ISSN: 0974-5823 Vol. 7 No. 2 February, 2022

International Journal of Mechanical Engineering

Machine Learning in Identification of Polygon Shapes for Recognition of Mechanical Engineering Drawings

Arundeep Murugan[1], Feleke Worku Tadesse[2], Esubalew Nega Wondirad[3], Mesfin Tura Bizuneh[4] and Abebe Firew Guadie[5]

Lecturer, School of Mechanical and Industrial Engineering, Institute of Technology,

Debre Markos University, Debre Markos, Ethiopia - 269

Abstract

In this research work, an attempt is made, how machine learning may be used to recognise 2D schematics for machine parts. As part of this technique, the recognition of simple geometric objects like polygons is essential. Polygons with random forms and segmented edges may be identified using machine learning methods. Uncertainty caused by segmented edges makes it difficult to determine how many sides a given object has. Various datasets with varied degrees of uncertainty are utilised in this study. When attempting to recognise forms, a variety of characteristics are used. These include point coordinates and slopes of lines, as well as geometrical metrics like area, perimeter, and centroid. It is decided to use the Random Forest Classifier, K-Nearest Neighbor Classifier, and Support Vector Classifier models for classification. These models are examined in terms of their ability to identify polygons.

Keywords—Deep Learning, Classifier, Machine learning, Pattern Recognition, Polygon.

1. INTRODUCTION

In all engineering specialties, sketching is a universal means of communication. This information is vital to ensuring the quality of produced items in the mechanical engineering sector, and is described in a set of manufacturing drawings. Existing goods often have hardcopy or 2D CAD drawings accessible. Modern manufacturing techniques need the creation of 3D CAD models from these designs. Drawing Reconciling 2D drawings of machine components with their 3D models may be done by identifying distinct parts in a drawing. Attempts to automate the process of drawing recognition have been made using various methodologies for more than two decades. Machine learning algorithms for drawing recognition have been more popular in recent years.

Machine learning is one of the hottest topics right now, and it's affecting a wide range of fields, particularly those that deal with private information. Its goal is to create and train models that improve with time and experience automatically and continually, allowing them to do their jobs more accurately. Machine learning may help us make crucial judgments and predictions without any human interaction thanks to the expanding capabilities of networked and mobile computing systems to amass large amounts of data sets. Fundamental scientific and technical concerns and the actual computer software they've developed, which has been deployed in several applications, both need investigation into how machine learning works. Computer vision, language translation, voice recognition and automated speech controller, robot control, pattern recognition, etc. are some of the applications of machine learning.

Automated systems and the requirement to do specific tasks without human intervention have led to the widespread application of machine learning in the mechanical engineering field, as shown below. As an example, in I the diagnosis of a reciprocating diesel engine, the most appropriate case is selected after calculating the similarity between the given case and the previous cases in the database, (ii) the prediction of flank tool wear in high speed machines is carried out using Random forests and statistical data created from parameters like cutting force, vibration and acoustic emissions collected from milling tests, (iii) Unsupervised clustering was used to characterise materials, and the research took into account the materials' thermal, electrical, physical, mechanical, and chemical compositions.

The use of machine learning algorithms such as Random Forests, Support Vector Machines, Logistic Regression, and Nave Bayes in the identification of acceptable design methodologies has been researched in design engineering. The model picks up new skills by studying examples of other designers' work in the database and applying what it learns to its own work. [4] discusses the importance of machine learning in the design of machines.

Recognition of Engineering Patterns using Pattern Recognition Machine learning may be used to create drawings. There has been a lot of interest in pattern recognition because of the wide variety of applications. To convert electrical schematics into an electronic format, analyse and validate electrical schematics in logic circuits, and recognise distinct entities in engineering drawings, symbol recognition, one of the pattern recognition segments, is actively utilised.

An crucial part of the drawing recognition process is identifying geometric forms, such as polygons, in 2D drawings. The recognition of polygon form when its edges include many interior points is difficult, despite the fact that this topic has been extensively studied in the literature. When a drawing item is fractured along its edge to make it easier to measure or fit another element, this scenario occurs. Machine learning is being used to recognise these polygons in a drawing in the present study.

