Data Augmentation of Engineering Drawings for Data-driven Component Segmentation

Carnegie Mellon University

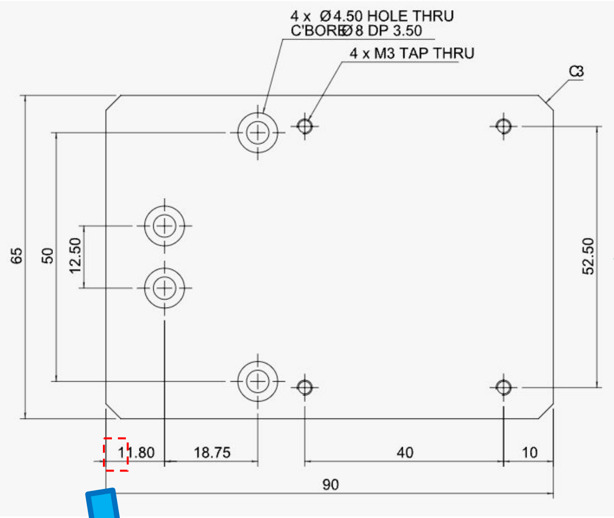
MiSUMi

Wentai Zhang, Quan Chen, Can Koz, Louise Xie, Amit Regmi, Soji Yamakawa, Tomotake Furuhata, Kenji Shimada, Levent Burak Kara

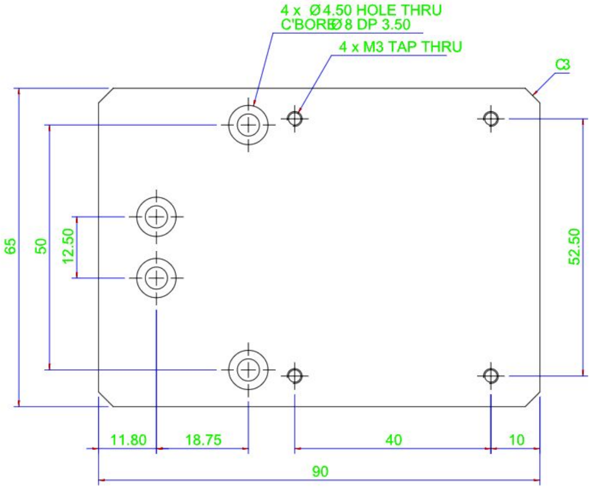
Carnegie Mellon University

Overview

Ultimate goal: Data-driven component segmentation of raster drawings



An automatic system

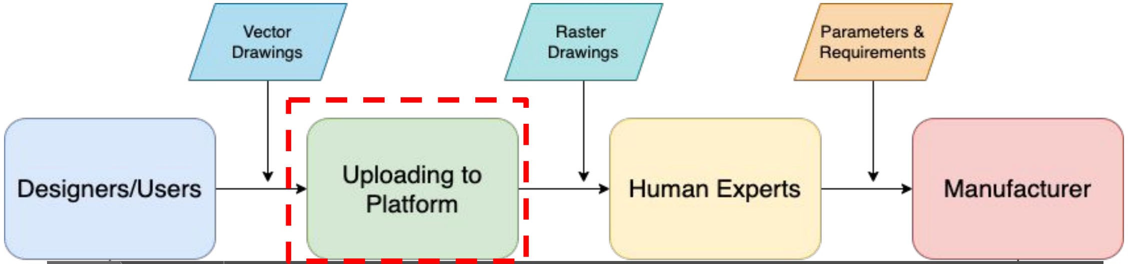


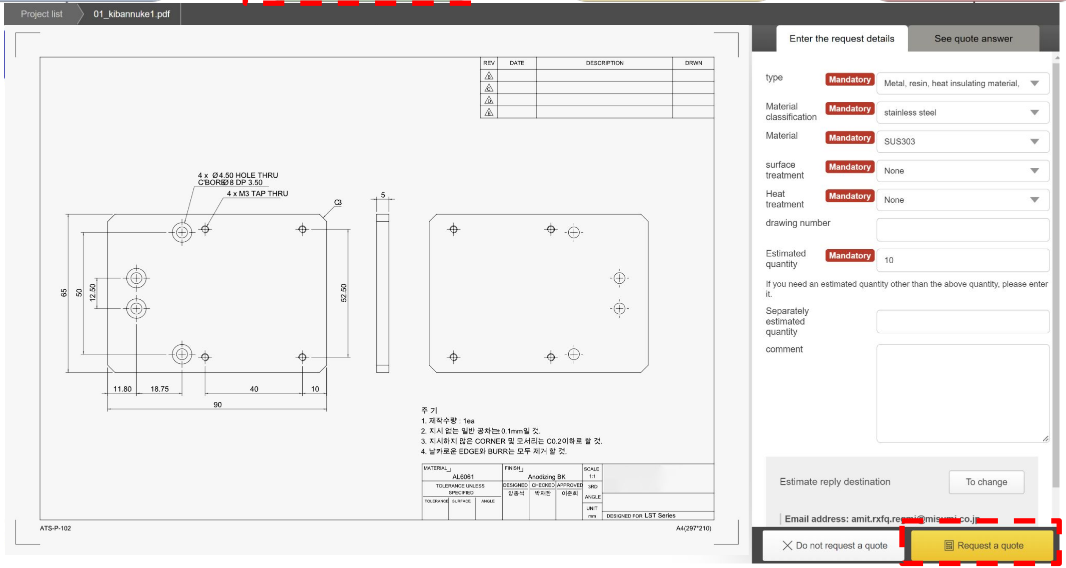
B&W raster drawing

Vectorized segmentation among contours, dimensions and texts

```
Line 0, x1, y1, x2, y2, contour
Line 1, x1, y1, x2, y2, dimension
Circle 0, x1, y1, r, contour
```

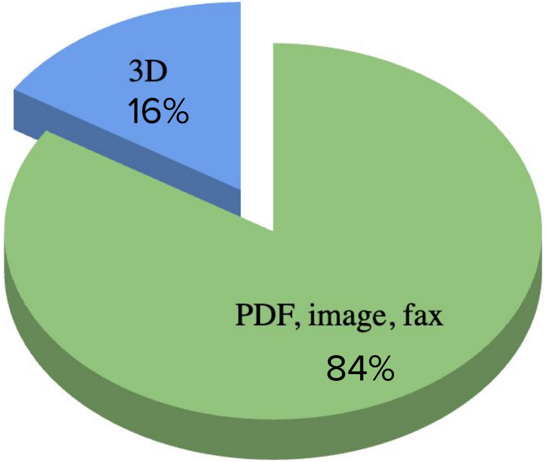
Motivation: Industrial Part Quotation Systems





