each of the sliding windows and use these features to train a supervised learning classifier. We experiment with a multiclass logistic regression classifier, a linear support vector machine, and a decision tree. (We avoided using a neural network since it is prone to over-fitting in the case of a small training set).

Then, for each bond in the test set, we extract the same sliding window HOG features and use the classifier to predict the type of bond that appears in each of the windows. Then we employ a voting system where each window gets a "vote" for the overall type of bond. A sample result is shown below in Fig. 12, and overall results are presented in Sec. 4.

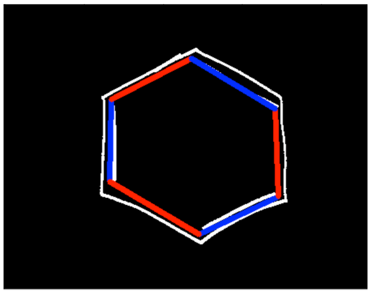


Fig. 12: Final result of bond classification for a benzene ring.

4 Results

4.1 Text Recognition Results

4.1.1 Scale-Invariant Template Matching

The results of text recognition are presented here. In order to optimize the tolerance of the scale-invariant template matching, we measured the precision and recall on the test set. The results are presented in Fig. 13. We chose the tolerance that maximizes the F1 score, 0.77.

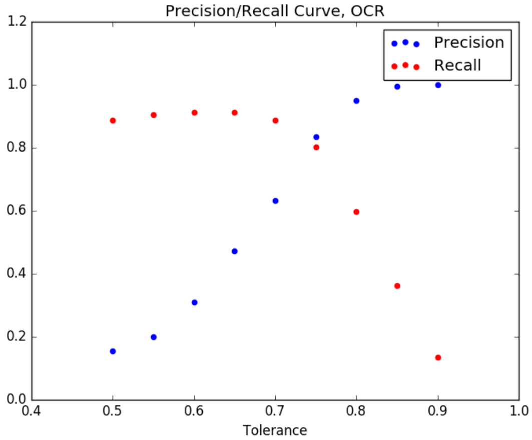


Fig. 13: OCR precision and recall on test set. The optimal value was found to be 0.77.

We then apply the template matching to the test set. By molecule and in total, the results are presented below in Table 1. Accuracy is by molecule image, so a molecule has to have all of its text completely recognized with no false positives for it to count positively towards the accuracy metric. Diagrams of the molecules associated with each molecule ID can be found in the Appendix.

Molecule ID	**Precision**	**Recall**	**Accuracy**
1	1.0	1.0	1.0
2	1.0	1.0	1.0
3	0.54	1.0	0.50
4	1.0	1.0	1.0
5	0.95	0.95	0.90
6	1.0	1.0	1.0
7	0.96	0.79	0.40
8	0.79	0.65	0.53
9	0.98	0.90	0.58
Total	**0.91**	**0.92**	**0.77**

Table 1: Results of scale-invariant template matching on test set.

While an accuracy of 77% is far from ideal, it is surprisingly effective considering we only used 5 training images to build the templates. With more examples, this method could perform even better in future work. We use the images where text was accurately identified from this stage in the further stages of the pipeline.

4.1.2 Supervised Classifiers

	Parameters	**Train Set**	**Cross-Validation**	**#Iter-ations**	**Avg. Acc.**
Logistic Regression with HOG features	Regularization coeff. = 1.0, L2 norm	1330 examples	10-fold	100	0.97
Linear Support Vector Machine with HOG features	Regularization coeff. = 1.0, L2 norm	1330 examples	10-fold	100	0.96
One-Layer Neural Network with HOG features	One hidden layer with 30 nodes	1330 examples	10-fold	100	0.99

Table 2: Results of supervised classifiers on the OCR training set.