The following is the structure of the paper. The next portion of the article briefly summarises the many techniques to drawing recognition in the Mechanical Engineering discipline, as well as the issues presently being addressed by researchers and the normal drawing recognition procedure. Machine learning is used to classify convex polygon forms, and the findings are then discussed.

2. Data-Driven Shape Analysis

Data-driven techniques to 3D shape processing are becoming more practical and valuable as 3D data becomes more widely available as a result of advancements in 3D sensor technology and 3D modelling software. With data-driven techniques, instead of focusing on individual objects, conventional approaches [Levy and Zhang 2011] look at groups of forms as a whole to discover significant mappings and relationships. Data-driven computational models that can reason about the features and connections of forms without depending on predefined rules or explicit instructions are also possible with these approaches. The discovery of geometric and structural patterns across collections of forms, patterns that serve as strong priors in different geometry processing applications, is made possible by data-driven algorithms that make use of shared information across numerous objects.

As far back as the 1980s, the notion of using data to help geometry processing has been widely used. There are a few publications based on this approach that go beyond the example-based paradigm, although they tend to focus on only one aspect of data-driven techniques: information transmission. Most of the time, the input to these issues comes in the form of a target shape and one or more exemplar forms having predefined or precomputed information of interest. A connection between the source and target shapes is generally established and important information is transferred from source to target. Such methodologies may be used for a wide range of purposes, including form analysis and shape synthesis.

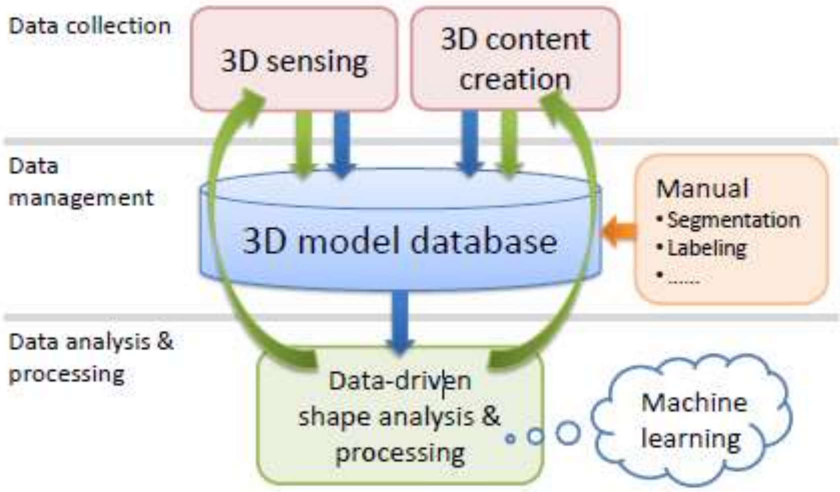

Figure 1: Data-driven shape processing and modeling provides a promising solution to the development of "big 3D data".

Data-driven methodologies are opening up new avenues for shape analysis and content production as the availability of 3D data grows. First, current shape repositories include a wide range of 3D material that may be directly reused to build new 3D models [Funkhouser et al. 2004]. Huge 3D geometric data may be used for 3D modelling by reusing content, which is the most basic use of big 3D geometric data. As a second example, high-level form knowledge may gain by co-analyzing a large number of shapes. Shape analysis is more accurate if it is supported by monitoring particular qualities over a range of semantically related forms rather of merely concentrating on a single item, according to many analysis tools. Co-analysis necessitates a crucial stage in which correlations between various forms in the input set must be found. This step differs significantly from constructing correlations pairwise. The consistency of the correlations over the whole set is an important term in co-analysis, which has both semantic and mathematical grounds for its existence.

For example, certain shapes may be labelled with semantic information (e.g., parts) that can then be applied to the rest of the forms via the use of learnt mappings. The idea of knowledge transfer across forms is being further developed as a result of this information propagation.