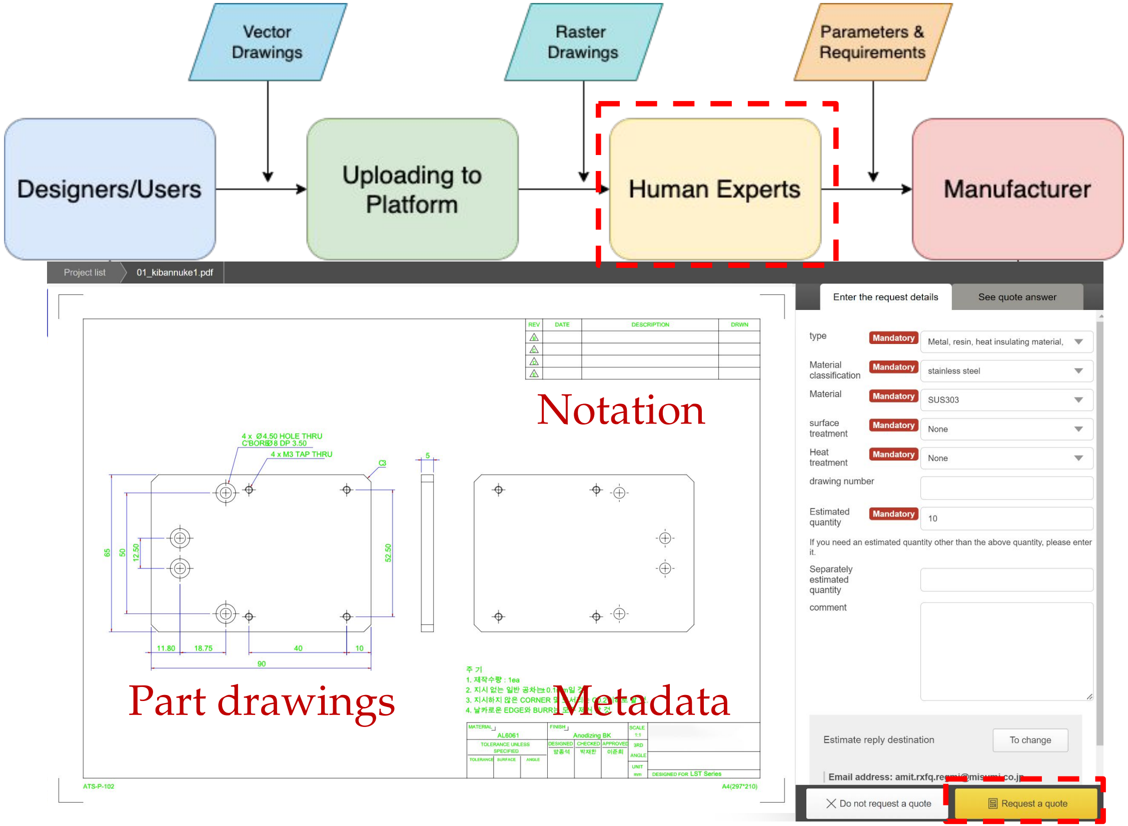
Average time for a part quote: 7 days

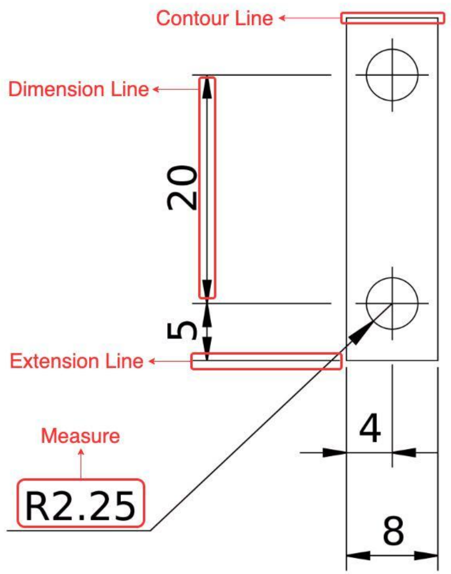
Drawing Format When an order is placed



A survey on the project and issues in Japan's manufacturing industry, 2017

Motivation: Industrial Part Quotation Systems





This step is time-consuming and dull.
We aim at building an automatic system to aid the inspector.

Literature Review: Engineering Drawing Analysis

Prior works mainly focus on pattern recognition, shape identification and drawing retrieval.

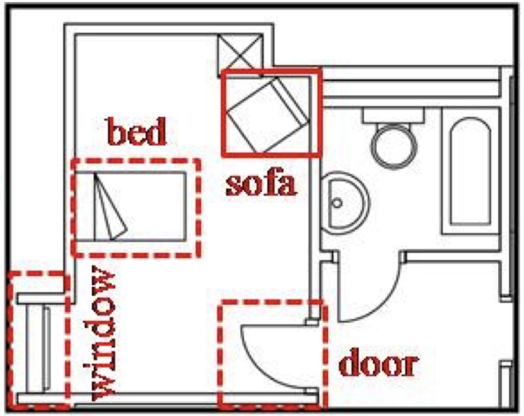
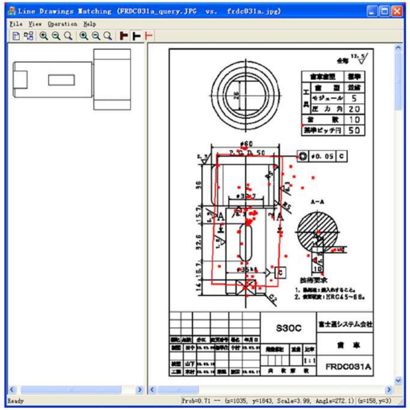
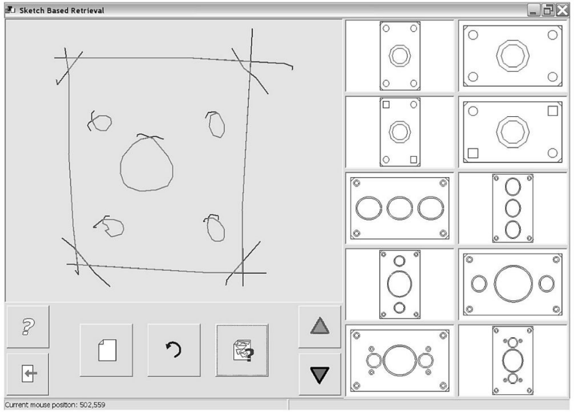


Diagram recognition in floor plans, flow charts and electric circuit diagrams and vibratory mechanical systems [Delalandre et. al, 2010, Kara et. al. 2008, Schafer et. al. 2021]



Shape identification for part drawings with sampled points, histograms or shape descriptors. [Liu et. al. 2009, Huet et. al. 2001]

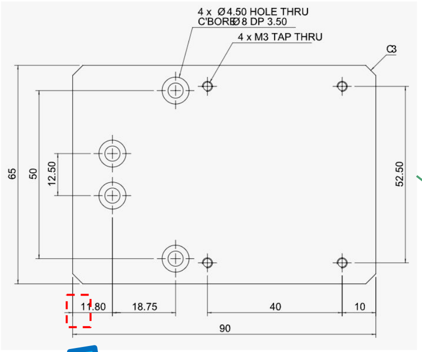


Drawing retrieval or matching using lines, pixel blocks or patches. [Mednonogov et. al. 2000, Jiao et. al. 2009, Sousa et. al. 2010]

Only achieve a partial interpretation of the drawings for a specific scenario. We aim at developing a data-driven system to analyze all the components for general analysis.

