Classification of LEGO parts using Topological features

Team members: Jiayue Xu, Srishti Saha

Abstract

Image Classification is a widely studied subject in computer vision and machine learning. Methods in the deep learning space are being continuously developed and tested to pre-process and classify images with high accuracy. On integrating topological data analysis with conventional data preparation techniques, it has been proven to provide additional information and features for image classification. This paper tests if the application of topological features to 2D part drawings of LEGO parts improves the classification algorithms built on this data-set. We aim to classify parts into their predefined categories. We demonstrate how the performance of classification algorithms is altered with and without topological features.

1 Introduction

1.1 Motivation

A lot of manufacturing and product companies use image data (mechanical drawings, CAD models and 3D image representations) of their products for a variety of applications like checking for anomalies in model consistency, checking for flaws in produced models, categorizing products into classes for inventory and supply purposes etc. These applications require building intelligent systems, a lot of which employ machine learning models. These models either augment the manual process of inspecting these parts or can even automate the complete procedure. With the vast amount of data and advancement in technologies, it is easy to extract simplistic features like size, orientation, colour etc. from these drawings. However, a lot of vital information is contained in the actual shape of the data. Our project aims to extract topological features from images of LEGO parts and then build classification algorithms using these topological features over and above the conventional features obtained from the 2D drawings of these LEGO parts.

1.2 Topological Data Analysis

Topological data analysis (TDA) comprises of powerful tools that can quantify shape and structure in any dataset (represented as point cloud) in order to answer questions from the data's domain. To do so, we represent some aspect of the structure of the data in a simplified topological signature. In topological analysis, data implies a collection of data points with each arising from different entities like objects, people, times, processes and and so on. TDA focusses on the proximity between these data points. The distance between these points is measured using distance metrics (like Euclidean distance). The main objective of TDA is to investigate the intrinsic shape of the data using a provided distance measure.

1.3 Relevant Work

M Fonseca et.al.[2] in their paper "Content-Based Retrieval of Technical Drawings" used topological feature extraction to plug them further into machine learning algorithms. Alshawabkeh, Y. et' al [3] converted an RGB image into a 3D point cloud which is then converted into a series of 2D depth maps. This was then used for further analysis of these images. Nathaniel Saul et al.[4] and Wei

Guo[5] used the Mapper Algorithm to select the most important features from a high-dimensional TDA dataset. This is a TDA based algorithm that selects key features based on distance-metrics calculated on a generated topological network.

1.4 Data

For the purpose of this project we used image data of LEGO parts ([1]). It consists of the LEGO Parts/Sets/Colors and Inventories of every official LEGO set in the Rebrickable database. The raw data source has 12 csv files with parts information and more than 80,000 PNG images of official LEGO parts of different colors. We are selecting image data for only black parts as that has the most number of parts i.e. 2,600 parts (see Figure 2)

The dataset has 2 sets of files: 3D drawings of LEGO parts labeled by their part ID and a structured dataset (.csv file) describing these parts by their name, material, and category label.

However, a major caveat is that some parts are mentioned in the CSV file do not have part drawings as they are simple manuals or sheets for describing other parts in the sets. To address this issue, we select only those parts (2388 images) which have an image as well as details on the CSV file. We then split the dataset by a 80-20 percent ratio to form our train and test datasets. The final training dataset has data for 1910 parts while the test set has data for 478 parts.

Moreover, the categorization of parts in the CSV file was done according to the business definition/functionality of these parts. This led to over 60 categories. To simplify the classification model requirements, in this project we used an intuitive sense to condense these categories to 6 major categories as shown in Figure 3.

2 Methodology

In this project, we followed a simple methodology (see Figure 1) to extract features from the image data and then build machine learning algorithms to classify these LEGO parts into categories.