The performance of the supervised learning classifiers using HOG features with 64 bins was evaluated using cross-validation, training a classifier with 90% of the 1100 negative test images and 60 positive test images per character, and testing on the remaining 10%. We conclude our training set is not large enough to provide accurate detections on the test set, recording less than 1% overall accuracy. This is because although the performance of the supervised learning classifiers on the cross-validation set is relatively good, perfect matching on the test set requires a correct match on each of the 1000+ sliding windows used for detection, so even the neural network, with 99% accuracy on the cross-validation set, is unable to perform well in the natural setting, with nearly 0% accuracy and several false positives.

For future work we would like to expand the size of the training set to improve the accuracy; but for now we use template matching.

4.2 Corner Detection Results

As shown in Table 3, we conclude that the Harris corner detector outperforms the baseline method of the MLOCSR polygon reconstruction method quite significantly, by approximately 15% on the molecule level (node level refers to the number of correctly detected nodes over the total number of nodes, molecule level refers to the number of correctly detected molecules with no false positives divided by the total number of molecules. There are several reasons that this result is the case. First, the polygon reconstruction method performs very poorly in the case of dashed bonds; whereas the Gaussian smoothing applied to the image before applying Harris corner detection “blends” dashed bonds into an edge before finding the corners. This kind of preprocessing is not feasible for the polygon reconstruction method since it relies on the narrow opposite contours that form the edges of the thick lines. The performance on a dashed molecule is demonstrated in Fig. []. While neither method performs particularly well on dashed bonds (polygon method performs at 15% on these dashed bonds, while Harris method performs at 45% on dashed bonds), which when blended are very wide, making corner detection difficult especially when they a dashed bond is near other corners.

	Molecule accuracy	**Overall Precision**	**Overall Recall**
Polygon Method	0.748	0.960	0.970
Harris Method	0.896	0.987	0.989

Table 3: Comparison of corner detection methods.

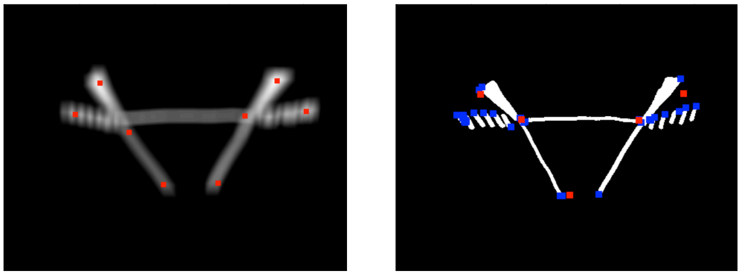


Fig. 14

Fig. 14: Harris corner detection on a molecule with dashed bonds (left) and polygon reconstruction method with initial corners in blue and clustered corners in red. (right). A wide Gaussian filter helps "blend" dashed bonds together. Since each dash of the dashed bond is a contour, many spurious corners in blue are detected with the polygon reconstruction method and make the agglomerative clustering inaccurate.

As we hypothesized, the Harris method also outperforms the polygon reconstruction method when bonds are not perfectly straight. This was particularly evident in the benzene rings, which the Harris corner detector (95% accuracy on benzene rings) was able to substantially outperform the polygon reconstruction method (50% accuracy on benzene rings). This effect is shown in Fig. 15.

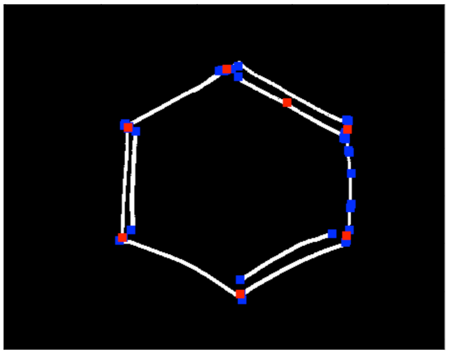


Fig. 15: The polygon reconstruction method detects several incorrect corners due to curved lines, as shown in blue. The Harris method does not suffer from this drawback.