Overview

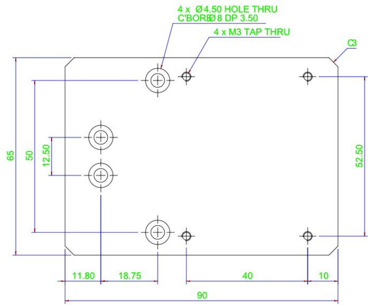
Ultimate goal: Data-driven component segmentation of raster drawings



B&W raster drawing

An automatic system with:

1. A vectorization preprocessing
2. A synthesis method to automatically construct a large labelled dataset
3. A data-driven model that predicts the type of each vectorized component



Vectorized segmentation

```
Line 0, x1, y1, x2, y2, contour
Line 1, x1, y1, x2, y2, dimension
Circle 0, x1, y1, r, contour
```

Requirements

A data synthesis method that:

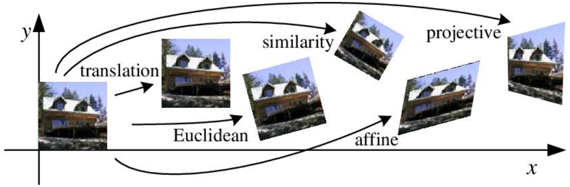
(1) Utilize the information stored in existing labelled examples

(2) Generate an arbitrarily large set of synthetic drawings to train a data-driven model for binary component segmentation (contour shape/dimension set)

(3) The generated drawings are subjected to validity of technical rules

Related work

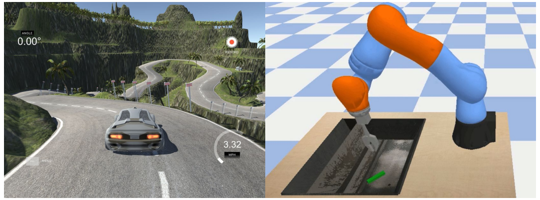
General Data Augmentation Methods in CV



Basic geometric transformation, filtering, random erasing and mixing. [Kang et. al. 2017, Zhong et. al. 2020, Chatfield et. al. 2014, Inoue 2018]

Most of these manipulations will result in invalid image data in the context of engineering drawings

Simulated Data Augmentation



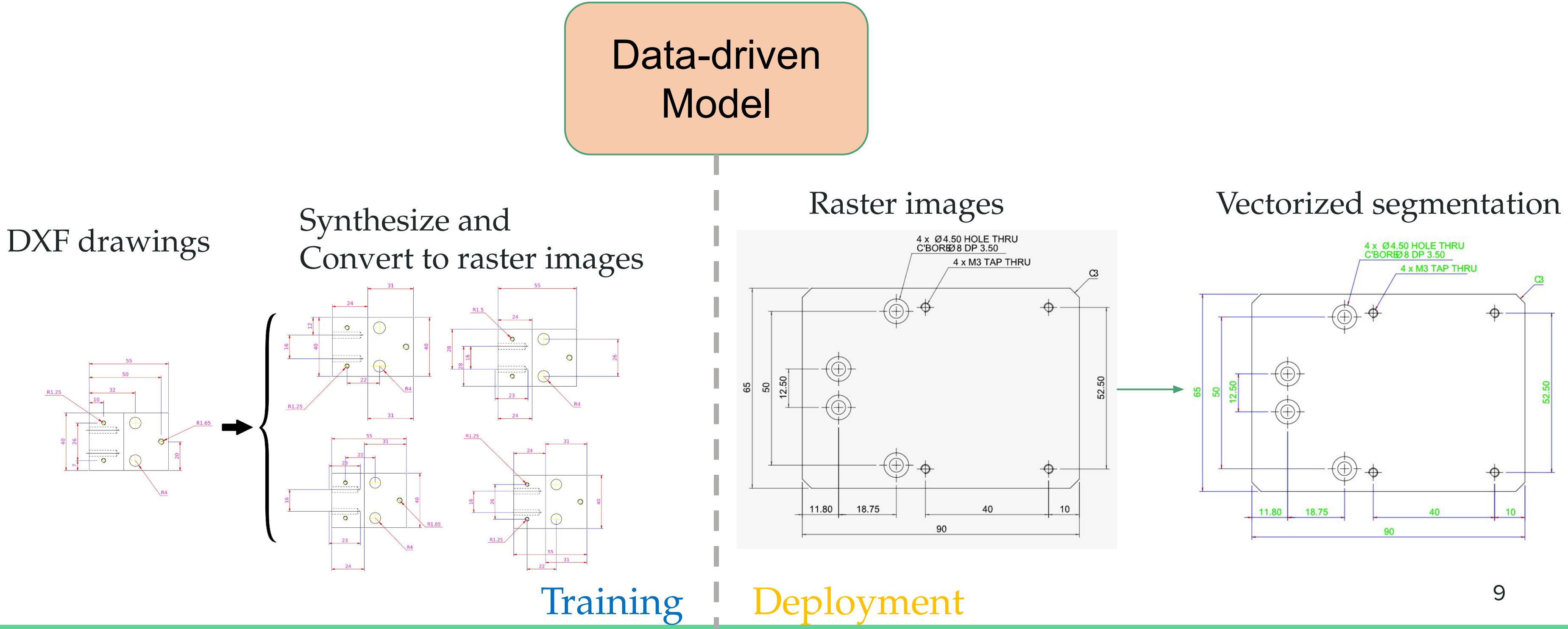
Carla[Alexey et. al. 2017],
Udacity[molyakov et. al. 2018],
Kuka[Lukaˇc et. al. 2018]

The simulated data can be generated with flexible experiment conditions in a reasonably short time

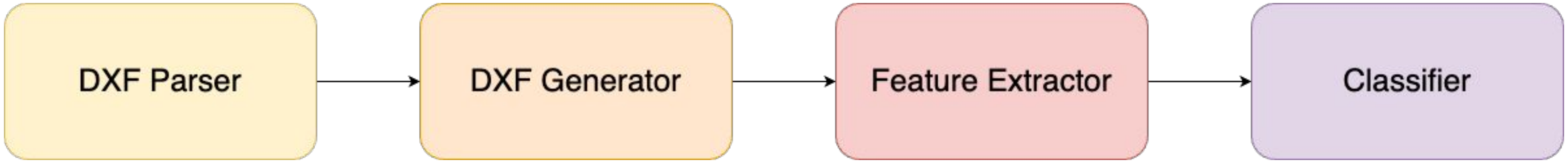
We aim to create a parametric drawing generator that can synthesize a pool of new drawings in a simulated manner with a handful of existing drawing examples