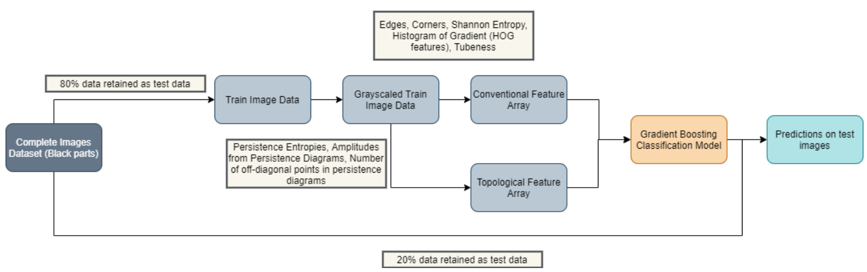

Figure 1: Overview of Methodology

2.1 Feature Extraction

2.1.1 Edge extraction (Canny Edge Detection)

Canny Edge Detector is a multi-stage edge detection algorithm with a high accuracy rate. The three main criteria of Canny edge detection include:

1. Detection of edges is done with a low error rate. This step of the detection algorithm should accurately catch as many edges shown in the image as possible.
2. The edge point detected from the operator accurately localizes on the center of the edge detected.
3. A given edge in the image should only be marked once, and where possible, image noise should not create false edges.

We obtained satisfactory results for the detected edges across images (see Figure 4).

2.1.2 Corner Detection (Harris Corner Detector)

Harris Corner detection is a commonly used algorithm in Computer Vision applications. A corner is a point whose local neighborhood stands in two dominant and different edge directions. In other words, a corner can be interpreted as the junction of two edges, where an edge is a sudden change in image brightness. Since corners are vital in image data to understand the orientation, depth, illumination, and size of any object in an image, it is an important feature to extract. We applied this transformation to all images (see Figure 5)

2.1.3 Texture Features

We use entropy, mean and standard deviation of the images to extract texture of the same. In 2016, Eri Matsuyama et al.[6] showed that information entropy of an image can be used as an effective featue for measuring textures of an image in addition to the commonly used features: mean and standard deviation.

1. Shannon Entropy: The information entropy is the log-base-2 of the number of possible outcomes for a message. For an image, local entropy is related to the complexity contained in a given neighborhood, typically defined by a structuring element. The entropy filter can detect subtle variations in the local gray level distribution. Since it can effectively measure the intensity level of each individual pixel, it computes the distribution and the occurrence of pixel values which serves as a proxy for the texture of the image.
2. Mean: Small mean values will indicate a coarse image texture having a grain size equal to or larger than the magnitude of the displacement vector. This helps measure the texture of the given image.
3. Standard Deviation: Standard Deviation is another commonly used texture feature as it measures of the dispersion of the gray-level differences at a certain distance.

All the above methods were applied to all the images in the dataset (see Figure 5).

2.1.4 HOG (Histogram of Oriented Gradient) transformation

HOG is a popular feature descriptor used in Computer Vision applications. It is effective as it is invariant to geometric and photometric transformations. This algorithm represents objects in images as histograms using intensity gradients or edge directions (see Figure 5).

Since the HOG transformation returns a high dimensional dataset for each image, this cannot be directly passed through a machine learning model. We have to reduce the dimensionality of the dataset while preserving the maximum amount of information in the data. To do so, we applied the **Principal Component Analysis (PCA)** to the HOG transformed dataset. We extracted 10 components from over 100,000 variables. Although this captured only 40 percent of the total information in the HOG transformed dataset (see Figure 6), we used only 10 components as it was a computationally expensive and time-intensive process.

2.1.5 SATO tubeness

This filter can be used to detect continuous ridges, e.g. tubes, wrinkles, rivers. It can be used to calculate the fraction of the whole image containing such features and is defined only for 2-D and 3-D images. This algorithm calculates the eigenvectors of the Hessian to compute the similarity of an image region to tubes (see Figure 7).