4.3 Bond Detection Results

The bond detection is the worst performing stage of the pipeline, but fortunately it is the most correctable by a higher-level Markov model as described in MLOCSR. Still, large improvements can still be made to the algorithm by applying more heuristics about how the various atoms bond. We did not take chemical knowledge about the topological structure of molecules into account when looking for bonds, but there exist further constraints that can reduce our false positive rate, which would let us adjust the tolerance thresholds on the Hough detector and the angle acceptance to reduce the false negative rate. The results are presented below in Table 4 and errors are characterized further in Figs. 16 and 17.

Molecule ID	**Molecule Accuracy**	**Overall Precision**	**Overall Recall**
1	1.0	1.0	1.0
2	0.31	0.94	0.95
3	0.0	1.0	0.78
4	0.26	1.0	0.79
5	0.70	1.0	0.88
6	0.82	1.0	0.94
7	0.22	0.86	0.97
8	0.70	1.0	0.90
9	0.10	0.90	0.83
Total	**0.55**	**0.96**	**0.91**

Table 4: Bond detection results.

Typical errors included a missing bond, as shown in Fig. 16, and false triangular closures with bonds at very wide angles (bonds that are nearly, but not quite collinear), as shown in Fig. 17.

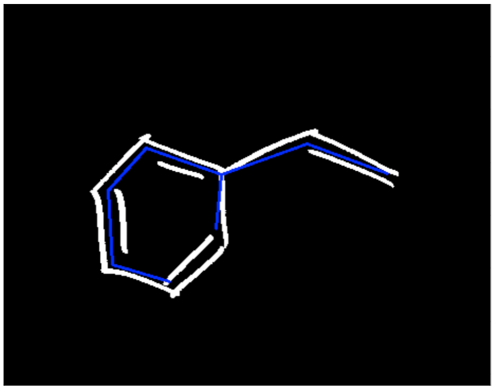


Fig. 16: A typical bond detection error, a missing bond, likely due to slight inaccuracy of corner detection. These "missing bonds" can be detected in later stages as long as most of the structure is correct.

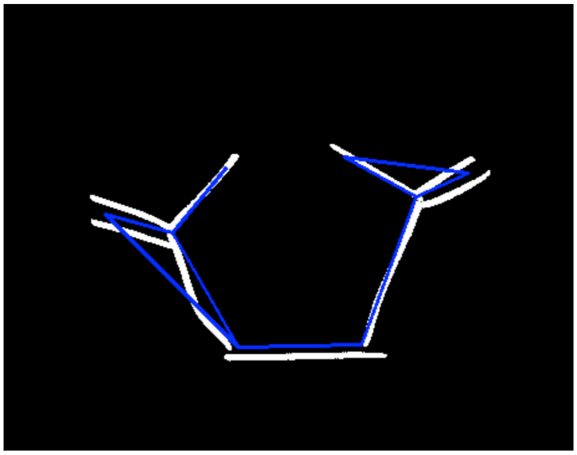


Fig. 17: Another common bond detection error, triangular closure of wide bonds, due to there being too much contamination in the bounding box. These can also be corrected later if most of the molecule is correct.

4.4 Bond Classification Results

4.4.1 Comparison of Classifiers

We split our training bonds randomly according to a 90-10 split and run cross-validation 10 times.

Classifier	**Cross-Validation**	**Cross-Validation Accuracy**
Logistic Regression	10-fold	0.85
Linear Support Vector Machine	10-fold	0.97
Decision Tree	10-fold	0.88

Table 5: Cross-Validation Results on a 90-10 training set split of known bond labels.

4.4.2 Performance on Test Set

Based on the results on the cross-validation set, we use the SVM for classification on the full set of molecules. We find that there is no great disparity in confusing one type for another; despite the only 5 training examples of triple bonds, we find that double bonds are no more often mistaken for single bonds as triple bonds, for example.

Molecule ID	**Accuracy By Bond**	**Accuracy By Molecule**
1	0.98	0.90
2	0.97	0.81
3	0.57	0.0
4	0.80	0.30
5	1.0	1.0
6	0.83	0.50
7	1.0	1.0
8	0.98	0.93
9	1.0	1.0
Total	0.94	0.75

Table 6: Test results by molecule using an SVM classifier.