Algorithm Overview

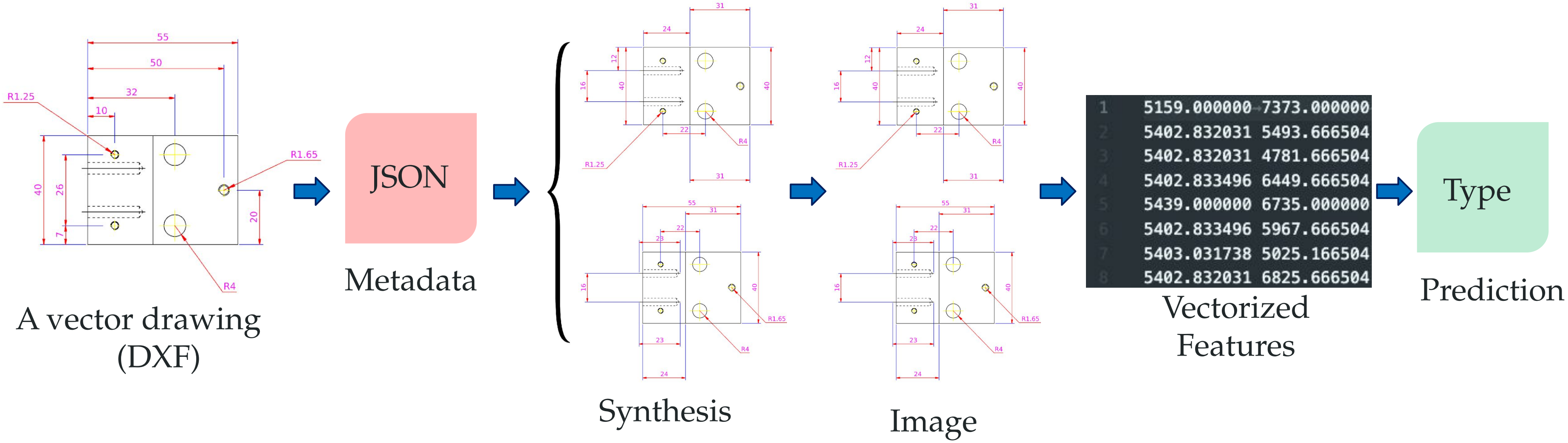
The labeling for such drawings requires humans with technical training. But vector drawings can serve to create a dataset for training a model that deploys on raster images.



Approach Overview

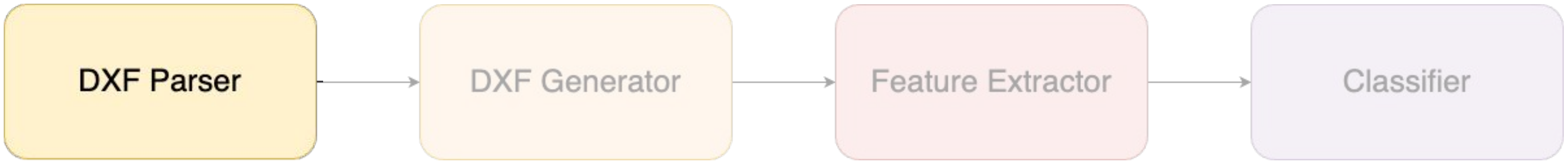


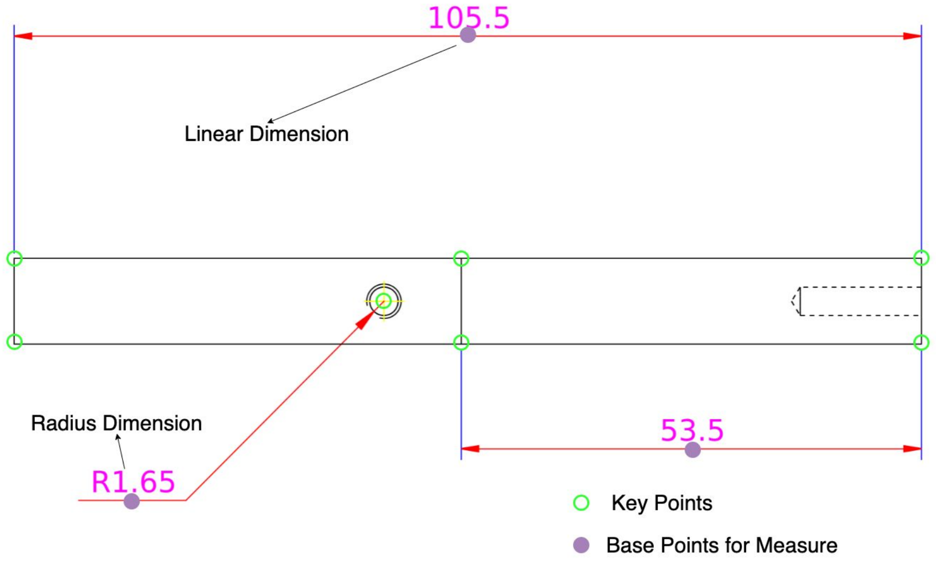
Four major modules to parse existing data, generate new data, vectorize the drawing, and predict the component type



(To prepare training raster drawings)

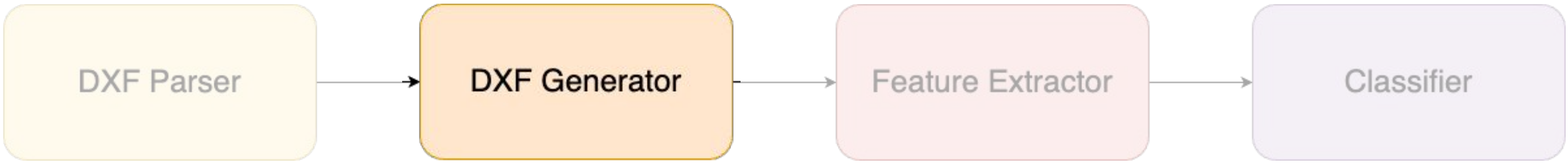
Our Algorithm Pipeline: Parser

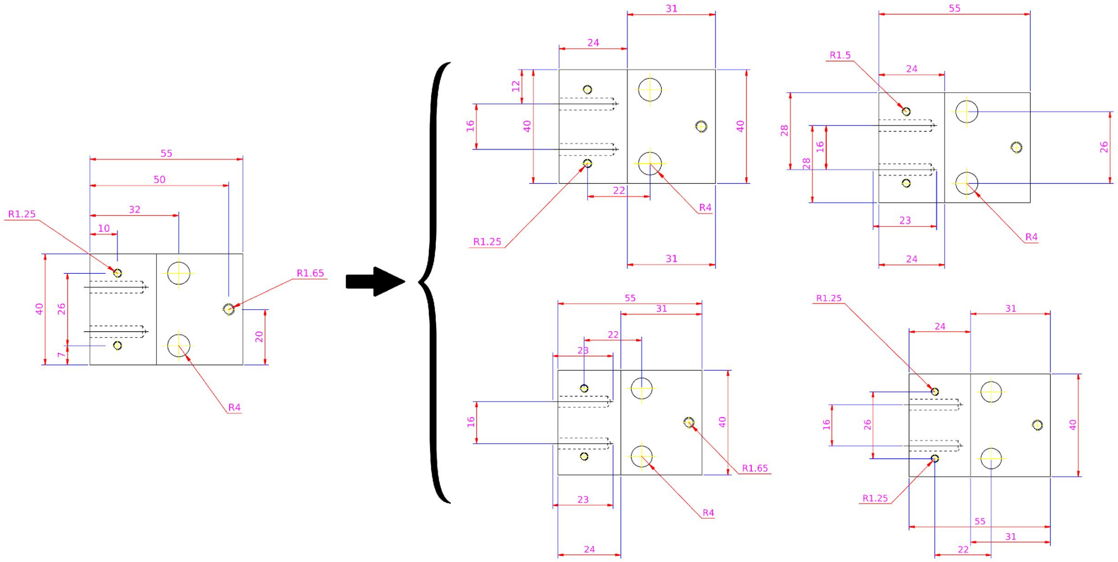




- Parse the information from a given DXF drawing
- Record each component based on base points and key points
- Save it as a JSON file