2.1.6 Persistence Entropies

We then extracted a few topological features. The first measure we extracted was the persistence entropy. This measure quantifies entropy of the points in a persistence diagram. If $D = [(b_i, d_i)]_{i \in I}$ is a persistence diagram with each $d < +\infty$, then the persistence entropy of D is defined by:

$$E(D) = -\Sigma_{i \in I} p_i log(p_i)$$

Given a persistence diagram consisting of birth-death-dimension triples $[b, d, q]$, subdiagrams corresponding to distinct homology dimensions are considered separately, and their respective persistence entropies are calculated as the (base 2) Shannon entropies of the collections of differences $d - b$ ("lifetimes"), normalized by the sum of all such differences.

2.1.7 Amplitudes from Persistence Diagrams

For each persistence diagram in a collection, a vector of amplitudes or a single scalar amplitude is calculated according to the following steps:

1. The diagram is partitioned into subdiagrams according to homology dimension.
2. The amplitude of each subdiagram is calculated according to the metric passed and metric parameters. This gives a vector of amplitudes, $a = (a_{q1}, ..., a_{qn})$ where the values of q_i range over the available homology dimensions.
3. The final result is either a itself or a norm of a, specified by the parameter order.

Using this methodology, we extracted amplitudes for 3 dissimilarity functions: L^p distance between Betti curves, L^p distance between persistence landscapes and for the Wasserstein distance measure.

2.1.8 Number of points from Persistence Diagrams

This process was used to estimate the number of off-diagonal points in persistence diagrams, per homology dimension.

Given a persistence diagram consisting of birth-death-dimension triples $[b, d, q]$, subdiagrams corresponding to distinct homology dimensions are considered separately, and their respective numbers of off-diagonal points are calculated. This estimation was done for all images in the dataset.

2.2 Learning Models

We then implemented 4 learning models to study and compare the impact of topological features on classification models.

2.2.1 Baseline Neural Network Model with direct images

The first model we tried was a Neural Network with a transfer learning of ResNet-50 architecture. We passed the images of the LEGO parts directly into this model and obtained a validation-set accuracy of 77.20 percent. Using this model, we were able to correctly classify a part from the 'Figures' category and 'Bricks and Plates' category. However, it could not identify parts from the mechanical category and transportation category (see Figure 8). This might be because these parts have high similarity to the parts from the predicted class.

2.2.2 Gradient Boosting Classification Model with only conventional features

We built a Gradient Boosting classification model using the conventional features only. Using this model, we were able to achieve a validation-set accuracy of 65.3 percent (error rate: 0.347). The test set accuracy was obtained to be 54 percent. We see that this model was performing fairly well in classifying some categories like figures and bricks (see Figure 10). However, it performed sub-optimally in other categories.

An example of an incorrect classification can be seen in figure (11). Here, a part that belongs to the brick category was incorrectly classified as a mechanical part.

We also did an evaluation of the feature importance for this model. We see that the top most important features (see Figure 9) are the PCA-derived HOG components.

2.2.3 Gradient Boosting Classification Model with both Conventional Features and Persistence Entropies

We built a Gradient Boosting classification model using the conventional features only. Using this model, we were able to achieve a validation-set accuracy of 66 percent. The test set accuracy was obtained to be 57 percent. We see that this model was performing fairly well in correctly identifying some categories like figures and bricks (see Figure 14). However, it performed suboptimally in other categories.

In the example below (see Figure 13), we see that this model was able to correct the previous incorrect classification. The part mentioned was correctly identified as a brick part.

We also did an evaluation of the feature importances for this model. We see that the top most important features (see Figure 12) are still the PCA-derived HOG components. However, the persistence entropy feature (1st component) appears in the top 8 features by significance. The second component appears in the least important features.

2.2.4 Gradient Boosting Classification Model with both conventional features and all topological features

The final model we implemented included all the conventional features in the previous models along with all the topological features i.e. persistence entropies, amplitudes from persistence diagrams and number of points derived from persistence diagrams. Using this model, we see that the validation-set accuracy is 68 percent (error rate of 0.32). We also saw that the test accuracy increased to 60.6 percent.

Using this model, we saw that the performance of classification improved significantly in a few categories like Bricks (as shown in Figure 16). Categories like Figures, Panels, Transportation and Mechanical also showed marginal improvement in accuracy. However, the category 'Others' showed a reduction in the performance.