4.5 Overall Results

When the overall pipeline is run on the entire set of molecules, 94 out of the original 360 molecules are correctly recognized in their entirety. While this accuracy may seem low, it is still higher than the performance of the "out of the box" existing optical structure recognition algorithms, the most well-known being OSRA, which when used on handwritten data have nearly 0% accuracy. We also find that even the *approach* of MLOCSR applied to the data, which relies on the Douglas-Peucker polygon fitting algorithm, does not even detect C- and T- points as successfully as our algorithm on our hand-written dataset. We also find that our supervised learning bond classification algorithm performs extremely well given the very small training data set, which was extracted from only 5 images of each molecule. We are optimistic that with more training data we will be able to obtain nearly 100% accuracy with this method in the future.

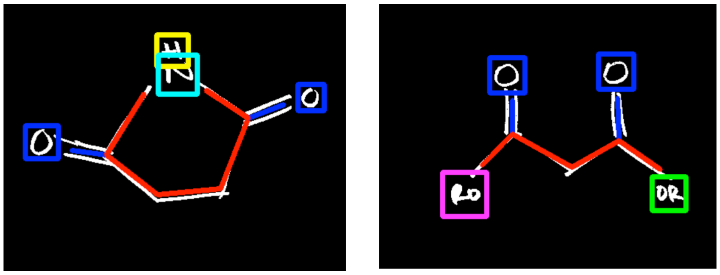


Fig. 18: Two examples of correctly recognized molecules after completion of the full pipeline. These can be easily converted to a standard chemical data format.

5 Conclusion

Although our overall accuracy is low, we believe that the work presented in this paper will lay the foundation for hand-drawn structure recognition in the future.

Much of the low accuracy can simply be attributed to a lack of training data. State-of-the-art OCR methods, for example, would boost the accuracy of text recognition from 77% to near perfect. We also believe that more training data will ultimately allow us to use a convolutional neural network for the bond classification stage rather than an SVM, and more data will be able to significantly improve the accuracy of bond classification as well.

Additionally, as mentioned previously, the focus of this project was on the low-level recognition of atoms and bonds; or the nodes and vertices that make up the overall graph. There are additional heuristics that can be applied in higher-level modules, such as bonding patterns like valence rules that we did not take into account, which will significantly improve the performance of bond detection.

Accuracy may also be a misleading metric for certain applications of hand-drawn structure recognition as well, in cases where more information is obtained. For example, in an electronic tablet drawing application, in a similar way to how Chinese and Japanese characters are recognized by OCR software, information about how the user is drawing the structure is also available. This can improve the localization of corners (using information about when the user picks up and puts down the pen) and identify bonds with much greater accuracy (based on speed of stroke, etc.). Additionally, if there is a limited subset of molecules that the engine is required to recognize, various molecule similarity algorithms can be used to compare the molecule against the database of possible molecules and return the one with the greatest similarity. This is often the case for simple molecules and could be very useful in chemistry education.

We conclude that handwritten structure recognition and analysis is a difficult problem, one that cannot be treated in the same way that computer-generated structure recognition is treated. More flexibility must be applied in accounting for the greater degree of variability in hand-drawn images, and we have accounted for that in this work with modern corner and line detection techniques. The key insight of this project was analyzing small cross-sections of bonds so the algorithm can gain a consensus from many cross-sections instead of trying to analyze bonds as a whole, as previous algorithms have done. Overall, there are many parts of this pipeline that can be improved as mentioned, but much progress has been made towards being able to apply these methods to a public application.

6 References

[1] Gaulton, A.; Overington, J. P. *Role of open chemical data in aiding drug discovery and design*. Future Med. Chem. 2010, 2, 903–7.

[2] Kind, T.; Scholz, M.; Fiehn, O. *How large is the metabolome? A critical analysis of data exchange practices in chemistry*. PLoS One 2009, 4, e5440.