Our Algorithm Pipeline: Generator





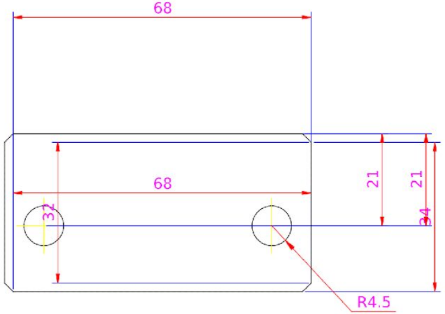
- Separate the dimensional elements from the object contour lines
- Expand the drawing set by generating new drawings wherein new dimensional elements are generated and placed in novel configurations.

How to ensure the validity?

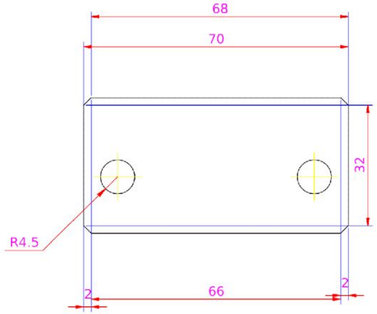
Dimension Sets Randomization

Two designed constraints:

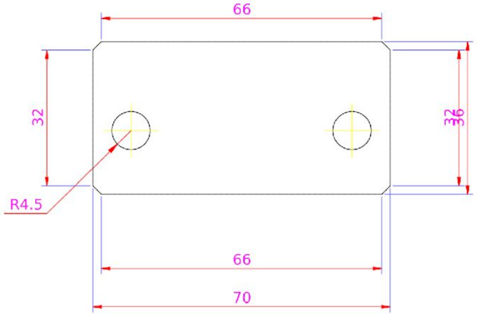
- **C1**: There should be no overlap between the generated dimension sets.
- **C2**: The dimensions should locate outside of the contour shape if possible.



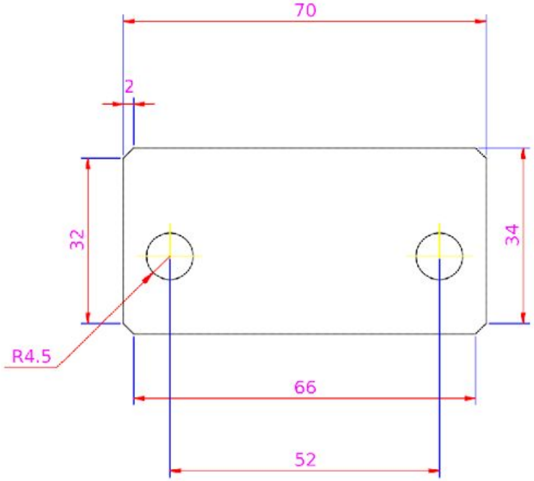
Unconstrained



C1 only



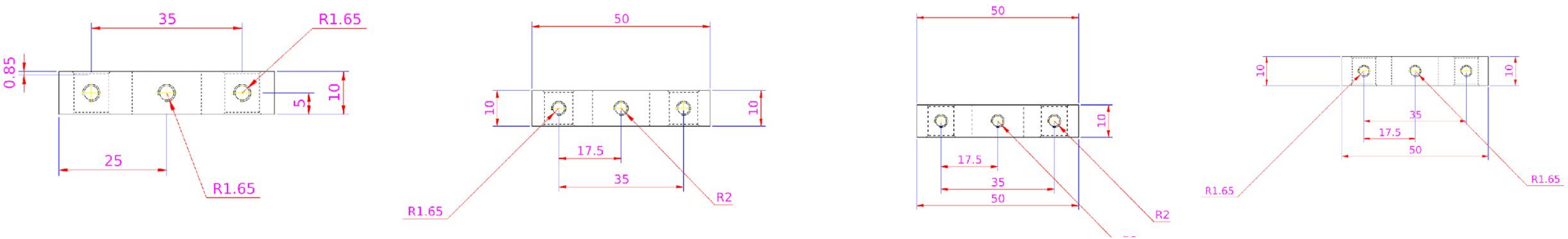
C2 only



Fully-constrained

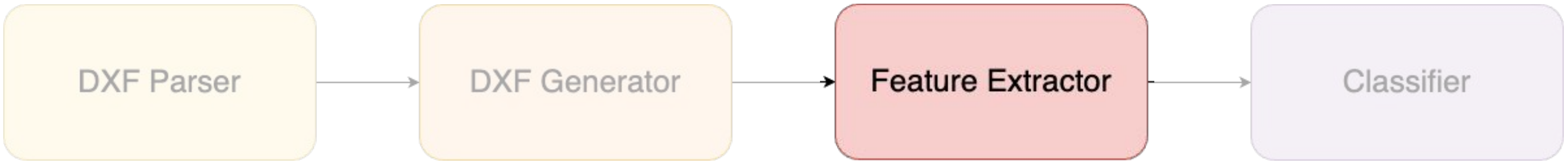
Dimension Set Randomization

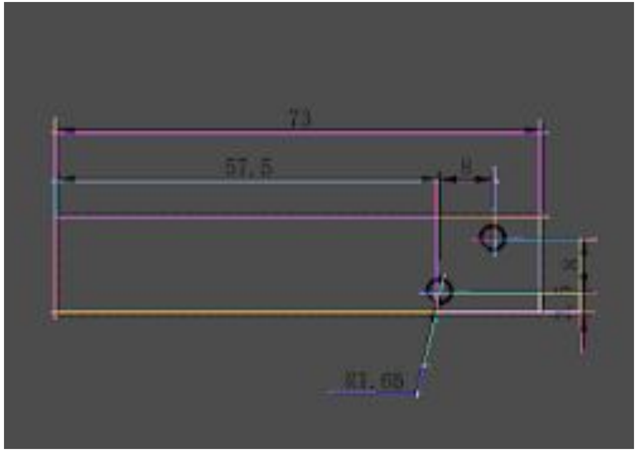
1. Parse the information from a previously saved JSON file.
2. Determine the number dimension sets to be generated (±20% from original drawing)
3. In an iterative manner:
    - choose a pair of key points
    - randomly generate a base point with random orientation
    - conflict check with all existing generated dimensions
    - conflict check with the bounding box of the contour shape.