On evaluating the feature importances of this model, we observed that the number of off-diagonal points in the persistence diagrams is the 7th most important feature. The next most important topological feature is the amplitude of the persistence landscapes.

3 Results and Conclusions

We see that the baseline neural network model shows a better classification accuracy than the models with extracted features from the image dataset. However, we can also see that the addition of persistence entropies as a feature in the model led to a significant increase in the test accuracy (by 4 percent). We also saw that one of the components of persistence entropies was a significant feature in the second model. Incorporating this feature led to an improvement in the model performance for some specific categories like Bricks, Transportation and Mechanical. This means that inclusion of topological features can definitely capture some information that the conventional features were unable to. As we progressed to the third model, we see that the test accuracy improved with the inclusion of more topological features. Moreover, more topological features like number of off-diagonal points from persistence diagrams and amplitudes became significantly important features.

4 Future Work

This work can be extended to improvise the model performance by including more exhaustive topological features. We also need to tune the model to avoid overfitting. The models can also be built using features extracted from the persistence landscapes directly. We could also try other dimensionality reduction techniques to extract more informative components. One of the challenges in the model was that the redefined categories are not reflective of the topological similarity of these parts (but the functional similarity of the parts). The data could then be altered to create more topologically-dependent categories. This would help the topological features correlate more with the target variable (category).

5 Appendix

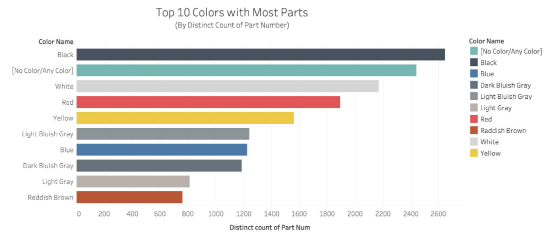

Figure 2: Distribution of LEGO parts image data across colours

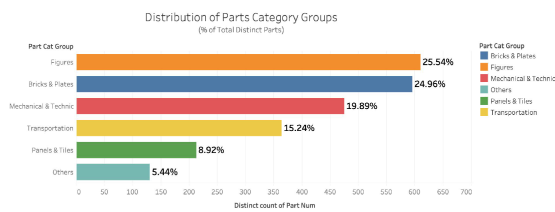

Figure 3: Redefining categories for LEGO parts

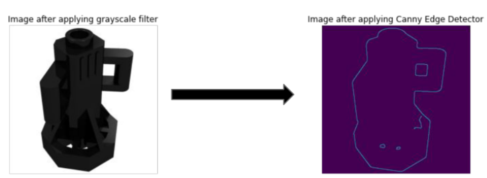

Figure 4: Use of the Canny Edge Detector Algorithm

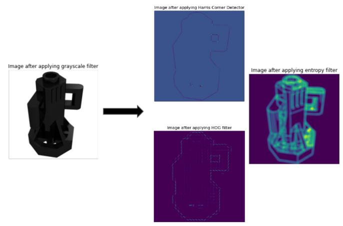

Figure 5: Applying Corner Detection, Entropy Filter and HOG filter to grayscale images

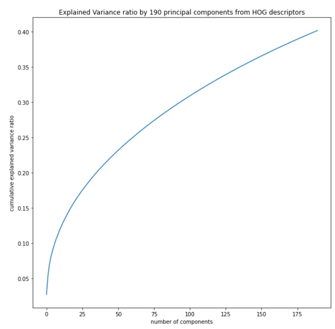

Figure 6: Explained Variance from 10 Principal Components of HOG features

References

[1] Rebrickable Database: https://rebrickable.com/api/

[2] Fonseca, Manuel J.; Ferreira , A; Jorge, Joaquim A. *Content-Based Retrieval of Technical Drawings* Int. J. of Computer Applications in Technology, Vol. 23, Nos. 2/3/4, 2005

[3] Alshawabkeh, Y. *Linear feature extraction from point cloud using color information.* Herit Sci 8, 28 (2020).