For the purpose of processing 3D forms and sceneries, data-driven methodologies are described in a high-level overview. Input data collection and processing, data representation and feature extraction, as well as learning and inference all share a number of components regardless of the method's specific application or purpose. In general, machine learning techniques rely heavily on representation, learning, and inference. When working with 3D geometric data, each of these components presents a number of intriguing and distinct challenges. Research on data-driven geometry processing has been a major factor in the development of computer vision and machine learning, as seen by the rising interest in 3D visual data from these domains. Data-driven 3D form and scene processing algorithms have certain features and problems that are discussed below. Figure 2 shows the most typical parts of these algorithms in a schematic form.

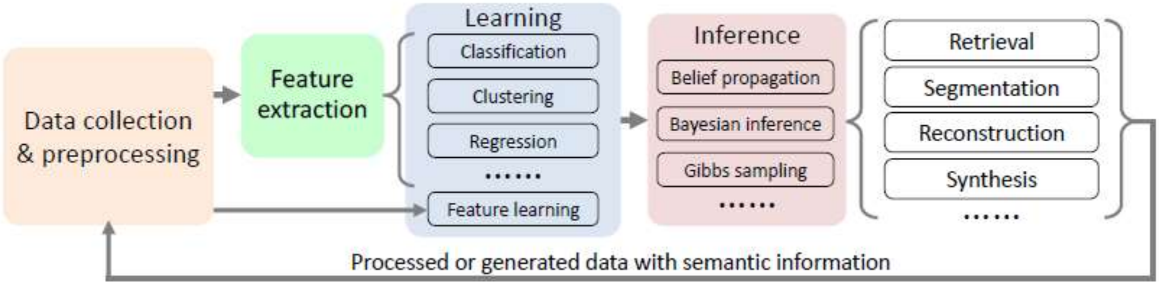

Figure 2: The general pipeline of data-driven geometry processing contains four major stages.

3. Drawing Recognition and Challenges Involved

Figure 3 depicts a typical machine component engineering drawing. Before turning the drawing to a 3D model, it is necessary to identify the various components.

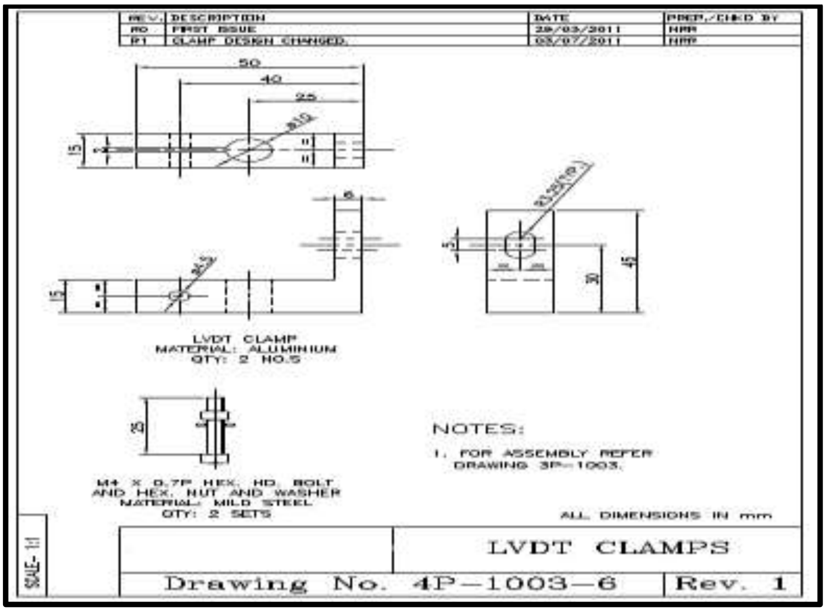

Figure. 3 Typical drawing for a machine component

3.1 Different Approaches for Drawing Recognition

Statistical and structural techniques to drawing recognition are two of the most important. Classification is accomplished by splitting feature space into multiple classes and then using statistical methods to classify each geometric element representing an element of a drawing. For this aim, classification techniques like Decision trees, Random Forests, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Naive Bayes, and Principal Component Analysis (PCA) are used. This technique uses geometric primitives to represent each symbol, and then an ideal model is developed utilising these primitives to match the input to the existing ideal models and classify it. Figure 4: Various Methods of Recognizing Drawings