The generated new vector drawings are converted to raster images as training data.

Our Algorithm Pipeline: Extractor





Line detection

Island detection

- Vectorized with a fine-tuned Hough line detector to find all the **straight lines**
- Extracted the non-line elements as **isolated islands** in the remaining pixel space

How to unify the input components?

Our Algorithm Pipeline: Extractor

Index	Symbol	Notion
1	X_1	x coordinate of the upper left corner points of the bounding box.
2	Y_1	y coordinate of the upper left corner points of the bounding box.
3	X_2	x coordinate of the lower right corner points of the bounding box.
4	Y_2	y coordinate of the lower right corner points of the bounding box.
5	L	Diagonal length of the bounding box of the component. The length is normalized by the diagonal length of the image.
6	r	Aspect ratio of the bounding box. length (x range)/height (y range) is used for consistency.
7	P_b	Percentage of black pixels within the bounding box.
8	P_{bp}	Percentage of black pixels in the projection of the components. The components are projected along the axis with smaller range.
9	D_a	Average distance of the 4 nearest neighboring components.
10	D_{std}	Standard deviation distance of the 4 nearest neighboring components.
11	COV	Coefficient of variation. The standard deviation of the distances from the black pixels in a component to its center of gravity. This feature is introduced to indicate the symmetry.
12	M_Z	Zernike Moments of the components. 8 degrees are utilized to generate 25 response features. These features are able to indicate the local gradient orientation of the components.

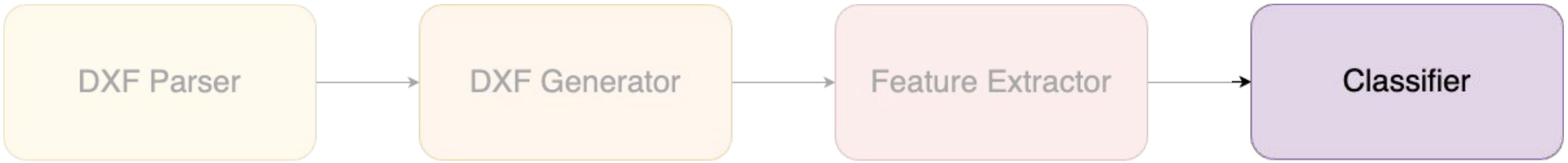
Inspired by [Yun, et. al. 2019, Ye et. al. 2016, Van et. al. 2016]

For each detected component (line/island), we design a series of features including:

- Basic geometric info (1-6)
- Density info (7,8)
- Contextual info (9,10)
- Symmetry (11)
- Local gradient (12)

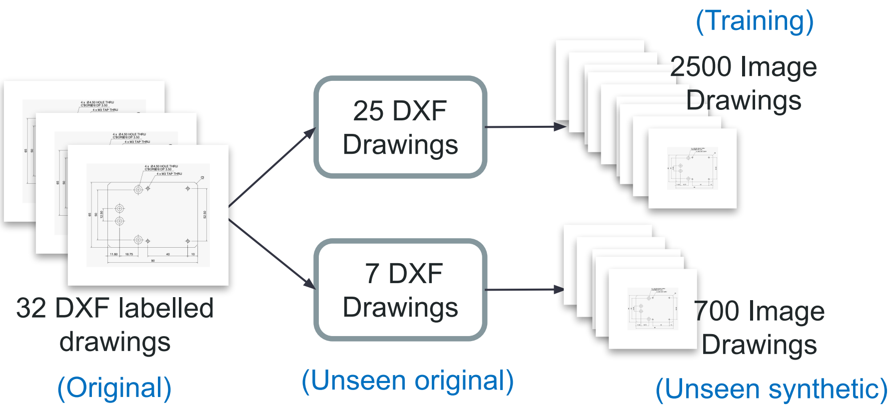
In the end, each vectorized component is converted to a 36 dimensional feature vector. The task becomes vector classification.

Our Algorithm Pipeline: Classifier



As a case study, we test our data augmentation method with 3 classifiers:

- **DT**: A decision tree model, max depth: 10, min split: 3, metric: Gini impurity.
- **RF**: A random forest model, 40 DT models above, No. of features: square root.
- **MLP**: A multi-layer perceptron model, 2 hidden layers with 100 nodes in each.



Task: Binary Component Segmentation (contour/dimension)
Training: 2500 synthetic drawings
Test set 1 (unseen original): 7 original drawings
Test set 2 (unseen synthetic): 700 synthetic drawings
Criterion: Accuracy of the predicted label from each model

Results

Validation Accuracy %	Multi-layer Perceptron	Decision Tree	Random Forrest
Unseen Synthetic	76.84	86.29	87.52
Unseen Original	74.72	82.71	83.78

Validation Accuracy %	Multi-layer Perceptron	Decision Tree	Random Forrest
No synthesis	45.74	56.50	58.19
Unconstrained	58.03	64.12	66.56
C1 only	70.65	81.31	82.12
C2 only	67.49	77.84	80.27
C1+C2	76.84	86.29	87.52

- The tree-based methods yield better results than the simple MLP model.
- The performance on the unseen synthetic dataset is better than on the unseen real dataset as expected

- A major improvement (like 87.52 vs 58.19 for RF) when our proposed synthesis method is introduced
- C1 and C2 contribute to a marked improvement by regularizing the random dimension sets with valid prior assumptions
- C1 results in a larger increase in accuracy compared to C2