[4] Saul N.; Arendt, Dustin L. *Machine Learning Explanations With Topological Data Analysis*

[5] Guo W. *Feature Extraction Using Topological Data Analysis for Machine Learning and Network Science Applications,* 2019

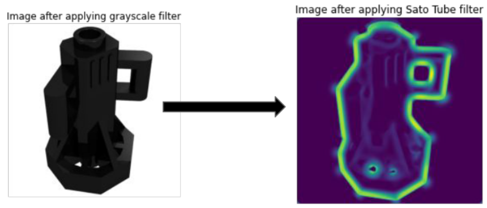

Figure 7: Using the SATO tubeness Filter

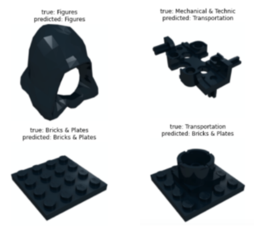

Figure 8: Using the Neural Network Result

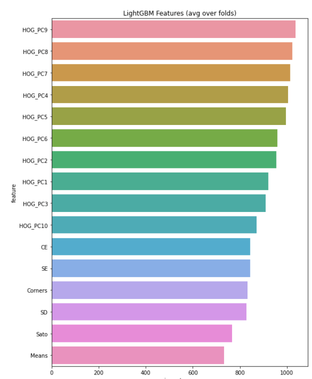

Figure 9: Feature Importances in the model with conventional features

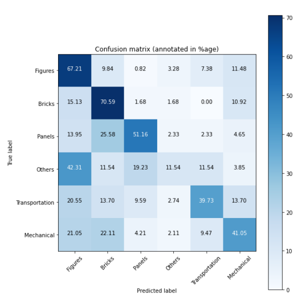

Figure 10: Confusion Matrix for Model with only Conventional Features

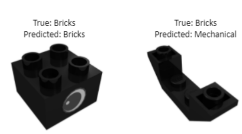

Figure 11: Incorrect Classification using the Model

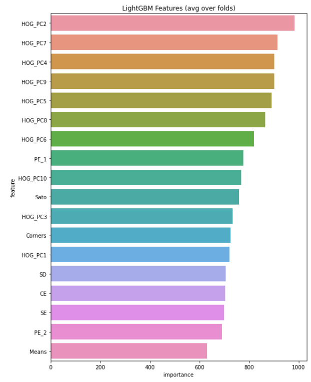

Figure 12: Feature Importances for the Model using Conventional Features and Persistence Entropies

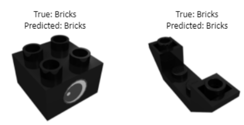

Figure 13: Classification results from model using conventional features and persistence entropies

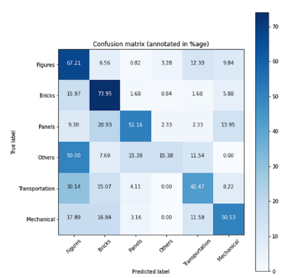

Figure 14: Conventional Features and Persistence Entropies

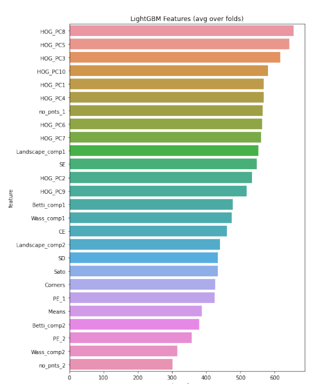

Figure 15: Feature Importances for the Model using Conventional Features and all Topological Features

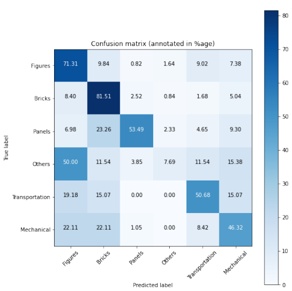

Figure 16: Confusion Matrix of the Model using Conventional Features and all Topological Features