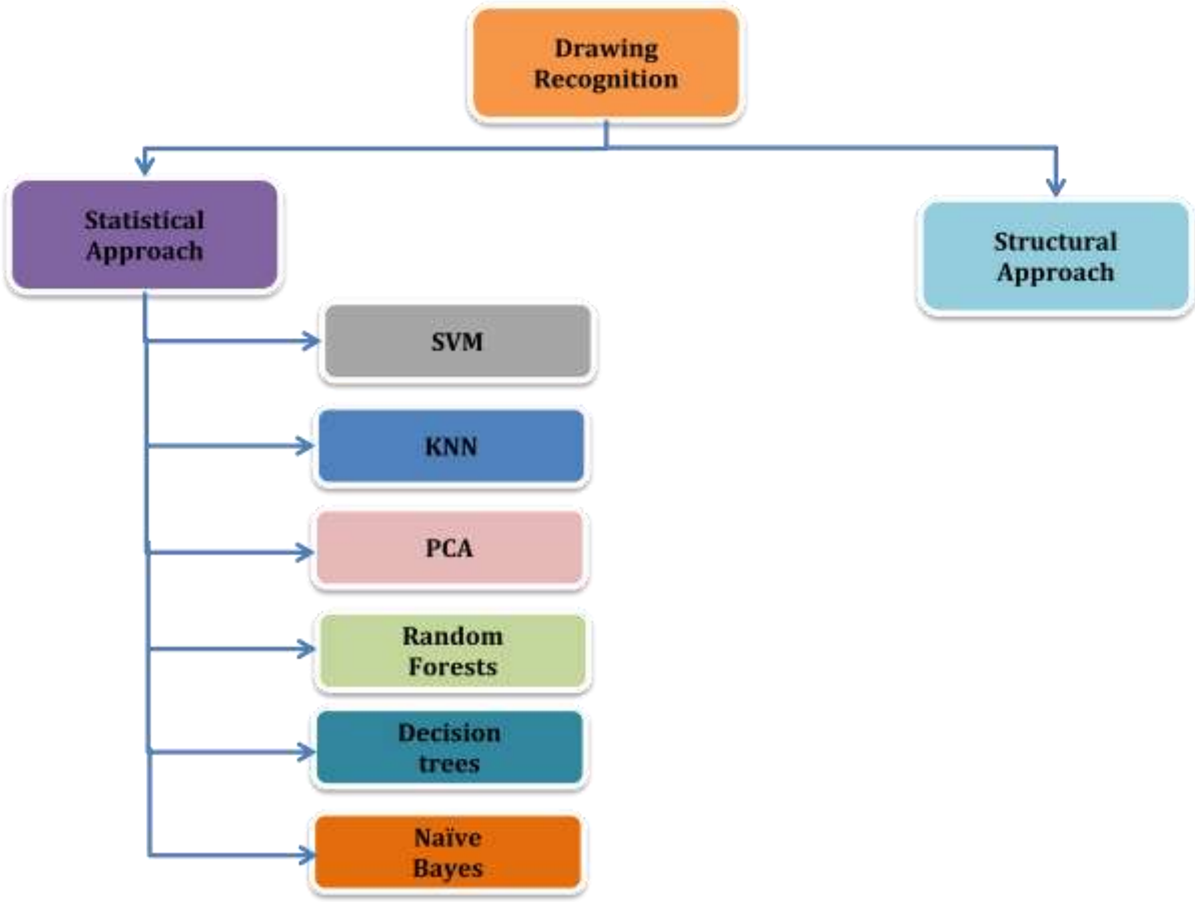

Figure. 4 Different approaches for drawing recognition.