Results

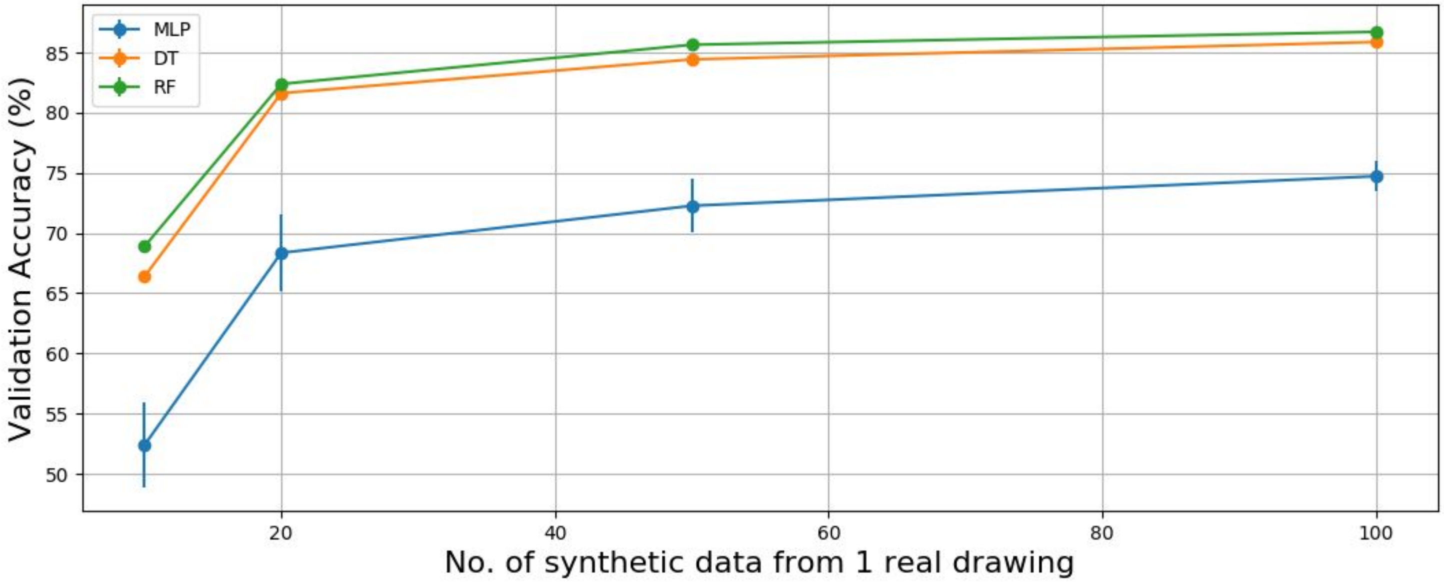
Validation Accuracy %	Multi-layer Perceptron	Decision Tree	Random Forrest
Unseen Synthetic	76.84	86.29	87.52
Unseen Original	74.72	82.71	83.78

Validation Accuracy %	Multi-layer Perceptron	Decision Tree	Random Forrest
No synthesis	45.74	56.50	58.19
Unconstrained	58.03	64.12	66.56
C1 only	70.65	81.31	82.12
C2 only	67.49	77.84	80.27
C1+C2	76.84	86.29	87.52

- The tree-based methods yield better results than the simple MLP model.
- The performance on the unseen synthetic dataset is better than on the unseen real dataset as expected

- A major improvement (like 87.52 vs 58.19 for RF) when our proposed synthesis method is introduced
- C1 and C2 contribute to a marked improvement by regularizing the random dimension sets with valid prior assumptions
- C1 results in a larger increase in accuracy compared to C2

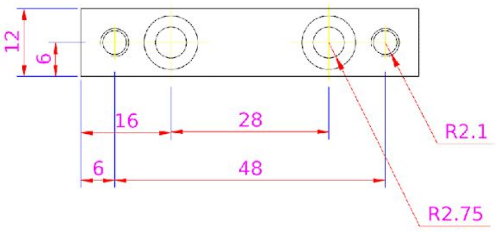
Results

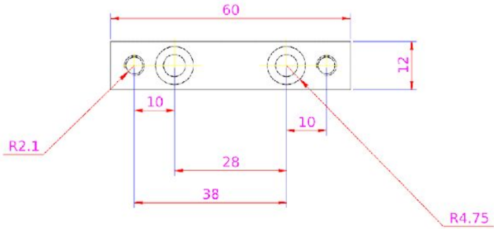


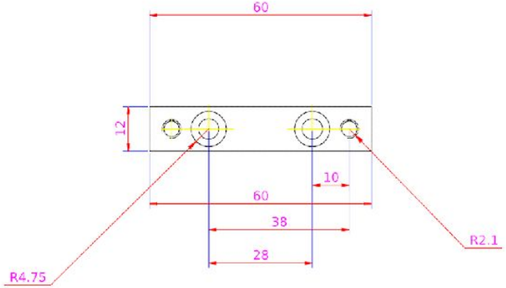
- A very similar trend in accuracy as the number of drawings increases.
- The rate of the increase in accuracy gradually levels out as more synthetic drawings are generated. Negligible improvement beyond 50.
- The standard deviation of tree-based methods is much less than MLP.

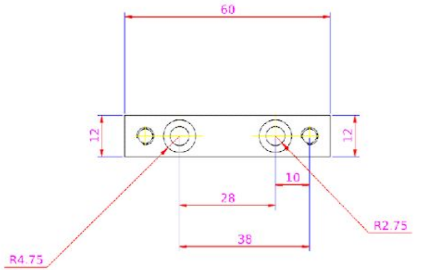
Takeaways

- A novel method to synthesize a large amount of engineering drawing images based on constrained dimension set randomization.
- Results show that the capacity of the trained model to generalize to unseen new geometries is considerably improved with only a handful of labelled examples.



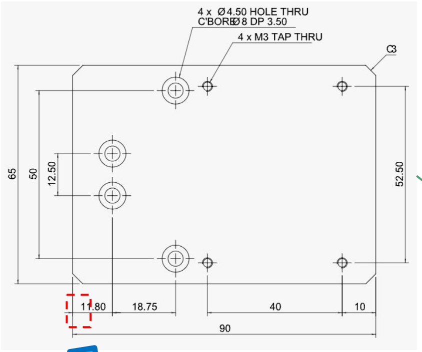






Future Work

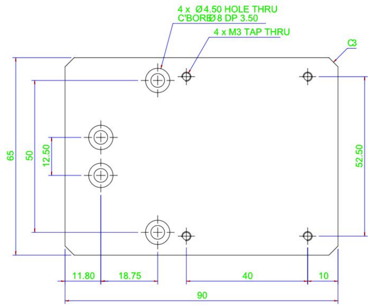
Ultimate goal: Data-driven component segmentation of raster drawings



B&W raster drawing

An automatic system with:

1. A vectorization preprocessing
2. A synthesis method to automatically construct a large labelled dataset
3. A data-driven model that predicts the type of each vectorized component