[3] G.R. Rosania, G. Crippen, P. Woolf, D. States, and K. Shedden, R. *A cheminformatic toolkit for mining biomedical knowledge*. Pharmaceutical Research, vol. 24, (no. 10), pp. 1791-1802, Oct 2007.

[4] Casey, R. et. al. *Optical Recognition of Chemical Graphics*. Document Analysis and Recognition: Proceedings of the 2nd International Conference on Document Analysis and Recognition. 1993.

[5] Ibison, P. et. al. *Chemical Literature Data Extraction: The CLiDE project*. Journal of Chemical Informatics and Computer Science, 33, pp. 338-344. 1993.

[6] Park, J. et. al. *Image-to-Structure Task by ChemReader*. Text Retrieval Conference, 2011.

[7] Filippov, I. and Marc Nicklaus. *Optical Structure Recognition Software to Recover Chemical Information: OSRA, An Open Source Solution*. J. Chem. Inf. Model., 49 (3), pp. 740-743, 2009.

[8] Frasconi, P. et. al. *Markov Logic Networks for Optical Chemical Structure Recognition*. Journal of Chemical Information and Modeling. 54, pp. 2380-2390, 2014.

[9] Douglas, D.; Peucker, T. *Algorithms for the reduction of the number of points required for represent a digitized line or its caricature*. Can. Cartogr. 1973, 10, 112–122.

[10] Navneet Dalal, Bill Triggs. *Histograms of Oriented Gradients for Human Detection*. Cordelia Schmid and Stefano Soatto and Carlo Tomasi. International Conference on Computer Vision & Pattern Recognition (CVPR '05), Jun 2005, San Diego, United States. IEEE Computer Society, 1, pp.886–893, 2005.

[11] T.E. de Campos, B.R. Babu and M. Varma. *Character Recognition in natural images.* In *Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal,* February 2009.

[12] Harris, C. and M. Stephens, *A Combined Corner and Edge Detector.* Plessey Research, 1988.

7 Code Access

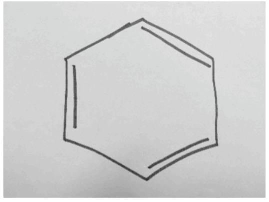
The code is available open-source. The repository is located at https://github.com/bradleyemi/chemtype2. Instructions for downloading data and usage are located on GitHub.

8 Acknowledgements

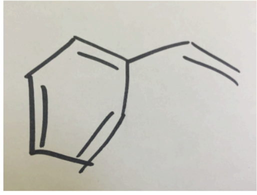
This project was implemented in Python 2.7 with additional use of the Anaconda distribution, Scikit-Learn (for machine learning classifiers), and OpenCV for CS 231A at Stanford University.

A Appendix: Molecule Table

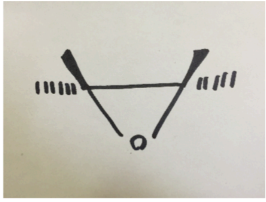
1



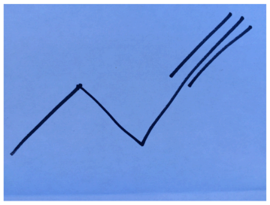
2



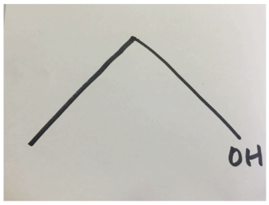
3



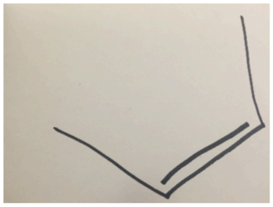
4



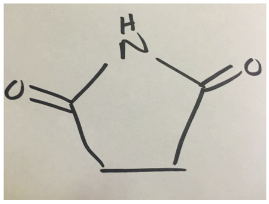
5



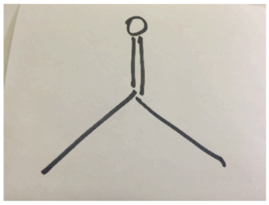
6



7



8



9