These techniques face a number of issues, such as the difficulty to deal with big datasets, the segmentation of drawings and an inability to distinguish deformed objects. The definition of symbol signatures, which enable indexing into the artwork to discover locations where the symbol is likely to occur, is one method that has been suggested. Researchers have experimented with other symbol properties, such as images, centroids, moments invariants, etc., to address additional issues.

3.2 Typical Process of Drawing Recognition

Lots of work has already been done on pattern matching and symbol recognition so that technical drawings may be analysed without human intervention. Pattern matching may be used to match two dissimilar drawings, as well as to match a 2-D drawing to its 3-D counterpart. The detection of certain traits is greatly aided by symbol recognition or pattern recognition in industrial processes. It may also be used to digitally transform drawings created on paper.

Recognizing and extracting valuable information from the drawings is called data extraction Raster and Vector are the two most used representations for spatial drawing data. A grid of cells or pixels is used to represent data in raster format, and each cell in the grid has a corresponding value. Data in vector format is made up of coordinates and details on how the coordinates are linked together. Unlike raster formats, vector formats allow data to be represented in its original resolution, but in raster formats, resolution is dictated by the cell sizes. For big datasets, the processing of cell properties in raster format might be difficult. To avoid raster errors, vector files are used instead. Different CAD software or geometric modelling programmes save drawing information in vector databases, such as the Data Exchange Format (DXF), the Standard for the Exchange of Product Model (STEP), and the Initial Graphic Exchange Specification (IGES). Raster images are transformed to vectors if they are available before further processing begins. DXF is a common data storage format. It is simpler to write programmes to retrieve information from these structures since they are more clearly defined. In [2] and [3], several ways of extracting CADD data from CADD designs are explored. Extracting data from Engineering Drawings, which are often made up of numerous entities, such as Drawing views, text, symbols that convey information about manufacturing or materials, and measurements, requires segmentation. Polygonal forms in the views are identified and classified in the first phase of the CAD model conversion process.

Conversion of data into features

Once the data has been gathered, it has to be transformed into characteristics that machine learning models can use to analyse technical drawings. They should be invariant to scale and translating for diverse viewpoints, texts, dimensions, and symbols.

Image normalisation and invariants are discussed by Flusser [5] as three techniques to dealing with this problem: brute force and image normalisation. Each class must be represented by at least one blurred, rotated, scaled, and deformed version in a training set for brute force approach. Using this approach will take a long time. Normalization is a technique that simplifies the process of categorising an item by transforming it into a standard position. One goal of the invariants method is to employ a collection of characteristics to describe the object that are insensitive to certain deformations. Moment invariants, mentioned in [4], are one such characteristic used to represent 2-D drawings. Moment invariants that can handle both geometric and radiometric aberrations are described in this paper. A Graph Theory-based technique, as explained in [5,] circumvents some of the drawbacks of using feature vectors. Structural Pattern Recognition followed on from Statistical Pattern Recognition.

Use of features in Machine Learning models. During the Pure, Impure, and Extreme phases, the development of machine learning algorithms for pattern recognition is recounted in [16]. In the Pure era, graph matching was done using optimum and suboptimal algorithms to achieve precise and inexact matches. These models include Nearest Neighbor (NN) classification, its variants, K-NN and K+NN classification, clustering and Learning Vector Quantization were created during the Impure era. For classification, these models need a labelled dataset. A variety of machine learning models were utilised for pattern identification throughout the severe era, including Support Vector Machines, Kernel Principal Component Analysis (KPCA), and Multilayer Perceptrons (MLP) (MLP). Symbols used in electrical drawings may be recognised using neural networks [6]. The multi-layer perceptron (MLP) classifier with back propagation (BP) learning model has been used effectively in numerous pattern recognition situations [7]. Neural Networks have the benefit of being able to adapt to changes in training. The collected characteristics may be subjected to any of these categorization models in order to identify points of view, texts, symbols, or dimensions. In order to create a 3D CAD model or bill of materials, all of these models may be combined.