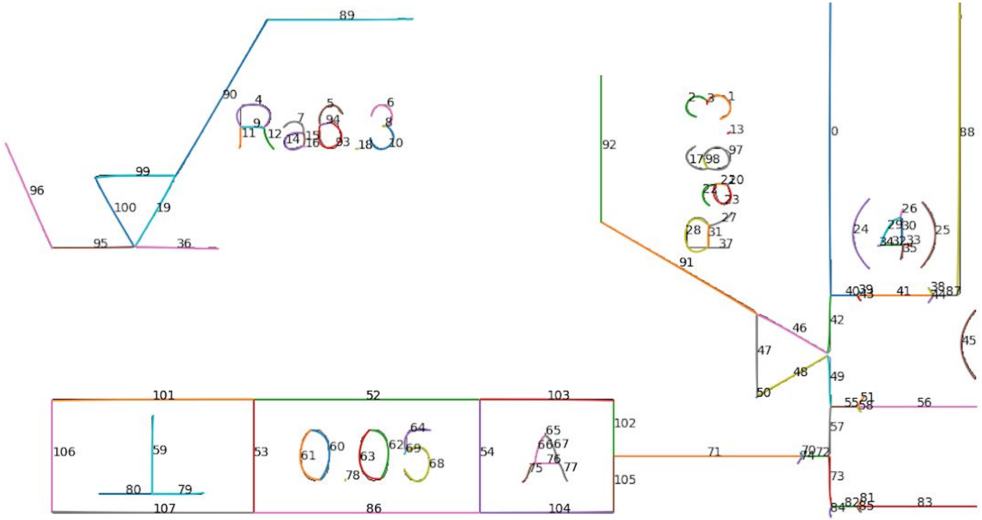


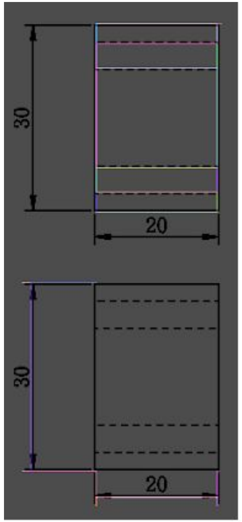
Vectorized segmentation

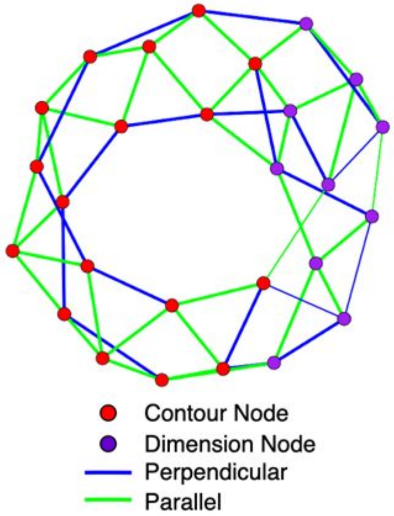
```
Line 0, x1, y1, x2, y2, contour
Line 1, x1, y1, x2, y2, dimension
Circle 0, x1, y1, r, contour
```

Currently Exploring

- Hierarchical line/curve fitting for vectorization

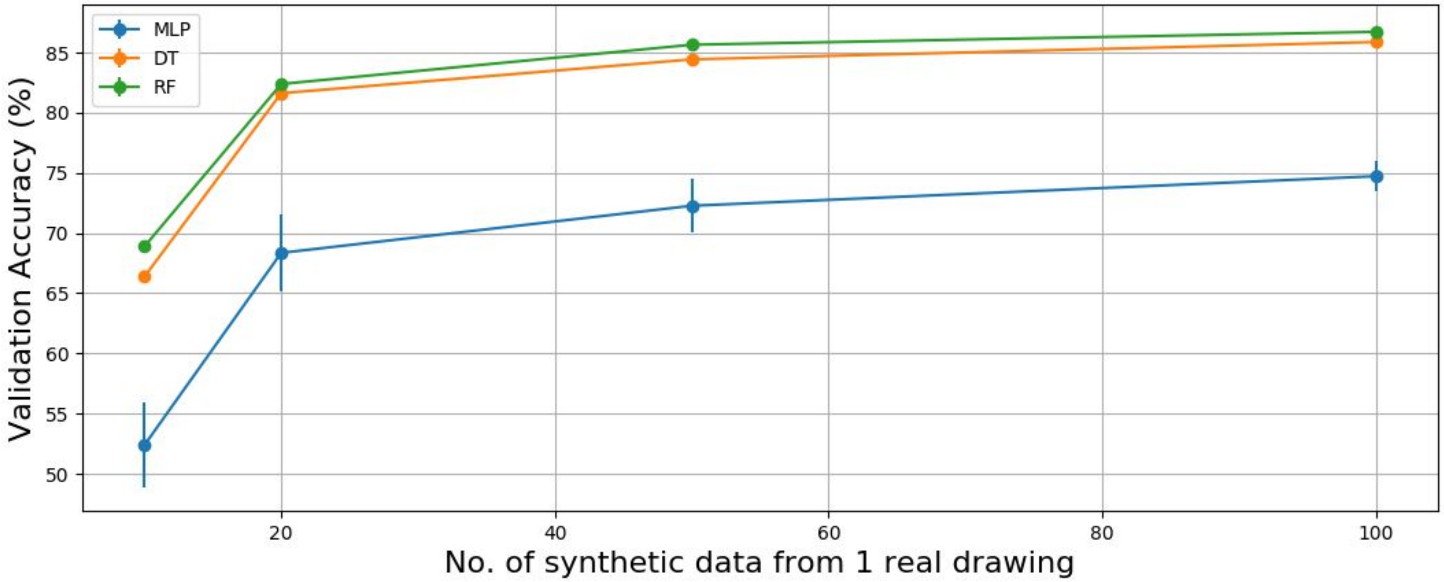






- Represent the vectorized results with component graphs based on connectivity. The task is converted to a graph nodal labelling problem.

Updated results



- Our preliminary model with hierarchical vectorizations and graph neural networks:

Models	Validation Accuracy
Best RF	87.52%
GraphSAGE+Vector Graph	90.90%

Acknowledgement

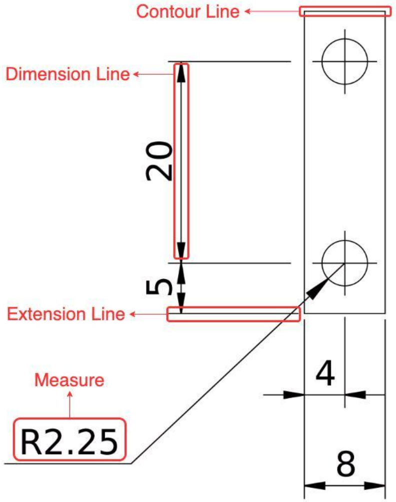
Thank you!

I would like to thank Quan Chen, Can Koz, Joe Joseph, Louise Xie, Yao Lu, Zheren Zhu, Zhuoran Cheng, Sam Yin and Run Wang for their support in the brainstorming, discussion and experiments.

I would like to thank MiSUMi Corporation for their provision of a contemporary engineering problem, guidance on the applicability of developed methods, and financial support.

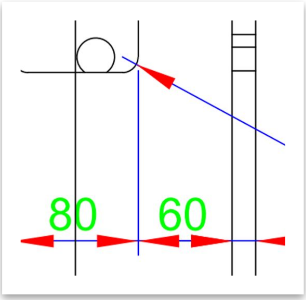
Major Challenges

Data Preparation

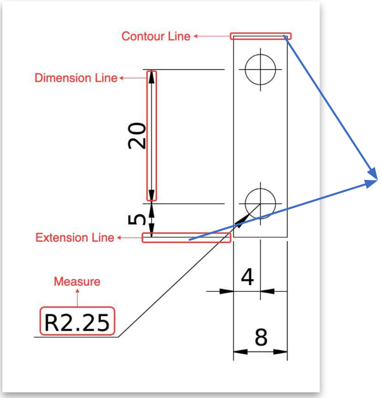


A lack of labelled data for such segmentations. Time-consuming and costly.

Feature Extraction



Unlike natural images, engineering drawings are extremely sparse. Only black and white pixels.



Both appear as a solid straight line

The local features in the pixel level cannot guarantee enough evidence for predicting the component type