4. Identification of Polygons Using Machine Learning

An engineering drawing is used to train a machine learning model to recognise various polygonal forms. In the machine learning model, you may incorporate things like photos, the coordinates of vertices, the slopes of lines, the centroids, and moment invariants. [8] Statistical or machine learning techniques may automatically learn from a large number of training samples in order to obtain decent identification abilities. Engineering drawings include a wide variety of items, thus it is hard to include all of them.

A. Generation of Polygon Dataset

To train the model, researchers employed datasets comprising the sequential coordinates of points arranged along 3–7-sided convex polygons produced at random. Figure 5 depicts a variety of common polygonal forms. Randomly sized polygons with randomly arranged interior points were generated using AutoCAD's Visual LISP function. The.csv format is used to export the coordinates of each point in turn.

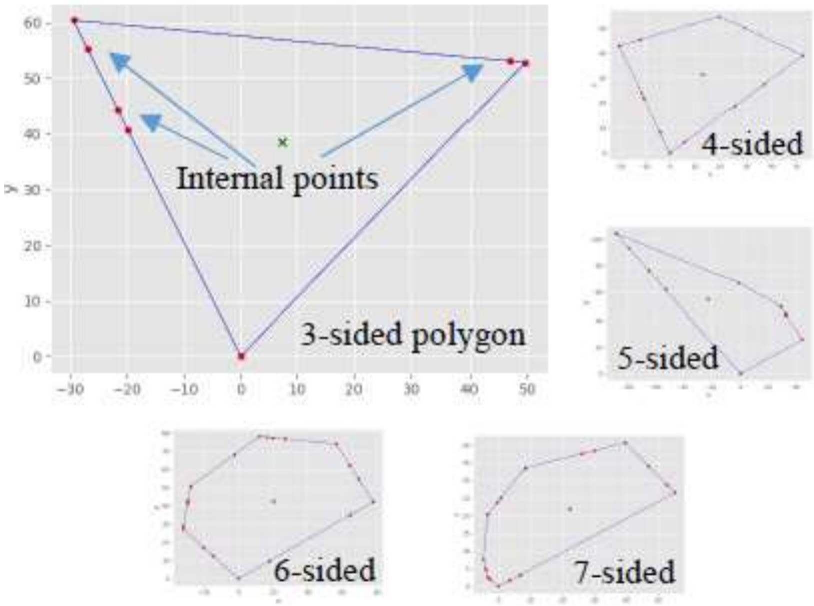

Figure. 5 Typical 3 to 7 sided polygons with internal points

Machine learning models were trained and tested on four different sets of data. Each dataset is meant to provide a different amount of difficulty in estimating the number of sides numerically.

Polygons with three to seven sides are included in the first dataset (POLY-1). There are no mid-side nodes in these polygons, since each side is made up of a single line element. It is possible to regulate the angle between neighbouring edges, which may vary from 5 to 175 degrees, and record the coordinates up to 8 decimal points. There is no uncertainty in this dataset when it comes to identifying the number of sides. One thousand polygons (POLY-2) with a number of sides between three and seven are included in the second dataset (POLY-2) (maximum 3). There is no control over these angles and the coordinates are kept to four decimal places. If two edges are almost perpendicular to one another, it is difficult to determine the precise number of sides.

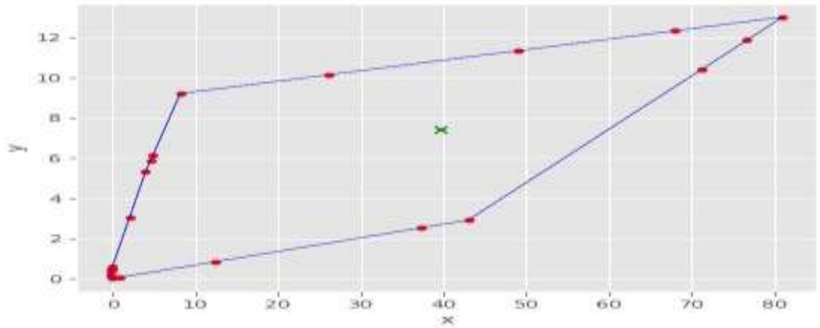

Figure. 6 Typical polygon with large uncertainity in detection of number of sides.

For instance, as seen in Figure 6, the angle between neighbouring sides of a polygon might vary from 5 degrees to 175 degrees. A reduction in erroneous counting or edge-dropping has been achieved, however internal points may still lead to an incorrect number of sides.

There are 2000 polygons in POLY-4, which was developed by merging POLY-1 and POLY-3. It is projected to perform better in terms of accuracy since it has a higher percentage of forms that are immediately recognisable. There are rows for numerous polygons, and columns for information on each of those polygons, in these databases. This table shows the number of sides, points, and x and y coordinates for each of the polygon's points in the order they appear in the table.

Feature Selection

For testing the polygon recognition model, we used the following features:. ii. The Cartesian coordinates of the polygons' consecutive points the polar coordinates of the polygon's next two points. It is important to note that x axis slopes of consecutive line segments. Each polygon has its own set of geometric parameters: total points, the number of nearly-collinear points, perimeter, area, a percentage of perimeter as a percentage of total area, x- and y-coordinates, the total area bound by x- and y-limits, bound-area percentage, and centroid distance from origin to origin in radial distance.

Selection of Machine Learning Model

The Python scikit-learn package was used to carry out the task. For this task, it was decided that a classifier model was necessary. The decision tree approach may be used, although it suffers from the problem of overfitting for small or medium-sized datasets.. As a result, the primary machine learning model was 'Random Forest Classifier.' There were also comparisons with the classification models of K Nearest Neighbor (KNN) and Support Vector Classifier (SVC).

Standard Scaler was used to pre-process the data. The dataset was divided at random such that 75% of the data could be used for training, and the remaining 25% could be used for testing.

5. Discussion

New data-driven and machine learning applications in geometry processing and computer graphics are not easy to come by. Some essential considerations need to be made while creating a data-driven technique, such as the complexity of computing and the scalability of the system. These difficulties are briefly addressed in this section.

Computational complexity. Algorithms for data-driven form analysis and processing often include many phases (Figure 2). Input shape resolution (number of faces, surface points, pixels, or voxels) and quantity and kind of geometric features all have a role in determining the difficulty of each step. It is common practise to do the feature extraction step for each shape, hence the time complexity increases linearly with the number of forms provided. As the number of primitives in the input shape representation increases, the extraction of local geometric characteristics such as surface curvature or PCA-based descriptors becomes linearly dependent on the number of these primitives. Shape diameter, geodesic distance-based features, and heat-kernel descriptors are all examples of geometric features that capture less local or global information about the shape and are thus computationally more costly, i.e., super-linear in the number of primitives.

Scalability. The input data is intrinsically linked to data-driven approaches. 3D shape and scene data repositories have grown exponentially over the last several years, and this has had a direct impact on the improvement of data-driven geometry processing. In data-driven geometry processing approaches, training forms have expanded from a few tens to several thousands [6-8] in number. Data-driven approaches' accuracy and generalizability may be enhanced by the growing accessibility of 3D data.

Scalability, on the other hand, becomes a problem as the system grows. Debugging these approaches becomes more difficult as more data accumulates and the processing time increases. Because of the highdimensionality and complexity of the 3D geometric feature representations, scaling becomes even more difficult. Debugging the pipeline of these approaches on smaller datasets before moving on to bigger ones, using simpler learning techniques before moving on to more complicated ones, or using computer clusters to perform offline phases are all possible remedies.

Scope of application. A data-driven approach isn't appropriate for all problems in shape analysis and processing. Non-data-driven techniques should be explored when the problem's basic rules, concepts, and parameters can be explicitly and clearly described. For example, a physics-based approach may be used to deform a shape using an elastic material and known physical parameters and forces, rather than a data-driven approach. Shape analysis and processing difficulties, on the other hand, might be difficult or impossible to solve using manual design rules and principles, if not impossible. Shape and scene recognition, high-level processing, structure parsing, co-analysis, reconstruction from noisy missing data, and modelling with high-level user input are all examples of situations where this is a common occurrence.. Co-segmentation, for example, involves predicting multiple potential geometric and semantic relationships among input forms, which would be almost difficult to capture by hand-designed rules. This is why shape coanalysis is so important. It is preferable to use data-driven algorithms that automatically calculate the geometric, semantic and structural links in the input shapes [9]. The challenge of analysing the form style is another good illustration.

It is difficult to manually quantify geometric criteria for modelling the aesthetic resemblance of forms despite the fact that individuals have an intuitive sense of style. Much more suited for this quantification is an algorithm for style analysis that learns its parameters from data [10].

6. Conclusion

Despite the extensive published work on drawing recognition in the mechanical engineering sector, no complete solution has yet been discovered. Convex polygons with three to seven sides and segmented edges were effectively identified using a machine learning-based technique. In the case of a well-structured collection of randomly formed polygons, 93.3 percent of accuracy is achieved. Poorly formed polygons have lower precision, although it is similar to that of standard Python code. Engineering drawings for machine components may be used to detect more complicated geometric forms using the approach outlined here.

References

1. M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects.," *Science (New York, N.Y.)*, vol. 349, no. 6245, American Association for the Advancement of Science, pp. 255–60, 17-Jul-2015.
2. Q. Huang, "Application of Artificial Intelligence in Mechanical Engineering Qi Huang," in *2nd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2017) Application*, 2017, vol. 74, no. Iccia, pp. 855–860.
3. D. Wu, C. Jennings, J. Terpenny, S. Kumara, and R. X. Gao, "Cloud-Based Parallel Machine Learning for Tool Wear Prediction," *J. Manuf. Sci. Eng.*, vol. 140, no. 4, p. 041005, Feb. 2018.
4. N. Ahmad and A. A. Haque, "Manufacturing feature recognition of parts using DXF files," *4th Int. Conf.Mech. Eng.*, no. April, p. VI-111-116, 2001.
5. A. S. Sobh, S. A. Salem, R. Darwish, M. Hussein, and O. Karam, "Unsupervised clustering of materials properties using hierarchical techniques," *Int. J. Collab. Enterp.*, vol. 5, no. 1/2, p. 74, 2015.
6. M. Fuge, B. Peters, and A. Agogino, "Machine Learning Algorithms for Recommending Design Methods," *J. Mech. Des.*, vol. 136, no. 10, p. 101103, Aug. 2014.
7. A. K. Noor, "AI and the Future of the Machine Design," *Mech. Eng.*, vol. 139, no. 10, p. 38, Oct.2017.
8. G. X. Gu, C.-T. Chen, and M. J. Buehler, "De novo composite design based on machine learning algorithm," *Extrem. Mech. Lett.*, vol. 18, pp. 19–28, Jan. 2018.
9. M. L. Dering, C. S. Tucker, and S. Kumara, "An Unsupervised Machine Learning Approach to Assessing Designer Performance During Physical Prototyping," *J. Comput. Inf. Sci. Eng.*, vol. 18, no. 1, p. 011002, Nov. 2017.
10. S. K. Chandrasegaran *et al.*, "The evolution, challenges, and future of knowledge representation in product design systems," vol. 45, no. 2, pp. 204–228, Feb. 2013.
11. J. Lladós, E. Valveny, G. Sánchez, and E. Martí, "Symbol Recognition: Current Advances and Perspectives," 2001.
12. M. F. A. Jabal, M. S. M. Rahim, N. Z. S. Othman, and Z. Jupri, "A Comparative Study on Extraction and Recognition Method of CAD Data from CAD Drawings," in *2009 International Conference on Information Management and Engineering*, 2009, pp. 709–713.